NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCES
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

PROGRAM OF POSTGRADUATE STUDIES

PhD THESIS

Nonconvex Optimization Algorithms for Structured Matrix Estimation in Large-Scale Data Applications

P. V. Giampouras

ATHENS

July 2018

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Αλγόριθμοι Μη-Κυρτής Βελτιστοποίησης για την Εκτίμηση Δομημένων Πινάκων σε Εφαρμογές Δεδομένων Μεγάλης Κλίμακας

Παρασκευάς Β. Γιαμπουράς

ΑΘΗΝΑ

Ιούλιος 2018

PhD THESIS

Nonconvex Optimization Algorithms for Structured Matrix Estimation in Large-Scale Data Applications

P. V. Giampouras

SUPERVISOR: Athanasios Rontogiannis, Research Director NOA

THREE-MEMBER ADVISORY COMMITTEE:

Athanasios Rontogiannis, Research Director NOA

Konstantinos Koutroumbas, Research Director NOA

Sergios Theodoridis, Professor NKUA

SEVEN-MEMBER EXAMINATION COMMITTEE

Athanasios Rontogiannis,
Research Director NOA

Konstantinos Koutroumbas,
Research Director NOA

Sergios Theodoridis,
Professor NKUA

Nikolaos Kalouptsidis,
Professor NKUA

Petros Maragos,
Professor NTUA

Konstantinos Berberidis,
Professor U. of Patras

Eleftherios Kofidis,
Assoc. Professor U. of Piraeus

Examination Date: July 9, 2018

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Αλγόριθμοι Μη-Κυρτής Βελτιστοποίησης για την Εκτίμηση Δομημένων Πινάκων σε Εφαρμογές Δεδομένων Μεγάλης Κλίμακας

Παρασκευάς Β. Γιαμπουράς

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: Αθανάσιος Ροντογιάννης, Διευθυντής Ερευνών ΕΑΑ

ΤΡΙΜΕΛΗΣ ΕΠΙΤΡΟΠΗ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ:

Αθανάσιος Ροντογιάννης, Διευθυντής Ερευνών ΕΑΑ

Κωνσταντίνος Κουτρούμπας, Διευθυντής Ερευνών ΕΑΑ

Σέργιος Θεοδωρίδης, Καθηγητής ΕΚΠΑ

ΕΠΤΑΜΕΛΗΣ ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ

Αθανάσιος Ροντογιάννης,
Διευθυντής Ερευνών ΕΑΑ

Κωνσταντίνος Κουτρούμπας,
Διευθυντής Ερευνών ΕΑΑ

Σέργιος Θεοδωρίδης,
Καθηγητής ΕΚΠΑ

Νικόλαος Καλουπτσίδης,
Καθηγητής ΕΚΠΑ

Πέτρος Μαραγκός,
Καθηγητής ΕΜΠ

Κωνσταντίνος Μπερμπερίδης,
Καθηγητής Παν. Πατρών

Ελευθέριος Κοφίδης,
Αν/της Καθηγητής Παν. Πειραιώς

Ημερομηνία Εξέτασης: 9 Ιουλίου 2018

ABSTRACT

Structured matrix estimation belongs to the family of learning tasks whose main goal is to reveal low-dimensional embeddings of high-dimensional data. Nowadays, this task appears in various forms in a plethora of signal processing and machine learning applications. In the present thesis, novel mathematical formulations for three different instances of structured matrix estimation are proposed. Concretely, the problems of a) simultaneously sparse, low-rank and nonnegative matrix estimation, b) low-rank matrix factorization and c) online low-rank subspace learning and matrix completion, are addressed and analyzed. In all cases, it is assumed that data are generated by a linear process, i.e., we deal with linear measurements. A suite of novel and efficient *optimization algorithms* amenable to handling *large-scale data* are presented. A key common feature of all the introduced schemes is *nonconvexity*. It should be noted that albeit nonconvexity complicates the derivation of theoretical guarantees (contrary to convex relevant approaches, which - in most cases - can be theoretically analyzed relatively easily), significant gains in terms of the estimation performance of the emerging algorithms have been recently witnessed in several real practical situations.

Let us first focus on simultaneously sparse, low-rank and nonnegative matrix estimation from linear measurements. In the thesis this problem is resolved by three different optimization algorithms, which address two different and novel formulations of the relevant task. All the proposed schemes are suitably devised for minimizing a cost function consisting of a least-squares data fitting term and two regularization terms. The latter are utilized for promoting sparsity and low-rankness. The novelty of the first formulation lies in the use, for the first time in the literature, of the sum of the reweighted ℓ_1 and the reweighted nuclear norms. The merits of reweighted ℓ_1 and nuclear norms have been exposed in numerous sparse and low-rank matrix recovery problems. As is known, albeit these two norms induce nonconvexity in the resulting optimization problems, they provide a better approximation of the ℓ_0 norm and the rank function, respectively, as compared to relevant convex regularizers. Herein, we aspire to benefit from the use of the combination of these two norms. The first algorithm is an incremental proximal minimization scheme, while the second one is an ADMM solver. The third algorithm's main goal is to further reduce the computational complexity. Towards this end, it deviates from the other two in the use of a matrix factorization based approach for modelling low-rankness. Since the rank of the sought matrix is generally unknown, a low-rank imposing term, i.e., the variational form of the nuclear norm, which is a function of the matrix factors, is utilized. In this case, the optimization process takes place via a block coordinate descent type scheme. The proposed formulations are utilized for modelling in a pioneering way a very important problem in hyperspectral image processing, that of hyperspectral image unmixing. It is shown that both sparsity and low-rank offer meaningful interpretations of inherent natural characteristics of hyperspectral images. More specifically, both sparsity and low-rankness are reasonable hypotheses that can be made for the so-called *abundance* matrix, i.e., the nonnegative matrix containing the fractions of presence of the different materials, called *endmembers*, at the region depicted by each pixel. The merits of the proposed algorithms over other state-of-the-art hyperspectral unmixing algorithms are corroborated in a wealth of simulated and real hyperspectral imaging data experiments.

In the framework of low-rank matrix factorization (LRMF) four novel optimization algorithms are presented, each modelling a different instance of it. All the proposed schemes share a common thread: they impose low-rank on both matrix factors and the sought matrix by a newly introduced regularization term. This term can be considered as a generalized weighted version of the variational form of the nuclear norm. Notably, by appropriately selecting the weight matrix, low-rank enforcement amounts to imposing joint column sparsity on both matrix factors. This property is actually proven to be quite important in applications dealing with large-scale data, since it leads to a significant decrease of the induced computational complexity. Along these lines, three well-known machine learning tasks, namely, denoising, matrix completion and low-rank nonnegative matrix factorization (NMF), are redefined according to the new low-rank regularization approach. Then, following the block successive upper bound minimization framework, alternating iteratively reweighted least-squares, Newton-type algorithms are devised accounting for the particular characteristics of the problem that each time is addressed. Lastly, an additional low-rank and sparse NMF algorithm is proposed, which hinges upon the same low-rank promoting idea mentioned above, while also accounting for sparsity on one of the matrix factors. All the derived algorithms are tested on extensive simulated data experiments and real large-scale data applications such as hyperspectral image denoising, matrix completion for recommender systems, music signal decomposition and unsupervised hyperspectral image unmixing with unknown number of endmembers.

The last problem that this thesis touches upon is online low-rank subspace learning and matrix completion. This task follows a different learning model, i.e., online learning, which offers a valuable processing framework when one deals with large-scale streaming data possibly under time-varying conditions. In the thesis, two different online algorithms are put forth. The first one stems from a newly developed online variational Bayes scheme. This is applied for performing approximate inference based on a carefully designed novel multi-hierarchical Bayesian model. Notably, the adopted model encompasses similar low-rank promoting ideas to those mentioned for LRMF. That is, low-rank is imposed via promoting jointly column sparsity on the columns of the matrix factors. However, following the Bayesian rationale, this now takes place by assigning Laplace-type marginal priors on the matrix factors. Going one step further, additional sparsity is independently modelled on the subspace matrix thus imposing multiple structures on the same matrix. The resulting algorithm is fully automated, i.e., it does not demand fine-tuning of any parameters. The second algorithm follows a cost function minimization based strategy. Again, the same low-rank promoting idea introduced for LRMF is incorporated in this problem via the use of a - modified to the online processing scenario - low-rank regularization term. Interestingly, the resulting optimization scheme can be considered as the deterministic analogue of the Bayesian one. Both the proposed algorithms present a favorable feature, i.e., they are competent to learn subspaces without requiring the a priori knowledge of their true rank. Their effectiveness is showcased in extensive simulated data experiments and in online hyperspectral image completion and eigenface learning using real data.

SUBJECT AREA: Signal Processing, Machine Learning

KEYWORDS: structured matrix estimation, nonconvex optimization, large-scale data, online learning

ΠΕΡΙΛΗΨΗ

Το πρόβλημα της εκτίμησης δομημένου πίνακα ανήκει στην κατηγορία των προβλημάτων εύρεσης αναπαραστάσεων χαμηλής διάστασης (low-dimensional embeddings) σε δεδομένα υψηλής διάστασης. Στις μέρες μας συναντάται σε μια πληθώρα εφαρμογών που σχετίζονται με τις ερευνητικές περιοχές της επεξεργασίας σήματος και της μηχανικής μάθησης. Στην παρούσα διατριβή προτείνονται νέοι μαθηματικοί φορμαλισμοί σε τρία διαφορετικά προβλήματα εκτίμησης δομημένων πινάκων από δεδομένα μεγάλης κλίμακας. Πιο συγκεκριμένα, μελετώνται τα ερευνητικά προβλήματα α) της εκτίμησης πίνακα που είναι ταυτόχρονα αραιός, χαμηλού βαθμού και μη-αρνητικός, β) της παραγοντοποίησης πίνακα χαμηλού βαθμού, και γ) της ακολουθιακής (online) εκτίμησης πίνακα υποχώρου (subspace matrix) χαμηλού βαθμού από ελλιπή δεδομένα. Για όλα τα προβλήματα αυτά προτείνονται καινοτόμοι και αποδοτικοί αλγόριθμοι βελτιστοποίησης (optimization algorithms). Βασική υπόθεση που υιοθετείται σε κάθε περίπτωση είναι πως τα δεδομένα έχουν παραχθεί με βάση ένα γραμμικό μοντέλο. Το σύνολο των προσεγγίσεων που ακολουθούνται χαρακτηρίζονται από μη-κυρτότητα. Όπως γίνεται φανερό στην παρούσα διατριβή, η ιδιότητα αυτή, παρά τις δυσκολίες που εισάγει στην θεωρητική τεκμηρίωση των προτεινόμενων μεθόδων (σε αντίθεση με τις κυρτές προσεγγίσεις στις οποίες η θεωρητική ανάλυση είναι σχετικά ευκολότερη), οδηγεί σε σημαντικά οφέλη όσον αφορά την απόδοσή τους σε πλήθος πραγματικών εφαρμογών.

Για την εκτίμηση πίνακα που είναι ταυτόχρονα αραιός, χαμηλού βαθμού και μη-αρνητικός, προτείνονται στην παρούσα διατριβή τρεις νέοι αλγόριθμοι, από τους οποίους οι δύο πρώτοι ελαχιστοποιούν μια κοινή συνάρτηση κόστους και ο τρίτος μια ελαφρώς διαφορετική συνάρτηση κόστους. Κοινό χαρακτηριστικό και των δύο αυτών συναρτήσεων είναι ότι κατά βάση αποτελούνται από έναν όρο προσαρμογής στα δεδομένα και δύο όρους κανονικοποίησης, οι οποίοι χρησιμοποιούνται για την επιβολή αραιότητας και χαμηλού βαθμού, αντίστοιχα. Στην πρώτη περίπτωση αυτό επιτυγχάνεται με την αξιοποίηση του αθροίσματος της επανασταθμισμένης ℓ_1 νόρμας (reweighted ℓ_1 norm) και της επανασταθμισμένης πυρηνικής νόρμας (reweighted nuclear norm), οι οποίες ευθύνονται για το μη-κυρτό χαρακτήρα της προκύπτουσας συνάρτησης κόστους. Από τους δύο προτεινόμενους αλγορίθμους που ελαχιστοποιούν τη συνάρτηση αυτή, ο ένας ακολουθεί τη μέθοδο καθόδου σταδιακής εγγύτητας και ο άλλος βασίζεται στην πιο απαιτητική υπολογιστικά μέθοδο ADMM. Η δεύτερη συνάρτηση κόστους διαφοροποιείται σε σχέση με την πρώτη καθώς χρησιμοποιεί μια προσέγγιση παραγοντοποίησης για τη μοντελοποίηση του χαμηλού βαθμού του δομημένου πίνακα. Επιπλέον, λόγω της μη εκ των προτέρων γνώσης του πραγματικού βαθμού, ενσωματώνει έναν όρο επιβολής χαμηλού βαθμού, μέσω της μη-κυρτής έκφρασης που έχει προταθεί ως ένα άνω αυστηρό φράγμα της (κυρτής) πυρηνικής νόρμας (σ.σ. στο εξής θα αναφέρεται ως εναλλακτική μορφή της πυρηνικής νόρμας). Και στην περίπτωση αυτή, το πρόβλημα που προκύπτει είναι μη-κυρτό λόγω του φορμαλισμού του μέσω της παραγοντοποίησης πίνακα, ενώ η βελτιστοποίηση πραγματοποιείται εφαρμόζοντας μια υπολογιστικά αποδοτική μέθοδο καθόδου συνιστωσών ανά μπλοκ (block coordinate descent). Το σύνολο των προτεινόμενων σχημάτων χρησιμοποιείται για τη μοντελοποίηση, με καινοτόμο τρόπο, του προβλήματος φασματικού διαχωρισμού υπερφασματικών εικόνων (ΥΦΕ). Όπως εξηγείται αναλυτικά, τόσο η αραιότητα όσο και ο χαμηλός βαθμός παρέχουν πολύτιμες ερμηνείες ορισμένων φυσικών χαρακτηριστικών των

ΥΦΕ, όπως π.χ. η χωρική συσχέτιση. Πιο συγκεκριμένα, η αραιότητα και ο χαμηλός βαθμός μπορούν να υιοθετηθούν ως δομές στον πίνακα αφθονίας (abundance matrix - ο πίνακας που περιέχει τα ποσοστά παρουσίας των υλικών στην περιοχή που απεικονίζει κάθε εικονοστοιχείο). Τα σημαντικά πλεονεκτήματα που προσφέρουν οι προτεινόμενες τεχνικές, σε σχέση με ανταγωνιστικούς αλγορίθμους, αναδεικνύονται σε ένα πλήθος διαφορετικών πειραμάτων που πραγματοποιούνται τόσο σε συνθετικά όσο και σε αληθινά υπερφασματικά δεδομένα.

Στο πλαίσιο της παραγοντοποίησης πίνακα χαμηλού βαθμού (low-rank matrix factorization) περιγράφονται στη διατριβή τέσσερις νέοι αλγόριθμοι, ο καθένας εκ των οποίων έχει σχεδιαστεί για μια διαφορετική έκφανση του συγκεκριμένου προβλήματος. Όλα τα προτεινόμενα σχήματα έχουν ένα κοινό χαρακτηριστικό: επιβάλλουν χαμηλό βαθμό στους πίνακες-παράγοντες καθώς και στο γινόμενό τους με την εισαγωγή ενός νέου όρου κανονικοποίησης. Ο όρος αυτός προκύπτει ως μια γενίκευση της εναλλακτικής έκφρασης της πυρηνικής νόρμας με τη μετατροπή της σε σταθμισμένη μορφή. Αξίζει να επισημανθεί πως με κατάλληλη επιλογή των πινάκων στάθμισης καταλήγουμε σε μια ειδική έκφραση της συγκεκριμένης νόρμας η οποία ανάγει την διαδικασία επιβολής χαμηλού βαθμού σε αυτή της από κοινού επιβολής αραιότητας στις στήλες των δύο πινάκων. Όπως αναδεικνύεται αναλυτικά, η ιδιότητα αυτή είναι πολύ χρήσιμη ιδιαιτέρως σε εφαρμογές διαχείρισης δεδομένων μεγάλης κλίμακας. Στα πλαίσια αυτά μελετώνται τρία πολύ σημαντικά προβλήματα στο πεδίο της μηχανικής μάθησης και συγκεκριμένα αυτά της αποθορυβοποίησης σήματος (denoising), πλήρωσης πίνακα (matrix completion) και παραγοντοποίησης μη-αρνητικού πίνακα (nonnegative matrix factorization). Χρησιμοποιώντας τη μέθοδο ελαχιστοποίησης άνω φραγμάτων συναρτήσεων διαδοχικών μπλοκ (block successive upper bound minimization) αναπτύσσονται τρεις νέοι επαναληπτικά σταθμισμένοι αλγόριθμοι τύπου Newton, οι οποίοι σχεδιάζονται κατάλληλα, λαμβάνοντας υπόψη τα ιδιαίτερα χαρακτηριστικά του εκάστοτε προβλήματος. Τέλος, παρουσιάζεται αλγόριθμος παραγοντοποίησης πίνακα ο οποίος έχει σχεδιαστεί πάνω στην προαναφερθείσα ιδέα επιβολής χαμηλού βαθμού, υποθέτοντας παράλληλα αραιότητα στον ένα πίνακα-παράγοντα. Η επαλήθευση της αποδοτικότητας όλων των αλγορίθμων που εισάγονται γίνεται με την εφαρμογή τους σε εκτεταμένα συνθετικά πειράματα, όπως επίσης και σε εφαρμογές πραγματικών δεδομένων μεγάλης κλίμακας π.χ. αποθορυβοποίηση ΥΦΕ, πλήρωση πινάκων από συστήματα συστάσεων (recommender systems) ταινιών, διαχωρισμός μουσικού σήματος και τέλος μη-επιβλεπόμενος φασματικός διαχωρισμός.

Το τελευταίο πρόβλημα το οποίο διαπραγματεύεται η παρούσα διατριβή είναι αυτό της ακολουθιακής εκμάθησης υποχώρου χαμηλού βαθμού και της πλήρωσης πίνακα. Το πρόβλημα αυτό εδράζεται σε ένα διαφορετικό πλαίσιο μάθησης, την επονομαζόμενη ακολουθιακή μάθηση, η οποία αποτελεί μια πολύτιμη προσέγγιση σε εφαρμογές δεδομένων μεγάλης κλίμακας, αλλά και σε εφαρμογές που λαμβάνουν χώρα σε χρονικά μεταβαλλόμενα περιβάλλοντα. Στην παρούσα διατριβή προτείνονται δύο διαφορετικοί αλγόριθμοι, ένας μπεϋζιανός και ένας ντετερμινιστικός. Ο πρώτος αλγόριθμος προκύπτει από την εφαρμογή μιας καινοτόμου ακολουθιακής μεθόδου συμπερασμού βασισμένου σε μεταβολές. Αυτή η μέθοδος χρησιμοποιείται για την πραγματοποίηση προσεγγιστικού συμπερασμού στο προτεινόμενο ιεραρχικό μπεϋζιανό μοντέλο. Αξίζει να σημειωθεί πως το μοντέλο αυτό έχει σχεδιαστεί με κατάλληλο τρόπο έτσι ώστε να ενσωματώνει, σε πιθανοτικό πλαί-

σιο, την ίδια ιδέα επιβολής χαμηλού βαθμού που προτείνεται για το πρόβλημα παραγοντοποίησης πίνακα χαμηλού βαθμού, δηλαδή επιβάλλοντας από-κοινού αραιότητα στους πίνακες-παράγοντες. Ωστόσο, ακολουθώντας την πιθανοτική προσέγγιση, αυτό πραγματοποιείται επιβάλλοντας πολύ-επίπεδες a priori κατανομές Laplace στις στήλες τους. Ο αλγόριθμος που προκύπτει είναι πλήρως αυτοματοποιημένος, μιας και δεν απαιτεί τη ρύθμιση κάποιας παραμέτρου κανονικοποίησης. Ο δεύτερος αλγόριθμος προκύπτει από την ελαχιστοποίηση μιας κατάλληλα διαμορφωμένης συνάρτησης κόστους. Και στην περίπτωση αυτή, χρησιμοποιείται η προαναφερθείσα ιδέα επιβολής χαμηλού βαθμού (κατάλληλα τροποποιημένη έτσι ώστε να μπορεί να εφαρμοστεί στο ακολουθιακό πλαίσιο μάθησης). Ενδιαφέρον παρουσιάζει το γεγονός πως ο τελευταίος αλγόριθμος μπορεί να θεωρηθεί ως μια ντετερμινιστική εκδοχή του προαναφερθέντος πιθανοτικού αλγορίθμου. Τέλος, σημαντικό χαρακτηριστικό και των δύο αλγορίθμων είναι ότι δεν είναι απαραίτητη η εκ των προτέρων γνώση του βαθμού του πίνακα υποχώρου. Τα πλεονεκτήματα των προτεινόμενων προσεγγίσεων παρουσιάζονται σε ένα μεγάλο εύρος πειραμάτων που πραγματοποιήθηκαν σε συνθετικά δεδομένα, στο πρόβλημα της ακολουθιακής πλήρωσης ΥΦΕ και στην εκμάθηση ιδιο-προσώπων κάνοντας χρήση πραγματικών δεδομένων.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Επεξεργασία Σήματος, Μηχανική Μάθηση

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: εκτίμηση δομημένου πίνακα, μη-κυρτή βελτιστοποίηση, δεδομένα μεγάλης κλίμακας, ακολουθιακή επεξεργασία

To Marina and Antigone...

ACKNOWLEDGEMENTS

I would like to give my foremost thanks to my supervisors Dr. Athanasios Rontogiannis and Dr. Konstantinos Koutroumbas. Their immense knowledge and wealth of ideas have made working with them a wonderful experience for me. Their advising style and the patience in working out details have motivated me to overcome all the difficulties I met during these fruitful PhD years. I am deeply indebted to them not only for their scientific guidance but also for their constant support and encouragement.

I am also grateful to Professor Sergios Theodoridis whose inspiring lectures during my Master's studies have cultivated my strong interest in the fields of signal processing and machine learning.

I especially thank my close collaborator Dr. Kostas Themelis to whom I accord credit for his contribution to a part of the research work of this thesis. My work has also been benefited by collaborating with Dr. Olga Sykioti and Dr. Spyridoula Xenaki to whom I owe special thanks.

I also thank Dr. Eleni Charou who motivated me to embark on academic research during the elaboration of my Master's thesis in NCSR Demokritos.

Throughout the five years in National Observatory of Athens, I am fortunate enough to have had great colleagues and friends and I would like to thank all of them.

Finally, my heartfelt gratitude goes to my family and in particular to my wife Marina for the invaluable support that she gave me all of these years.

LIST OF PUBLICATIONS

Book Chapter

1. P. V. Giampouras, K. E. Themelis, A. A. Rontogiannis, K. D. Koutroumbas, “Structured Abundance Matrix Estimation for Land Cover Hyperspectral Image Unmixing,” *Compressive Sensing of Earth Observations*, C.H. Chen (Editor), CRC Press Signal and Image Processing of Earth Observations Book Series, 2017.

Refereed Journal Papers

1. P. V. Giampouras, A. A. Rontogiannis, K. D. Koutroumbas, “Alternating Iteratively Reweighted Least Squares Minimization for Low-Rank Matrix Factorization,” *IEEE Transactions on Signal Processing*, under review, 2018.

2. I. C. Tsaknakis, P. V. Giampouras, A. A. Rontogiannis, K. D. Koutroumbas, “A Computationally Efficient Tensor Completion Algorithm,” (to appear) *IEEE Signal Processing Letters*, 2018.

3. P. V. Giampouras, A. A. Rontogiannis, K. E. Themelis, K. D. Koutroumbas, “Online Sparse and Low-Rank Subspace Learning from Incomplete Data: A Bayesian View,” *Signal Processing*, vol. 137, pp. 199-212, Aug. 2017.

4. P. V. Giampouras, K. E. Themelis, A. A. Rontogiannis, K. D. Koutroumbas, “Simultaneously Sparse and Low-Rank Abundance Matrix Estimation for Hyperspectral Image Unmixing,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, issue.8, pp. 4775-4789, Aug. 2016.

Refereed Conference Papers

1. P. V. Giampouras, A. A. Rontogiannis, K. D. Koutroumbas, “Robust PCA via Alternatingly Iteratively Reweighted Low-Rank Matrix Factorization,” to be presented in the *IEEE International Conference on Image Processing (ICIP)*, Athens, Oct. 2018.

2. P. V. Giampouras, A. A. Rontogiannis , K. D. Koutroumbas, “Low-rank and Sparse NMF for Joint Endmembers’ Number Estimation and Blind Unmixing of Hyperspectral Images,” In proceedings of the *25th European Signal Processing Conference (EUSIPCO)*, Kos Island, Sept. 2017.

3. P. V. Giampouras, A. A. Rontogiannis, K. D. Koutroumbas, “ℓ_1/ℓ_2 Regularized Nonconvex Low-rank Matrix Factorization,” In proceedings of the *Signal Processing with Adaptive Sparse Structured Representations (SPARS) Workshop*, Lisbon, June 2017.

4. P. V. Giampouras , A. A. Rontogiannis, K. D. Koutroumbas, “Online Low-Rank Subspace Learning from Incomplete Data Using Rank Revealing ℓ_1/ℓ_2 Regularization,” In proceedings of the *IEEE Statistical Signal Processing Workshop (SSP)*, Palma de Mallorca, June 2016.

5. P. V. Giampouras, A. A. Rontogiannis, K. E. Themelis, K. D. Koutroumbas, “Online Bayesian Low-Rank Subspace Learning from Partial Observations,” In proceedings of the *23th European Signal Processing Conference (EUSIPCO)*, Nice, Sept. 2015.

6. P. V. Giampouras, A. A. Rontogiannis, K. D. Koutroumbas, K. E. Themelis, “A Sparse Reduced-Rank Regression Approach for Hyperspectral Image Unmixing,” In proceedings of the *3rd International Workshop on Compressive Sensing Theory and its Applications in Radar, Sonar and Remote Sensing (CoSeRa)*, Pisa, June 2015. (2nd Best Student Paper Award winning paper)

7. P. V. Giampouras, K. E. Themelis, A. A. Rontogiannis, K. D. Koutroumbas, “Hyperspectral Image Unmixing via Simultaneously Sparse and Low-Rank Abundance Matrix Estimation,” In proceedings of the *7th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, Tokyo, June 2015.

8. P. V. Giampouras, K. E. Themelis, A. A. Rontogiannis, K. D. Koutroumbas, “A Variational Bayes Algorithm for Joint-Sparse Abundance Estimation,” In proceedings of the *6th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, Lausanne, June 2014.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

PREFACE

This research has been conducted at the facilities of the Institute for Astronomy, Astrophysics, Space Applications and Remote Sensing (IAASARS) of the National Observatory of Athens (NOA).

It has been financed in part by the European Union (European Social Fund - ESF) and Greek national funds through a) the Operational Programme "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: ARISTEIA- HSI-MARS-1413, b) the Operational Programme "Competitiveness, Entrepreneurship and Innovation" (NSRF 2014-2020) (European Regional Development Fund): PROTEAS II - Advanced Space Applications for Exploring the Universe of Space and Earth (MIS 5002515), which is implemented under the "Action Reinforcement of the Research and Innovation Infrastructure" and, finally, c) by the PHySIS project (http://www.physis-project.eu/), contract no. 640174, within the H2020 Framework Program of the European Commission.

1. INTRODUCTION

Nowadays, there exists a flood of information generated by different sources such as web services, sensor networks, remote sensing applications, communication networks, biological systems, etc., [69]. The generated data may be *big* and *high-dimensional* thus rendering the respective processing and estimation tasks quite intriguing and challenging. Modern learning tools shall efficiently handle intrinsic barriers and benefit out of using a certain amount of prior knowledge that may be available. In that vein, parsimonious models and representations of data have been shown up lately. Structured matrix estimation is at the heart of various applications of that kind, that are commonly addressed in signal processing and machine learning literature. Put simply, structured matrix estimation refers to the task of recovering a matrix (or more than one matrices) that is characterized by specific structure(s) such as sparsity, group-sparsity, low-rankness, by suitably processing the data at hand. In this introductory chapter, we present the main structured matrix estimation tasks that are addressed in this thesis. In this regard, the original formulations of *supervised* problems, concerning the estimation of sparse or low-rank matrices from linear measurements, are provided first. Next, the more challenging *simultaneously* sparse, low-rank and nonnegative matrix estimation task is defined. Finally, the matrix factorization problem is discussed and insight is provided on the disparate variants of this approach (e.g., dictionary learning, sparse PCA, etc.), which arise when additional structures on the matrix factors are imposed. After formulating the problems of interest, the most relevant methods reported in the literature are overviewed. The chapter concludes with the presentation of the contribution of the thesis and a brief description of the content of each chapter.

1.1 Structured matrix estimation: an overview

The task of estimating structured matrices encapsulates a diversity of machine learning problems such as sparse matrix regression, reduced-rank regression, low-rank matrix factorization, to name just a few. In all cases, the successful accomplishment of this task calls for the adoption of specific hypotheses. The most important one concerns the “mechanism” which conveys information as to the underlying phenomena that determine the data generation process. In this thesis, the main hypothesis that embraces all the algorithms to be presented next is that data are generated following a linear model. In mathematical terms, this can be stated as follows,

$$\mathbf{Y} = \mathbf{A}\mathbf{X}. \tag{1.1}$$

where **Y** denotes a $l \times n$ matrix which contains the data, **X** is an $m \times n$ matrix that we seek to estimate and **A** is a $l \times m$ matrix which linearly maps **X** to **Y**. Eq. (1.1) gives rise to two large classes of a) supervised and b) unsupervised problems. The distinction between these two classes can be roughly described as follows: if both **A** and **Y** are known the problem is called *supervised*, whereas if one knows only the data matrix **Y**, then the

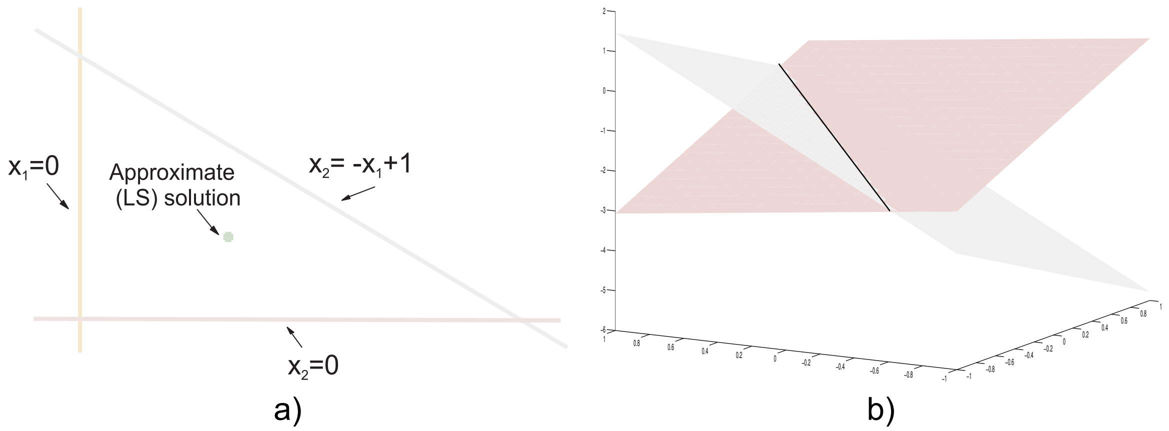


Figure 1: The geometry of a) an overdetermined and b) an underdetermined linear system. a) An overdetermined system in a 2d-space corresponding to three lines. Obviously, there is no exact solution since there is no common intersection point of the three lines. In that case, an approximate solution may be obtained based on a certain criterion (e.g. least-squares). b) The intersection of the two 2d-hyperplanes in a 3d-space is depicted by the black line segment. All points belonging to this line are candidate solutions of this underdetermined linear system.

problem is *unsupervised*. The former problem calls for finding an efficient way to estimate matrix **X** given **Y** and **A**, while the latter requires the estimation of both matrices **A** and **X** given **Y**. However, albeit the linearity assumption may enhance mathematical tractability, there do exist inherent obstacles in the estimation process that one must get around.

Let us first focus on the easiest case, i.e., the supervised scenario. By (1.1) it appears that we shall deal with a system of linear equations. A crucial point that hence arises is that of the relation between the dimensions l and m of matrix **A**. Concretely, the system of equations may be either *overdetermined* if $l > m$ (see Fig. 1a for an intuitive geometrical view of an overdetermined[1] system of linear equations) or *underdetermined* if $l < m$ (see Fig 1b). Both aforementioned cases give rise to *ill-posed* problems, that is, to problems that do not have either any solution at all (in case of *inconsistent* overdetermined systems) or a unique solution (underdetermined systems). The situation, in terms of the ill-posed nature of the problem, is even harsher in the unsupervised scenario, i.e., the matrix factorization problem. This is so, since there exist infinitely many pairs of matrices $\mathbf{A}, \mathbf{X}$ that produce **Y**, i.e., for any $\bar{\mathbf{A}} = \mathbf{AW}$ and $\bar{\mathbf{X}} = \mathbf{W}^{-1}\mathbf{X}$ we have $\bar{\mathbf{A}}\bar{\mathbf{X}} = \mathbf{AWW}^{-1}\mathbf{X} = \mathbf{AX}$ (assuming any invertible square matrix **W**), which is known as *change of basis ambiguity*. That being said, we are confronted with problems which, in most cases, necessitate the inclusion of prior knowledge in an effort to find "decent" approximate solutions. The rationale behind this strategy is the following: since there is no either exact or unique solution when it comes to the matrix that we wish to estimate by solving the system of linear equations defined in (1.1), either find an approximate solution or restrict the solution set, based on the a priori knowledge we have regarding the structure of the sought matrix[2]. The word "structure"

[1] Note that for a linear system to be overdetermined, it is not sufficient to have $l > m$. In fact, this property depends on the relation between the ranks of the coefficient and the augmented matrices, i.e., **A** and [**A** | **Y**], respectively.

[2] The linear model assumption is also a sort of prior knowledge that has been used for modelling the data

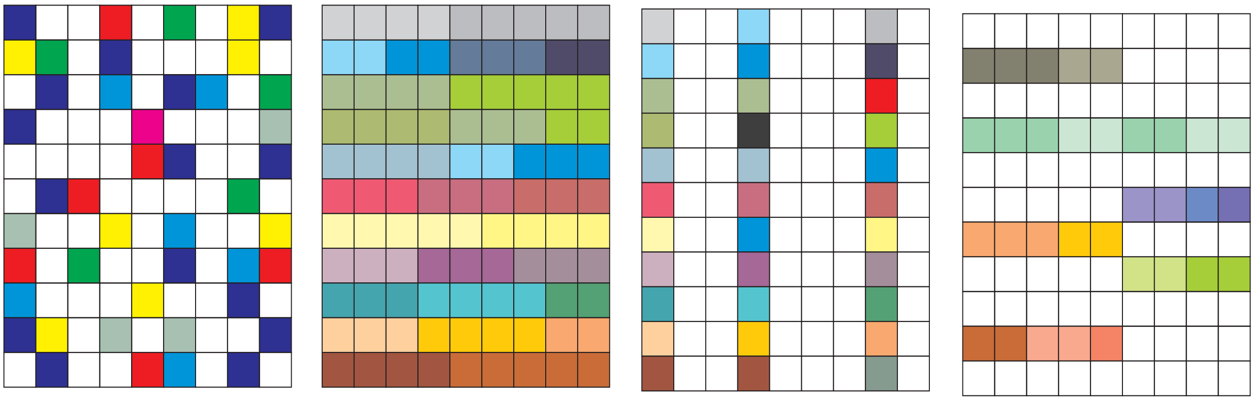


Figure 2: Examples of structured matrices. A sparse, a low-rank, a column-sparse and a simultaneously sparse and low-rank matrix (seen from left to right). White cells correspond to zero values.

refers to particular characteristics that a matrix may convey, subject to the physics of the problem that is addressed each time. For instance, a matrix may be *sparse*, that is, it consists of a number of zero entries. Matrices might also have a specific sparsity pattern, e.g., column/row sparsity. Alternatively, the matrix columns/rows may be highly correlated. In such case, a low-rank matrix structure can be assumed. Furthermore, a matrix may be simultaneously sparse and low-rank. Examples of such matrix structures are shown in Fig. 2. Finally, there exist many applications in which the matrix elements are bounded to a specific range of values, e.g., the set of nonnegative reals. That said, we next describe the problems of a) sparse, low-rank or nonnegative matrix estimation, b) simultaneously sparse, low-rank and nonnegative matrix estimation and c) sparse and low-rank matrix factorization, that revolve around this thesis.

1.1.1 Sparse and low-rank matrix estimation

The problem of matrix estimation/regression imposing either a sparsity or low-rank or nonnegativity constraint belongs to the general class of supervised problems. Next, we focus on the linear measurements' case, i.e., data are assumed to be generated following a linear process (as given in (1.1)), and formally define these problems.

Let us begin with a formal statement of *sparse matrix estimation*. Given $\mathbf{Y}$ and $\mathbf{A}$, the sparse matrix $\mathbf{X}$ is recovered by solving the following minimization problem,

$$\min_{\mathbf{X}\in\mathcal{R}^{m\times n}} \text{card}\{\text{supp}(\mathbf{X})\} \quad \text{subject to} \quad \mathbf{Y} = \mathbf{AX}, \tag{1.2}$$

where $\text{supp}(\mathbf{X})$ is the support set[3] of matrix $\mathbf{X}$ and $\text{card}\{\cdot\}$ returns the cardinality of a set[4]. It can be readily shown that $\text{card}\{\text{supp}(\mathbf{X})\}$ is identical to the ℓ_0 quasinorm. The problem in (1.2), as a direct generalization from vectors to matrices of the well-studied problem of

generation process.

[3] The set that contains the indexes of the elements of $\mathbf{X}$ corresponding to nonzero values.

[4] The number of elements in the set.

sparse regression, shares the same issues with the latter in terms of the computational complexity and the recovery guarantees. Concretely ℓ_0 quasinorm minimization is NP-hard and uniqueness of the solution does not generally hold, [53]. For the latter, it has been shown that if certain conditions are satisfied by the measurements matrix **A** based on its mutual coherence, spark, etc., [48], theoretical recovery guarantees can be provided. In such cases, based on various relaxations, e.g., the ℓ_1 norm (that will be elaborated in Section 1.2), polynomial time algorithms have been developed for making the problem tractable.

Departing from sparsity, we next move onto *affine rank minimization* (ARM), i.e., the task of recovering low-rank matrices from linear measurements, [121]. ARM is analogous to the sparse matrix estimation problem (1.2) and is mathematically formulated as,

$$\min_{\mathbf{X}\in\mathcal{R}^{m\times n}} \text{rank}(\mathbf{X}) \quad \text{subject to} \quad \mathbf{Y} = \mathbf{A}\mathbf{X}. \tag{1.3}$$

This analogy can be illustrated by considering the singular value decomposition (SVD) of matrix **X** (assuming $m \leq n$),

$$\mathbf{X} = \sum_{i=1}^{m} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T, \tag{1.4}$$

where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_m \geq 0$ are the singular values of **X** and $\boldsymbol{u}_i \in \mathcal{R}^m$, $\boldsymbol{v}_i \in \mathcal{R}^n$ are the left and right singular vectors, respectively. Matrix **X** is of rank d if and only if the vector of singular values, $\boldsymbol{\sigma}(\mathbf{X}) = [\sigma_1, \sigma_2, \dots, \sigma_m]^T$, is d-sparse, i.e., card$\{$supp$(\boldsymbol{\sigma}(\mathbf{X}))\} \equiv \|\boldsymbol{\sigma}(\mathbf{X})\|_0 = d$, where $\|\cdot\|_0$ denotes the ℓ_0 quasinorm. Problem (1.3) is tantamount to solving the ℓ_0 minimization problem on the singular values of **X** and is also NP-hard. To this end various relaxation schemes have come to the scene in literature in an effort to render the problem tractable. In that spirit the so-called nuclear norm, which is actually the ℓ_1 norm of the vector of singular values, as well as a wealth of other approaches have been extensively utilized in the literature, [51], and will be detailed in the next section.

Interestingly, even by solving the convex relaxations of ℓ_0 quasinorm and rank function minimization problems, using the ℓ_1 and the nuclear norm, respectively, exact or approximate recovery of low-rank matrices can be obtained. In particular, exact recovery is theoretically guaranteed if specific conditions are satisfied by the measurements matrix **A**, such as the restricted isometry property and the nullspace property, [53].

1.1.2 Simultaneously sparse, low-rank and nonnegative matrix estimation

Having described the sparse and low-rank matrix estimation tasks, we now stick to the more challenging case of estimating multiple structured matrices out of linear measurements. Concretely, we focus on the intriguing task of recovering matrices that are *at the same time* sparse, low-rank and nonnegative. It is worth noticing that while sparse matrix estimation and low-rank matrix estimation are two distinct problems that have been extensively studied in the literature during the last two decades, this is not the case when both sparsity and low-rank constraints are simultaneously imposed on the matrix estimation problem. In addition, this task has become essential to many applications lately.

Focusing on a specific set of these applications, we restrict our attention to those that concern nonnegative data values. All in all, the problems that we address can be formulated as follows,

$$\min_{\mathbf{X}\in\mathcal{R}_{+}^{m\times n}} \lambda\text{card}\{\text{supp}(\mathbf{X})\} + \gamma\text{rank}(\mathbf{X}) \quad \text{subject to} \quad \mathbf{Y} = \mathbf{AX}, \tag{1.5}$$

where $\mathcal{R}_{+}^{m\times n}$ denotes the $m \times n$ dimensional nonnegative orthant of real numbers and $\lambda \geq 0$ and $\gamma \geq 0$ are parameters that are used for controlling the contribution of each one of the two terms in the sum. Again, Problem (1.5) is intractable since it derives from the combination of problems (1.2) and (1.3), which are both NP-hard, [51]. To this end, both convex (e.g., ℓ_1 and nuclear norm) and tractable nonconvex approaches (e.g., ℓ_p and Schatten-p -with $0 < p < 1$- quasinorms), have come to the scene that will be elucidated in the sequel.

1.1.3 Sparse, low-rank and nonnegative matrix factorization

Matrix factorization (MF) has long been at the core of numerous machine learning problems that have been studied in the literature. Among other applications, matrix factorization has been utilized as an efficient reformulation of the matrix rank minimization problems offering scalability with respect to the data size (see Fig. 3) and, thus, giving rise to efficient low computational complexity solvers. Since it is an inherently ill-posed problem owing to the invariances it presents, a number of constraints have been used for transforming it to a computationally tractable one. Along these lines, constraints such as sparsity and low-rankness have been included in a slew of optimization problems. These involve either one or both the unknown matrix factors. Interestingly, each constraint models a specific problem. In the following, it is assumed that matrix **A** is the $m \times m$ identity matrix $\mathbf{I}_m$ and matrix **X** is factorized as $\mathbf{X} = \mathbf{UV}^T$, where $\mathbf{U} \in \mathcal{R}^{m\times d}$ and $\mathbf{V} \in \mathcal{R}^{n\times d}$. To provide an insight on this we next list some of the cases that have gained extreme popularity over the last few years.

Dictionary learning (DL). DL refers to an omnipresent task in machine learning whose goal is to find a linear subspace called *dictionary* in which training data admit a sparse representation. DL has been used in a variety of problems ranging from compressed sensing to blind source separation. In mathematical terms, let us assume that the data at hand are contaminated by additive zero-mean Gaussian i.i.d. noise **E**, that is, they may be modelled as $\mathbf{Y} = \mathbf{X} + \mathbf{E} \equiv \mathbf{UV}^T + \mathbf{E}$, where the columns of **U** define the dictionary. Then, DL is formulated as the following optimization problem

$$\min_{\mathbf{U}\in\mathcal{R}^{m\times d}, \mathbf{V}\in\mathcal{R}^{n\times d}} \|\mathbf{Y} - \mathbf{UV}^T\|_F^2 \quad \text{subject to} \quad \text{card}\{\text{supp}(\mathbf{V})\} \leq \kappa, \quad \|\boldsymbol{u}_i\|_2 \leq 1 \;\; \forall i = 1, 2, \ldots, d \tag{1.6}$$

where $\|\cdot\|_F$ and $\|\cdot\|_2$ denote the Frobenius and the ℓ_2 norm of a matrix and a vector,

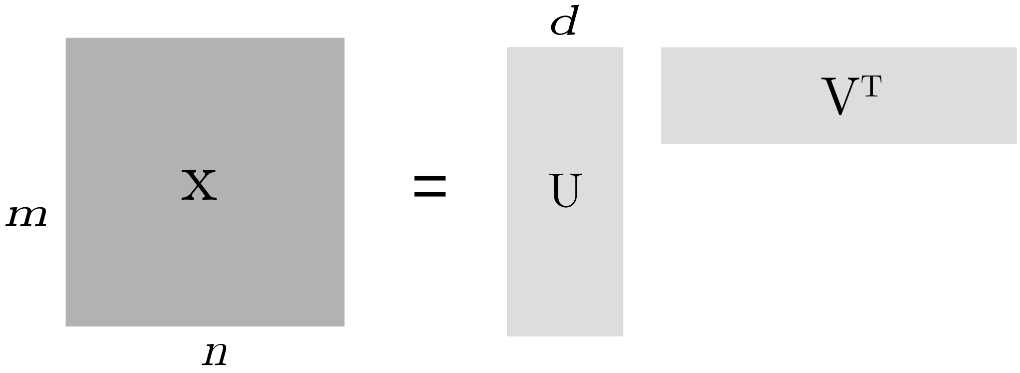


Figure 3: A $m \times n$ matrix X of rank $d \leq (m,n)$ can be factorized using two smaller matrices, i.e., an $m \times d$ matrix U and an $n \times d$ matrix V, thus offering significant computational merits.

respectively, and κ is a cardinal number, while $\boldsymbol{u}_i$ is the ith column vector of matrix **U**. Note that the constraint $\|\boldsymbol{u}_i\|_2 \leq 1$ is utilized for resolving scaling ambiguities.

Sparse principal component analysis (sPCA). sPCA is an extension of the principal component analysis (PCA) method, which is mostly used in the framework of dimensionality reduction, where the aim is to identify a low-dimensional orthogonal subspace for representing high-dimensional data. PCA identifies the space where the first axis retains the maximum possible variance of the data, the next axis, which is perpendicular to the previous retains the maximum possible of the remained variance of the data, etc. These axes are defined by the eigenvectors resulting from the solution of a suitably defined eigenvalue/eigenvector problem. Dimensionality reduction is achieved by retaining the (few) axes that capture the maximum variance of the data, [86]. sPCA differs from PCA by accounting also for sparsity on this subspace. In other words, sPCA seeks sparse eigenvectors. By following the so-called synthesis approach, [85], sPCA can be cast as a constrained matrix factorization problem. This way, the problem becomes nonconvex[5]. Moreover the orthogonality constraint is often dropped and the resulting optimization problem is expressed as,

$$\min_{\mathbf{U} \in \mathcal{R}^{m \times d}, \mathbf{V} \in \mathcal{R}^{n \times d}} \|\mathbf{Y} - \mathbf{U}\mathbf{V}^T\|_F^2 \quad \text{subject to} \quad \text{card}\{\text{supp}(\mathbf{U})\} \leq \kappa. \tag{1.7}$$

As it has been shown, [170], this approach is of particular interest in cases where the number of unknown parameters is comparable or even exceeds the number of the data samples.

Matrix completion (MC). Matrix completion refers to the task of recovering a data matrix from a sample of its entries. Clearly, this is impossible unless certain conditions are satisfied. Along these lines, a necessary condition is that there exists a certain degree of correlation in the matrix that we wish to recover. More specifically, the degrees of freedom[6] of the data matrix must be much less than the total number of its available elements.

[5] Convex formulations of sPCA have been also proposed based on the so called *analysis* formulation of the problem, e.g., $\max_{\boldsymbol{u}_i} \boldsymbol{u}_i^T \mathbf{X}^T \mathbf{X} \boldsymbol{u}_i$ subject to $\|\boldsymbol{u}_i\|_2 = 1, \;\; \|\boldsymbol{u}_i\|_1 = \kappa$, [40].

[6] Degrees of freedom refer to the number of free parameters that are needed in order to specify the matrix,

Moreover, the positions of these elements must be uniformly distributed. Matrix completion has been widely formulated as a rank minimization problem, i.e,

$$\min_{\mathbf{X}\in\mathcal{R}^{m\times n}} \text{rank}(\mathbf{X}) \quad \text{subject to} \quad [\mathbf{Y}]_{ij} = [\mathbf{X}]_{ij} \quad \text{for} \quad i,j \in \Omega \tag{1.8}$$

where Ω is the set of indexes of matrix **Y** where information is present. Since the rank minimization problem defined above is NP-hard, various convex and nonconvex relaxation approaches have been proposed in the literature, in order to derive algorithms with polynomial time complexity. Amongst the latter, matrix factorization (MF) formulations have gained popularity lately. These are based on the so-called *Burer-Monteiro* heuristic, i.e., the fact that any rank-r matrix can be written in a factorized form $\mathbf{X} = \mathbf{U}\mathbf{V}^T$ where $\mathbf{U} \in \mathcal{R}^{m\times r}$ and $\mathbf{V} \in \mathcal{R}^{n\times r}$. As explained later, albeit nonconvex problems are generally more difficult to be solved, MF based matrix completion methods present substantial merits as compared to other approaches especially in large volume and/or dimension data (big data), since they significantly decrease the size of the arising optimization problem.

Nonnegative matrix factorization (NMF). NMF is a matrix decomposition technique, which possesses a prominent position in various fields of machine learning and signal processing, where nonnegative data, such as image and video, are involved. This is so, since similarly to sPCA, it offers meaningful interpretable decompositions while at the same time it is a perfect choice for decomposing nonnegative data. Similarly to the DL and sPCA problems, NMF takes the form of a matrix factorization setting. However, it deviates from the others since the matrix factors are now constrained to have only nonnegative values. NMF can be defined as a constrained optimization problem, as is shown below

$$\min_{\mathbf{U}\in\mathcal{R}_{+}^{m\times d}, \mathbf{V}\in\mathcal{R}_{+}^{n\times d}} \|\mathbf{Y} - \mathbf{U}\mathbf{V}^T\|_F^2. \tag{1.9}$$

NMF has sparked a lot of research focusing on the investigation of conditions for the uniqueness (up to positive scaling and permutation) of the derived decompositions. This is an important aspect in many applications such as *blind source separation*.

1.2 Related work

The problems described so far have attracted a great deal of interest in the literature during the past few years. Especially, during the last decade, structured matrix estimation has been at the heart of the research endeavors, which can be classified into two main classes, that is, a) those which focus on theoretical underpinnings of these problems (e.g., recovery guarantees, sample complexity, algorithmic aspects of the problems, etc.) and b) those which focus on the development and application of relevant techniques on various fields such as image and video processing, collaborative filtering, etc. This thesis can be considered as a middle-of-the-road approach. For this reason, the literature review of the related work in the field of interest will selectively refer to both popular theoretical works

[162].

and others that concern diverse aspects emanating from the application of minimization solvers in appropriately formulated practical problems.

1.2.1 Sparse and low-rank matrix estimation

Sparse matrix estimation, as a natural extension of the sparse vector recovery problem, has been at the core of the research endeavors in machine learning and signal processing field during the recent years. This is so, because sparse recovery constitutes the backbone of the field of *compressed sensing*, which has been lately emerged and has a great impact in many groundbreaking and paradigm-shifting applications in the areas of image processing, sampling theory, etc.

As mentioned in Section 1.1.1, the originally formulated problem in (1.2) is NP-hard, [53]. This fact has motivated the proposition of various alternative formulations of the problem[7], which aim at achieving exact or approximate recovery of sparse matrices requiring the minimum possible amount of measurements. ℓ_1 norm minimization was the first effort made towards this direction. ℓ_1 norm is proven to be the convex envelope of the ℓ_0 quasinorm, [48]. Thus, by incorporating this into the optimization problem, the originally NP-hard problem is now relaxed to a tractable convex program called *basis pursuit*, which is formulated as

$$\min_{\mathbf{X}\in\mathcal{R}^{m\times n}} \|\mathbf{X}\|_1 \text{ subject to } \mathbf{Y} = \mathbf{A}\mathbf{X}. \tag{1.10}$$

This optimization problem can be efficiently solved via various off-the-shelf tools, e.g., interior point methods. In an attempt to increase the computational efficiency and generate fast algorithms, algorithmic tools specialized on ℓ_1 minimization subject to linear measurements, have been put forward. In that respect, the homotopy method, the least angle regression (LARS, which resembles a modified version of the greedy type orthogonal matching pursuit (OMP) method), [47], and the primal-dual approaches (see Section 2.1.3) such as the Champolle-Pock's algorithm, [32], have been widely applied.

Sparse vector/matrix recovery has been also seen through an iterative reweighted least squares (IRLS$_p$, with $p \geq 0$ a user defined parameter) approach, [41], i.e.,

$$\min_{\mathbf{X}\in\mathcal{R}^{m\times n}} \|\mathbf{X}\|^2_{F,\mathbf{D}_k} \text{ subject to } \mathbf{Y} = \mathbf{A}\mathbf{X}, \tag{1.11}$$

where

$$\|\mathbf{X}\|^2_{F,\mathbf{D}_k} = \sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij}^2 d_{ij,k}, \tag{1.12}$$

is the weighted Frobenius norm and matrix $\mathbf{D}_k \in \mathcal{R}^{m\times n}$, which is updated at each iteration

[7] Besides the optimization based methods, greedy and thresholding based approaches have also been proposed. However, these methods are out of the scope of this thesis. Interested readers may refer to [53] for a detailed review of these methods.

k, contains the reweighting coefficients $d_{ij,k}$'s defined as

$$d_{ij,k} = (x_{ij,k} + \eta_k^2)^{\frac{p}{2}-1}, \tag{1.13}$$

which are updated at each iteration k. For $\eta_k \neq 0$, IRLS$_p$, contrary to ℓ_1 minimization, offers a smooth formulation of the sparse vector/matrix recovery problem. As shown in [41], for $p = 1$ and under similar to the ℓ_1 minimization problem conditions that must be satisfied by the measurements' matrix **A** (nullspace property), the minimizer obtained by IRLS$_1$ coincides with that of (1.10).

Beyond the ℓ_1 norm, and in an effort to further enhance the recovery performance, nonconvex approaches have been proposed. Along these lines, the ℓ_p norm for $0 < p < 1$ has been utilized giving rise to optimization problems in the form,

$$\min_{\mathbf{X} \in \mathcal{R}^{m \times n}} \|\mathbf{X}\|_p^p \text{ subject to } \mathbf{Y} = \mathbf{AX}. \tag{1.14}$$

As shown in Fig. 4, the ℓ_p quasinorm approximates the ℓ_0 quasinorm as $p \to 0$ and hence is a better surrogate thereof. However, (1.14) is nonconvex and intractable since it belongs to the class of NP-hard problems when it comes to finding a globally optimal solution, [56]. On the other hand, locally optimal solutions can be obtained by polynomial time algorithms, resulting from relaxed versions of the original problem, where ℓ_p norms are adopted. Interestingly, in many practical situations, ℓ_p norm minimization (with $0 < p < 1$) has showcased superior recovery performance of sparse vectors/matrices over the convex ℓ_1 minimization, [33, 124]. In the same nonconvex setting, other approaches have been also proposed such as the weighted ℓ_1 norm minimization defined as

$$\min_{\mathbf{X} \in \mathcal{R}^{m \times n}} \|\mathbf{X}\|_{1,\mathbf{D}} \text{ subject to } \mathbf{Y} = \mathbf{AX}, \tag{1.15}$$

where,

$$\|\mathbf{X}\|_{1,\mathbf{D}} = \sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} |x_{ij}|, \tag{1.16}$$

and $d_{ij} \geq 0$ are the weighting coefficients, which if wisely set ensure a better approximation of the ℓ_0 norm (as compared to plain ℓ_1 minimization). For instance, in the so-called *reweighted* ℓ_1 norm method proposed in [30], the weighting coefficients are updated at iteration $k+1$ based on the value of the respective $x_{ij,k}$ obtained at iteration k and are set as

$$d_{ij,k+1} = \frac{1}{|x_{ij,k}| + \eta^2} \tag{1.17}$$

where $\eta \neq 0$ ensures that the denominator does not vanish. Note that the reweighted ℓ_1 scheme is related to a majorization-minimization approach (see Section 2.1.4.2) which arises by subsuming a first-order Taylor expansion of the concave log$(\cdot)$ function on **X**. Interestingly, reweighted ℓ_1 can be also viewed as a special instance of the reweighted

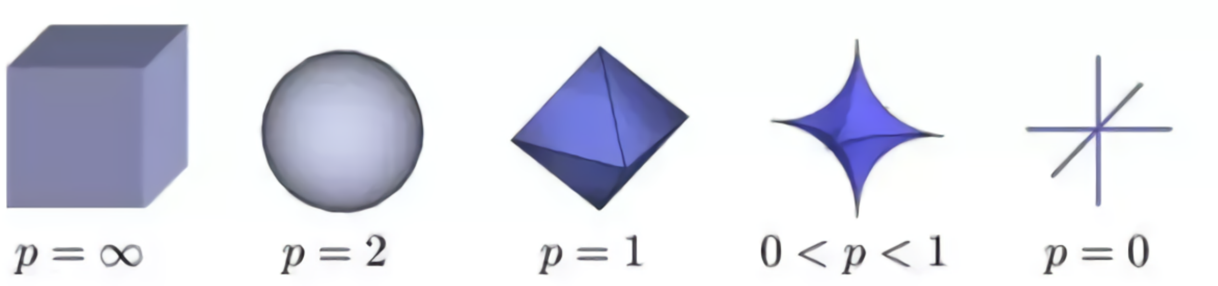


Figure 4: ℓ_p norm unit balls for different values of p in a 2-dimensional space (A unit ball $\mathcal{B}_1$ of an ℓ_p norm centered at 0 in the n-dimensional Euclidean space $\mathcal{R}^n$, is defined as $\mathcal{B}_1 = \{\mathbf{x} \in \mathcal{R}^n : \|\mathbf{x}\|_p \leq 1\}$), [1].

Frobenius norm defined in (1.12), which arises for $p = 0$ in the weighting coefficients given in (1.13), [34].

As explained in Section 1.1.1, the low-rank matrix estimation problem draws strong parallels with the sparse vector/matrix recovery problem. Hence, it is obvious that the different formulations of the affine rank minimization problem proposed in the literature, are based on similar premises, all aiming at finding tractable alternatives to the rank minimization problem. In that respect, the nuclear norm, which is the convex envelope of the rank, [51], has been utilized as a proxy of the NP-hard rank minimization task. In an effort to better approximate the rank, the Schatten-p quasinorm defined as

$$\|\mathbf{X}\|_{\mathcal{S}_p} = \|\boldsymbol{\sigma}(\mathbf{X})\|_p, \tag{1.18}$$

with $0 < p \leq 1$, has been proposed. For $p = 1$, the Schatten-p quasinorm reduces to the nuclear norm $\|\mathbf{X}\|_*$. Schatten-p quasinorms have played a significant role in numerous cases, [111], involving relaxation of the rank minimization problem of (1.3) expressed as follows

$$\min_{\mathbf{X} \in \mathcal{R}^{m \times n}} \|\mathbf{X}\|_{\mathcal{S}_p}^p \quad \text{subject to} \quad \mathbf{A}\mathbf{X} = \mathbf{Y}. \tag{1.19}$$

Nowadays, Schatten-p quasinorm based minimization has been seen via a more intriguing perspective, i.e., using an iterative reweighting approach. In this vein, inspired by the previously mentioned IRLS$_p$ method for imposing sparsity, in [107, 52] the authors propose to minimize a *reweighted* version of the Frobenius norm. The equivalence of the Schatten-p quasinorm and those minimized in [107, 52] is mathematically expressed as follows,

$$\begin{aligned} \|\mathbf{X}\|_{\mathcal{S}_p}^p &= \text{tr}\{\left(\mathbf{X}^T\mathbf{X}\right)^{p/2}\} = \text{tr}\{\left(\mathbf{X}^T\mathbf{X}\right)\left(\mathbf{X}^T\mathbf{X}\right)^{p/2-1}\} \\ &= \text{tr}\{\left(\mathbf{X}^T\mathbf{X}\right)\mathbf{W}\} = \|\mathbf{X}\mathbf{W}^{\frac{1}{2}}\|_F^2, \end{aligned} \tag{1.20}$$

where $\mathbf{W}$ is the symmetric weight matrix

$$\mathbf{W} = \left(\mathbf{X}^T\mathbf{X}\right)^{p/2-1}. \tag{1.21}$$

It should be noted that, as in the case of IRLS$_p$ method described above, the weight matrix **W** is computed from the estimate of **X** obtained in the previous iteration. Such schemes have been shown to offer significant merits in terms of computational complexity, estimation performance and rate of convergence.

Lastly, inspired by the weighted version of the ℓ_1 norm mentioned above, a weighted version of the nuclear norm was recently proposed in [74], defined as

$$\|\mathbf{X}\|_{*,\mathbf{w}} = \sum_{i=1}^{m} w_i \sigma_i(\mathbf{X}). \tag{1.22}$$

By appropriately selecting the weighting coefficients w_i's, as in the weighted ℓ_1 norm, this approach addresses the inherent drawback of the nuclear norm, i.e., the equal penalization of the singular values irrespective of their values. In [96], a reweighted version was put forward by setting

$$w_i = \frac{1}{\sigma_i(\mathbf{X}) + \eta^2}. \tag{1.23}$$

Notably, despite the nonconvexity of the reweighted scheme, algorithms that converge to stationary points of the cost function were derived, which in many cases outperform nuclear norm minimization approaches in terms of estimation performance, [96].

1.2.2 Simultaneously sparse and low-rank matrix estimation

The estimation of matrices that are *either* sparse *or* low-rank has attracted the attention of several studies in the literature during the last decade, since it is met in diverse modern applications. As mentioned previously, these two problems have been tackled by following various approaches such as convex and nonconvex optimization based methods, greedy techniques, etc., assuming solely one structure for the sought matrix, i.e., either sparsity or low-rankness. Beyond these traditional approaches, the concept of additive decompositions of data **Y** as the sum of a sparse (**S**) and a low-rank **L** component, i.e., $\mathbf{Y} = \mathbf{L} + \mathbf{S}$, came into the scene, [28]. This problem, named after Robust PCA, is useful for applications such as background subtraction in video, graph clustering, [23], etc., and is beyond the scope of the present thesis.

A more recent problem is that of estimating matrices which are simultaneously *sparse and low-rank*. This problem was initially formulated as a convex optimization problem in [126], using a mixed penalty consisting of the combination of the sparsity inducing ℓ_1 and the nuclear norm. The respective optimization problem is expressed as

$$\min_{\mathbf{X} \in \mathcal{R}^{m \times n}} \quad l(\mathbf{Y}, \mathbf{X}) + \lambda \|\mathbf{X}\|_1 + \gamma \|\mathbf{X}\|_*, \tag{1.24}$$

where $l(\mathbf{Y}, \mathbf{X})$ serves as the *loss function* which measures the fitting between data **Y** and the sought matrix **X**. Moreover, $\|\cdot\|_1$ denotes the ℓ_1 norm, $\|\cdot\|_*$ is the nuclear norm and λ, γ are the corresponding regularization parameters. Taking into account the application each time at hand, the fitting term is appropriately set. Focusing on linear measurements

Table 1: Sample complexity for the recovery of sparse vectors, low-rank and simultaneously sparse and low-rank matrices from linear measurements. For the simultaneously sparse and low-rank matrix (last row) it is assumed that all elements of the rank-r matrix outside a $k_1 \times k_2$ submatrix are zero (for details see [113]).

Model	Degrees of freedom	Nonconvex recovery	Convex recovery
Sparse vector	k	$\mathcal{O}(k)$	$\mathcal{O}(k\log\frac{n}{k})$
Low-rank matrix	$r(2n-r)$	$\mathcal{O}(rn)$	$\mathcal{O}(rn)$
Low-rank & sparse matrix	$\mathcal{O}(r(k_1+k_2))$	$\mathcal{O}(r(k_1+k_2)\log n)$	$\Omega(rn)$

corrupted by i.i.d. Gaussian noise (which is of our interest in this thesis), $l(\mathbf{Y},\mathbf{X})$ reduces to the squared Frobenius norm.

Theoretical aspects related to the sample complexity (i.e., the minimum number of measurements required for successful recovery of the sought matrix) of simultaneously sparse and low-rank matrix estimation were studied in [113]. Therein, it was shown that in the compressed sensing scenario, assuming linear measurements and no presence of noise, convex formulations such as the one proposed in [126] can offer no better results (in terms of sample complexity) than incorporating into the optimization problem only one of the two constraints. Interestingly, the case is quite different in the nonconvex setting. Concretely, in [113] it was shown that only a few measurements (in the order of the degrees of freedom!) are needed for the recovery of the simultaneously structured models when nonconvex regularizers are used (assuming that global minima can be found). This sample complexity gap as to the performance of convex methods and their nonconvex counterparts is illustrated in Table 1. In [126], the merits of simultaneous sparse and low-rank matrix estimation were experimentally shown in three different settings (each corresponding to a different loss function). Specifically, the simultaneously sparse and low-rank approaches provided enhanced estimation performance as compared to traditional only sparse and only low-rank imposing algorithms. Around the same period of time, a similar sparse and low-rank formulation was proposed in the context of multitask learning in [104]. A similar setting has also been used for addressing the problems of subspace clustering in [155]. Recently, a novel approach for imposing simultaneously sparse and low-rank structures on the sought matrix was proposed in [114], whose main idea is the use of a nonconvex regularization function for imposing both sparsity and low-rankness in place of the convex ℓ_1 and nuclear norms, respectively. The ultimate goal of that approach is to benefit from the merits of nonconvex formulations over their convex relevant, while retaining the favorable properties of the latter, i.e., guaranteed convergence to global minimum. To this end, novel nonconvex regularizers for the imposition of sparsity and low-rank were introduced. These were then appropriately parameterized in order for their combination to provide a convex cost function.

1.2.3 Sparse, low-rank and nonnegative matrix factorization

Matrix factorization (MF) based methods have flourished in recent years. The reason for this is that MF based problems appear in a wide range of applications. A glimpse on these applications was given in Section 1.1.3 by listing four ubiquitous problems where MF is applied, i.e., dictionary learning, sparse PCA, matrix completion and nonnegative matrix factorization (NMF). In this section, a literature review of the different approaches that have been proposed is provided. Interestingly, all the above-mentioned problems can be put under a common umbrella, i.e., they can be expressed as constrained matrix factorization problems. In the framework of this thesis, we restrict our attention to the constraints of sparsity, low-rank and nonnegativity.

1.2.3.1 Sparse matrix factorization

Let us first focus on the sparsity constraint. Sparsity may be imposed on both the matrix factors, i.e., the coefficients' matrix and the dictionary, which are denoted as **V** and **U**, respectively. The former case, known as dictionary learning (DL) for sparse coding, [149], has attracted great interest since it constitutes one of the backbone methods in the areas of image processing and compressed sensing. The main scope of DL is to learn an overcomplete dictionary[8], which leads to a sparse representations of data. This problem is actually addressed following a two-step alternating procedure: a) estimate the dictionary **U** and b) find a sparse representation (i.e., matrix **V**) of the data on the basis spanned by the columns of the dictionary **U**[9]. The most popular DL algorithm is the so-called KSVD, [4], which is based on a generalization of the ubiquitous K-means clustering algorithm for learning the dictionary and a greedy OMP-based approach for solving the sparse coding step. Overall, KSVD solves the following minimization problem

$$\min_{\mathbf{U}\in\mathcal{R}^{m\times d},\mathbf{V}\in\mathcal{R}^{n\times d}} \|\mathbf{Y}-\mathbf{U}\mathbf{V}^T\|_F^2, \text{ subject to } \|\mathbf{V}\|_0 \leq \kappa,\ \ \|\boldsymbol{u}_i\|_2 \leq 1,\ \ \forall i = 1,2,\ldots,d, \tag{1.25}$$

where $\|\mathbf{V}\|_0$ is the ℓ_0 norm of **V** and κ is a cardinal number. Since the greedy scheme (OMP) used in the sparse coding step of KSVD is computationally prohibitive for large-scale tasks, alternative formulations of DL have been introduced. In that respect, the convex surrogate of the ℓ_0 norm, i.e., the ℓ_1 norm has been utilized for imposing sparsity on the coefficients' matrix, [101, 84], formulating DL as follows,

$$\min_{\mathbf{U}\in\mathcal{R}^{m\times d},\mathbf{V}\in\mathcal{R}^{n\times d}} \|\mathbf{Y}-\mathbf{U}\mathbf{V}^T\|_F^2 + \lambda\|\mathbf{V}\|_1,\ \ \|\boldsymbol{u}_i\|_2 \leq 1,\ \ \forall i = 1,2,\ldots,d. \tag{1.26}$$

It should be noted, that the use of the ℓ_1 norm in place of the ℓ_0 pseudo-norm now makes the sparse coding step a convex one, which can be solved by a variety of computationally efficient schemes such as the ones mentioned in Section 1.2.1 for the case of the sparse matrix estimation problem.

Another class of problems that, among other approaches, has also been viewed through the lens of the sparse MF framework in the frame of the so-called *synthesis* approach,

[8] An overcomplete dictionary actually gives rise to an underdetermined system of linear equations, see Section 1.1.

[9] Note that, in contrast to the traditional approaches whereby the dictionary is selected to be a known basis, e.g., a wavelet transform basis, in DL schemes the dictionary is *learned* from the training data.

[85], is that of sparse PCA. The key difference of sparse PCA from the plain PCA is the following: in PCA, each principal component (PC) is defined as a linear combination of all the m original variables (which correspond to the m dimensions of the $m \times 1$ data samples). Moreover, the PCs are selected so as to retain the maximum variance of the data. However, a major shortcoming of PCA is the PCs that define the transformed space are not physically meaningful. Sparse PCA circumvents this impediment by favoring PCs which arise by the linear combination of a *subset* of the original variables. This way, the resulting PCs are better connected to the initial physically interpretable variables. Nevertheless, the price to be paid for that is the loss of orthogonality among PCs and, as a consequence, the fact that the sparse PCs do not capture the dimensions of maximum variance.

Based on the above description, it becomes evident that sparse PCA can be formulated as a MF problem with the additional imposition of sparsity on the *subspace matrix* **U** (called dictionary in the context of DL). Similarly to the DL problem stated above, among other convex approaches corresponding to the *analysis* framework, sparse PCA has been also viewed as a matrix factorization problem. Along these lines, in [170] sparse PCA was formulated as the following optimization problem,

$$\min_{\mathbf{U} \in \mathcal{R}^{m \times d}, \mathbf{V} \in \mathcal{R}^{n \times d}} \|\mathbf{Y} - \mathbf{U}\mathbf{V}^T\|_F^2 + \lambda \|\mathbf{V}\|_1, \quad \|\boldsymbol{u}_i\|_2^2 + \|\boldsymbol{u}_i\|_1 \leq 1, \quad \forall i = 1, 2, \ldots, d, \tag{1.27}$$

where $\boldsymbol{v}_i$ is the ith column of matrix **V**. It should be noted that when it comes to the subspace matrix **U** (which represents the coefficients - also called *loadings* - that correspond to each principal component in the sPCA framework), sparsity is imposed along with the ℓ_2 regularization on its columns according to the so-called *elastic net* approach, [169]. Similar formulations have been also used in [100], yet in an online learning setting.

1.2.3.2 Low-rank matrix factorization

Broadly speaking, in the literature low-rank matrix factorization (LRMF) refers to the bilinear representation of a low-rank matrix (e.g., rank-d with $d \ll \min(m, n)$) $\mathbf{X} \in \mathcal{R}^{m \times n}$ as $\mathbf{X} = \mathbf{U}\mathbf{V}^T$ (with $\mathbf{U}, \mathbf{V}$ defined as above). This approach has been utilized for solving various problems involving low-rank matrix identification, leading to optimization schemes that scale well with the size of the data. In this sense, LRMF has been the cornerstone of one of the most popular problems in statistical signal processing and machine learning, that of matrix completion (MC). Departing from the usual convex formulation of matrix completion through the minimization of the nuclear norm, the authors in [83] provided an alternating minimization algorithm for recovering the missing entries. Interestingly, the size of the optimization problem (i.e., the number of involved parameters to be estimated) has been significantly reduced ($\mathcal{O}(2d \times (m + n))$) and hence the complexity of the emerging algorithms is much lower than that of their convex relevant. The reason why this happens is related to the fact that alternating minimization schemes avert the need for - computationally heavy in high-dimensional and large-scale data applications - singular value decomposition (SVD) of the data matrix. That said, matrix completion via LRMF can be stated as,

$$\min_{\mathbf{U}\in\mathcal{R}^{m\times d},\mathbf{V}\in\mathcal{R}^{n\times d}} \|\mathcal{P}_\Omega(\mathbf{U}\mathbf{V}^T) - \mathbf{Y}\|_F^2, \quad \mathbf{X} = \mathbf{U}\mathbf{V}^T. \tag{1.28}$$

Efficient algorithms which guarantee convergence to the global minimum of (1.28) under certain conditions on the linear *sampling operator* $\mathcal{P}_\Omega(\cdot)$ w.r.t. the set of indexes Ω and the assumption that the rank of **X** is known, were provided in [83]. It should also be noted that despite the nonconvexity of the arising optimization problems, recent studies have advocated the absence of spurious local minima in MF based schemes, [57, 167, 18].

Since the hypothesis that the rank of matrix **X** is known beforehand (which is a necessary condition for the algorithms in [83] to converge to a globally optimal solution) is rather strong, low-rank *decomposition norms* (i.e., norms that apply on the matrix factors **U** and **V** with the aim to penalize the rank of **X**) have been recently proposed in the literature, [8]. In view of this, a common practice is to initialize the rank of **X** with an overestimate of it, and then utilize an appropriate regularization (rank-penalization) term[10] that will gradually reduce the initial estimate of the rank (hopefully) towards the true one. The most popular low-rank promoting term is the so-called *variational form of the nuclear norm*, i.e.,

$$\operatorname*{argmin}_{\mathbf{X}=\mathbf{U}\mathbf{V}^T,\mathbf{U}\in\mathcal{R}^{m\times d},\mathbf{V}\in\mathcal{R}^{n\times d}} \frac{1}{2}\left(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2\right), \tag{1.29}$$

which is actually an upper bound of the nuclear norm of **X**, i.e., $\|\mathbf{X}\|_*$. Eq. (1.29) came up in literature as a low-rank regularizer in [134]. In [122] it was theoretically shown that the minimization of (1.29) is equivalent to minimizing the nuclear norm. In addition, the factorized formulation gives rise to algorithms that do not involve the computationally demanding SVD step any longer, which makes them serious candidates for many challenging settings involving large-scale data. Moreover, state-of-the-art research has shown that despite the nonconvex nature of the problems that include the variational form of the nuclear norm as a low-rank regularizer, convergence to the global minimum can still be established, [94].

1.2.3.3 Nonnegative matrix factorization

The last MF based problem which is addressed in the framework of this thesis is nonnegative matrix factorization (NMF). NMF has gained popularity in recent years mainly due to the interpretability it offers which is of crucial importance in various applications. The main premise of NMF differs from that of plain matrix factorization, since it restricts the solution set to that containing only nonnegative matrix factors (that is, matrices with nonnegative entries). This is the reason why NMF has been at the core of several applications dealing with nonnegative data, such as images processing, [156]. Indeed, in a variety of signal processing and machine learning tasks, such as blind source separation, the main objective is to decompose a given data matrix **X** into two *nonnegative* factors **U** and **V** up to scaling and permutation indeterminacy.

[10] This class of problems is closely related to the model selection schemes, [129].

As shown in [154], NMF is NP-hard, and the nonnegative matrix factors cannot be uniquely recovered in general. However, uniqueness (up to permutation) arises under certain conditions related to the generative model of the original data, e.g., if data have been generated by sufficiently sparse and nonnegative factors **U** and **V**, [70].

NMF was first put forth by the seminal work in [91] that proposed a first-order algorithm for solving the suitably formulated optimization problem. Since then, a large variety of algorithms have been presented differing both in the formulation of the NMF problem as well as in the proposed algorithmic aspects. Based on the latter, NMF can be categorized into four different classes, as proposed in [156], i.e., a) simple NMF, b) constrained NMF, c) structured NMF and d) generalized NMF. In this thesis, we focus on the constrained NMF (CNMF) class which refers to NMF which is equipped with constraints. That said, CNMF has been formulated by simultaneously incorporating sparsity constraints on both matrix factors in [80]. In a similar vein, ℓ_0 norm constraints were used in the NMF algorithm of [117].

1.3 Contribution

This thesis addresses the various structured matrix estimation tasks mentioned above innovating by proposing novel *nonconvex* formulations of the respective problems and new efficient algorithmic tools for solving them. Next, the contribution of the thesis is outlined with regard to each problem that is addressed.

- **Simultaneously sparse, low-rank and nonnegative matrix estimation**: As mentioned in Section 1.2.2, simultaneously multiple structured models offer no added value in terms of sample complexity and recovery performance when convex approaches, such as the one based on (1.24), are utilized, [113]. Capitalizing on this shortcoming of convex approaches, we go one step beyond the state-of-the-art by proposing a formulation that combines the reweighted versions of the ℓ_1 and nuclear norms for recovering matrices that are simultaneously sparse, low-rank and nonnegative, [67, 66]. The proposed *nonconvex* formulation leads to two iterative algorithms based on a) the incremental proximal and b) the alternating direction method of multipliers philosophy (outlined in Section 2.1), that efficiently solve the related optimization problem. Moreover, in an attempt to reduce the computational complexity and thus produce an algorithm amenable to handling large volumes of data, an innovative alternative formulation of simultaneously sparse low-rank and nonnegative matrix estimation is proposed, [63, 68]. This is based on the so-called Burer-Monteiro heuristic and the utilization of a matrix factorization representation of a low-rank matrix. Since the rank of the sought matrix is not *a priori* known, an overestimate of it is assumed and the variational form of the nuclear norm is utilized for penalizing the rank. Then, based on this formulation, a computationally efficient alternating minimization algorithm is derived. The proposed simultaneously sparse, low-rank and nonnegative matrix estimation framework is utilized for formulating hyperspectral image unmixing (to be discussed in Section 2.2.2) in a pioneering way.

As is shown in extensive simulated and real data experiments, the derived algorithms outperform their state-of-the-art counterparts showing the effectiveness of the proposed framework over other existing approaches.

- **Sparse, low-rank and nonnegative matrix factorization**: In the framework of this thesis a novel mathematical formulation of low-rank matrix factorization is introduced. This innovative scheme is sparked by the need to further improve existing low-rank MF approaches by providing scalable algorithms amenable to dealing with large-scale data. The proposed formulation is based on ideas stemming from the iterative reweighted least squares (IRLS) framework. In this regard, a novel alternating reweighted scheme for low-rank promotion in MF problems, is derived. As is shown, the recent low-rank MF schemes, such as the variational form of the nuclear norm, can be cast as special occasions of the proposed formulation by suitably selecting the reweighting matrices applied on the matrix factors. Going one step further, we propose the selection of a common reweighting matrix that couples the matrix factors and leads to a joint column sparsity promoting regularization term, [59, 61]. In doing so, low-rank promotion now reduces to the task of jointly annihilating columns of the matrix factors. In an effort to address inherent obstacles related to the nonseparability and nonsmoothness of the introduced low-rank promoting terms, we resort to the *block successive upper bound minimization* framework (see Section 2.1.4). In this regard, novel iteratively reweighted least squares (IRLS) type denoising, matrix completion and NMF algorithms are devised that rely exclusively on computationally efficient matrixwise updates. In addition, to further reduce complexity, a column pruning procedure is incorporated that removes the matrix factor columns whose power has become negligible, thus gradually reducing the size of the optimization problems towards that of the actual rank of the sought matrix. The connection of the proposed schemes with previously reported IRLS algorithms is also established. Analysis regarding the convergence of the algorithms to stationary points and their rates of convergence is also provided. The proposed low-rank promoting term is further extended to the problem of low-rank and sparse NMF, [60]. The merits of the proposed algorithms in terms of estimation performance and computational complexity, compared to relevant state-of-art algorithms, are illustrated on simulated and real data experiments. In order to test the effectiveness of the algorithms on real applications involving large-scale data, the problems of hyperspectral image denoising, matrix completion in movies recommender systems, music signal decomposition and unsupervised hyperspectral unmixing with unknown number endmembers, are employed.

- **Online low-rank subspace learning and matrix completion**: Capitalizing on previous work on online (group) sparse linear regression [142], [143] and leveraging the low-rank promotion idea mentioned above for the problem of sparse and low-rank MF, we devise two new online low-rank subspace learning and matrix completion algorithms, [64, 65, 58]. The first approach sticks to the Bayesian philosophy, while the second one can be viewed as its deterministic analogue. Concerning the Bayesian approach, a novel Bayesian model is first defined. It is worth emphasizing that the proposed Bayesian model incorporates exponentially weighted data and parameter priors, which facilitate online inference in a time-varying environment. Moreover, both column-sparsity and sparsity are enforced on the subspace matrix, i.e. *multiple constraints* are imposed *simultaneously* on the same data structure. This is a strategy in its very infancy in the Bayesian literature that, among others, makes the algorithm capable of addressing the sparse dictionary learning problem. Regarding the inference procedure, a novel online variational Bayes approximation scheme is developed, which induces complexity similar to that of state-of-the-art first-order stochastic approximation based schemes, [103]. The second online deterministic algorithm is based on the minimization of an exponentially weighted regularized cost function. Assuming the same column-sparsity promoting ideas with the Bayesian scheme, we are led to an efficient alternating minimization based algorithm for online matrix completion and low-rank subspace learning. It should be noted that in order to avoid matrix inversions (a desirable property that is of crucial importance in online applications) a Gauss-Seidel approach is adopted. As demonstrated on simulated and real data experiments, both the proposed algorithms present superior estimation performance than other state-of-the-art relevant algorithms. To validate this, the hyperspectral image denoising and the eigenface learning problems are examined, corroborating the effectiveness and higher reconstruction performance of the proposed algorithms on real data.

1.4 Outline of the thesis

In **Chapter 1**, the structured matrix estimation problem from linear measurements was introduced and the basic topics that are addressed in the thesis were presented. More specifically, first we described the original formulations of these estimation tasks and next, we provided a brief, yet comprehensive, review of the various solutions that have recently come to the scene in the literature.

In the first part of **Chapter 2**, we present the basic algorithmic tools that are employed in the thesis. These tools range from traditional convex optimization algorithms, such as proximal minimization methods and the alternating direction method of multipliers, to cutting-edge tools belonging to the block successive upper bound minimization framework. Beyond the optimization schemes, the basic principles of the variational Bayes inference method are presented. Finally, we outline the basic premise of *online learning* in particular within the online variational Bayes framework. The second part of Chapter 2, describes the main large-scale data applications with which we are concerned in the thesis, i.e., hyperspectral image unmixing and denoising.

Chapter 3 presents two different *nonconvex* formulations for addressing the simultaneously sparse, low-rank and nonnegative matrix regression problem, namely, a) a reweighted ℓ_1 and nuclear norm minimization one and b) a matrix factorization based approach. An incremental proximal minimization algorithm and an alternating direction method of multipliers algorithm are then derived for solving the first minimization problem associated with the first formulation. In addition, an alternating minimization scheme is presented for tackling the matrix factorization based problem. Next, the task of hyperspectral image unmixing is formulated as a simultaneously sparse, low-rank and nonnegative matrix estimation problem which can be dealt with the derived algorithms. The merits of this approach are next showcased in both simulated and real hyperspectral imaging data.

In **Chapter 4**, a new low-rank matrix factorization based formulation of matrix completion, denoising, low-rank nonnegative matrix factorization and sparse and low-rank nonnegative matrix factorization is presented. Next, novel alternatingly iterative reweighted least squares algorithms springing from the block successive upper bound minimization framework are derived, in order to solve this problem. A convergence analysis of these algorithms is also provided. Finally, the problems of hyperspectral image denoising, matrix completion in recommender systems, music signal decomposition and unsupervised unmixing of hyperspectral data are utilized for verifying the effectiveness of the proposed algorithms in real data applications.

Chapter 5 presents a) a variational Bayes algorithm and b) a cost function minimization one, for online low-rank subspace learning from incomplete data. As far as a) is concerned, first a multi-hierarchical Bayesian model is defined. Based on this model, a batch variational Bayes scheme is derived which constitutes the basis of the subsequently presented online variational Bayes algorithm. In the frame of b), a deterministic relevant scheme is presented. The performance of both algorithms is subsequently analyzed in a variety of simulated and real data experiments.

Finally, concluding remarks and directions for future research are provided in **Chapter 6**, the last chapter of the thesis.

2. OPTIMIZATION TOOLS AND APPLICATIONS

This chapter consists of two parts. In the first part a review is provided of the optimization and Bayesian inference methods that have been used in the frame of this thesis for addressing the various structured matrix estimation problems presented in Chapter 1. In the second part, a detailed description is given of the main practical applications that were studied, i.e., hyperspectral image unmixing and hyperspectral image denoising.

2.1 Optimization algorithms

A certain amount of contribution of this thesis lies in the derivation of novel efficient estimation algorithms. Herein, the basic methodologies upon which the introduced algorithms have been built are presented. First, proximal operators, which are at the core of two of the proposed algorithms, are presented.

2.1.1 Proximal operators and proximal based algorithms

Next, a formal definition of the proximal operator is given, [115].

Definition 2.1. *Let Γ_0 denote the class of closed*[1]*, convex and proper*[2] *(hence lower-semicontinuous) functions from $\mathcal{R}^N$ to $]-\infty, +\infty]$. The proximal operator of $\mathbf{x} \in \mathcal{R}^N$ with respect to an $f \in \Gamma_0$, is defined as*

$$\text{prox}_f(\mathbf{x}) = \underset{\mathbf{y}\in\mathcal{R}^N}{\text{argmin}} \;\; f(\mathbf{y}) + \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_2^2. \tag{2.1}$$

and is unique.

The proximal operator reduces to the projection operator when $f(\mathbf{x}) \equiv I_C(\mathbf{x})$ where $I_C(\mathbf{x})$ is the indicator function of a set C. Thus, it can be considered as a generalized version of the latter. Next an interesting interpretation of the proximal operator is presented. In this regard, let us first define the *infimal convolution* of the closed proper convex functions f and g on $\mathcal{R}^N$, denoted by $f\Box g$, as,

$$(f\Box g)(\mathbf{x}) = \inf_{\mathbf{y}} \left(f(\mathbf{y}) + g(\mathbf{x} - \mathbf{y})\right), \tag{2.2}$$

where the domain of the resulting function $f\Box g$ is the union of the domains of f and g. By setting $g(\cdot) = \frac{1}{2\lambda}\|\cdot\|_2^2$ we get the so-called Moreau envelope of f with parameter $\lambda > 0$

[1] A function $f : \mathcal{R}^N \mapsto \mathcal{R}$ is closed if for each $\alpha \in \mathcal{R}$, the sublevel set $\{\mathbf{x} \in \textbf{dom}\ f : f(\mathbf{x}) \leq a\}$ (where **dom** f denotes the domain of f) is a closed set.

[2] A convex function is proper if its effective domain set is nonempty and it never attains -∞.

denoted as $M_{\lambda f}$ (also known as Moreau-Yosida regularization), defined as

$$M_{\lambda f}(\mathbf{x}) = \inf_{\mathbf{y}}(f(\mathbf{y}) + \frac{1}{2\lambda}\|\mathbf{y} - \mathbf{x}\|_2^2). \tag{2.3}$$

The Moreau envelope $M_{\lambda f}$ provides a *smoothed* approximation of f, that is, it is continuously differentiable even if f is not. Moreover it holds some other favorable properties, i.e., it is strictly convex and hence it has a unique minimum. Based on the above, a proximal operator can be viewed as the point that attains the infimum of the Moreau envelope of a convex function f. This in turn implies that the proximal operation corresponds to a gradient step of the Moreau envelope of f at **x**, [115], i.e.,

$$\text{prox}_{\lambda f}(\mathbf{x}) = \mathbf{x} - \lambda \nabla M_{\lambda f}(\mathbf{x}). \tag{2.4}$$

Proximal operators present favorable properties such as the firm nonexpansive property, i.e.,

$$\|\text{prox}_f(\mathbf{x}) - \text{prox}_f(\mathbf{y})\|_2^2 \leq (\mathbf{x} - \mathbf{y})^T(\text{prox}_f(\mathbf{x}) - \text{prox}_f(\mathbf{y})) \tag{2.5}$$

$\forall \mathbf{x}, \mathbf{y} \in \mathcal{R}^N$. Moreover, it can be easily shown that the fixed points of the proximal operator coincide with the minimizers of the function f, [115]. By virtue of these two key features, various *proximal algorithms* have been devised whose goal is to find the minimizers of a cost function via pursuing the fixed points of the respective proximal operators.

In that respect the proximal point algorithm (also known as proximal minimization algorithm) emerges by repeatedly applying the proximal operator starting with an initial point $\mathbf{x}_0$. The resulting algorithm thus consists of iterations of the form

$$\mathbf{x}_{k+1} = \text{prox}_{\lambda f}(\mathbf{x}_k). \tag{2.6}$$

where k denotes the iteration number. If the set of minimizers of f is nonempty, $\mathbf{x}_k$ converges to a member of it, [115].

The proximal operator lies at the heart of another ubiquitous algorithm in the signal processing field that is the *proximal gradient algorithm*, [39]. Let us assume that $f : \mathcal{R}^N \rightarrow \mathcal{R}$ and $g : \mathcal{R}^N \rightarrow \mathcal{R}$ are closed, convex and proper functions, with f being also differentiable. The proximal gradient algorithm can be utilized for solving minimization problems in the form

$$\min_{\mathbf{x} \in \mathcal{R}^N} \; f(\mathbf{x}) + g(\mathbf{x}) \tag{2.7}$$

by applying the following iteration scheme

$$\mathbf{x}_{k+1} = \text{prox}_{\lambda^k g}(\mathbf{x}_k - \lambda^k \nabla f(\mathbf{x}_k)) \tag{2.8}$$

where $\lambda^k > 0$ is the step size at iteration k. The proximal gradient algorithm is also referred in the literature as the *forward-backward splitting algorithm* since it can be considered as a combination of a gradient (forward) step with a proximal (backward) step[3].

2.1.2 Incremental proximal minimization

Now that the proximal operator and the most popular proximal operator based algorithms have been introduced, we proceed by presenting the family of *incremental proximal minimization algorithms*, [17], focusing on a specific member of it, employed in Chapter 3 of this thesis.

In general, incremental algorithms have been devised for addressing minimization problems that include several constraints thus giving rise to cost functions consisting of multiple additive components. The arising minimization problem is mathematically formulated as

$$\min_{\mathbf{x}\in\mathcal{C}} \sum_{i=1}^{\rho} f_i(\mathbf{x}). \tag{2.9}$$

The main idea of incremental minimization is to deal with a single component $f_i(\mathbf{x})$ at each iteration instead of considering the whole cost function. Actually, this scheme leads to a remarkable decrease of the induced computational complexity, especially in cases where the number ρ of components $f_i(\mathbf{x})$ is large.

The first incremental type algorithms that came into the scene were the incremental gradient algorithms, which are based on the assumption that all f_i's are differentiable, [17]. Specifically, incremental gradient algorithms consist of iterations in the form

$$\mathbf{x}_{k+1} = P_{\mathcal{C}}(\mathbf{x}_k - \lambda^k \nabla f_{i_k}(\mathbf{x}_k)), \tag{2.10}$$

where $P_{\mathcal{C}}(\cdot)$ is the projection operator on the set $\mathcal{C}$ and the subscript i_k is the index of the component of the cost function that is iterated on. The stepsize parameter λ^k can be defined in various ways, each leading to a different scheme. The order of the components of the cost function may also be random as is the case in the *randomized* versions of the algorithms. However, the computational merits of incremental gradient methods come at a price, that is, a slow (sublinear) asymptotic rate of convergence. This is not only because they are first-order methods, but also due to the need for adopting diminishing step-sizes ($\lambda^k \rightarrow 0$, as $k \rightarrow \infty$) in order to avoid oscillations that otherwise occur in the case of constant step-sizes (actually diminishing step sizes are required also for the other incremental schemes that are mentioned next), [16]. It should be though noted that in most cases the initial convergence rate of incremental methods is quite fast thus making them an efficient tool when it comes to large-scale data processing, whereby the accuracy of the estimations is less important than the need for lower algorithmic demands concerning the computational resources.

Along similar lines, incremental subgradient methods were introduced as simple extensions of the gradient counterparts for the case of nondifferentiable cost functions. In that

[3] The notions of forward and backward are coming from the gradient flow interpretation of the proximal operators by applying the Euler discretization, [115].

case, an arbitrary subgradient $\tilde{\nabla} f_{i_k}(\mathbf{x})$ (the subgradient $\tilde{\nabla} f_{i_k}(\mathbf{x})$ is selected by the subdifferential set (i.e., the set of the subgradients) of f at **x** denoted as $\partial f_{i_k}(\mathbf{x})$) is used in place of the gradient hence the iterations of the arising algorithm are defined as

$$\mathbf{x}_{k+1} = P_{\mathcal{C}}(\mathbf{x}_k - \lambda^k \tilde{\nabla} f_{i_k}(\mathbf{x}_k)). \tag{2.11}$$

Similarly to the incremental gradient algorithms, incremental subgradient methods require a decreasing stepsize λ^k in order to ensure convergence. However, in this case this property is not a shortcoming of the incremental rationale, since the same rule is required also for the case of the non-incremental subgradient algorithms.

Now that the incremental gradient and subgradient methods have been described, we move onto the more recently introduced *incremental proximal methods*. As implied by their name, the same incremental philosophy utilized in the previously described incremental gradient and subgradient methods is now adapted for proximal minimization. That said, incremental proximal methods perform the following iterations

$$\mathbf{x}_{k+1} = \underset{\mathbf{x}\in\mathcal{C}}{\text{argmin}}\{f_{i_k}(\mathbf{x}) + \frac{1}{2\lambda^k}\|\mathbf{x} - \mathbf{x}_k\|_2^2\}. \tag{2.12}$$

Obviously, the righthand side of (2.12) equals to the proximal operator of f_{i_k} at $\mathbf{x}_k$, i.e., $\text{prox}_{\lambda^k f_{i_k}}(\mathbf{x}_k)$. The motivation behind the use of incremental proximal methods lies in the easily obtained closed form expressions for the proximal operators for a wide range of functions $f_{i_k}(\mathbf{x})$ along with the fact that proximal iterations are considered more stable compared to their gradient and subgradient counterparts. This favorable stability property can be conjectured by the fact that, in the non-incremental setting, proximal algorithms contrary to gradient methods, converge for any choice of λ^k. That said, it can be easily understood that in many cases such as ill-conditioned problems, the use of proximal type methods instead of either gradient or subgradient ones may be preferred.

Recently, incremental gradient, subgradient and proximal methods have been considered under a unified framework, [16]. This generalized view of the incremental optimization strategy is of significant importance in cases where the components f_i's differ in their form, i.e., for some of the f_i's their respective proximal operators may be expressed in closed form whereas for others this may not be the case. In the latter cases, the use of either gradient or subgradient steps may clearly be a better choice. Let us now define the following optimization problem

$$\min_{\mathbf{x}\in\mathcal{C}} \sum_{i=1}^{\rho} F_i(\mathbf{x}), \tag{2.13}$$

where $F_i(\mathbf{x}) = f_i(\mathbf{x}) + h_i(\mathbf{x})$, $f_i : \mathcal{R}^N \mapsto \mathcal{R}$ and $h_i : \mathcal{R}^N \mapsto \mathcal{R}$ are convex functions, and $\mathcal{C}$ is a nonempty closed convex set. Assuming that the proximal operators of f_i's can be easily obtained whereas h_i's suit better to subgradient steps, the unified incremental framework gives rise to algorithms consisting of iterations in the form

$$\mathbf{z}_k = \underset{\mathbf{x}\in\mathcal{C}}{\text{argmin}}\{f_{i_k}(\mathbf{x}) + \frac{1}{2\lambda^k}\|\mathbf{x} - \mathbf{x}_k\|_2^2\}, \tag{2.14}$$

$$\mathbf{x}_{k+1} = P_{\mathcal{C}}(\mathbf{z}_k - \lambda^k\tilde{\nabla}h_{i_k}(\mathbf{z}_k)), \tag{2.15}$$

where $\tilde{\nabla}h_{i_k}(\mathbf{z}_k)$ denotes an arbitrary subgradient of the nonempty subdifferential set ∂h_i estimated at $\mathbf{z}_k$.

Convergence analysis of the incremental algorithms described above is provided in [16]. It should be noted that convergence behavior depends on the update rule (cyclic or randomized order) that is followed regarding the minimization of the components f_i's and h_i's.

2.1.3 Primal-dual optimization and the alternating direction methods of multipliers

Before presenting the main principles and philosophy of the alternating direction method of multipliers (ADMM) that is utilized in Chapter 3, the general concepts of the primal - dual optimization strategy, which is at the core of ADMM, are briefly described.

An optimization problem is described in its primal form (also called as standard form) as follows:

$$\text{min} f_0(\mathbf{x}) \;\text{ subject to }\; f_i(\mathbf{x}) \leq 0, \quad i = 1, 2, \ldots, m, \;\; h_j(\mathbf{x}) = 0, \quad j = 1, 2, \ldots, n \tag{2.16}$$

with $\mathbf{x} \in \mathcal{R}^N$. The inequalities $f_i(\mathbf{x}) \leq 0$ are called the *inequality constraints* and the equalities $h_j(\mathbf{x}) = 0$ are the so-called *equality constraints*. The domain $\mathcal{D}$ of the optimization problem is defined as

$$\mathcal{D} = \bigcap_{i=0}^{m} \textbf{dom} f_i \cap \bigcap_{j=1}^{n} \textbf{dom} h_j, \tag{2.17}$$

where **dom** denotes the domain set of a function. The *feasibility set* is defined as the subset of $\mathcal{D}$, which contains points $\mathbf{x}$ that satisfy all the constraints, i.e., $f_i(\mathbf{x}) \leq 0$ and $h_j(\mathbf{x}) = 0$. Assuming that the feasibility set is nonempty, the optimal value of the problem denoted as $p^\star$ is attained as the infimum of the objective function $f_0(\mathbf{x})$ on the feasibility set, i.e.,

$$p^\star = \text{inf}\{f_0(\mathbf{x}) \mid f_i(\mathbf{x}) \leq 0, i = 1, 2, \ldots, m, h_j(\mathbf{x}) = 0, j = 1, 2, \ldots, n\}. \tag{2.18}$$

Note that by convention, $p^\star = +\infty$ when the feasibility set is empty, and $p^\star = -\infty$ when there exist a sequence of points $\mathbf{x}_k$ with $f_0(\mathbf{x}_k) \rightarrow -\infty$ as $k \rightarrow +\infty$, that is, the objective function is unbounded below.

The Lagrangian $L : \mathcal{R}^N \times \mathcal{R}^m \times \mathcal{R}^n \rightarrow \mathcal{R}$ of the problem (2.16) is defined as

$$L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = f_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i f_i(\mathbf{x}) + \sum_{j=1}^{n} \nu_j h_j(\mathbf{x}), \tag{2.19}$$

where $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_m]^T$, $\boldsymbol{\nu} = [\nu_1, \nu_2, \ldots, \nu_n]^T$. Parameters λ_i's and ν_j's related to f_i's and h_j's, respectively, are called *Lagrange multipliers* or *dual variables* of the optimization problem.

The dual function (also called *Lagrangian dual* function) $g : \mathcal{R}^m \times \mathcal{R}^n \to \mathcal{R}$ of the dual variables $(\boldsymbol{\lambda}, \boldsymbol{\nu})$ with $\boldsymbol{\lambda} \geq \mathbf{0}$, is defined as

$$g(\boldsymbol{\lambda}, \boldsymbol{\nu}) = \inf_{\mathbf{x} \in \mathcal{D}} \left(L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) \right). \tag{2.20}$$

It can be easily shown that for any pair $(\boldsymbol{\lambda}, \boldsymbol{\nu})$ the dual function yields a lower bound of the optimal value $p^\star$ of the primal optimization problem, [25]. Since there is no meaning in the cases that the dual function takes the value of $-\infty$, attention is given in the so-called *dual feasible* pairs $(\boldsymbol{\lambda}, \boldsymbol{\nu})$, i.e., the pairs $(\boldsymbol{\lambda}, \boldsymbol{\nu}) \in \mathbf{dom} g$.

The lower-bound property of the dual function and the fact that it is a concave function irrespective of the convex (or nonconvex) nature of f_i's[4] has given way to the formulation of the *dual optimization* problem of (2.16) defined as

$$\max_{\boldsymbol{\lambda}, \boldsymbol{\nu}} \; g(\boldsymbol{\lambda}, \boldsymbol{\nu}) \;\; \text{subject to} \;\; \boldsymbol{\lambda} \geq \mathbf{0}. \tag{2.21}$$

Let us denote by $d^\star$ the optimal value of the problem (2.21). In case that the optimal values of the primal and the dual problems coincide, i.e., $d^\star = p^\star$, then the so-called *strong duality* holds, [25]. Strong duality implies zero *optimal duality gap* ($d^\star - p^\star = 0$) and usually (but not always) holds when the respective primal optimization problem is convex. Various conditions that establish strong duality, known as *constraint qualifications*, have been proposed in the literature. Among them, the Slater's theorem states that strong duality holds if the problem is convex and there exists an $\mathbf{x} \in \text{relint}\mathcal{D}$, where relint stands for the relative interior of a set, [25], such that the inequality constraints hold in a strict sense, i.e., $f_i(\mathbf{x}) < 0, i = 1, 2, \ldots, m$.

In the sequel, we clarify further the notions of strong duality and optimal duality gap. To this end, it is not difficult to verify that the optimal value $p^\star$ of the primal problem can be expressed in terms of the Lagrangian function as,

$$p^\star = \inf_{\mathbf{x}} \sup_{\boldsymbol{\lambda} \geq \mathbf{0}, \boldsymbol{\nu}} L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) \tag{2.22}$$

while, taking into account (2.20) and (2.21), the optimal value of the respective dual function $d^\star$ is given by

$$d^\star = \sup_{\boldsymbol{\lambda} \geq \mathbf{0}, \boldsymbol{\nu}} \inf_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}). \tag{2.23}$$

By (2.22) and (2.23) it becomes clear that strong duality allows us to switch the order of the optimization w.r.t. $\mathbf{x}$ and $(\boldsymbol{\lambda}, \boldsymbol{\nu})$ without affecting the result. This property is known as

[4] This is a consequence of the fact that the dual function is by construction the infimum of an affine function of $(\boldsymbol{\lambda}, \boldsymbol{\nu})$.

strong max-min property or *saddle property*. The latter name actually stems from the fact that the primal optimum $\mathbf{x}^\star$ and the dual optimum pair $(\boldsymbol{\lambda}^\star, \boldsymbol{\nu}^\star)$ actually correspond to a saddle point of the Lagrangian function, i.e.,

$$L(\mathbf{x}^\star, \boldsymbol{\lambda}, \boldsymbol{\nu}) \leq L(\mathbf{x}^\star, \boldsymbol{\lambda}^\star, \boldsymbol{\nu}^\star) \leq L(\mathbf{x}, \boldsymbol{\lambda}^\star, \boldsymbol{\nu}^\star). \tag{2.24}$$

Having said that we may proceed by briefly describing two basic precursors of ADMM, the *dual ascent method* and the *method of multipliers*.

2.1.3.1 The dual ascent method

Next, for ease of understanding, the general forms of the primal and dual optimization problems are simplified by keeping only the equality constraints and dropping the inequality ones (the simplified analysis can be easily generalized). In this case, the resulting optimization problem reduces to

$$\min_{\mathbf{x}} f(\mathbf{x}) \text{ subject to } \mathbf{A}\mathbf{x} - \mathbf{b} = \mathbf{0}, \tag{2.25}$$

where $\mathbf{A} \in \mathcal{R}^{n \times N}$ and $\mathbf{b} \in \mathcal{R}^n$. Assuming that strong duality holds and $L(\mathbf{x}, \boldsymbol{\nu})$ is uniquely minimized w.r.t. $\mathbf{x}$ (i.e., it is strictly convex), optimal points $\mathbf{x}^\star$ and $\boldsymbol{\nu}^\star$ of the primal and the dual problems, respectively, can be recovered as

$$\mathbf{x}^\star = \operatorname*{argmin}_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\nu}^\star) \tag{2.26}$$

and

$$\boldsymbol{\nu}^\star = \operatorname*{argmax}_{\boldsymbol{\nu}} g(\boldsymbol{\nu}) \quad (\equiv \operatorname*{argmax}_{\boldsymbol{\nu}} L(\mathbf{x}^\star, \boldsymbol{\nu})). \tag{2.27}$$

where

$$g(\boldsymbol{\nu}) \equiv L(\mathbf{x}^\star, \boldsymbol{\nu}) = f(\mathbf{x}^\star) + \boldsymbol{\nu}^T(\mathbf{A}\mathbf{x}^\star - \mathbf{b}). \tag{2.28}$$

Since the dual function $g(\boldsymbol{\nu})$ is differentiable and due to its concavity, the update of the dual variable $\boldsymbol{\nu}$ can be accomplished via a gradient ascent step[5]. For the dual function given in (2.28) the gradient is $\nabla g(\boldsymbol{\nu}) = \mathbf{A}\mathbf{x}^+ - \mathbf{b}$, where $\mathbf{x}^+ = \text{argmin} L(\mathbf{x}, \boldsymbol{\nu})$, hence the dual ascent algorithm is composed of iterations for $\mathbf{x}$ and $\boldsymbol{\nu}$ in the following form

$$\begin{aligned} \mathbf{x}_{k+1} &= \operatorname*{argmin}_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\nu}_k) \\ \boldsymbol{\nu}_{k+1} &= \boldsymbol{\nu}_k + \alpha^k(\mathbf{A}\mathbf{x}_{k+1} - \mathbf{b}). \end{aligned} \tag{2.29}$$

Under certain assumptions, it can be shown that the generated sequences of $\mathbf{x}_k$ and $\boldsymbol{\nu}_k$ converge to the optimal points $\mathbf{x}^\star$ and $\boldsymbol{\nu}^\star$.

A key feature of the dual ascent method is its decomposability. More specifically, by as-

[5] In nondifferentiable occasions, a subgradient is used in place of the gradient.

suming $\mathbf{x} = [\mathbf{x}_1^T, \mathbf{x}_2^T, \ldots, \mathbf{x}_\rho^T]^T$ and a separable objective function

$$f(\mathbf{x}) = \sum_{i=1}^{\rho} f_i(\mathbf{x}_i), \tag{2.30}$$

the minimization step of the Lagrangian can be split into ρ steps, which offers the possibility of applying distributed optimization schemes, [24].

2.1.3.2 The method of multipliers

In an effort to relax the assumptions needed for the dual ascent method to converge to the optimal $\mathbf{x}^\star$ and $\boldsymbol{\nu}^\star$ such as strict convexity, and thus address inherent weaknesses of the dual ascent method, the method of multipliers has been proposed. To this end, a variant version of the Lagrangian function, termed *augmented Lagrangian*, has been utilized. The augmented Lagrangian of the problem in (2.25) is defined as

$$L_a(\mathbf{x}, \boldsymbol{\nu}) = f(\mathbf{x}) + \boldsymbol{\nu}^T(\mathbf{Ax} - \mathbf{b}) + \frac{\mu}{2}\|\mathbf{Ax} - \mathbf{b}\|_2^2, \tag{2.31}$$

where μ is a penalty parameter. The minimization of the Lagrangian (2.31), can be easily reformulated as the following optimization problem

$$\min_{\mathbf{x}} \ f(\mathbf{x}) + \frac{\mu}{2}\|\mathbf{Ax} - \mathbf{b}\|_2^2 \ \text{ subject to } \ \mathbf{Ax} - \mathbf{b} = \mathbf{0}. \tag{2.32}$$

From (2.31), it is obvious that, for any feasible point $\mathbf{x}$ which satisfies the equality constraint, the resulting primal optimization problem (2.32) is tantamount to (2.25) since the added regularization term becomes zero. Interestingly, the regularization induced by following the approach of the method of multipliers makes the dual function $g_a(\boldsymbol{\nu}) = \inf_{\mathbf{x}} (L_a(\mathbf{x}, \boldsymbol{\nu}))$ differentiable under rather mild conditions. All in all, the method of multipliers results by applying the dual ascent method on the modified problem (2.32), and consists of iterations of the form

$$\begin{aligned} \mathbf{x}_{k+1} &= \operatorname*{argmin}_{\mathbf{x}} L_a(\mathbf{x}, \boldsymbol{\nu}_k) \\ \boldsymbol{\nu}_{k+1} &= \boldsymbol{\nu}_k + \mu(\mathbf{Ax}_{k+1} - \mathbf{b}). \end{aligned} \tag{2.33}$$

Note that the regularization parameter μ has been used in place of the stepsize parameter a^k of (2.29)[6]. Interestingly, the method of multipliers is characterized by some favorable features as compared to the dual ascent method including that a) the objective function $f_0(\mathbf{x})$ in not required to be strictly convex in order to guarantee uniqueness of the minimizer and b) it converges even if the objective function takes on the value $+\infty$, [24]. It should be though noted that the aforementioned favorable properties of the method of multipliers come at a price: the minimization of the modified Lagrangian is no longer decomposable even if the objective function is separable.

[6] By using μ as the step size in the dual update, the iterate $\boldsymbol{\nu}_{k+1}$ is dual feasible, [24].

2.1.3.3 The alternating direction method of multipliers

The main objective of the alternating direction method of multipliers is to blend the benefits of the dual ascent method, i.e., *decomposability*, and the robust convergence properties of the method of multipliers. To put it in mathematical terms, ADMM is applied to optimization problems that have the form

$$\underset{\mathbf{x},\mathbf{z}}{\text{argmin}} \;\; f(\mathbf{x}) + g(\mathbf{z}) \;\; \text{subject to} \;\; \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{c}, \tag{2.34}$$

where $\mathbf{x} \in \mathcal{R}^N, \mathbf{z} \in \mathcal{R}^M, \mathbf{A} \in \mathcal{R}^{n\times N}$, $\mathbf{B} \in \mathcal{R}^{n\times M}$ and $\mathbf{c} \in \mathcal{R}^n$. The augmented Lagrangian of the optimization problem in (2.34) is given as

$$L_a(\mathbf{x}, \mathbf{z}, \boldsymbol{\nu}) = f(\mathbf{x}) + g(\mathbf{z}) + \boldsymbol{\nu}^T \left(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}\right) + \frac{\mu}{2}\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}\|_2^2. \tag{2.35}$$

The iterations of ADMM are expressed as follows

$$\begin{aligned} \mathbf{x}_{k+1} &= \underset{\mathbf{x}}{\text{argmin}} \; L_a(\mathbf{x}, \mathbf{z}_k, \boldsymbol{\nu}_k) \\ \mathbf{z}_{k+1} &= \underset{\mathbf{z}}{\text{argmin}} \; L_a(\mathbf{x}_{k+1}, \mathbf{z}, \boldsymbol{\nu}_k) \\ \boldsymbol{\nu}_{k+1} &= \boldsymbol{\nu}_k + \mu \left(\mathbf{A}\mathbf{x}_{k+1} + \mathbf{B}\mathbf{z}_{k+1} - \mathbf{c}\right). \end{aligned} \tag{2.36}$$

It becomes clear from (2.35) that ADMM iterations differ from those of the method of multipliers in the following sense: the method of multipliers updates $(\mathbf{x}, \mathbf{z})$ jointly (i.e., $(\mathbf{x}, \mathbf{z}) = \underset{(\mathbf{x},\mathbf{z})}{\text{argmin}} \; L(\mathbf{x}, \mathbf{z}, \boldsymbol{\nu})$). On the other hand, ADMM alternatingly updates $\mathbf{x}$ and $\mathbf{z}$ hence the name alternating direction. That said, ADMM resembles a *single-pass Gauss-Seidel* version of the method of multipliers.

It has been shown in [25] that by assuming that a) both $f(\mathbf{x})$ and $g(\mathbf{z})$ are closed, proper and convex and b) strong duality holds, the following ADMM convergence results hold:

- $\mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{z}_k - \mathbf{c} \rightarrow 0$ as $k \rightarrow +\infty$, i.e., the sequence $\mathbf{x}_k, \mathbf{z}_k$ convergences to a feasible point
- $f(\mathbf{x}) + g(\mathbf{z}) \rightarrow p^\star$, which implies convergence of the primal objective to the primal optimal value
- $\boldsymbol{\nu}_k \rightarrow \boldsymbol{\nu}^\star$ that is, the dual variable converges to the dual optimal point.

2.1.4 Block successive upper bound minimization

Block successive upper bound minimization (BSUM) has been recently appeared in the literature as an invaluable optimization tool for large-scale data applications. BSUM actually serves as a generalization of block coordinate descent (BCD) schemes, [159]. That is, similarly to BCD, BSUM is based on the following premise: solve an optimization problem which involves a possibly huge number of variables by solving a sequence of “smaller” - easier to handle - problems, each one focusing on a subset (block) of the variables.

Next, a concise description of the BCD philosophy, which is at the core of BSUM, is provided. Let us consider the following optimization problem

$$\text{min} f(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) \quad \mathbf{x}_i \in \mathcal{C}_i, \;\; i = 1, 2, \ldots, n, \tag{2.37}$$

where f is a continuous objective function and each $C_i \subset \mathcal{R}^{m_i}$ is a closed convex set. The $(k+1)$th iteration of BCD entails an update of the following form

$$\mathbf{x}_{i,k+1} \in \underset{\mathbf{x}_i}{\text{argmin}} f(\mathbf{x}_{1,k}, \mathbf{x}_{2,k}, \ldots, \mathbf{x}_{i-1,k}, \mathbf{x}_i, \mathbf{x}_{i+1,k}, \ldots, \mathbf{x}_{n,k}), \tag{2.38}$$

that is f is minimized w.r.t. $\mathbf{x}_i$, $i = 1, 2, \ldots, n$ considering all the remaining blocks of variables fixed to their most recent estimates. It becomes evident that BCD is quite appealing and very simple to implement. For this reason, BCD has been extensively used in several signal processing and machine learning applications where one should deal with optimization problems which are characterized by increased computational complexity. Recently, BCD has been applied to additive cost functions consisting of sums of smooth and nonsmooth terms, which are widely met in numerous applications such as compressed sensing, image processing, etc. Moreover, theoretical analysis of BCD guarantees convergence to optimal points under certain conditions, [151].

Capitalizing on the favorable computational merits of the BCD strategy, BSUM offers a generalized optimization framework which encloses a great many algorithms that are commonly met in the literature, [79]. The crux of BSUM is the following: *instead of the exact updates defined in (2.38), at each iteration minimize a local tight upper bound of the cost function* (see Fig. 5). That is, while BCD hinges on the blockwise minimization of the exact cost function, BSUM follows a different approach by proposing the use of approximate versions of the cost function at each step of the algorithm. This approach is of particular importance in numerous occasions where the objective functions may be nonconvex and exact minimization required by BCD can not be accomplished. Moreover, BSUM provides milder conditions for convergence as compared to BCD.

For ease of notation we define $\mathbf{x}_{\neg i,k} = [\mathbf{x}_{1,k}^T, \mathbf{x}_{2,k}^T, \ldots, \mathbf{x}_{i-1,k}^T, \mathbf{x}_{i+1,k}^T, \ldots, \mathbf{x}_{n,k}^T]^T$ and denote $f(\mathbf{x}_{1,k}, \mathbf{x}_{2,k}, \ldots, \mathbf{x}_{i-1,k}, \mathbf{x}_i, \mathbf{x}_{i+1,k}, \ldots, \mathbf{x}_{n,k})$ as $f(\mathbf{x}_i, \mathbf{x}_{\neg i,k})$. For each $f(\mathbf{x}_i, \mathbf{x}_{\neg i,k})$, approximate functions $l_i(\mathbf{x}_i, \mathbf{x}_k) : \mathcal{C}_i \rightarrow \mathcal{R}$ are defined where $\mathbf{x}_k = [\mathbf{x}_{1,k}^T, \mathbf{x}_{2,k}^T, \ldots, \mathbf{x}_{n,k}^T]^T$[7]. BSUM performs updates of the form

$$\mathbf{x}_{i,k+1} \in \underset{\mathbf{x}_i}{\text{argmin}} \;\; l_i(\mathbf{x}_i, \mathbf{x}_k), \forall i = 1, 2, \ldots, n. \tag{2.39}$$

Obviously, at each iteration k the ith block is updated given the latest known estimates. It should be noted though that, in slight deviation from BCD, the update of the ith block of BSUM now possibly utilizes the most recent estimate of the same block that is updated.

[7] It should be highlighted that $\mathbf{x}_i$ denotes the ith block whereas, as defined, $\mathbf{x}_k$ is a vector containing all the n blocks ($i = 1, 2, \ldots, n$) as they are known at iteration k.

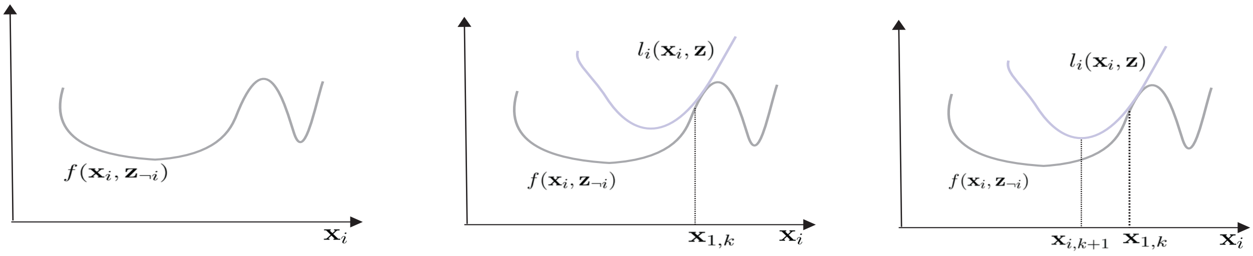

$f(\mathbf{x}_i, \mathbf{z}_{\neg i})$
$\mathbf{x}_i$
$l_i(\mathbf{x}_i, \mathbf{z})$
$f(\mathbf{x}_i, \mathbf{z}_{\neg i})$
$\mathbf{x}_{1,k}$
$\mathbf{x}_i$
$l_i(\mathbf{x}_i, \mathbf{z})$
$f(\mathbf{x}_i, \mathbf{z}_{\neg i})$
$\mathbf{x}_{i,k+1}$
$\mathbf{x}_{1,k}$
$\mathbf{x}_i$

Figure 5: Graphical illustration of the block successive upper bound minimization philosophy.

The selection of the blocks per iteration can be determined by disparate rules such as the cyclic rule with $i = (k \mod n) + 1$, which is the classical one and is followed in this thesis, the Gauss-Southwell rule, the Maximum Block Improvement (MBI), [79], etc.

2.1.4.1 Characteristics of upper bound functions and convergence behavior

As mentioned above, approximate functions $l_i(\mathbf{x}_i, \mathbf{x}_k)$ are local tight upper bounds of the original cost function. Since the ultimate goal of BSUM is to guarantee the descent of the original cost function at each update, any candidate approximate function must satisfy some additional conditions listed below:

- $l_i(\mathbf{x}_i, \mathbf{z}) = f(\mathbf{z}), \ \forall \mathbf{z} \in \mathcal{C}, \ \forall i$
- $l_i(\mathbf{x}_i, \mathbf{z}) \geq f(\mathbf{x}_i, \mathbf{z}_{\neg i}) \ \forall \mathbf{x}_i \in \mathcal{C}_i, \ \forall \mathbf{z} \in \mathcal{C}, \ \forall i$
- $l_i^{'}(\mathbf{x}_i, \mathbf{z}; \mathbf{d}_i) \mid_{\mathbf{x}_i = \mathbf{z}_i} = f^{'}(\mathbf{z}; \mathbf{d}) \ \forall \mathbf{d} = [\mathbf{0}^T, \dots, \mathbf{d}_i^T, \dots, \mathbf{0}^T]^T$ s.t. $\mathbf{z}_i + \mathbf{d}_i \in \mathcal{C}_i, \forall i$
- $l_i(\mathbf{x}_i, \mathbf{z})$ is continuous on $(\mathbf{x}_i, \mathbf{z}) \ \forall i$

The first two conditions actually declare the global upper bound property of the approximate functions l_i's, while the third one states that the first-order variations of the original cost function and those of the upper bounds should be the same at the point of approximation. This actually certifies the desirable descent direction of the original cost function which is attained from the updates obtained by minimizing its approximate functions.

Similarly to all algorithms that follow the coordinate-wise optimization rationale, convergence of BSUM methods necessitate what is called *regularity condition* of the objective function at the coordinate-wise minima, [151]. Regularity is formally defined as follows,
Definition 2.2. *Let* $\mathbf{z} \in \mathbf{dom} f$ *denote a coordinate-wise minimum point of the cost function* f*, i.e.,* $f(\mathbf{z} + [\mathbf{0}^T, \dots, \mathbf{d}_i^T, \dots, \mathbf{0}^T]^T) \geq f(\mathbf{z})$ *for all* $\mathbf{z} + [\mathbf{0}^T, \dots, \mathbf{d}_i^T, \dots, \mathbf{0}^T]^T \in \mathcal{C}$*,* $\forall i = 1, 2, \dots, n$*.* $\mathbf{z}$ *is a regular point of* f *iff*

$$f'(\mathbf{z}; \mathbf{d}) \geq 0, \quad \forall \mathbf{d} = [\mathbf{d}_1^T, \dots, \mathbf{d}_n^T]^T, \ \textit{such that} \ \mathbf{z} + \mathbf{d} \in \mathcal{C}. \tag{2.40}$$

According to this definition regularity implies that coordinate-wise minima are stationary points of the cost functions. This property is inherently valid for smooth functions, however it does not in general hold for nonsmooth and nonseparable ones, [120]. Assuming that a) the regularity conditions hold for all the points of the sequence generated by BSUM, b) the adopted approximate functions satisfy the four conditions given above and are quasi-convex and c) the solution of each subproblem is unique, then as stated in [79] (Theorem 1), any limit point of the generated sequence is a stationary point of the original cost function. Moreover, assuming that the level sets $\mathcal{X}^0 = \{\mathbf{x}|f(\mathbf{x}) \leq f(\mathbf{x}_0)\}$ of the cost function f are compact, then the sequence $\{\mathbf{x}_k\}$ generated by BSUM under the cyclic rule updating scheme converges to the set of stationary points of f, i.e., $\lim\limits_{k\to\infty} \quad d(\mathbf{x}_k, \mathcal{X}^\star) = 0$, where $d(\mathbf{x}, \mathcal{X}^\star) \triangleq \min\limits_{\mathbf{x}^\star \in \mathcal{X}^\star} \|\mathbf{x} - \mathbf{x}^\star\|_2$ and $\mathcal{X}^\star$ is the set of the stationary points.

2.1.4.2 Majorization-minimization and the proximal point algorithms

In an effort to show the pervasive nature of the BSUM framework in the optimization field we next give a glance at some well-established methods that can be seen as special instances of the BSUM framework.

Majorization-Minimization (MM) can be considered as a single block case of BSUM. That is, assuming a single-block function $f(\mathbf{x})$, MM is an iterative method that generates a sequence $\mathbf{x}_k$ which successively minimizes at each iteration an upper bound $l(\mathbf{x}, \mathbf{x}_k)$ of $f(\mathbf{x})$ for which it holds

$$l(\mathbf{x}, \mathbf{x}_k) \geq f(\mathbf{x}), \quad l(\mathbf{x}_k, \mathbf{x}_k) = f(\mathbf{x}_k). \tag{2.41}$$

One of the most popular MM techniques is the Expectation-Maximization (EM) algorithm. Let us denote as $\mathbf{y}$ the observed variable, which models the data at hand. In a statistical framework, the Maximum Likelihood (ML) estimate of the vector of unknown parameters $\mathbf{x}$ given $\mathbf{y}$ is defined as

$$\mathbf{x}_{k,ML} = \underset{\mathbf{x}}{\text{argmax}} \ \ln p(\mathbf{y}|\mathbf{x}) \equiv \underset{\mathbf{x}}{\text{argmin}}[-\ln p(\mathbf{y} \mid \mathbf{x})]. \tag{2.42}$$

Assuming now that there exists a hidden/unobserved variable $\mathbf{z}$, the EM algorithm consists of two-step iterations of the following forms: a) E-step: estimate $l(\mathbf{x}, \mathbf{x}_k) = -\langle \ln p(\mathbf{y}, \mathbf{z}|\mathbf{x})\rangle_{\mathbf{z}|\mathbf{y},\mathbf{x}_k}$ (where $\langle\cdot\rangle_{\mathbf{z}|\mathbf{y},\mathbf{x}_k}$ denotes the expectation operator w.r.t. the posterior distribution $p(\mathbf{z} \mid \mathbf{y}, \mathbf{x}_k)$) and b) M-step: $\mathbf{x}_{k+1} = \text{argmin}_{\mathbf{x}} \ l(\mathbf{x}, \mathbf{x}_k)$. By using the Jensen's inequality it can be easily shown that $l(\mathbf{x}, \mathbf{x}_k)$ majorizes (locally around $\mathbf{x}_k$) the negative log-likelihood function $-\ln p(\mathbf{y} \mid \mathbf{x})$ and hence EM can be considered as a MM algorithm.

Finally, proximal point algorithms described by the iterative equation, $\mathbf{x}_{k+1} = \text{prox}_{\lambda f}(\mathbf{x}_k)$ (where $f(\mathbf{x})$ is assumed to be a convex function) can also be viewed as MM schemes. Recall that the proximal operator of the cost function f is the minimizer of the infimal convolution of f with the function $\frac{1}{2}\|\mathbf{x} - \mathbf{x}_k\|_2^2$ and as a result, for $\lambda \geq 0$, it is an upper bound of f. Hence, the proximal point algorithm is, likewise MM, a BSUM approach and the particular approximate function is called as proximal upper bound, [79].

2.1.5 Variational Bayes approximation

Next, the variational Bayes approximation method is outlined. Before delving into the core of variational Bayes approximation, a brief outline of the main principles of the Bayesian inference philosophy is given.

2.1.5.1 Bayesian Inference

So far, the various optimization methods that have been presented share a common thread: they return single point estimates for the unknown parameters of interest, which are obtained from the available observations (data). No doubt, data in all occasions are uncertain and noisy. Under a statistical perspective, data may be considered as samples (in most cases i.i.d.) of a random variable.

In Bayesian statistics, the unknown model parameters that govern the probability distribution of the data are considered as random variables[8]. These random variables are *unobserved* and hence are called *latent* or *hidden*. That said, instead of retrieving single point estimates of the model parameters based on the data at hand, the goal now is to infer the distribution of latent variables (i.e., to perform Bayesian inference) by conditioning them on data using their prior distribution and the likelihood, which encapsulates the underlying mechanism that relates the observed and the latent variables. To put it in mathematical terms, let us denote as $\mathbf{y} = [y_1, y_2, .., y_n]^T$ the random vector of observations and $\mathbf{x} = [x_1, x_2, \dots, x_m]^T$ the m-dimensional vector of latent variables. The joint probability distribution of $\mathbf{y}$ and $\mathbf{x}$ equals

$$p(\mathbf{y}, \mathbf{x}) = p(\mathbf{y}|\mathbf{x})p(\mathbf{x}), \tag{2.43}$$

where $p(\mathbf{y}|\mathbf{x})$ and $p(\mathbf{x})$ denote the likelihood function and the prior distributions respectively. Bayesian inference utilizes the Bayes rule for computing the posterior distribution $p(\mathbf{x}|\mathbf{y})$ as follows

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{y})}. \tag{2.44}$$

However, in complex Bayesian models that are usually met in machine learning and signal processing applications, the exact computation of the posterior distribution can not be computed exactly. This is due to the intractability of the marginal likelihood $p(\mathbf{y})$ also called *evidence* which amounts to the computation of the integral

$$p(\mathbf{y}) = \int p(\mathbf{y}, \mathbf{x}) d\mathbf{x}. \tag{2.45}$$

In many practical situations, the evidence integral either can not be computed in polynomial time or is not available in closed form. Thus, one has to resort to approximation schemes.

[8] This random nature of the parameters reflects the uncertainty we have regarding their true values.

2.1.5.2 Approximate inference and the evidence lower bound

The first approximate inference method that came into the scene several decades ago is the ubiquitous Markov Chain Monte Carlo (MCMC) method. MCMC lies at the heart of many landmark algorithms such as the Gibbs sampler and Metropolis Hasting, [146]. MCMC algorithms provide guarantees as to the production of exact samples from the target distribution asymptotically. However, despite this favorable characteristic, MCMC algorithms are not the best choice when one should deal with large-scale datasets or highly complex models. In such occasions, the need for computationally efficient approximate inference algorithms becomes imperative, even at the expense of the absence of theoretical guarantees.

An alternative to MCMC that departs from the sampling approach and follows an optimization viewpoint instead, is the Variational Bayes (VB) approach, [21, 152]. The crux of VB is to approximate the exact posterior distribution of the latent variables, by suitably formulating a constrained optimization problem. As it will be further explained later, VB does not enjoy theoretical guarantees as is the case with MCMC algorithms. However, it is much faster than MCMC and thus suits better to large-scale datasets.

More specifically, all candidate approximate posterior distributions, denoted as $q(\mathbf{x})$, are assumed to originate from a specific family of distributions $\mathcal{L}$. In VB approximation, the optimal posterior distribution $q^\star(\mathbf{x})$ is the one that minimizes the *Kullback Leibler (KL) divergence criterion* with respect to the exact posterior distribution, i.e.,

$$q^\star(\mathbf{x}) = \operatorname*{argmin}_{q(\mathbf{x})\in\mathcal{L}} \; KL(q(\mathbf{x}) \parallel p(\mathbf{x}|\mathbf{y})). \tag{2.46}$$

KL divergence (also called *relative entropy function*) is defined as

$$KL(q(\mathbf{x}) \parallel p(\mathbf{x}|\mathbf{y})) = \langle \ln q(\mathbf{x})\rangle - \langle \ln p(\mathbf{x}|\mathbf{y})\rangle \tag{2.47}$$

where expectations (denoted by $\langle\cdot\rangle$) are taken w.r.t. $q(\mathbf{x})$. Expanding the exact posterior of Eq. (2.47) we get

$$KL(q(\mathbf{x}) \parallel p(\mathbf{x}|\mathbf{y})) = \langle \ln q(\mathbf{x})\rangle - \langle \ln p(\mathbf{y},\mathbf{x})\rangle + \ln p(\mathbf{y}). \tag{2.48}$$

The above equation shows that KL minimization involves again the evidence $p(\mathbf{y})$ and thus it can not be computed. To this end, a lower bound of the logarithm of the evidence is maximized in place of the KL divergence. This lower bound is called *Evidence Lower Bound (ELBO)* and equals the negative KL plus log $p(\mathbf{y})$, i.e.,

$$\text{ELBO}(q(\mathbf{x})) = \langle \ln p(\mathbf{y},\mathbf{x})\rangle - \langle \ln q(\mathbf{x})\rangle. \tag{2.49}$$

2.1.5.3 Variational Bayes inference using the mean field approximation

As mentioned above, the approximate inference procedure which is adopted by VB, leads to the maximization of the ELBO subject to $q(\mathbf{x}) \in \mathcal{L}$, i.e., the approximate posterior is restricted to belong to a family of distributions denoted as $\mathcal{L}$. In fact, the VB scheme assumes that this family is the so-called mean-field variational family of distributions. Assuming $\mathbf{x} = [\mathbf{x}_1^T, \mathbf{x}_2^T, \dots, \mathbf{x}_m^T]^T$, $q(\mathbf{x})$ can be written in the form

$$q(\mathbf{x}) = \prod_{i=1}^{m} q_i(\mathbf{x}_i), \tag{2.50}$$

i.e., the coordinates of the vector **x** are assumed to be statistically independent random vectors. A visualization of the mean-field approximation can be seen in Fig. 6 for the case of Gaussian distributions.

By adhering to the mean-field approximation, it now becomes much easier to maximize the ELBO. This is actually achieved by following a BCD strategy, i.e., a single latent vector-variable $\mathbf{x}_i$ is updated at each iteration (see the BCD algorithm of Section 2.1.4). More specifically, by exploiting the statistical independence assumption of the mean-field approximation we may rewrite the ELBO given in (2.49) as a function of the ith coordinate $q_i(\mathbf{x}_i)$ as follows

$$\text{ELBO}(q_i(\mathbf{x}_i)) = \langle\langle \ln p(\mathbf{y}, \mathbf{x}_i, \mathbf{x}_{\neg i})\rangle_{\neg i}\rangle_i - \langle \ln q_i(\mathbf{x}_i)\rangle_i + \text{const} \tag{2.51}$$

where the constant term (const) absorbs all the remaining terms that are independent of the ith coordinate. Note that $\langle\cdot\rangle_{\neg i}$ denotes expectation w.r.t. the approximate posterior of $\mathbf{x}_{\neg i}$. The expression in (2.51) equals the negative KL divergence up to an added constant. It can be shown that the optimal $q_i^{\star}(\mathbf{x}_i)$ can be obtained in closed-form and is proportional to the exponentiated expected (w.r.t. to $q_{\neg i}(\mathbf{x}_{\neg i})$) logarithm of the joint probability distribution, i.e.,

$$q_i^{\star}(\mathbf{x}_i) \propto \exp\{\langle \ln p(\mathbf{y}, \mathbf{x}_i, \mathbf{x}_{\neg i})\rangle_{\neg i}\}. \tag{2.52}$$

Variational Bayes inference is then performed in an iterative fashion, where each iteration involves m steps. At each one of them, a posterior distribution $q_i(\mathbf{x}_i)$ is estimated via (2.52), based on the most recent estimates of the others. Thus, at the completion of each iteration, a new estimate of $q(\mathbf{x})$ is obtained. Due to the nonconvexity of the KL divergence measure which is maximized, the sequence of the VB updates converges to a local optimal point of it.

2.1.5.4 Relation to the EM algorithm and the BSUM framework

It can be seen from Eq. (2.49) that the first term of ELBO is the expected joint loglikelihood w.r.t. the approximate posterior $q(\mathbf{x})$. Taking into account the E-step of the EM[9] algorithm described in the previous section, it can be easily derived that EM is a special case of the ELBO maximization framework which is followed in the VB approximation schemes. More specifically, EM arises for the case that the approximate posterior $q(\mathbf{x})$ equals to the

[9] **x** is now used for denoting the latent variable, instead of **z**, which is utilized for the EM algorithm in Section 2.1.4.2. Moreover variable **y** is now assumed to contain both **y** and **x** of the EM algorithm.

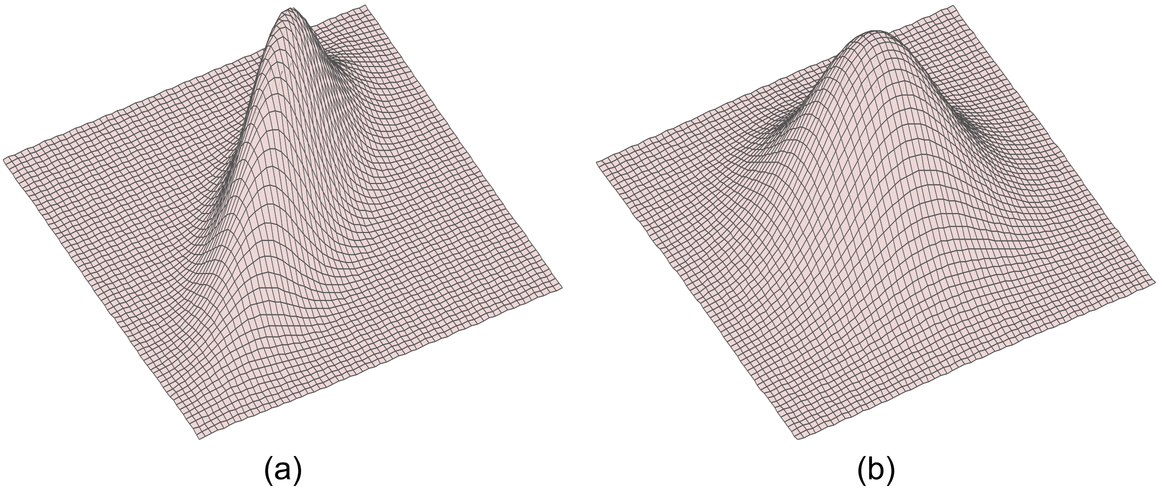


Figure 6: An example of the mean field approximation as illustrated in a 2d-space for the case of a Gaussian distribution. (a) True posterior pdf and (b) approximate posterior pdf.

exact posterior $p(\mathbf{x}|\mathbf{y})$, which in the EM algorithm is considered to be known[10]. Moreover, similarly to the EM algorithm, variational Bayes is related to the MM algorithms and as a result to the BSUM framework which was the subject of Section 2.1.4. This is so, since each update of VB arises by a maximization step of a lower-bound of the evidence function, which is equivalent to minimizing an upper bound function.

2.1.6 Online learning, stochastic approximation and empirical risk minimization

All *structured matrix estimation* problems presented in Chapter 1 are implicitly assumed to be *ill-posed* since, in general, unknown matrices can not be *uniquely* estimated. This fact may be ascribed either to the models that have been adopted (for instance the matrix factorization problem is ill-posed by definition) or to the *insufficient* information conveyed by the available data[11]. An additional underlying assumption that has been made is that available data are i.i.d. samples of a random variable whose probability distribution is *unknown*. Hence, it is obvious that data that we use for estimating the unknown matrices are merely *snapshots of the real world*. Our aim is to exploit as much as we can these empirical observations (which play the role of the training samples) so as to get reliable estimates of the unknown parameters (vectors/matrices).

Let us denote as $\boldsymbol{y}$ the random variable that models the data and $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n]$ the set of available data. For ease of notation we focus on a vector of unknown parameters $\mathbf{x}$, which can be obtained by minimizing an objective function $f(\mathbf{x})$. This function can be considered as a measure of the deviation between the reconstructed model, based on the

[10] Note that the M-step analogue of the EM algorithm is usually obsolete in variational Bayes schemes since all parameters are treated as latent variables.

[11] For instance, in the supervised scenario outlined in Chapter 1, this insufficiency may come from the presence of noise or from the fact that the number of measurements is less than the number of the unknown parameters.

estimate $\hat{\mathbf{x}}$ of $\mathbf{x}$, and the true model, represented here by $\mathbf{y}$. That said, $f(\mathbf{x})$ can be defined as the expectation of a certain loss function $l(\mathbf{y}, \mathbf{x})$ w.r.t. the *unknown* probability density function of $\mathbf{y}$, i.e.,

$$f(\mathbf{x}) \overset{def}{=} \langle l(\boldsymbol{y}, \mathbf{x}) \rangle. \tag{2.53}$$

An estimate of $\mathbf{x}$ is thus obtained by minimizing $f(\mathbf{x})$, i.e.,

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\text{argmin}} \;\; f(\mathbf{x}). \tag{2.54}$$

However, since the distribution of $\boldsymbol{y}$ is unknown the expected risk (cost) function (2.53) cannot be *exactly* obtained and hence (2.54) is insolvable. To deal with this issue, a surrogate of (2.54) may be used, called *empirical risk*, [130], and defined as $f(\mathbf{x}) \approx \sum_{t=1}^{n} l(\mathbf{y}_t, \mathbf{x})$. This gives rise to the following *empirical risk minimization* problem

$$\min_{\mathbf{x}} \;\; \frac{1}{n} \sum_{t=1}^{n} l(\mathbf{y}_t, \mathbf{x}). \tag{2.55}$$

General theorems provided in [153] have shown that empirical risk minimization provides a good estimate of the minimum of the expected risk for sufficiently large size n of the training set.

Empirical risk minimization may take place by applying a *batch* gradient descent algorithm[12], which gives rise to updates of $\mathbf{x}$ in the following form

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \lambda^k \nabla_{\mathbf{x}} \left(\frac{1}{n} \sum_{t=1}^{n} l(\mathbf{y}_t, \mathbf{x}_k) \right) \equiv \mathbf{x}_k - \lambda^k \frac{1}{n} \sum_{t=1}^{n} \nabla_{\mathbf{x}} l(\mathbf{y}_t, \mathbf{x}_k), \tag{2.56}$$

which converge to a local minimum of the empirical risk for small enough step sizes λ^k, [15]. It can be seen from (2.56) that the updates of the batch gradient descent algorithm involve the task of calculating the average of the gradients of the cost function over the entire training set, which becomes computationally cumbersome especially in the case where data are of large-scale and/or high-dimensional, since this fact necessitates the availability of huge computational and memory resources.

Departing from the aforementioned *batch* type of processing, online learning schemes were introduced in the early 1950's in both engineering (in the form of *recursive adaptive algorithms*) and the field of learning systems, [22]. The whole mathematical framework that online learning is built upon is named after *stochastic approximation*. In the case of the gradient descent algorithm, by applying stochastic approximation we are led to the so-called *online* (also known as stochastic) *gradient descent algorithm*, which deviates from the batch one in the following sense: $\mathbf{x}$ is updated by using each time just one $\mathbf{y}_t$ and the

[12] Differentiability of $l(\mathbf{y}_t, \mathbf{x})$ is next assumed for simplicity reasons.

expected risk $f(\mathbf{x})$ is approximated at each iteration t by $l(\mathbf{y}_t, \mathbf{x})$[13], i.e.,

$$f(\mathbf{x}) = \langle l(\mathbf{y}, \mathbf{x}) \rangle \approx l(\mathbf{y}_t, \mathbf{x}). \tag{2.57}$$

Hence gradient iterations are modified as follows,

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \lambda_t \nabla_{\mathbf{x}} l(\mathbf{y}_t, \mathbf{x}_t). \tag{2.58}$$

Note that iterations now become sample dependent (this is notationally shown by replacing the iteration index k with t) and the averaging operation of the gradient of the empirical risk is now dropped.

Apart from its advantages in dealing with large-scale and/or high-dimensionality problems, this online learning scheme is of critical importance when one is dealing with cases evolving in nonstationary environments where the statistics of data change over time. For this reason, this stochastic approximation based formulation has been at the core of adaptive algorithms whose premise is to a) simultaneously process an observation and b) learn to perform better, by embedding the knowledge carried by the previously processed observation. Clearly, computations are now much simpler hence the derived algorithms are of much lower computational complexity while at the same time memory requirements are eliminated. Finally, it is noted that convergence analysis of online algorithms has been analytically studied in [22].

2.1.6.1 Stochastic variational inference

Stochastic approximation ideas described above have been applied in the variational Bayes approximation scheme presented in Section 2.1.5 with the goal to devise algorithms amenable to processing massive amounts of data. The derived scheme arises by first viewing the coordinate ascent updates of variational Bayes approximation (which leads to expressions (2.52)) as *natural gradient steps*[14], [5] with appropriately selected step size, [125]. Then, the natural gradient of the ELBO defined in (2.51) is relaxed by using stochastic approximation. The derived scheme belongs to the online learning framework inheriting all the merits described above with provable convergence guarantees, [78].

2.2 Application to hyperspectral image processing

So far the main problems that are addressed in this thesis have been presented in Chapter 1, while in the first part of the present chapter the utilized optimization tools were described. Herein, we outline the applications that we will focus on, which give birth to a part of the afore-said problems, i.e., hyperspectral image unmixing and denoising.

[13] Obviously, $l(\mathbf{y}_t, \mathbf{x})$ is an unbiased estimate of $f(\mathbf{x})$.

[14] Natural gradient steps are iterates in the form $\mathbf{x}_{k+1} = \mathbf{x}_k - \lambda^k \mathbf{G}^{-1}(\mathbf{x}_k)\nabla f(\mathbf{x}_k)$, where $\mathbf{G}$ is a Riemannian metric. In the VB case $\mathbf{G}$ corresponds to the Fisher information matrix of the approximate posterior distribution $q(\mathbf{x})$, [21]. The term $\mathbf{G}^{-1}(\mathbf{x}_k)\nabla f(\mathbf{x}_k)$ is called the natural gradient, [5].

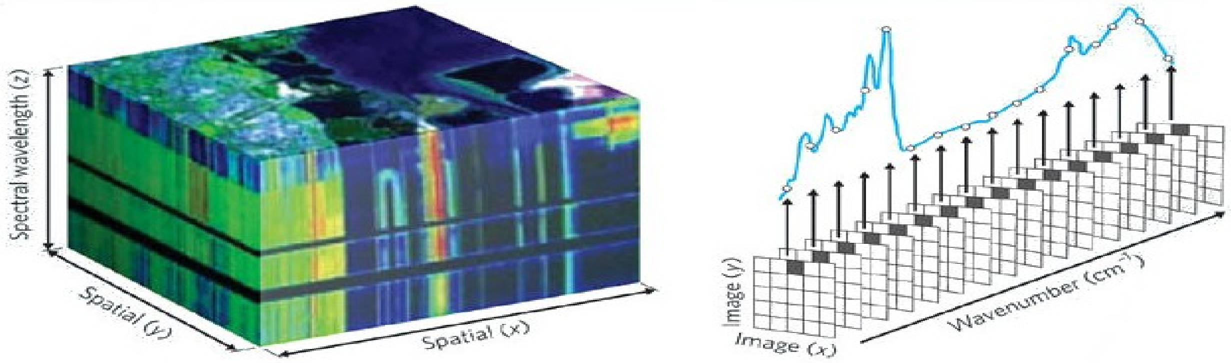


Figure 7: A hyperspectral image (left) and the pixel's spectral signature (right), [11].

2.2.1 Hyperspectral images

A *hyperspectral image* (HSI) is a collection of multiple grayscale images captured at many contiguous spectral bands (channels) with wavelengths ranging from the visible to the infrared spectrum (0.4–2.5 μm), thus forming a so-called *spectral cube* (Fig. 7). As a result of this, each pixel in a HSI is represented by a vector of size equal to the number of spectral bands called *spectral signature* of the pixel. The entries of this vector are the (nonnegative) radiance or reflectance values of the spatial area corresponding to the pixel in all spectral channels. Obviously, HSIs provide much more detailed information about the depicted scene as compared to conventional RGB images, which capture only three spectral channels (red, green, blue) and multispectral images that comprise a few (usually less than ten) spectral channels.

The rich spectral information of HSIs can be proved valuable in numerous tasks such as material identification, object detection, etc. These tasks are at the core of many application fields, such as earth observation and remote sensing, mineral detection, medical image processing, food quality assessment, etc., [99]. This is the reason why HSIs have gained extreme popularity over the past few decades.

The refined spectral information, which is provided by HSIs, usually comes at a price. Specifically, HSIs are *high-dimensional and large-scale data* (the number of the pixels is usually in the order of tens to hundreds of thousands and the number of the spectral bands is in the order of hundreds to thousands for modern hyperspectral sensors) requiring high computational cost for processing and extraction of information. Moreover, in many HSI applications such as remote sensing, the high spectral resolution is overshadowed by low spatial resolution. These two caveats are at the core of several hyperspectral imaging problems that are addressed by various algorithmic procedures. Along these lines, the problems of hyperspectral unmixing and hyperspectral denoising are next described.

2.2.2 Hyperspectral image unmixing

Hyperspectral unmixing (HU) has attracted considerable attention in recent years both in research and applications. HU is based on the assumption that each pixel in the image stems from the mixing of a set of some basic spectral signatures corresponding to pure materials (*endmembers*) (see Fig. 8). It generally involves the following two processing stages a) identification of the spectral signatures of the pure materials (endmembers) whose mixing generates the *mixed* pixels of HSI (endmembers' extraction stage - EE) and b) estimation of their corresponding fractions (*abundances*) in the formation of each HSI pixel (abundance estimation stage - AE), [99]. The latter constitutes the so-called *abundance vector* of the pixel (the matrix that contains the collection of abundance vectors corresponding to the pixels of the HSI is called *abundance matrix*). This two step procedure has given rise to a plethora of methods tackling either one or both of these two tasks. Diverse statistical and geometrical approaches have been lately put forward in literature addressing the first step, i.e., endmembers' extraction -EE (e.g. [110, 93]). On the other hand, there have been many research works that assume that the spectral signatures of the endmembers are available and focus on the abundance estimation task. The latter case is known as supervised scenario, in contrast to the general form of the HU problem given above, which corresponds to the unsupervised scenario.

Algorithms that fall into this class need to make a fundamental assumption concerning the inherent mixing process that generates the spectral signatures of the HSI pixels. In view of the latter, the linear mixing model (LMM) can be defined as

$$\mathbf{Y} = \mathbf{A}\mathbf{X} + \mathbf{E}, \tag{2.59}$$

where $\mathbf{Y}$ contains the l-dimensional spectral signatures of the n HSI's pixels, $\mathbf{A}$ is the $l \times m$ endmembers' matrix (consisting of m endmembers' spectral signatures) and $\mathbf{X}$ is the $m \times n$ abundance matrix which contains the set of the abundance vectors $\boldsymbol{x}_i$ corresponding to the n pixels of the HSI. Moreover, matrix $\mathbf{E}$ stands for the i.i.d. zero-mean Gaussian noise matrix. LMM holds a dominant position being widely adopted in numerous state-of-the-art unmixing algorithms (see e.g., [99] and the references therein). Abundance estimation is henceforth treated as a linear regression problem. The LMM has prevailed over other models, due to its conceptual simplicity and mathematical tractability. Physical considerations that naturally arise impose various constraints on the unmixing problem. In this context, the so-called *abundance nonnegativity* and the *abundance sum-to-one* constraints are usually adopted. That said, unmixing can be viewed as a constrained linear regression problem which is mathematically formulated as

$$\min_{\mathbf{X}} \ \|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F^2 \quad \text{subject to} \quad \mathbf{X} \geq \mathbf{0}, \ \ \|\boldsymbol{x}_i\|_2 = 1, \forall i = 1, 2, \ldots, n. \tag{2.60}$$

In an attempt to achieve better abundance estimation results, recent novel ideas promote the incorporation of further prior knowledge in the unmixing problem. In light of this, several methods bring into play the *sparsity* assumption on the abundance vector (or matrix), [141, 19, 81, 145]. Its adoption is justified by the fact that only a few of the available endmembers participate in the formation of a given mixed pixel, especially in the case of large size endmembers' dictionaries. Put it in other terms, it is envisaged that pixels' spectral signatures accept sparse representations with respect to a given endmembers' dictionary.

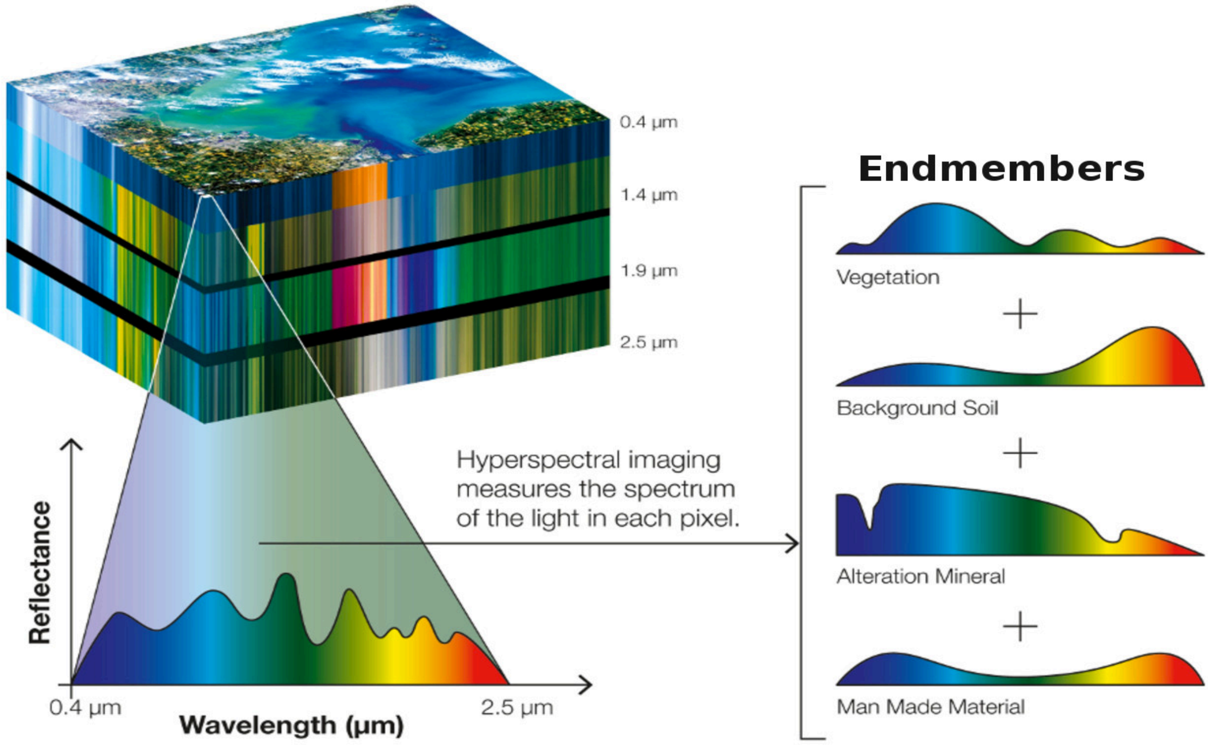


Figure 8: A visual illustration of hyperspectral unmixing, [2].

Furthermore, one could also say that the abundance vectors corresponding to the pixels of HSIs are deemed having only a few nonzero values. Practically speaking, sparsity is imposed on abundances by means of ℓ_1 norm regularization, [141, 19, 81] when a deterministic approach is followed. On the other hand, in Bayesian schemes appropriate sparsity inducing priors are adopted for the abundance vectors, [145, 123]. *Spatial correlation* is another constraint that has recently been incorporated in the unmixing process, offering stimulating results, [45, 119, 82]. In that vein, the redundant information that exists in homogeneous regions of HSIs is subject to exploitation. Actually, in such regions, there is a high degree of correlation among the spectral signatures of neighboring pixels. It is hence anticipated that there should also be correlation among the abundance vectors corresponding to these pixels. This has led to the development of novel unmixing schemes, whereby the information provided by the neighboring pixels is taken into account in the abundance estimation of each single pixel.

In case that the endmembers' dictionary is considered also unknown, linear unmixing becomes much more challenging since it now becomes an *unsupervised* estimation task. In this case, there are two main routes to follow. According to the first one, the EE and AE processes are performed one after the other independently. Considering the EE stage, a large number of works have come into the scene in the literature for addressing this problem. These can be classified into two main categories: a) geometrical and b) statistical methods. The former exploit geometrical features of the mixtures by assuming that under the LMM, the spectral signatures of the mixed pixels form a simplex whose vertices correspond to the endmembers' signatures, [110, 158]. Simplex based methods are based on the assumption that there exist pure pixels in the scene (known as PPI - pure pixel assumption), which is rather strong in several cases in practice. After the completion of the EE stage, that is, after estimating matrix **A**, the AE stage is performed in order to estimate the abundances of each pixel, based on **A**. According to the second route, EE and AE are treated simultaneously. In other words, HU is viewed as a blind source separation (BSS) problem. In that framework, HU has been formulated as an independent component analysis scheme which assumes statistical independence for both the abundance matrix and the endmembers' dictionary, [12, 109]. Taking into account the nonnegativity of HSI data, HU can be formulated as a nonnegative matrix factorization (NMF) problem, i.e.,

$$\min_{\mathbf{A},\mathbf{X}} \ \|\mathbf{Y}-\mathbf{A}\mathbf{X}\|_F^2, \ \text{ subject to } \ \mathbf{A} \geq \mathbf{0}, \ \ \mathbf{X} \geq \mathbf{0}, \|\mathbf{x}_i\|_2 = 1, \forall i = 1,2,\ldots,n. \tag{2.61}$$

NMF HU problems suffer from inherent obstacles related to the non-uniqueness of the factorization **AX**. To overcome these, several constrained versions of the NMF have been put forth in the literature specialized on the hyperspectral unmixing problem. In that framework, sparsity, [98], structured sparsity, [118], and volume constraints, [105], have been incorporated in the unsupervised HU problem thus transforming it to a multiple constrained NMF problem.

2.2.3 Hyperspectral image denoising

Hyperspectral image denoising is a preprocessing step in most of the HSI applications. Depending on the application, the noise that corrupts a HSI may have different characteristics. These partly determine the data generation process, giving rise to the adoption of diverse models. The most common assumptions that are made in practice are a) the noise is of additive Gaussian i.i.d. type and b) HSIs are grossly corrupted by spikes of noise which occur at random spectral and spatial positions, [163]. In mathematical terms the two different sorts of noise can be modelled as follows,

$$\mathbf{Y} = \mathcal{P}_\Omega(\mathbf{X}) + \mathbf{E}, \tag{2.62}$$

where the set Ω contains the indexes of **Y** that are assumed to contain uncorrupted by spikes of noise information. It is quite clear that Eq. (2.62) implies that HSI denoising can be modelled as a generalized version of the matrix completion problem (see Section 1.2.3.2).

Denoising is in general an ill-posed problem and hence necessitates the exploitation of prior knowledge that we may have, [166]. Fortunately, HSIs are intrinsically characterized by high coherence in both the spectral and the spatial domain. That said, the low-rank structure can be effectively leveraged in HSI denoising. These properties actually render HSIs highly compressible, i.e., HSIs can be represented in a given, e.g., wavelet or learned dictionary in a parsimonious way. HSI denoising can thus be formulated as a structured matrix estimation problem.

3. SIMULTANEOUSLY SPARSE, LOW-RANK AND NONNEGATIVE MATRIX ESTIMATION

In this chapter, we address the problem of simultaneously sparse, low-rank and nonnegative matrix estimation. Two novel formulations of the problem are proposed. The first one introduces a mixed penalty term, which consists of the sum of the weighted ℓ_1 and the weighted nuclear norm of the sought matrix. This penalty term is then used to regularize a conventional quadratic cost function and impose simultaneously sparsity and low-rankness on the sought matrix, [67]. The second approach follows a different paradigm when it comes to low-rank imposition. That is, low-rank is now enforced via assuming a matrix factorization (MF) representation of the matrix to be estimated. Since the inner dimension of the factorization -which is related to the rank of the factorized matrix- is in general unknown, the variational form of the nuclear norm is utilized for penalizing the rank thereof, [67]. Both approaches are utilized for formulating in a pioneering way the problem of hyperspectral unmixing (HU) (described in Section 2.2.2). It is shown that both sparsity and low-rankness may be incorporated in HU accounting for physical considerations of this problem such as spatial correlation. Then, three different optimization algorithms are introduced. Specifically, the regularized cost function arising by the first formulation is minimized by a) an *incremental proximal sparse and low-rank* unmixing algorithm and b) an algorithm based on the *alternating direction method of multipliers* (ADMM). Moreover, a block coordinate descent type algorithm is used for minimizing the emerging nonconvex MF based cost function of the second approach. The main premise of this approach is that the induced computational complexity is further reduced by virtue of the MF formulation which significantly decreases the “size” of the optimization problem. The effectiveness of the proposed algorithms is illustrated in experiments conducted on a wealth of simulated and real HSI data experiments.

3.1 Problem formulations

In the following, it is assumed that nonnegative data are generated by a linear model and contaminated by additive Gaussian i.i.d. noise, i.e.,

$$\mathbf{Y} = \mathbf{A}\mathbf{X} + \mathbf{E}, \tag{3.1}$$

where $\mathbf{Y} \in \mathcal{R}_{+}^{l\times n}$ is the data matrix, $\mathbf{A} \in \mathcal{R}_{+}^{l\times m}$ is a known matrix which corresponds to a given dictionary, $\mathbf{X} \in \mathbf{R}_{+}^{m\times n}$ is the coefficients matrix and $\mathbf{E} \in \mathcal{R}^{l\times n}$ is a Gaussian i.i.d. noise matrix. As it was clearly explained in Chapter 1, the problem of estimating matrix $\mathbf{X}$ given $\mathbf{Y}$ and $\mathbf{A}$ is, in general, *ill-posed* for $l \neq m$ and $\mathbf{E} = \mathbf{0}$ (noiseless case), let alone in the noisy case of (3.1). This fact urges us to exploit any prior knowledge we may have regarding $\mathbf{X}$ so as to find “good” approximate solutions for it.

Along these lines, we next focus on the estimation of a matrix $\mathbf{X}$ that is assumed to be simultaneously sparse, low-rank and nonnegative (see Fig. 9). The task of simultane-

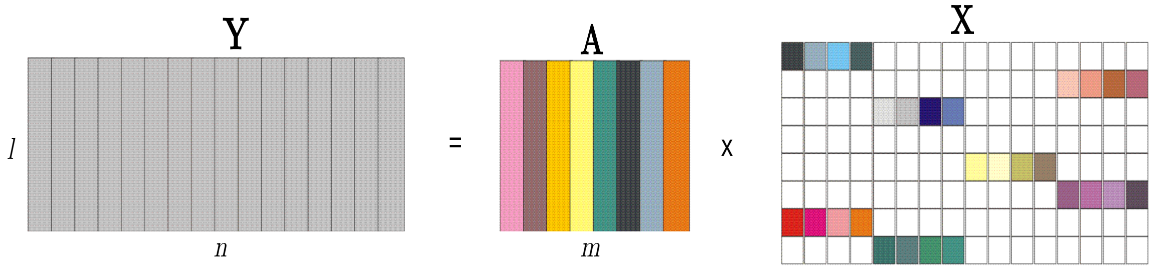


Figure 9: Sparse and low-rank matrix estimation under the linear model (white cells correspond to zero values).

ously sparse and low-rank matrix estimation and the proposed in the literature formulations thereof have been presented in sections 1.1.2 and 1.2.2, respectively. As it was described, convex combinations of the low-rank imposing nuclear norm and the sparsity promoting ℓ_1 norm have been reported in the literature for providing polynomial time algorithms that approximately solve the originally NP-hard optimization problem that involved the sum of the ℓ_0 quasinorm and the rank functions (see Section 1.2.2).

3.1.1 A weighted ℓ_1 and weighted nuclear norm minimization approach

Next we aim at exploiting the recently shown benefits arising by the use of weighted versions of the ℓ_1 and nuclear norms as compared to their nonweighted counterparts, when it comes to sparse and low-rank imposition respectively, [168, 30, 95, 88]. To this end, we propose to use a combination of the *weighted* ℓ_1 and nuclear norms, defined in Section 1.2.1, for efficiently addressing the problem of simultaneously sparse and low-rank matrix estimation. Since weighted versions of ℓ_1 and nuclear norm (if the weights are suitably selected) approximate better the ℓ_0 quasinorm and the rank, [30, 95], such an approach is expected to further enhance the sparsity on both the elements of **X**, x_{ij}, and the singular values $\sigma_i(\mathbf{X})$.

We hence come up with the following novel formulation of simultaneously sparse low-rank and nonnegative matrix estimation,

$$(\text{P1}):\ \hat{\mathbf{X}} = \underset{\mathbf{X}\in\mathcal{R}_+^{m\times n}}{\text{argmin}} \left\{\frac{1}{2}\|\mathbf{Y}-\mathbf{AX}\|_F^2 + \gamma\|\mathbf{X}\|_{1,\mathbf{D}} + \tau\|\mathbf{X}\|_{*,\mathbf{w}}\right\}. \tag{3.2}$$

where $\gamma, \tau \geq 0$ are parameters that control the trade-off between the sparsity and rank regularization terms and the data fidelity term. Being parametrized, (P1) becomes flexible enough to impose either one of the two structures on **X**. For example, by setting $\gamma = 0$, (P1) results in searching for a low-rank matrix. Accordingly, setting $\tau = 0$ is tantamount to searching for a sparse matrix. The flexibility of the proposed model provides certainly an advantage over either low-rank or sparse estimation methods, as it will also be demonstrated later in the experimental results section of this chapter.

Assumption 3.1. *For the weighting coefficients w_i of the nuclear norm it holds that $w_i = w$, $i = 1, 2, \ldots, \min(m, n)$.*
Remark 3.1. *Under Assumption 3.1, the weighted nuclear norm is convex [74, 95], while the weighted ℓ_1 norm is always convex for nonnegative* **D**. *Thus, the overall cost function of (P1) is convex.*
Note that (P1) is a nontrivial problem to solve, due to the nondifferentiable form of the ℓ_1 and nuclear norm regularizers, [7]. In Section 3.3, we suitably explore two optimization tools (presented in Chapter 2) to tackle this problem; an incremental proximal minimization method and an ADMM based technique. Note that both algorithms are derived in a convex setting under the Assumption 3.1. Alternative options for the selection of parameters **D** and **w**, such as reweighting schemes, that render the problem nonconvex, yet offering enhanced estimation performance, are discussed in Section 3.3.3.

3.1.2 A matrix factorization (MF) based approach

Next, an alternative formulation of simultaneously sparse, low-rank and nonnegative matrix estimation problem is presented. The motivation behind the new approach is to provide a formulation which gives rise to an optimization algorithm of reduced computational complexity. It should be emphasized that the minimization of the nuclear norm (and its weighted version) requires a singular value decomposition (SVD) step, whose complexity when applied on a matrix of size $m \times n$ (assuming $m \leq n$) is $\mathcal{O}(m^2 n + m^3)$. This fact can be a serious impediment for high-dimensional and large-scale data applications, such as hyperspectral unmixing, which will be the subject of the next section.

Capitalizing on this, we propose a different way for minimizing the rank of **X** by utilizing the variational form of the nuclear norm described in Section 1.2.3.2. Recall that this form is based on a bilinear representation of **X**, i.e., **X** may now be written as the product of two matrices $\mathbf{U} \in \mathcal{R}^{m \times d}$ and $\mathbf{V} \in \mathcal{R}^{n \times d}$, i.e., $\mathbf{X} = \mathbf{U}\mathbf{V}^T$, where d is an overestimate of the rank of **X**. As mentioned in Section 1.2.3.2, the variational form of the nuclear norm is a tight upper bound of it, i.e.,

$$\|\mathbf{X}\|_* = \min_{\mathbf{U} \in \mathcal{R}^{m \times d}, \mathbf{V} \in \mathcal{R}^{n \times d}, \mathbf{X} = \mathbf{U}\mathbf{V}^T} \frac{1}{2} \left(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2 \right). \tag{3.3}$$

The minimization of the righthand side of (3.3) (in place of the nuclear norm), gives rise the following optimization problem,

$$(\text{P2}) : \left\{ \hat{\mathbf{U}}, \hat{\mathbf{V}}, \hat{\mathbf{X}} \right\} = \underset{\mathbf{U}, \mathbf{V}, \mathbf{X} \geq \mathbf{0}}{\operatorname{argmin}} \; L(\mathbf{U}, \mathbf{V}, \mathbf{X}), \tag{3.4}$$

where,

$$L(\mathbf{U}, \mathbf{V}, \mathbf{X}) = \left\{ \frac{1}{2} \|\mathbf{Y} - \mathbf{A}\mathbf{U}\mathbf{V}^T\|_F^2 + \frac{\tau}{2} \left(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2 \right) + \gamma \|\mathbf{X}\|_{1,\mathbf{D}} + \frac{\mu}{2} \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2 \right\}. \tag{3.5}$$

It should be noted that the term $\|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2$ that appears in (3.5) is associated with the constraint $\mathbf{X} = \mathbf{U}\mathbf{V}^T$, with μ being the corresponding Lagrange multiplier parameter. Problem (P2) is nonconvex with respect to matrices $\mathbf{X}, \mathbf{U}$ and $\mathbf{V}$. Moreover, the presence of the ℓ_1 norm induces the objective function a nonsmooth behavior that must be suitably handled. In Section 3.3.4, a novel block coordinate descent (BCD) algorithm, [15], is presented that solves (P2) for abundance matrix estimation in hyperspectral unmixing.

3.2 HU as a sparse, low-rank and nonnegative matrix estimation problem

Before we proceed with the presentation of the algorithms that solve the above problems, we will show the potential of the above formulation in the framework of a real world application, namely, the abundance matrix estimation task. That is, we will show how the latter problem can be written in terms of the previously defined formulation. It is noted that the abundance matrix estimation task is, for the first time in the literature, viewed as a simultaneously sparse, low-rank and nonnegative matrix estimation problem. In our analysis we will adhere to the linear mixing model (LMM) presented in Section 2.2.2.

Note that due to physical considerations, matrix $\mathbf{X}$, which now contains the abundance coefficients, should satisfy two constraints, namely, the *abundance nonnegativity* and the *abundance sum-to-one* constraints, [87], i.e.,

$$\mathbf{X} \geq \mathbf{0}, \text{ and } \mathbf{1}^T\mathbf{X} = \mathbf{1}^T. \tag{3.6}$$

Nevertheless, in the following we relax the sum-to-one constraint based on the reasoning presented in [81]. That said, the general problem that is dealt with can be described as: "given the spectral measurements $\mathbf{Y}$ and the endmember matrix $\mathbf{A}$, estimate the abundance matrix $\mathbf{X}$ subject to the nonnegativity constraint". This is a typical inverse problem, which has been addressed via many methods in the signal processing literature. However, the efficacy of the proposed approach lies on the exploitation of intrinsic structural characteristics of $\mathbf{X}$, namely, sparsity and low-rankness, as it will be explained next. Before we proceed with the proposed HU formulation, in the next subsection we pass briefly over the related work.

3.2.1 Related work

In an attempt to achieve better abundance estimation results, recent novel ideas promote the incorporation of further prior knowledge in the unmixing problem. In light of this, several methods bring into play the *sparsity* assumption, [141, 19, 81, 145]. Its adoption is justified by the fact that (in practice) only a few of the available endmembers participate in the formation of a given pixel, especially in the case of large size endmembers' dictionaries. Put it in other terms, it is envisaged that pixels' spectral signatures accept sparse representations with respect to a given endmembers' dictionary; that is, the corresponding abundance vectors are deemed to have only a few nonzero values. Practically, sparsity is imposed on abundances via the ℓ_1 norm regularization term, [141, 19, 81], when a deterministic approach is followed. On the other hand, in Bayesian schemes appropriate sparsity inducing priors are adopted for the abundance vectors, [145, 123].

Spatial correlation is another constraint that has recently been incorporated in the unmixing process, offering stimulating results, [45, 119, 82]. In that vein, the additional information that exists in homogeneous regions of HSIs is subject to exploitation. Actually, in such regions, there is a high degree of correlation among the spectral signatures of neighboring pixels. It is hence anticipated that there should also be correlation among the abundance vectors corresponding to these pixels. This has led to the development of novel unmixing schemes, whereby the information related to the neighboring pixels of each single pixel is taken into account in the abundance estimation of the latter.

In this spirit, a collaborative deterministic scheme, termed CLSUnSAL, was recently proposed in [82], which uses a wealth of information stemming from all the pixels of the examined HSI. CLSUnSAL adopts dictionaries consisting of a large number of endmembers. Then it assumes that spatial correlation translates into abundance vectors sharing the same support set, i.e., presenting a similar sparsity pattern. Thus, the corresponding abundance matrix should meaningfully be of a joint-sparse structure. To impose joint-sparsity, CLSUnSAL applies a $\ell_{1,2}$ norm on the sought abundance matrix **X**, which is then used to penalize a suitably defined quadratic cost function. Minimization of the resulting regularized cost function is performed by an alternating direction method of multipliers (ADMM), [24]. A similar perspective is followed in [119], however in a “localized” fashion. Specifically, [119] proposes the use of a 3×3 square window that slides in the spatial dimensions of the image. The abundance vector of the central pixel is then inferred by taking into account the spectral signatures of the adjacent pixels contained in the window. Based on this idea, two algorithms are derived. First the MMV-ADMM, which in a similar to CLSUnSAL fashion, seeks joint-sparse abundance matrices utilizing the $\ell_{1,2}$ norm, and second the LRR algorithm that promotes a low-rank structure on the abundance matrix. Actually, the LRR algorithm presents an alternative way of modelling the spatial correlation among neighboring pixels. That is, it assumes that the correlation among pixels’ spectral signatures is reflected as linear dependence among their corresponding abundance vectors. Apparently, the matrix formed by these abundance vectors should be of low rank. That said, a nuclear norm is imposed on the abundance matrix, and a properly adapted augmented Lagrangian cost function is minimized in an alternating minimization fashion.

3.2.2 Proposed HU formulation

Herein, we impose concurrently two naturally justified *structural constraints* on the abundance matrix **X**, that promote *low-rankness* and *sparsity*.

It is worth mentioning that the sparsity of **X** does by no means invalidate its low-rankness. On the contrary, both structural hypotheses are assumed to hold *simultaneously* on **X**, although low-rankness implicitly imposes some kind of “regular” structure on sparsity[1]. So

[1] That is, there exist subsets of columns of **X**, $\mathbf{X}_i$, with the following property: The columns of a certain $\mathbf{X}_i$ exhibit zero values at certain places (mainly due to the low-rank constraint). However, some of them may

far, reports in the spectral unmixing literature explore either the sparsity, e.g., [145, 140], or the low-rankness property of $\mathbf{X}$, e.g., [119]. To the best of our knowledge this is the first time that spectral unmixing is formulated as a simultaneously sparse and low-rank matrix estimation problem. That is, we seek a matrix $\mathbf{X} \geq \mathbf{0}$ that, apart from fitting the data well in the least squares sense, it has minimum rank and the minimum number of positive elements.

3.3 Proposed algorithms

After the short parenthesis of Section 3.2, where it has been shown how the HU application can be written in terms of the proposed problem formulation (Section 3.1), we proceed with the next issue that naturally arises, namely, the algorithms that solve the above problem. More specifically, in this section we present three algorithms to address the nonsmooth, constrained, optimization problems described in Section 3.1, developed within the context of hyperspectral unmixing. When it comes to the weighted norms based formulation, two algorithms are proposed: the first one comes from the family of incremental proximal algorithms, presented in Section 2.1.2, and makes use of the proximal operators of all the terms appearing in $(\mathrm{P1})$ (3.2), while the second exploits the splitting strategy of the ADMM philosophy described in Section 2.1.3.3. Moreover, a BCD algorithm (see Section 2.1.4) is devised for solving the MF based formulation of the simultaneously sparse, low-rank and nonnegative matrix estimation problem (P2) defined through (3.4) and (3.5).

3.3.1 Incremental proximal sparse and low-rank unmixing algorithm

Let us first recall from (2.1) that the proximal operator of a function $f(\cdot)$ is defined as,

$$\mathrm{prox}_{\lambda f(\cdot)}(\mathbf{X}) = \underset{\mathbf{W}}{\mathrm{argmin}} \left(f(\mathbf{W}) + \frac{1}{2\lambda} \|\mathbf{W} - \mathbf{X}\|_F^2 \right), \tag{3.7}$$

where $\mathbf{X} \in \mathcal{R}^{m \times n}$ and $\mathbf{W} \in \mathbf{dom} f$. As detailed in Section 2.1.2, incremental proximal algorithms suit perfectly to minimization problems in the form

$$\min_{\mathbf{X} \in \mathcal{C}} \sum_{i=1}^{\rho} f_i(\mathbf{X}) \tag{3.8}$$

where $f_i(\mathbf{X}), i = 1, 2, \ldots, \rho$ are convex functions and $\mathcal{C} \subseteq \mathcal{R}^{m \times n}$ is a closed convex set. Recall also that proximal operators of all f_i's are first derived and then a sequential scheme is defined, in which the proximal operator of $f_i(\mathbf{X})$ is evaluated at the point provided by its predecessor (the proximal operator of $f_{i-1}(\mathbf{X})$), for $i = 2, 3, \ldots, \rho$.

As we may observe, $(\mathrm{P1})$ in (3.2) with $\mathbf{w} = w\mathbf{1}$ has exactly the same form with the minimization problem in (3.8), with respect to $\mathbf{X}$. Embedding the nonnegativity to the cost

exhibit zero values at some additional places (attributed exclusively to the sparsity constraint).

function in (3.2) we obtain the following regularized quadratic loss function,

$$L_1(\mathbf{X}) = \frac{1}{2}\|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F^2 + \gamma\|\mathbf{X}\|_{1,\mathbf{D}} + \tau\|\mathbf{X}\|_{*,\mathbf{w}} + I_{\mathcal{R}_+}(\mathbf{X}), \tag{3.9}$$

where the nonnegativity constraint is now replaced by the indicator function $I_{\mathcal{R}_+}(\mathbf{X})$, which is zero when all $x_{ij} \geq 0, i = 1, 2, \ldots, m, j = 1, 2, \ldots, n$, and $+\infty$ if at least one x_{ij} is negative. Typically, we wish to minimize $L_1(\mathbf{X})$ with respect to $\mathbf{X}$. Notice that $L_1(\mathbf{X})$ is the sum of four convex functions and the incremental proximal algorithm of [17] can be applied directly in our problem. Next, the proximal operators of all four convex functions are obtained. Starting with the least squares fitting term, we readily get

$$\text{prox}_{\lambda\frac{1}{2}\|\mathbf{Y}-\mathbf{A}\cdot\|_F^2}(\mathbf{X}) = (\mathbf{A}^T\mathbf{A} + \lambda^{-1}\mathbf{I}_m)^{-1}(\mathbf{A}^T\mathbf{Y} + \lambda^{-1}\mathbf{X}). \tag{3.10}$$

In contrast to the fitting term, the remaining three terms are nondifferentiable. Thus, the standard optimization tools for differentiable functions cannot be applied for them. The minimization of these terms involves the notion of the soft-thresholding operator. Specifically, the soft-thresholding operator on matrix $\mathbf{X} = [x]_{ij}$ is defined as

$$\text{SHR}_{\mathbf{\Psi}}(\mathbf{X}) = \text{sign}(\mathbf{X}) \max(\mathbf{0}, |\mathbf{X}| - \mathbf{\Psi}), \tag{3.11}$$

where $\mathbf{\Psi} = [\psi]_{ij}$ is the matrix that contains thresholding parameters. Note that the soft-thresholding in (3.11) is performed in an elementwise manner, i.e., $\text{SHR}_{\psi_{ij}}(x_{ij}) = \text{sign}(x_{ij}) \max(0, |x_{ij}| - \psi_{ij})$. Notably, when we apply the soft-thresholding operator on a diagonal matrix, we shrink only the elements belonging to its diagonal. These elements are assumed to be shrinked by thresholding parameters contained in a vector. In the above spirit, we define the singular value thresholding operation by

$$\text{SVT}_{\boldsymbol{\psi}}(\mathbf{X}) = \mathbf{U}_{\mathbf{X}}\, \text{SHR}_{\boldsymbol{\psi}}(\Sigma_{\mathbf{X}})\, \mathbf{V}_{\mathbf{X}}^T$$

where $\mathbf{X} = \mathbf{U}_{\mathbf{X}}\Sigma_{\mathbf{X}}\mathbf{V}_{\mathbf{X}}^T$ is the singular value decomposition (SVD) of $\mathbf{X}$, and $\boldsymbol{\psi}$ is the vector whose entries are the thresholding parameters that reduce the corresponding diagonal elements of matrix $\Sigma_{\mathbf{X}}$. Finally, we define the projection operator on the set of nonnegative real numbers,

$$\mathcal{P}_{\mathcal{R}_+}(v) = \arg\min_{x\in\mathcal{R}_+} |x - v| = \begin{cases} 0, & v < 0 \\ v, & v \geq 0 \end{cases}, \tag{3.12}$$

which is naturally extended to matrices in an elementwise manner.

Utilizing the above definitions, we can compute the proximal operators for all regularizing convex functions in (3.9). Specifically, $\text{prox}_{\gamma\|\cdot\|_{1,\mathbf{D}}}(\mathbf{X})$ is computed by soft-thresholding matrix $\mathbf{X}$ with $\gamma\mathbf{D}$ as follows,

$$\text{prox}_{\gamma\|\cdot\|_{1,\mathbf{D}}}(\mathbf{X}) = \text{SHR}_{\gamma\mathbf{D}}(\mathbf{X}). \tag{3.13}$$

Similarly, the proximal operator of the nuclear norm can be expressed via a soft thresholding operation on the singular values of $\mathbf{X}$, i.e.,

$$\text{prox}_{\tau\|\cdot\|_{*,\mathbf{w}}}(\mathbf{X}) = \text{SVT}_{\tau\mathbf{w}}(\mathbf{X}). \tag{3.14}$$

Moreover, the computation of $\text{prox}_{I_{\mathcal{R}_+}(\cdot)}(\mathbf{X})$ reduces to a projection operation, i.e.,

$$\text{prox}_{I_{\mathcal{R}_+}(\cdot)}(\mathbf{X}) = \mathcal{P}_{\mathcal{R}_+}(\mathbf{X}). \tag{3.15}$$

The proposed incremental proximal sparse, low-rank unmixing (IPSpLRU) algorithm iterates among the proximal operators (3.10), (3.13), (3.14) and (3.15) in a cyclic order until convergence, [17]. IPSpLRU is summarized in Algorithm 3.1.

The incremental proximal approach employed above for deriving IPSpLRU is closely related to the incremental subgradient method, [17] and the parameters λ, γ and τ can be seen as the step sizes of the corresponding subgradient steps. By invoking Proposition 3.2 of [17], it arises that for fixed values of these parameters the incremental proximal algorithm converges to a neighborhood of the optimum, which shrinks to zero as their values are closer to zero. On the other hand, exact convergence to the optimal solution of the cost function is achieved when the values of these step sizes diminish over iterations, while they additionally satisfy certain conditions described in [17]. Herein, the parameters λ, γ and τ are selected to be fixed to positive constants during the execution of the algorithm. In doing so, we sacrifice the accuracy of the estimations in favor of faster convergence.

Algorithm 3.1: The proposed IPSpLRU algorithm

```
Inputs $\mathbf{Y}$, $\mathbf{A}$
Select parameters $\mathbf{D}, \mathbf{w}, \lambda, \tau, \gamma$
Set $\mathbf{R} = (\mathbf{A}^T\mathbf{A} + \lambda^{-1}\mathbf{I}_m)^{-1}$, $\mathbf{P} = \mathbf{A}^T\mathbf{Y}$, $\mathbf{Q} = \mathbf{RP}$
Initialize $\mathbf{X}^0$ and set $k = 0$
repeat
  $\mathbf{X}_{k+1} = \mathbf{Q} + \lambda^{-1}\mathbf{R}\mathbf{X}_k$
  $\mathbf{X}_{k+1} = \text{prox}_{\gamma\|\cdot\|_{1,\mathbf{D}}}(\mathbf{X}_{k+1})$ (3.13)
  $\mathbf{X}_{k+1} = \text{prox}_{\tau\|\cdot\|_{*,\mathbf{w}}}(\mathbf{X}_{k+1})$ (3.14)
  $\mathbf{X}_{k+1} = \text{prox}_{I_{\mathcal{R}_+}(\cdot)}(\mathbf{X}_{k+1})$ (3.15)
until convergence
Output : Estimated matrix $\hat{\mathbf{X}} = \mathbf{X}_{k+1}$
```

Concerning the computational complexity of IPSpLRU, the most complex step is the SVD of the abundance matrix $\mathbf{X}_{k+1}$, which takes place at each iteration and is of the order of $\mathcal{O}(nm^2 + m^3)$, [72]. Note that matrices $\mathbf{R} = (\mathbf{A}^T\mathbf{A} + \lambda^{-1}\mathbf{I}_m)^{-1}$, $\mathbf{P} = \mathbf{A}^T\mathbf{Y}$ and $\mathbf{Q} = \mathbf{RP}$ are computed only once at the initialization stage and thus the first step in the repeat-until loop of the algorithm just requires a fast matrix-by-matrix multiplication. The algorithm converges rapidly and terminates when either the following stopping criterion is satisfied,

$$\frac{||\mathbf{X}_{k+1} - \mathbf{X}_k||_F^2}{||\mathbf{X}_k||_F^2} < \epsilon \tag{3.16}$$

where ϵ is a predefined threshold value, or a preset maximum number of iterations is reached. In the following section we present an alternative approach to solve the same problem by employing a primal-dual ADMM type technique.

3.3.2 Alternating direction method of multipliers for sparse and low-rank unmixing

In this section, we develop an instance of the alternating direction method of multipliers that solves the abundance matrix estimation problem (P1) defined in (3.2). To proceed, we utilize the auxiliary matrix variables $\mathbf{\Omega}_1, \mathbf{\Omega}_2, \mathbf{\Omega}_3$ and $\mathbf{\Omega}_4$ of proper dimensions (similar to [82, 140]), and reformulate the original problem (P1) into its equivalent ADMM form, [24],

$$(\text{P3}): \min_{\mathbf{\Omega}_1,\mathbf{\Omega}_2,\mathbf{\Omega}_3,\mathbf{\Omega}_4} \left\{ \frac{1}{2}\|\mathbf{\Omega}_1 - \mathbf{Y}\|_F^2 + \gamma\|\mathbf{\Omega}_2\|_{1,\mathbf{D}} + \tau\|\mathbf{\Omega}_3\|_{*,\mathbf{w}} + I_{\mathcal{R}_+}(\mathbf{\Omega}_4) \right\} \tag{3.17}$$

$$\text{subject to } \mathbf{\Omega}_1 - \mathbf{AX} = \mathbf{0},\ \mathbf{\Omega}_2 - \mathbf{X} = \mathbf{0},\ \mathbf{\Omega}_3 - \mathbf{X} = \mathbf{0},\ \mathbf{\Omega}_4 - \mathbf{X} = \mathbf{0}$$

Based on (P3), the following augmented Lagrangian function is optimized w.r.t. $\mathbf{X}, \mathbf{\Omega}_1, \mathbf{\Omega}_2$, $\mathbf{\Omega}_3$ and $\mathbf{\Omega}_4$,

$$\begin{aligned} L_2(\mathbf{X}, \mathbf{\Omega}_1, \mathbf{\Omega}_2, \mathbf{\Omega}_3, \mathbf{\Omega}_4) &= \frac{1}{2}\|\mathbf{\Omega}_1 - \mathbf{Y}\|_F^2 + \gamma\|\mathbf{\Omega}_2\|_{1,\mathbf{D}} + \tau\|\mathbf{\Omega}_3\|_{*,\mathbf{w}} + I_{\mathcal{R}_+}(\mathbf{\Omega}_4) \\ &+ \operatorname{tr}\left[\mathbf{\Delta}_1^T(\mathbf{\Omega}_1 - \mathbf{AX})\right] + \operatorname{tr}\left[\mathbf{\Delta}_2^T(\mathbf{\Omega}_2 - \mathbf{X})\right] + \operatorname{tr}\left[\mathbf{\Delta}_3^T(\mathbf{\Omega}_3 - \mathbf{X})\right] + \operatorname{tr}\left[\mathbf{\Delta}_4^T(\mathbf{\Omega}_4 - \mathbf{X})\right] \\ &+ \frac{\mu}{2}\Big(\|\mathbf{AX} - \mathbf{\Omega}_1\|_F^2 + \|\mathbf{X} - \mathbf{\Omega}_2\|_F^2 + \|\mathbf{X} - \mathbf{\Omega}_3\|_F^2 + \|\mathbf{X} - \mathbf{\Omega}_4\|_F^2\Big) \end{aligned} \tag{3.18}$$

where the $l \times n$ matrix $\mathbf{\Delta}_1$, and the $m \times n$ matrices $\mathbf{\Delta}_2, \mathbf{\Delta}_3, \mathbf{\Delta}_4$ are the Lagrange multipliers and $\mu > 0$ is a positive penalty parameter. Note that again the nonnegative weights $\mathbf{D}$ and $\mathbf{w}$ are considered to be constant and Assumption 3.1 for $\mathbf{w}$ also holds here. Defining

$$\mathbf{\Omega} = \begin{bmatrix} \mathbf{\Omega}_1 \\ \mathbf{\Omega}_2 \\ \mathbf{\Omega}_3 \\ \mathbf{\Omega}_4 \end{bmatrix},\ \mathbf{G} = \begin{bmatrix} \mathbf{A} \\ \mathbf{I}_m \\ \mathbf{I}_m \\ \mathbf{I}_m \end{bmatrix} \text{ and } \mathbf{B} = \begin{bmatrix} -\mathbf{I}_l & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I}_m & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{I}_m & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I}_m \end{bmatrix}, \tag{3.19}$$

(3.18) can be written in the following equivalent more compact form as

$$L_3(\mathbf{X}, \mathbf{\Omega}, \mathbf{\Lambda}) = \frac{1}{2}\|\mathbf{\Omega}_1 - \mathbf{Y}\|_F^2 + \gamma\|\mathbf{\Omega}_2\|_{1,\mathbf{D}} + \tau\|\mathbf{\Omega}_3\|_{*,\mathbf{w}} + I_{\mathcal{R}_+}(\mathbf{\Omega}_4) + \frac{\mu}{2}\|\mathbf{GX} + \mathbf{B\Omega} - \mathbf{\Lambda}\|_F^2, \tag{3.20}$$

where $\mathbf{\Lambda} = \left[\mathbf{\Lambda}_1^T\ \mathbf{\Lambda}_2^T\ \mathbf{\Lambda}_3^T\ \mathbf{\Lambda}_4^T\right]^T, \mathbf{\Lambda}_i = (1/\mu)\mathbf{\Delta}_i, i = 1, \ldots, 4$, contains the scaled Lagrange multipliers. Having expressed the augmented Lagrangian function as in (3.20), the ADMM proceeds by minimizing $L_3(\mathbf{X}, \mathbf{\Omega}, \mathbf{\Lambda})$ sequentially, each time with respect to a single matrix variable, keeping the remaining variables fixed at their latest values. The dual variables (Lagrange multipliers) are also updated via a gradient ascent step at the end of each alternating minimization cycle.

To elaborate further on the updating steps of the ADMM, the optimization of $L_3(\mathbf{X}, \mathbf{\Omega}, \mathbf{\Lambda})$

with respect to **X** gives

$$\begin{aligned}\mathbf{X}_{k+1} &= \underset{\mathbf{X}}{\operatorname{argmin}}\ L_3(\mathbf{X}, \boldsymbol{\Omega}_k, \boldsymbol{\Lambda}_k)\\ &= \left(\mathbf{A}^T\mathbf{A} + 3\mathbf{I}_m\right)^{-1}[\mathbf{A}^T(\boldsymbol{\Omega}_{1,k} + \boldsymbol{\Lambda}_{1,k}) + \boldsymbol{\Omega}_{2,k} + \boldsymbol{\Lambda}_{2,k} + \boldsymbol{\Omega}_{3,k} + \boldsymbol{\Lambda}_{3,k} + \boldsymbol{\Omega}_{4,k} + \boldsymbol{\Lambda}_{4,k}].\end{aligned} \tag{3.21}$$

Next, the optimization with respect to $\boldsymbol{\Omega}_1$ is performed as

$$\boldsymbol{\Omega}_{1,k+1} = \underset{\boldsymbol{\Omega}_1}{\operatorname{argmin}}\ L_3(\mathbf{X}_{k+1}, \boldsymbol{\Omega}, \boldsymbol{\Lambda}_k) = \frac{1}{1+\mu}\left(\mathbf{Y} + \mu\left(\mathbf{A}\mathbf{X}_{k+1} - \boldsymbol{\Lambda}_{1,k}\right)\right). \tag{3.22}$$

The remaining auxiliary variables $\boldsymbol{\Omega}_2, \boldsymbol{\Omega}_3$, and $\boldsymbol{\Omega}_4$ are involved in nondifferentiable norms, namely, the weighted ℓ_1 norm, the weighted nuclear norm, and the indicator function, respectively. In this regard, the minimization task with respect to these variables resolves to computing some of the proximity operators that we introduced in the previous section. Minimizing (3.20) with respect to $\boldsymbol{\Omega}_2$ yields

$$\boldsymbol{\Omega}_{2,k+1} = \underset{\boldsymbol{\Omega}_2}{\operatorname{argmin}}\ L_3(\mathbf{X}_{k+1}, \boldsymbol{\Omega}, \boldsymbol{\Lambda}_k) = \mathsf{SHR}_{\gamma\mathbf{D}}(\mathbf{X}_{k+1} - \boldsymbol{\Lambda}_{2,k}). \tag{3.23}$$

In the same vein, $\boldsymbol{\Omega}_3$ is computed by a shrinkage operation,

$$\boldsymbol{\Omega}_{3,k+1} = \underset{\boldsymbol{\Omega}_3}{\operatorname{argmin}}\ L_3(\mathbf{X}_{k+1}, \boldsymbol{\Omega}, \boldsymbol{\Lambda}_k) = \mathsf{SVT}_{\tau\mathbf{w}}(\mathbf{X}_{k+1} - \boldsymbol{\Lambda}_{3,k}). \tag{3.24}$$

Next, for the auxiliary variable $\boldsymbol{\Omega}_4$, a projection onto the nonnegative orthant is required,

$$\boldsymbol{\Omega}_{4,k+1} = \underset{\boldsymbol{\Omega}_4}{\operatorname{argmin}}\ L_3(\mathbf{X}_{k+1}, \boldsymbol{\Omega}, \boldsymbol{\Lambda}_k) = \mathcal{P}_{\mathcal{R}_+}(\mathbf{X}_{k+1} - \boldsymbol{\Lambda}_{4,k}). \tag{3.25}$$

After the updating of the variables $\mathbf{X}, \boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_4$, at a given updating cycle, the scaled Lagrange multipliers in $\boldsymbol{\Lambda}$ are also updated by performing gradient ascent on the dual problem [24], as follows,

$$\begin{aligned}\boldsymbol{\Lambda}_{1,k+1} &= \boldsymbol{\Lambda}_{1,k} - \mathbf{A}\mathbf{X}_{k+1} + \boldsymbol{\Omega}_{1,k+1}\\ \boldsymbol{\Lambda}_{i,k+1} &= \boldsymbol{\Lambda}_{i,k} - \mathbf{X}_{k+1} + \boldsymbol{\Omega}_{i,k+1}, \quad i = 2,3,4\end{aligned} \tag{3.26}$$

The proposed algorithm, termed as Alternating Direction Sparse and Low-Rank Unmixing (ADSpLRU) algorithm, is summarized in Algorithm 3.2. An iteration of ADSpLRU consists of the update steps given in (3.21), (3.22), (3.23), (3.24), (3.25), and (3.26). Its computational complexity per iteration is $\mathcal{O}(lmn + nm^2)$ per iteration, slightly higher than that of IPSpLRU, since it usually holds $l > m$. However, as verified by the simulations of the next section, ADSpLRU requires fewer iterations than IPSpLRU to converge[2], while its convergence is also guaranteed as explained in [46]. Moreover, it achieves a slightly lower steady-state error as compared to IPSpLRU. Recall that under Assumption 3.1 the ob-

[2] The reason for this may be that ADSpLRU manipulates the whole cost function at each step, while IPSpLRU splits the cost function and treats each term individually at every step of the algorithm.

Algorithm 3.2: The proposed ADSpLRU algorithm

```
Inputs $\mathbf{Y}$, $\mathbf{A}$
Select parameters $\mathbf{D}, \mathbf{w}, \mu, \tau, \gamma$
Set $\mathbf{R} = \left(\mathbf{A}^T\mathbf{A} + 3\mathbf{I}_m\right)^{-1}$
Initialize $\boldsymbol{\Omega}_0 = [\boldsymbol{\Omega}_{1,0}, \boldsymbol{\Omega}_{2,0}, \boldsymbol{\Omega}_{3,0}, \boldsymbol{\Omega}_{4,0}]$, $\boldsymbol{\Lambda}_0 = [\boldsymbol{\Lambda}_{1,0}, \boldsymbol{\Lambda}_{2,0}, \boldsymbol{\Lambda}_{3,0}, \boldsymbol{\Lambda}_{4,0}]$ and set $k = 0$
repeat
  $\mathbf{X}_{k+1} = \mathbf{R}[\mathbf{A}^T(\boldsymbol{\Omega}_{1,k} + \boldsymbol{\Lambda}_{1,k}) + \boldsymbol{\Omega}_{2,k} + \boldsymbol{\Lambda}_{2,k} + \boldsymbol{\Omega}_{3,k} + \boldsymbol{\Lambda}_{3,k} + \boldsymbol{\Omega}_{4,k} + \boldsymbol{\Lambda}_{4,k}]$
  $\boldsymbol{\Omega}_{1,k+1} = 1/(1+\mu)\left(\mathbf{Y} + \mu\left(\mathbf{A}\mathbf{X}_{k+1} - \boldsymbol{\Lambda}_{1,k}\right)\right)$
  $\boldsymbol{\Omega}_{2,k+1} = \mathsf{SHR}_{\gamma\mathbf{D}}(\mathbf{X}_{k+1} - \boldsymbol{\Lambda}_{2,k})$
  $\boldsymbol{\Omega}_{3,k+1} = \mathsf{SVT}_{\tau\mathbf{w}}(\mathbf{X}_{k+1} - \boldsymbol{\Lambda}_{3,k})$
  $\boldsymbol{\Omega}_{4,k+1} = \mathcal{P}_{\mathcal{R}_+}(\mathbf{X}_{k+1} - \boldsymbol{\Lambda}_{4,k})$
  $\boldsymbol{\Lambda}_{1,k+1} = \boldsymbol{\Lambda}_{1,k} - \mathbf{A}\mathbf{X}_{k+1} + \boldsymbol{\Omega}_{1,k+1}$
  $\boldsymbol{\Lambda}_{i,k+1} = \boldsymbol{\Lambda}_{i,k} - \mathbf{X}_{k+1} + \boldsymbol{\Omega}_{i,k+1}$, $i = 2, 3, 4$
until convergence
Output : Estimated matrix $\hat{\mathbf{X}} = \mathbf{X}_{k+1}$
```

jective function $L_1(\mathbf{X})$ in (3.9) is convex. In that case, and since matrix $\mathbf{G}$ has full column rank, the convergence conditions defined in [46] are met and if an optimal solution exists, ADSpLRU converges to the global optimum, for any $\mu > 0$. This in turn implies that for the primal and dual residuals $\mathbf{r}_k$, $\mathbf{d}_k$ given by

$$\mathbf{r}_k = \mathbf{G}\mathbf{X}_k + \mathbf{B}\boldsymbol{\Omega}_k,$$
$$\mathbf{d}_k = \mu\mathbf{G}^T\mathbf{B}\left(\boldsymbol{\Omega}_k - \boldsymbol{\Omega}_{k-1}\right)$$

it holds that, $\mathbf{r}_k \to 0$ and $\mathbf{d}_k \to 0$, respectively, as $k \to \infty$. In this work, ADSpLRU terminates when either the following termination criterion

$$\|\mathbf{r}_k\|_2 \leq \zeta \text{ and } \|\mathbf{d}_k\|_2 \leq \zeta \tag{3.27}$$

is satisfied for the primal and dual residuals, where $\zeta = \sqrt{(3m+l)n}\zeta^{rel}$, [24] (the relative tolerance $\zeta^{rel} > 0$ takes its value depending on the application, and in our experimental study has been empirically determined to 10^{-4}), or the maximum number of iterations is reached.

3.3.3 Selection of weighting coefficients and regularization parameters for IPSpLRU and ADSpLRU

As mentioned previously, in both IPSpLRU and ADSpLRU the weighting coefficients $\mathbf{D}$ and $\mathbf{w}$ are predetermined, they remain constant during the execution of the algorithms and satisfy certain constraints. As is widely known, [168, 30, 74], a proper selection of these parameters is quite crucial as for the accuracy of the estimations. In view of this, two potential choices are

a) to select the weighting coefficients based on the least squares estimate[3] $\mathbf{X}^{LS}$ of $\mathbf{X}$ as follows,

$$d_{ij} = \left(\frac{1}{x_{ij}^{LS} + \eta^2}\right) \text{ and } w_i = \left(\frac{1}{\sigma_i(\mathbf{X}^{LS}) + \eta^2}\right), \tag{3.28}$$

where $\eta^2 = 10^{-16}$ is a small constant added to avoid singularities or

b) to allow the adaptation of the weighting coefficients from iteration to iteration based on the current estimate of $\mathbf{X}$ as defined in Section 1.2.1, i.e.,

$$d_{ij}^{k+1} = \left(\frac{1}{x_{ij}^{k+1} + \eta^2}\right) \text{ and } w_i^{k+1} = \left(\frac{1}{\sigma_i(\mathbf{X}_{k+1}) + \eta^2}\right), \tag{3.29}$$

Note that by adopting the latter option we end up with the so-called reweighting norm minimization versions of the problem (P1). It should be noted that both these two options render the minimization problem (P1) nonconvex, since the weighted nuclear norm is known to be convex only if the weights $w_i,\ i = 1, 2, \ldots, \min(m, n)$ are nonnegative and in a nonascending order, [74, 43]. Additionally, the reweighting norm minimization version of the problem is known to be inherently nonconvex, [30], while its theoretical convergence analysis for these cases is difficult to be established[4]. Nevertheless, numerous research works advocate the positive impact of these nonconvex weighted norms on the performance of general constrained estimation tasks [43, 30, 74, 95] as well as in hyperspectral unmixing, [35, 55]. Along this line of thought, the algorithms presented in the previous section are modified by adopting the reweighting scheme given by (3.29). As verified in our empirical study presented in the next section, such an option enhances to a large degree the effectiveness of the proposed algorithms, while no numerical issues have been encountered in our experiments.

As far as the remaining parameters λ and μ are concerned, which control the convergence behavior of IPSpLRU and ADSpLRU, respectively, they take positive values, with μ close to zero and λ on the order of 1. In all our experiments we fixed $\mu = 0.01$ and $\lambda = 0.5$. On the other hand, the low-rank and sparsity promoting parameters τ and γ are chosen via fine-tuning, as is commonly done in relevant deterministic schemes. This is so because the optimal set of these parameters depends on the unknown in advance particular structure of the sought abundance matrix, an issue which is further explained in Section 3.4.

3.3.4 A BCD algorithm for matrix factorization based simultaneously sparse, low-rank and nonnegative matrix estimation

In the present section we focus on the minimization problem (P2) defined in (3.4) and (3.5), where $\mathbf{X}$ is assumed to accept a matrix factorization representation and we derive a relative

[3] The $\mathbf{X}^{LS}$ estimate is the solution of the problem $\min_{\mathbf{X}} \frac{1}{2}||\mathbf{AX} - \mathbf{Y}||_F^2$.

[4] It should be noted that the convergence results for the incremental proximal algorithms provided in [17], do not hold for nonconvex f_i's.

algorithm to solve this problem. Due to the separability of the nonconvex and nonsmooth minimization problem (P2) we are permitted to split it into three distinct subproblems, which can be expressed as follows,

$$\begin{aligned}
&(\mathrm{P2a}) : \hat{\mathbf{U}} = \underset{\mathbf{U}}{\operatorname{argmin}} L(\mathbf{U}, \mathbf{V}, \mathbf{X}),\\
&(\mathrm{P2b}) : \hat{\mathbf{V}} = \underset{\mathbf{V}}{\operatorname{argmin}} L(\mathbf{U}, \mathbf{V}, \mathbf{X}),\\
&(\mathrm{P2c}) : \hat{\mathbf{X}} = \underset{\mathbf{X}}{\operatorname{argmin}} L(\mathbf{U}, \mathbf{V}, \mathbf{X}).
\end{aligned}$$

Each subproblem is convex and can be solved independently, in the frame of a block coordinate descent strategy. As is shown below, the solutions of the above-described problems are interrelated, thus paving the way for an iterative scheme, which converges to a local minimum of the cost function defined in Eq. (3.5).

Solution of (P2a). Since $L(\mathbf{U}, \mathbf{V}, \mathbf{X})$ is differentiable with respect to **U**, **U** can be obtained as the solution of the following equation

$$\hat{\mathbf{U}} : \frac{\partial L(\mathbf{U}, \mathbf{V}, \mathbf{X})}{\partial \mathbf{U}} = \mathbf{0}. \tag{3.30}$$

Calculating the derivative in (3.30) yields

$$\left(\mathbf{A}^T\mathbf{A} + \mu\mathbf{I}_m\right)\mathbf{U}\mathbf{V}^T\mathbf{V} + \tau\mathbf{U} = \left(\mathbf{A}^T\mathbf{Y} + \mu\mathbf{X}\right)\mathbf{V}. \tag{3.31}$$

Next, by setting $\mathbf{P} = \mathbf{A}^T\mathbf{A} + \mu\mathbf{I}_m$, $\mathbf{Q} = \mathbf{V}^T\mathbf{V}$ and $\mathbf{C} = \left(\mathbf{A}^T\mathbf{Y} + \mu\mathbf{X}\right)\mathbf{V}$, (3.31) can be compactly written as

$$\mathbf{PUQ} + \tau\mathbf{U} = \mathbf{C}. \tag{3.32}$$

Eq. (3.32) belongs to the class of the so-called *Stein matrix equations* that, among others, have been widely used in the field of control, [3]. To solve the Stein equation (3.32), we adopt the robust algorithm proposed in [71]. In this algorithm, matrix **P** is reduced to its Hessenberg form $\mathbf{H} = \mathbf{OPO}^T$ and matrix **Q** is suitably replaced by its Schur representation $\mathbf{S} = \mathbf{TQT}^T$, where **O** and **T** are orthogonal matrices. Favorably, the symmetry of both **O** and **T**, renders **H** and **S** to be tri-diagonal and diagonal matrices (as it can be shown by simple algebraic manipulations), respectively. By multiplying both sides of (3.32) from the left and the right by $\mathbf{O}^T$ and **T** respectively, and defining $\mathbf{W} = \mathbf{OUT}^T$, (3.32) is rewritten as:

$$\mathbf{HWS} + \tau\mathbf{W} = \mathbf{F}, \tag{3.33}$$

where $\mathbf{F} = \mathbf{O}^T\mathbf{CT}$. Let us denote by $\mathbf{w}_i$, $\mathbf{f}_i$ the ith columns of **W**, **F** and by s_{ii} the ith diagonal element of **S**. Then, we get from (3.33) the following system of equations

$$\left(s_{ii}\mathbf{H} + \tau\mathbf{I}_m\right)\mathbf{w}_i = \mathbf{f}_i, \; i = 1, ..., d \tag{3.34}$$

which can be solved for $\mathbf{w}_i$ with only $\mathcal{O}(m)$ operations, [42], due to the tri-diagonal form of

H. After estimating **W**, column by column, matrix **U** is obtained by the inverse transform

$$\hat{\mathbf{U}} = \mathbf{O}^T\mathbf{W}\mathbf{T}. \tag{3.35}$$

Solution of (P2b). Similarly to (P2a), the minimization problem (P2b) can be solved as

$$\hat{\mathbf{V}} : \frac{\partial L(\mathbf{U}, \mathbf{V}, \mathbf{X})}{\partial \mathbf{V}} = \mathbf{0}. \tag{3.36}$$

Utilizing $\mathbf{P}, \mathbf{C}$ defined above, (3.36) results to the closed form expression

$$\hat{\mathbf{V}} = \mathbf{C}^T\mathbf{U}\left(\mathbf{U}^T\mathbf{P}\mathbf{U} + \tau\mathbf{I}_d\right)^{-1} \tag{3.37}$$

that requires the computation of a small sized ($d \times d$) matrix inversion.

Solution of (P2c). The optimization problem (P2c) is employed in order to estimate matrix **X**. Considering **U** and **V** as constants, minimization of $L(\mathbf{U}, \mathbf{V}, \mathbf{X})$ with respect to **X** leads to

$$\hat{\mathbf{X}} = \underset{\mathbf{X}}{\text{argmin}}\left\{\|\mathbf{X}\|_{1,\mathbf{D}} + \frac{\mu}{2\gamma}\|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2\right\} \tag{3.38}$$

which is the proximal operator of the weighted ℓ_1 norm on $\mathbf{U}\mathbf{V}^T$. Writing (3.38) as

$$\hat{\mathbf{X}} = \underset{\mathbf{X}}{\text{argmin}}\sum_{i=1}^{m}\sum_{j=1}^{n}\left(d_{ij}|x_{ij}| + \frac{\mu}{2\gamma}\left(x_{ij} - \mathbf{u}_i^T\mathbf{v}_j\right)\right), \tag{3.39}$$

where $\mathbf{u}_i^T$ denotes the ith row of matrix **U** and $\mathbf{v}_j^T$ the jth row of **V**, $\hat{\mathbf{X}}$ can be determined via elementwise soft-thresholding, [115]. Thus, we have that

$$\hat{x}_{ij} = \text{SHR}_{d_{ij}\gamma/\mu}(\mathbf{u}_i^T\mathbf{v}_j). \tag{3.40}$$

In this case we stick to the reweighted version of the ℓ_1 norm thus again elements d_{ij}'s of **D** are selected as defined in (3.29). The solution of (3.38) can be written in a more compact form as

$$\hat{\mathbf{X}} = \text{SHR}_{\mathbf{D}(\gamma/\mu)}(\mathbf{U}\mathbf{V}^T) \tag{3.41}$$

The concluding scheme dubbed ALternating Minimization Sparse Low-Rank Unmixing (ALMSpLRU) algorithm is summarized in Algorithm 3.3, below. It should be noted that the aforementioned nonnegativity constraint is imposed by projecting the estimate of **X** produced after performing the steps associated with the minimization of the previous three sub-problems, onto the nonnegative orthant of $\mathcal{R}^{m\times n}$, i.e., $\mathcal{R}_+^{m\times n}$. Note that the most computationally demanding step is the calculation of matrix **C** requiring $\mathcal{O}(mnd)$ operations per iteration. Finally, as verified by extensive simulations presented in the next Section, the proposed algorithm is robust, and converges after a small number of iterations.

Algorithm 3.3: The proposed ALMSpLRU algorithm

```
Inputs $\mathbf{Y}$, $\mathbf{A}$
Select parameters $\mathbf{D}, \gamma, \tau, \mu$
Set $\mathbf{P} = \mathbf{A}^T\mathbf{A} + \mu\mathbf{I}_m$
Set $[\mathbf{O}, \mathbf{H}] = \text{hess}(\mathbf{P})$                ▷ Hessenberg form of $\mathbf{P}$
Set $\mathbf{B} = \mathbf{A}^T\mathbf{Y}$
Initialize $\mathbf{U}_0, \mathbf{V}_0, \mathbf{X}_0 = \mathbf{U}_0\mathbf{V}_0^T$ and set $k = 0$
repeat
  $\mathbf{Q}_{k+1} = \mathbf{V}_k^T\mathbf{V}_k,$
  $[\mathbf{T}_{k+1}, \mathbf{S}_{k+1}] = \text{schur}(\mathbf{Q}_{k+1}),$          ▷ Schur form of $\mathbf{Q}$
  $\mathbf{C}_{k+1} = (\mathbf{B} + \mu\mathbf{X}_k)\,\mathbf{V}_k,$
  $\mathbf{F}_{k+1} = \mathbf{O}^T\mathbf{C}_{k+1}\mathbf{T}_{k+1},$
  $(s_{ii,k+1}\mathbf{H} + \tau\mathbf{I}_m)\,\mathbf{w}_{i,k} = \mathbf{f}_{i,k+1},$          ▷ $i = 1, 2, \ldots, d$
  $\mathbf{W}_{k+1} = \mathbf{O}\mathbf{U}_k\mathbf{T}_{k+1}^T$
  $\mathbf{U}_{k+1} = \mathbf{O}^T\mathbf{W}_{k+1}\mathbf{T}_{k+1}$
  $\mathbf{V}_{k+1} = \mathbf{C}_{k+1}^T\mathbf{U}_{k+1}\left(\mathbf{U}_{k+1}^T\mathbf{P}\mathbf{U}_{k+1} + \tau\mathbf{I}_d\right)^{-1},$
  $\mathbf{X}_{k+1} = \text{SHR}_{\mathbf{D}(\gamma/\mu)}(\mathbf{U}\mathbf{V}^T),$
  $\mathbf{X}_{k+1} = \mathcal{P}_{\mathcal{R}_+}(\mathbf{X}_{k+1})$
until convergence
Output : Estimated matrix $\hat{\mathbf{X}} = \mathbf{X}_{k+1}$
```

3.4 Experimental results

This section unravels the performance characteristics of the proposed IPSpLRU, ADSpLRU and ALMSpLRU algorithms via experiments conducted both on simulated and real data. As mentioned above, we focus on hyperspectral image unmixing applications. Thus we compare the proposed algorithms with three well-known state-of-the-art unmixing algorithms, namely, the nonnegative constraint sparse unmixing by variable splitting and augmented Lagrangian algorithm (CSUnSAL), [19], the recently reported nonnegative constraint joint-sparse method (MMV-ADMM), [119], and, finally, the (fast) Bayesian inference iterative conditional expectations (BiICE) unmixing algorithm, [123]. The computational complexity (in terms of the number of multiplications) of all tested algorithms is given in Table 2. As shown in the table, the SVD operation, which is required by the proposed IPSpLRU, ADSpLRU algorithms, leads to a higher complexity thereofs as compared to CSUnSAL and BiICE. This actually shows that the exploitation of spatial correlation comes at a certain cost. Moreover, it is noticed that between IPSpLRU and ADSpLRU, the former has lower computational complexity than the latter per iteration, resulting from its more simplistic incremental approach. Notably, ALMSpLRU presents the lowest computational complexity among the proposed algorithms, which is attributed to the matrix factorization formulation that has been adopted.

In what follows, we first refer to the parameters' setting established for all the involved algorithms, and the performance evaluation metrics that are utilized in the experimental

Table 2: Computational complexity of n pixels per iteration.

Algorithm	IPSpLRU	ADSpLRU	ALMSpLRU	CSUnSAl [19]	MMV-ADMM [119]	BiICE [123]
Comput. complex.	$\mathcal{O}(nm^2+m^3)$	$\mathcal{O}(nm^2+nlm)$	$\mathcal{O}(nmd)$	$\mathcal{O}(nm^2)$	$\mathcal{O}(nm^2+nlm)$	$\mathcal{O}(nm^2)$

Table 3: Parameters setting.

Algorithm	τ (rank regularization parameter)	γ (sparsity regularization parameter)	μ	λ
IPSpLRU	$\{0,10^{-10},10^{-9},\ldots,10^{-1}\}$	$\{0,10^{-10},10^{-9},\ldots,10^{-1}\}$	Not applicable	0.5
ADSpLRU	$\{0,10^{-10},10^{-9},\ldots,10^{-1}\}$	$\{0,10^{-10},10^{-9},\ldots,10^{-1}\}$	10^{-2}	Not applicable
ALMSpLRU	$\{0,10^{-10},10^{-9},\ldots,10^{-1}\}$	$\{0,10^{-10},10^{-9},\ldots,10^{-1}\}$	10^{-1}	Not applicable
CSUnSAL	Not applicable	$\{0,10^{-10},10^{-9},\ldots,10^{-1}\}$	10^{-2}	Not applicable
MMV-ADMM	Not applicable	$\{0,10^{-10},10^{-9},\ldots,10^{-1}\}$	10^{-2}	Not applicable

procedure. To corroborate the effectiveness and robustness of the proposed algorithms we execute different types of synthetic data experiments whose detailed description is given below. Finally, we empirically compare the abundance maps as revealed by all examined algorithms, when applied on real hyperspectral images[5].

3.4.1 Setting of parameters and performance evaluation criteria

For simplicity reasons, we use γ for the sparsity imposing parameter in all tested algorithms (except BiICE which has no regularization parameters, [145]), μ for the Lagrange multiplier regularization parameter of the ADMM-type techniques and λ for the (relevant to μ) regularization parameter of IPSpLRU. Additionally, the low-rank promoting parameter of the proposed algorithms is denoted by τ. Parameters τ and γ are fine tuned with 10 different values, as shown in Table 3. On the other hand, the Lagrange multiplier regularization parameter μ and the regularization parameter λ of IPSpLRU, which influence to a less extend the efficiency of the corresponding algorithms, are set to a fixed value. In order to assess the performance of the proposed algorithms and the competing ones, we consider two metrics for the experiments conducted on synthetic data. First, the root mean square error (RMSE),

$$\text{RMSE} = \sqrt{\frac{1}{mn}\sum_{i=1}^{n}\|\hat{\mathbf{x}}_i - \mathbf{x}_i\|_2^2}, \tag{3.42}$$

where $\hat{\mathbf{x}}_i$ and $\mathbf{x}_i$ represent the estimated and actual abundance vectors of the ith pixel respectively, n is the total number of the pixels in the image under study, and m, as mentioned in Section 3.2, stands for the number of endmembers. The second metric, is the signal-to-reconstruction error (SRE),[81], which is defined as the ratio between the power

[5] The MATLAB code of the proposed algorithms is provided at `http://members.noa.gr/parisg/demo_splr_unmixing.zip`

of the signal and the power of the estimation error, and is given by the following formula

$$\text{SRE} = 10\text{log}_{10}\left(\frac{\frac{1}{n}\sum_{i=1}^{n}\|\hat{\mathbf{x}}_i\|_2^2}{\frac{1}{n}\sum_{i=1}^{n}\|\hat{\mathbf{x}}_i - \mathbf{x}_i\|_2^2}\right). \tag{3.43}$$

3.4.2 Experiments on simulated datacubes

In the sequel, m endmembers are randomly selected from the USGS library $\mathbf{Z} \in \mathcal{R}_+^{224\times 498}$, [38], so as to form our endmembers' dictionary **A**. Their reflectance values correspond to $l = 224$ spectral bands, uniformly distributed in the interval $0.4 - 2.5\mu m$. The LMM (see Eq. (3.1)) is then utilized for generating spectral signatures subject to given, different in each experiment, abundance matrices **X**'s.

3.4.2.1 Reweighting coefficients efficiency and convergence behavior of IPSpLRU and ADSpLRU

Herein, we aspire to demonstrate the merits emerging from the utilization of reweighting of **D** and **w** from (3.29), on the estimation performance of the proposed algorithms. In light of this, we consider an abundance matrix of rank 3 and sparsity level 10% (i.e., 10% of its entries are nonzero) corresponding to $m = 50$ endmembers, $n = 9$ pixels. Then, $n = 9$ spectral signatures are generated according to the LMM and contaminated by Gaussian noise such that SNR = 30dB.

For $a = 100$ realizations, Fig. 10 depicts the normalized mean squared estimation error (NMSE) (defined as $\text{NMSE}(k) = \frac{1}{a}\sum_{i=1}^{a}\frac{\|\hat{\mathbf{X}}_{i,k} - \mathbf{X}_i\|_F^2}{\|\mathbf{X}_i\|_F^2}$, where $\mathbf{X}_i$ is the true matrix of the ith realization and $\hat{\mathbf{X}}_{i,k}$ its estimate at the kth iteration) as k evolves over 2000 iterations. Three different cases are investigated, corresponding to: a) updating weighting coefficients from (3.29), b) keeping fixed the weighting coefficients based on (3.28) and c) no weighting coefficients, i.e., the weighted norms degenerate to their nonweighted versions by setting $\mathbf{w} = \mathbf{1}$ and $\mathbf{D} = [\mathbf{1}, \mathbf{1}, \ldots \mathbf{1}]$. As it is clearly evident in Fig. 10, both IPSpLRU and ADSpLRU achieve remarkably higher estimation accuracy in terms of NMSE, when using reweighting as compared to the case that fixed or no weights are employed. It is thus empirically verified that the enhanced efficiency of the reweighted ℓ_1 and nuclear norms, emphatically advocated in [168, 95, 30], is retained when using the sum of these two norms. The price to be paid is that such an option might increase the possibility of numerical instabilities, since the problem is rendered nonconvex and (yet) no theoretical convergence analysis has been established. Nevertheless, it is worthy to mention that, despite the fact that convergence is not theoretically guaranteed, in all our experiments both IPSpLRU and ADSpLRU exhibited a very robust convergence behavior.

It is also noticed that ADSpLRU needs less iterations to converge as compared to IPSpLRU and it converges to a slightly lower NMSE. This results from the inherent nature of the two proposed algorithms, as explained in Section 3.3.2. Interestingly, the faster con-

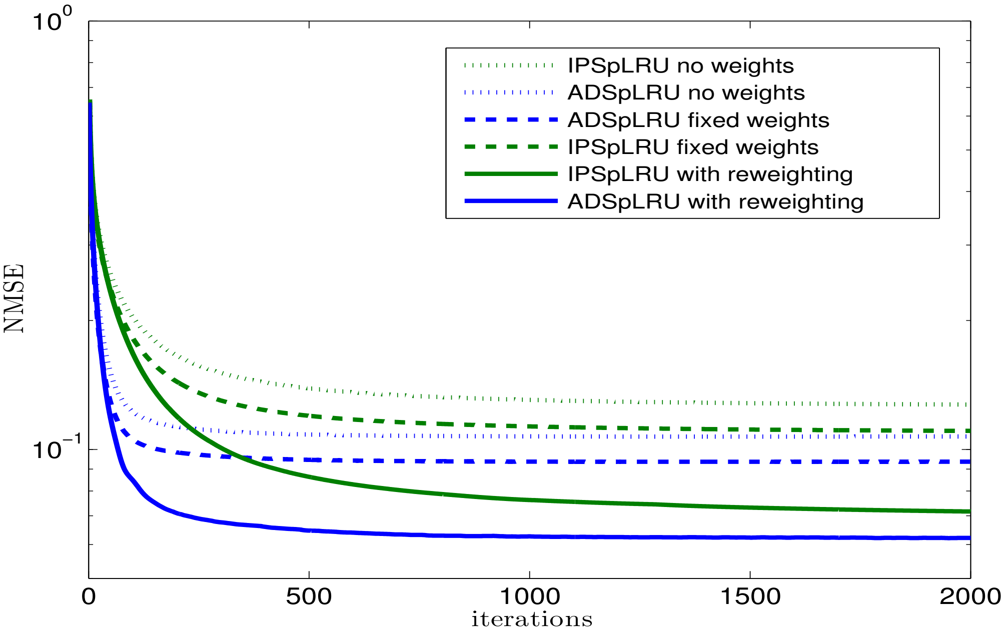


Figure 10: Convergence curves of IPSpLRU and ADSpLRU for a) updating weighting coefficients b) fixed weighting coefficients and c) no weighting coefficients.

vergence rate of ADSpLRU with reweighting comes at the price of its higher per iteration computational complexity as compared to that of IPSpLRU.

3.4.2.2 Two toy examples

In the following experiments our goal is to highlight the significance of the approach followed in this work, i.e., the simultaneous incorporation of sparsity and low-rankness on the abundance estimation problem. To this end, we initially derive the single prior counterparts of IPSpLRU and ADSpLRU. We first focus on the low-rankness assumption, thus the sparsity imposing norm is ignored ($\gamma = 0$). IPSpLRU and ADSpLRU are then reduced to their modified versions, namely, IPLRU and ADLRU respectively. As implied by their names, the aforementioned methods rely exclusively on the low-rank assumption. Similarly, IPSpU and ADSpU are formed by accounting solely for sparsity. That said, IPSpU and ADSpU emerge after dropping the low-rank prior constraint ($\tau = 0$). Next, we generate a $m \times n$ (where $m = 50$ and $n = 9$) simultaneously sparse and low-rank abundance matrix **X** of rank 2 with sparsity level 20%, which is graphically illustrated in Fig. 11a. Using this **X** we generate the $l \times n$ observations matrix **Y** via the LMM in Eq. (3.1), where the noise matrix **E** is Gaussian i.i.d. with SNR=35dB.

Fig. 11 shows the merits of the proposed IPSpLRU and ADSpLRU algorithms. Specifically, it appears that the concurrent exploitation of sparsity and low-rankness leads to significantly more accurate abundance matrix estimates, as compared to their single constraint counterparts, namely, IPLRU, IPSpU and ADLRU, ADSpU respectively. This is clearly seen in terms of the RMSE, as well as from a careful visual inspection of both the recovered abundance matrices and their residuals with the true abundance matrix (i.e. $|\hat{\mathbf{X}} - \mathbf{X}|$), depicted in pair from Fig. 11b - Fig. 11d.

Next, we study the performance of the proposed ALMSpLRU algorithm as compared to the state-of-the-art unmixing algorithms mentioned above. To this end, we randomly select $m = 60$ endmembers from the USGS spectral library and then we generate an $m \times n$ abundance matrix of rank 12 and with $n = 1000$ pixels. The sparsity level is now 10%. Again, the spectral signatures are produced via the LMM and noise of SNR=35dB contaminates the data. The superior performance of ALMSpLRU is illustrated in Fig. 12. As is clearly shown both in terms of RMSE and visually from the recovered abundance matrices and the residual errors (defined as above), ALMSpLRU is proven more efficient than its rivals in estimating the sought abundance matrix. This is so, since ALMSpLRU accounts for both sparsity and low-rankness contrary to the rest sparsity-aware unmixing algorithms.

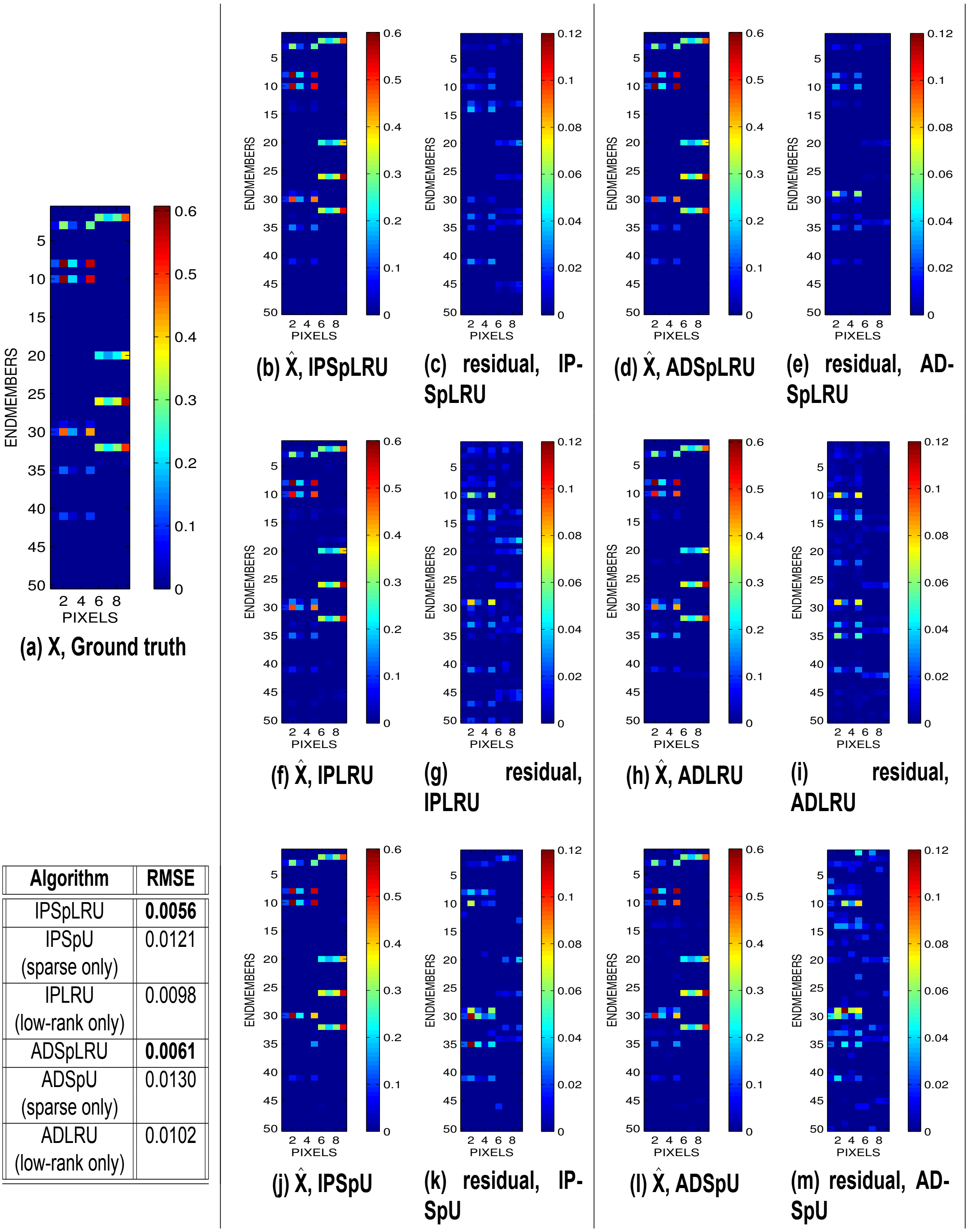
3.4.2.3 The key role of the parameters γ, τ

As explained earlier, parameters γ and τ control the imposition of sparsity and low-rankness, respectively, on the abundance matrix **X**. Herein, we unveil the dependency of the optimal (with respect to RMSE minimization) set of these parameters on the inherent structure of the sought abundance matrix. In this vein, five different types of abundance matrices are generated, each reflecting a specific combination of rank and sparsity level. Next, $n = 9$ linearly mixed pixels are produced, corrupted with Gaussian i.i.d. noise and SNR=35dB. A number of 100 independent realizations is run for each of the five experiments, and the average RMSE of IPSpLRU and ADSpLRU is demonstrated as a function of τ and γ. As shown in Fig. 13, in the first case (Figs. 13a and 13e), which corresponds to solely low-rank abundance matrices (without any presence of sparsity), the sparsity promoting parameter γ does not affect the estimation accuracy. In a similar manner, in the fourth experiment (Figs. 13d and 13h), where the abundance matrix is considered full-rank and sparse, the low-rank promoting parameter has no impact on the estimation performance. Notably, in the other two cases (columns 2 and 3) where both sparse and low-rank abundance matrices are considered, RMSE is minimized for nonzero values of both τ and γ. Such a result is consistent with the fundamental premise of our algorithms, which is the improvement in the abundance matrix estimation by simultaneously exploiting sparsity and low-rankness.

Moreover, the above results indicate that the optimal choice of τ, γ depends on the particular structure (sparse and/or low-rank) of the abundance matrix. Thus, a proper selection of these parameters shall involve fine-tuning schemes, which are commonplace when it comes to algorithms dealing with regularized inverse problems.

3.4.2.4 Performance in the presence of noise

In this experiment we aim at exhibiting the performance of the proposed algorithms in the presence of white and correlated noise corruption. To this end, we first stick with a



(a) X, Ground truth

(b) $\hat{X}$, IPSpLRU

(c) residual, IP-SpLRU

(d) $\hat{X}$, ADSpLRU

(e) residual, AD-SpLRU

(f) $\hat{X}$, IPLRU

(g) residual, IPLRU

(h) $\hat{X}$, ADLRU

(i) residual, ADLRU

(j) $\hat{X}$, IPSpU

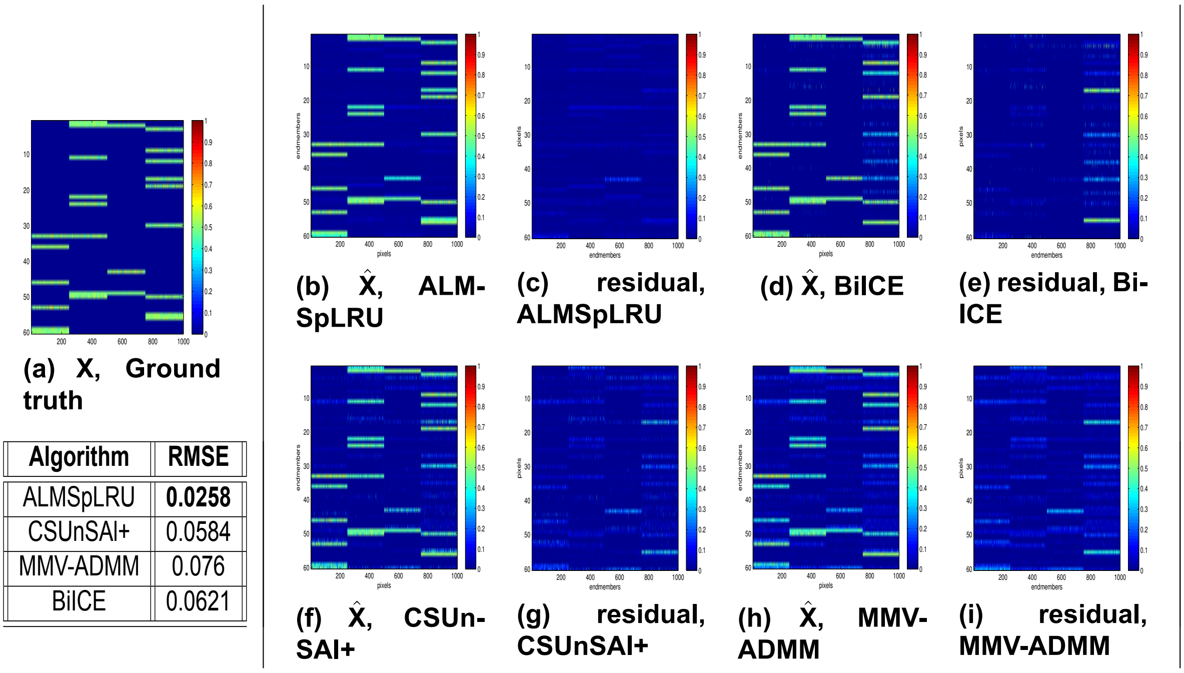
(k) residual, IP-SpU

(l) $\hat{X}$, ADSpU

(m) residual, AD-SpU

Algorithm	RMSE
IPSpLRU	**0.0056**
IPSpU (sparse only)	0.0121
IPLRU (low-rank only)	0.0098
ADSpLRU	**0.0061**
ADSpU (sparse only)	0.0130
ADLRU (low-rank only)	0.0102

Figure 11: Comparison of the results of the proposed sparse and low-rank algorithms versus their sparse only and low-rank only counterparts.



(a) X, Ground truth

Algorithm	RMSE
ALMSpLRU	**0.0258**
CSUnSAl+	0.0584
MMV-ADMM	0.076
BiICE	0.0621

(b) $\hat{X}$, ALMSpLRU

(c) residual, ALMSpLRU

(d) $\hat{X}$, BiICE

(e) residual, BiICE

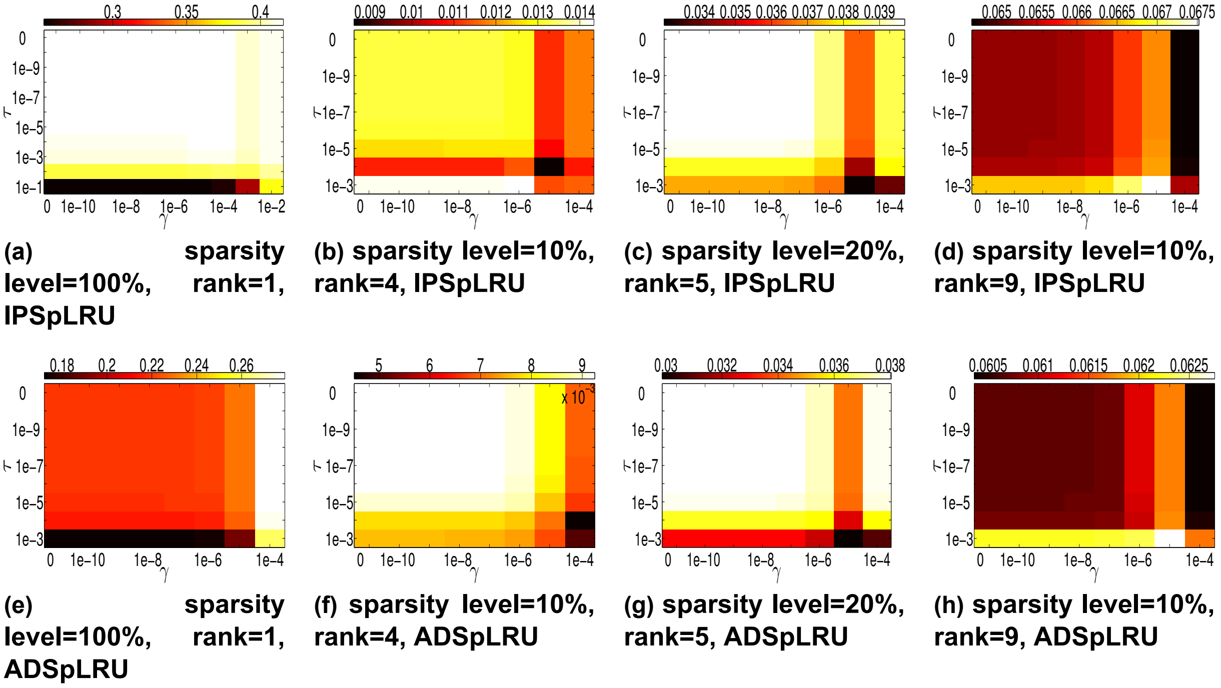
(f) $\hat{X}$, CSUnSAl+

(g) residual, CSUnSAl+

(h) $\hat{X}$, MMV-ADMM

(i) residual, MMV-ADMM

Figure 12: Performance comparison of ALMSpLRU and CSUnSAl+, MMV-ADMM and BiICE in estimating simultaneously sparse and low-rank abundance matrices.



(a) sparsity level=100%, rank=1, IPSpLRU

(b) sparsity level=10%, rank=4, IPSpLRU

(c) sparsity level=20%, rank=5, IPSpLRU

(d) sparsity level=10%, rank=9, IPSpLRU

(e) sparsity level=100%, rank=1, ADSpLRU

(f) sparsity level=10%, rank=4, ADSpLRU

(g) sparsity level=20%, rank=5, ADSpLRU

(h) sparsity level=10%, rank=9, ADSpLRU

Figure 13: RMSE as a function of the low-rankness and the sparsity regularization parameters τ and γ, respectively.

Table 4: RMSE and SRE vs SNR comparison between ALMSpLRU and the competing schemes.

Algorithm	SNR = 15dB		SNR = 20dB		SNR = 25dB		SNR = 30dB		SNR = 35dB	
	RMSE	SRE	RMSE	SRE	RMSE	SRE	RMSE	SRE	RMSE	SRE
ALMSpLRU	0.141	2.30	0.116	3.80	0.096	5.45	0.064	8.90	0.058	9.84
CSunSAL	0.174	0.467	0.148	1.73	0.134	2.61	0.093	5.63	0.068	8.37
MMV-ADMM	0.146	1.95	0.1246	3.25	0.122	3.40	0.093	5.60	0.074	7.71
BiICE	0.221	-1.78	0.165	0.26	0.134	2.50	0.089	5.78	0.087	6.36

specific simultaneously sparse and low-rank abundance matrix **X** of sparsity level 20% and rank 3. Based on this **X**, $n = 9$ linearly mixed pixels are generated, in the same way as described in Section 3.4.2. Then, depending on the case, white or colored Gaussian noise contaminates the data. Sixteen SNR values are considered ranging from 10 to 40 dB, while 100 realizations are run for each SNR value, and the mean of the RMSE and SRE metrics is calculated.

White Gaussian Noise: Fig. 14 shows the RMSE and SRE curves obtained for the proposed IPSpLRU, ADSpLRU and the three competing algorithms, namely, CSUnSAL, MMV-ADMM and BiICE. It is easily seen that both IPSpLRU and ADSpLRU attain remarkably better results comparing to CSUnSAL, MMV-ADMM and BiICE in all the examined SNR values. Additionally, we note that ADSpLRU performs slightly better as compared to IPSpLRU, especially for SNR values greater than 32dB. The price to be paid is that the computational complexity per iteration of ADSpLRU is higher than that of IPSpLRU. It is hence shown that sparse and low-rank methods are robust to different levels of white noise. At the same time, IPSpLRU and ADSpLRU outperform the sparse only CSUnSAL and BiICE algorithms as well as the joint-sparse MMV-ADMM algorithm, provided that the abundance matrix exhibits both sparsity and low-rankness. Next we focus on the robustness of the proposed ALMSpLRU algorithm to noise corruption. Towards this, the same process detailed above is followed for generating $n = 60$ linearly mixed pixels, out of $m = 60$ randomly selected from the USGS library endmembers, utilizing $m \times n$ simultaneously sparse and low-rank abundance matrices of rank 4 and sparsity level 10%. The experiment is executed 10 times for SNR values 15,20,25,30 and 35. In Table 4, the average RMSE and SRE values corresponding to each SNR is given. As it is easily observed, ALMSpLRU is proven again competent in estimating more accurately the simultaneously sparse and low-rank abundance matrix than the other state-of-the-art unmixing algorithms, in all tested cases corresponding to corruption of data by disparate noise levels.

Colored Gaussian Noise: Actually, in real hyperspectral images the noise that corrupts the data is rather structured than white. Thus, to assess the behavior of the proposed methods in such realistic conditions, we simulate correlated Gaussian noise that adds up to the linearly mixed pixels. Fig. 15 illustrates the effectiveness of the tested algorithms in terms of RMSE and SRE, for different SNR values. Therein as well, we can see that IPSpLRU and ADSpLRU achieve better results than their competing algorithms in the whole range of the examined SNRs. Furthermore, ADSpLRU performs better for high SNR values (> 32dB), as compared to IPSpLRU. As a result, the robustness of our proposed methods is also corroborated in the presence of correlated noise with different magnitude.

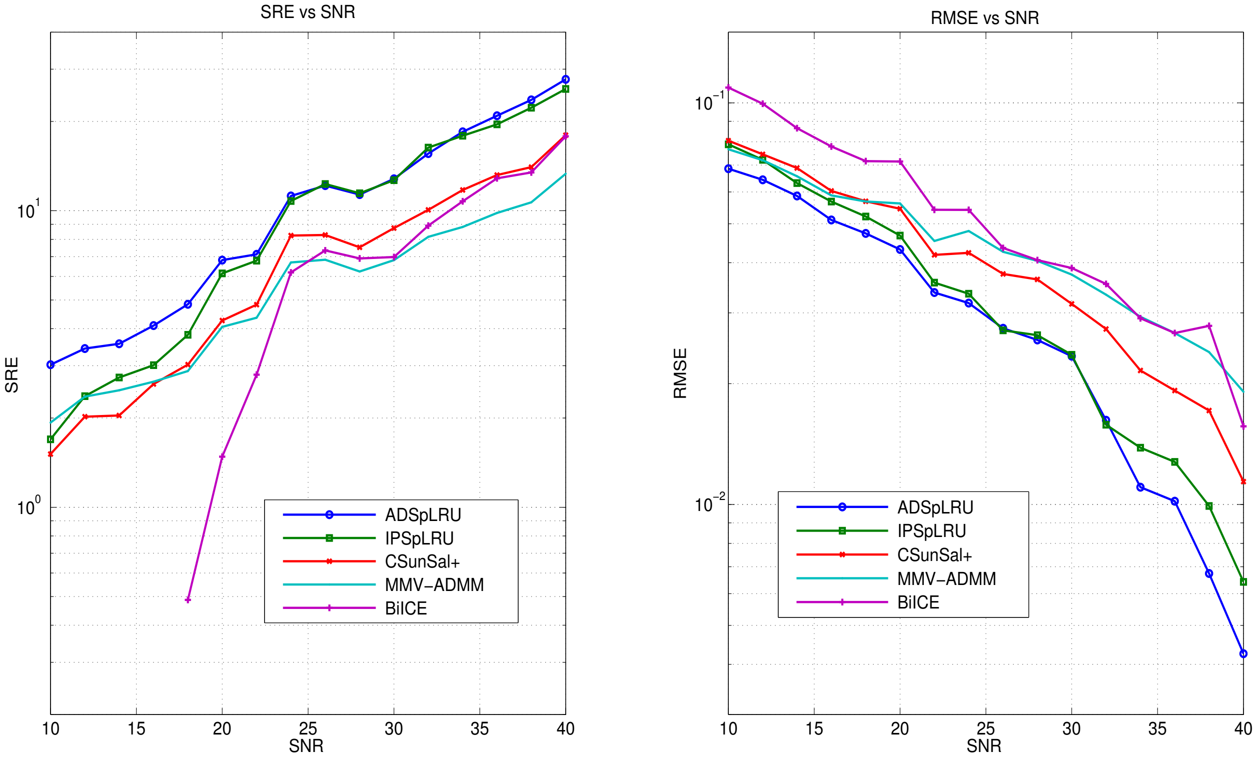


Figure 14: Performance in the presence of white noise (SRE & RMSE).

3.4.2.5 Synthetic Image

This experiment highlights the effectiveness of the proposed IPSpLRU and ADSpLRU methods in estimating sparse, low-rank or both sparse and low-rank abundance matrices. Focused on this purpose we form a simulated hyperspectral image using the LMM Eq. (3.1) and the same above-mentioned endmembers' dictionary **A**. As shown in Fig. 16a, the simulated hyperspectral image consists of 4 rows each consisting of 4 10×10 blocks of pixels. Each of the "block rows" is generated by abundance matrices of a distinct structure. To be more specific, the first row is generated by joint-sparse **X**'s, the second by solely low-rank **X**'s, while rows 3 and 4 are produced by simultaneously sparse and low-rank abundance matrices. The pixels in each block correspond to abundance matrices of a particular combination of sparsity level and rank. The detailed description of these structures is depicted in the table of Fig. 16b. The linearly mixed pixels are corrupted by white Gaussian i.i.d. noise such that SNR = 30dB. The table in Fig. 16c contains the obtained RMSE and SRE for all algorithms tested. It is worth pointing out that our proposed IPSpLRU and ADSpLRU algorithms outperform their rivals, not only in the "both sparse and low-rank" rows 3 and 4, but also in rows 1 and 2 that correspond to either sparse only or low-rank only **X**'s.

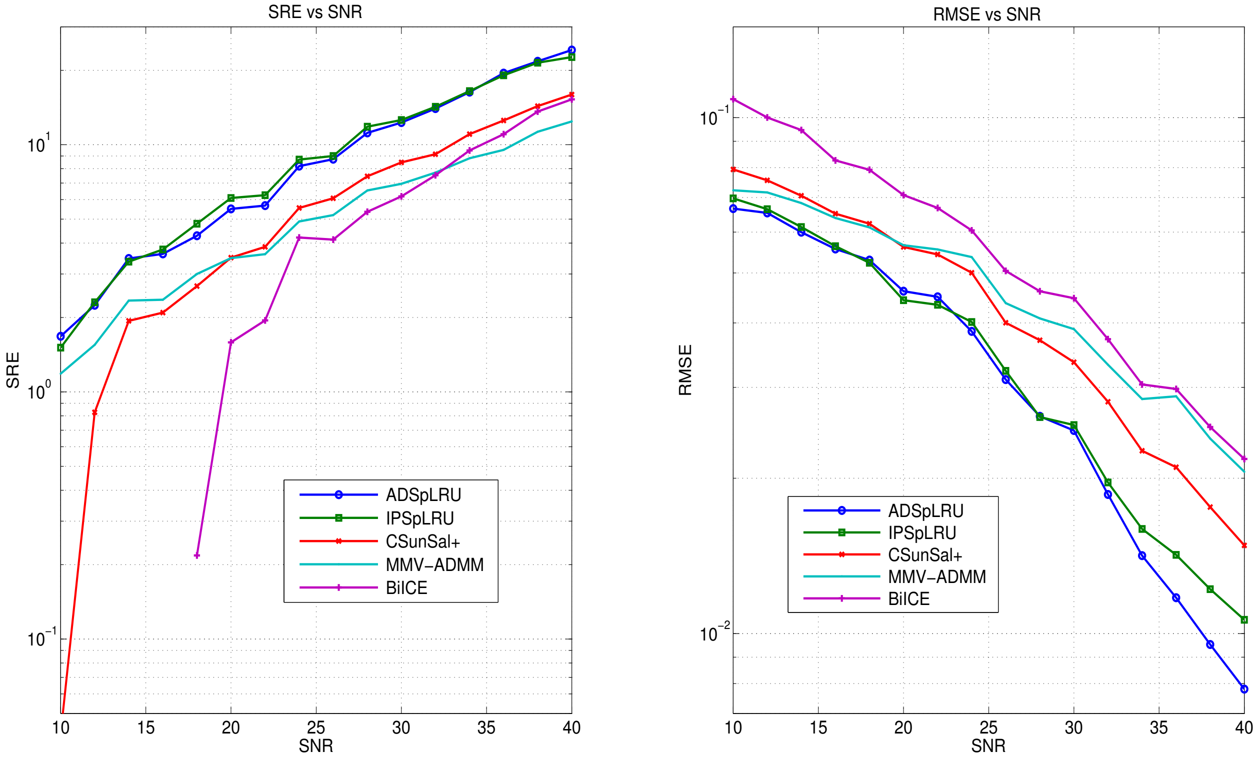
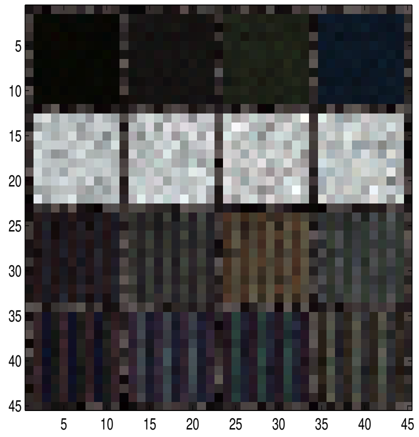


Figure 15: Performance in the presence of colored noise (SRE & RMSE).



(a) Synthetic image consisting of 16 blocks of size 10×10 pixels each.

row	column			
	1^{st}	2^{nd}	3^{rd}	4^{th}
joint sparse - 1^{st}	$(4,1)$	$(8,2)$	$(12,3)$	$(16,4)$
low-rank - 2^{nd}	$(100,1)$	$(100,2)$	$(100,3)$	$(100,4)$
sparse & low-rank - 3^{rd}	$(4,2)$	$(8,2)$	$(12,2)$	$(16,2)$
sparse & low-rank - 4^{th}	$(4,3)$	$(8,3)$	$(12,3)$	$(16,3)$

(b) Explanation of the structure of X in each block of the synthetic image. Each cell contains the pair (sparsity-level%, rank($\mathbf{X}$).)

Algorithm	1^{st} row		2^{nd} row		3^{rd} row		4^{th} row	
	RMSE	SRE	RMSE	SRE	RMSE	SRE	RMSE	SRE
ADSpLRU	0.009	28.96	0.078	16.62	0.032	18.71	0.029	19.62
IPSpLRU	0.008	28.39	0.081	16.41	0.026	21.01	0.030	19.81
CSunSAL	0.026	19.81	0.117	12.39	0.052	13.88	0.047	14.99
MMV-ADMM	0.030	18.00	0.105	12.99	0.061	12.32	0.056	13.16
BiICE	0.028	21.71	0.263	6.72	0.043	17.83	0.060	15.81

(c) RMSE and SRE (dB) results on synthetic image for each block row.

Figure 16: Structure of the synthetic image and results.

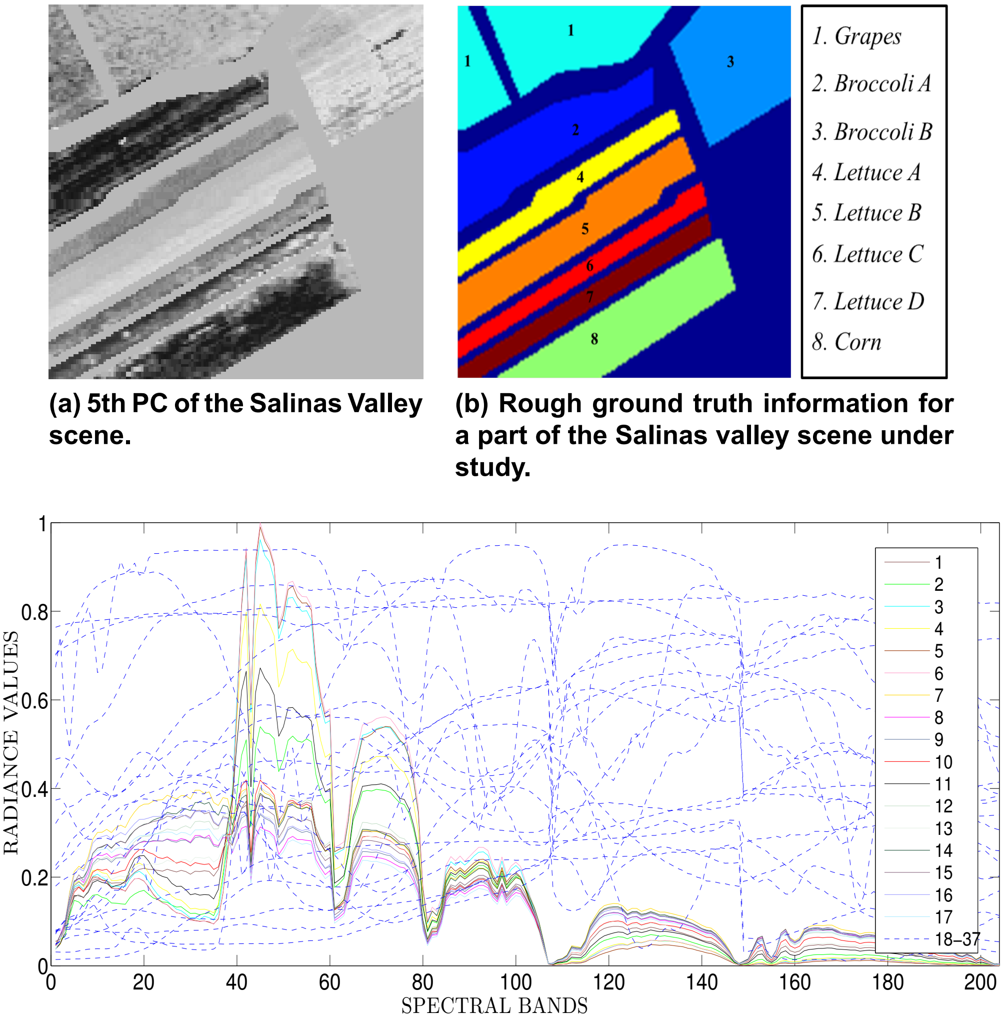
3.4.3 Experiments on real data

This section illustrates the performance of the proposed algorithms when applied on various real hyperspectral images. Specifically, Salinas Valley HSI captured by AVIRIS hyperspectral sensor and three HSIs each depicting a different region of the surface of Mars as obtained by Omega instrument, are studied.

3.4.3.1 Salinas Valley HSI

The hyperspectral scene under study is a portion of the widely used Salinas vegetation scene acquired by AVIRIS sensor over Salinas Valley in California. This scene contains eight different vegetation species, namely grapes, broccoli_A, broccoli_B, lettuce_a, lettuce_b, lettuce_c, lettuce_d, corn, as shown in Fig. 17b. Salinas hyperspectral image consists of $l = 204$ spectral bands (after excluding 20 noisy bands) and its spatial resolution is 3.7 meters. Taking the principal components (PCs) of the image, it can be seen that most of the information of the image is retained in the first six PCs. Focusing on them, we can see that the first PCs give more rough information about the formation of the vegetation, while less significant PCs give more refined information about the vegetation formation, [147]. Fig. 17a shows the 5th PC of the scene under study, where most of the various vegetation species are differentiated from each other. To make things more interesting, the endmembers dictionary **A** is composed of 37 pure spectral signatures, 17 of them manually selected from the image, as in [108], and 20 randomly chosen from the USGS library, [38]. As depicted in Fig. 17c, the 20 USGS's endmembers (blue dashed curves) differ significantly from the other 17 pure pixel spectral signatures selected from the image. This is so, since those signatures correspond to materials (minerals, organic and volatile compounds, etc.) nonexisting in the region under study, while the rest 17 endmembers correspond to the various vegetation types existing in the scene. However, USGS's endmembers were purposely included in the dictionary for investigating the competence of the proposed algorithms in distinguishing the present endmembers over the nonpresent ones, by exploiting sparsity in the abundance matrices.

Fig. 18 shows abundance maps corresponding to the region of interest, as obtained by the proposed IPSpLRU, ADSpLRU, ALMSpLRU and the three state-of-the-art competing algorithms namely CSUnSAL and MMV-ADMM and BiICE for $\gamma = 10^{-3}$, $\tau = 10^{-4}$, $\lambda = 0.5$ and $\mu = 10^{-2}$. Specifically, four different abundance maps are depicted for each algorithm, corresponding to four vegetation species, namely: grapes, broccoli_a, broccoli_b and corn. It is worth pointing out that since detailed ground truth information is not available, the evaluation is carried out in qualitative terms. From a careful visual inspection of the generated maps, we can see that the abundances obtained by IPSpLRU, ADSpLRU and ALMSpLRU present patterns which are closer to those revealed by the first five principal components of the hyperspectral image provided in [108]. This is particularly clear for the maps corresponding to broccoli_a and broccoli_b. More specifically, it is shown that the presence of these two species, which is mainly located in two distinct regions, is better emphasized by the proposed algorithms. Remarkably, the erroneous detection of

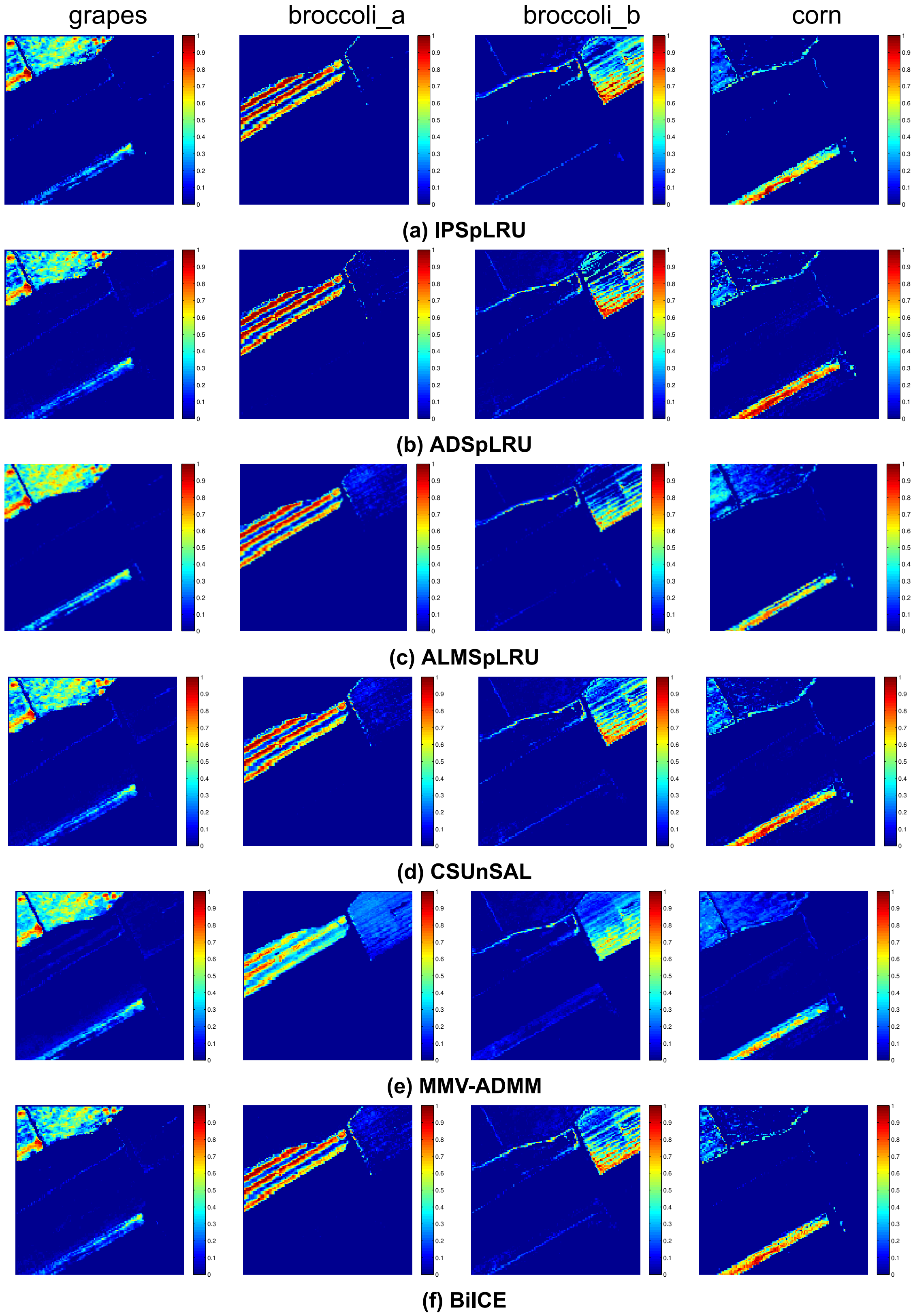


(a) 5th PC of the Salinas Valley scene.

(b) Rough ground truth information for a part of the Salinas valley scene under study.

(c) Spectral signatures of the 37 endmembers, 17 of them manually selected from the scene as pure pixels and 20 (dashed curves) randomly chosen from the USGS library, [38].

Figure 17: Salinas valley image and endmembers' dictionary.



(a) IPSpLRU

(b) ADSpLRU

(c) ALMSpLRU

(d) CSUnSAL

(e) MMV-ADMM

(f) BiICE

Figure 18: Abundance maps of Salinas hyperspectral image.

these vegetation types is eliminated more effectively by IPSpLRU, ADSpLRU and ALMSpLRU, as also verified by comparing Figs 17a and 18. Hence, it is corroborated that the exploitation of both sparsity and the inherent spatial correlation existing in hyperspectral images, can lead us to qualitatively better results, thus verifying the significance of our approach.

3.4.3.2 OMEGA Mars data

In this section we apply HU to three different hyperspectral images corresponding to regions of the Mars' surface, as estimated by IPSpLRU, ADSpLRU and ALMSpLRU algorithms. All the hyperspectral images were captured by the OMEGA instrument, a spectrometer on-board OMEGA's Mars Express satellite. The OMEGA instrument provides hyperspectral images with spatial resolution from 300m to 4km. To this end, it utilizes 96 wavelength channels in the visible band and 256 wavelength channels in the near infrared. Three detectors are used, with spectral resolutions about 7.5nm in the 0.35-1.05 μm and an average of 21 nm from 2.65 to 5.2 μm, respectively.

The first hyperspectral image under study is the observation ORB422_4 of Syrtis Major scene; the second one is a datacube obtained by the South Polar Cap of Mars and, finally, the third one is the observation labeled as ORB898_1.

ORB422_4 (Syrtis Major). ORB422_4 is a single hyperspectral datacube depicting the Syrtis Major region (Fig. 19). Syrtis Major is known to contain well identified areas with a significant presence of mafic minerals (pyroxenes, olivines) and phylosilicates, [128]. The spatial dimensions of the image are 183×63 pixels and the spectral bands are $l = 110$. The data cube has been radiometrically calibrated and the atmospheric gas transmission has been empirically corrected using the volcano scan method, [50, 90]. The endmembers' dictionary used, consists of 32 spectra of minerals that are known to be present in the surface of Mars and the Moon (Fig. 20). Fig. 21 illustrates the results obtained by the ADSpLRU algorithm. It should be noted that the abundance maps of Hypersthene, Diopside, Olivine, Phyll, Oxide, Maghemite and Phyll are in good agreement with those obtained from other state-of-the-art algorithms, [144, 128]. Hence, the particular assumptions upon which ADSpLRU is based, (i.e., sparsity and spatial correlation) are proven to be meaningful in this real case scenario.

South Polar Cap. This hyperspectral image depicts the South Polar Cap of Mars in the local summer (January 2004) (Fig. 22). The spatial size of the data cube is 871×128pixels. The spectral signatures of these pixels are made up of two channels: 128 spectral planes from 0.93 to 2:73 mm with a resolution of 14 nm and 128 spectral planes from 2.55 to 5:11 mm with a resolution of 21 nm. Noisy bands were excluded, and 156 out of the 250 initial bands were finally utilized in the region from 0.93 to 2:98 mm to avoid the thermal emission spectral range. Fig. 23 shows the spectral signatures of the 3 endmembers contained in the used dictionary, namely a) CO_2 ice b) H_2O and c) dust. These endmembers have been detected by the the Wavanglet method presented in [127]. In Fig. 24, the abundance maps obtained by ALMSpLRU are illustrated. It is notes that results are in full

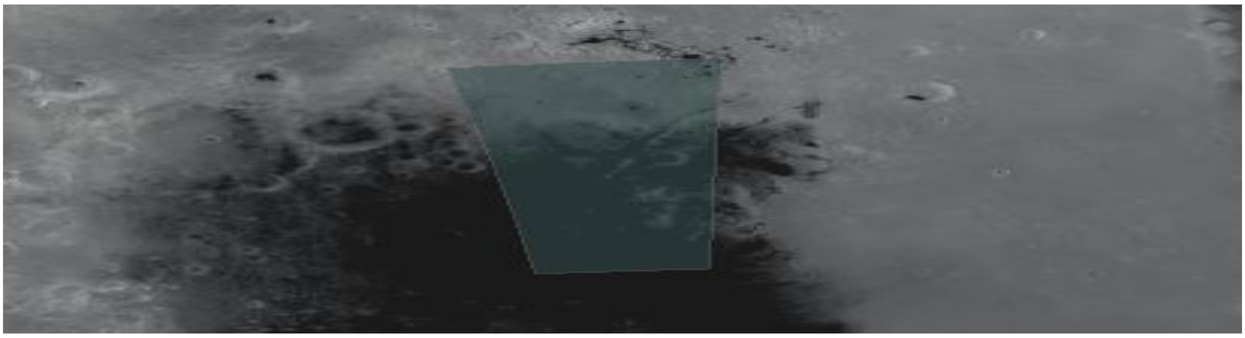


Figure 19: Syrtis Major (shadowed region).

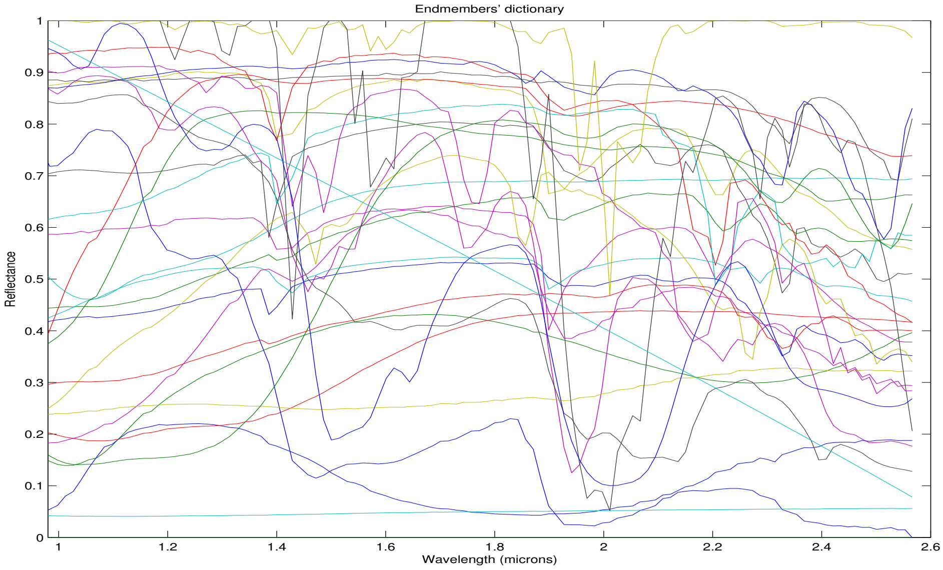
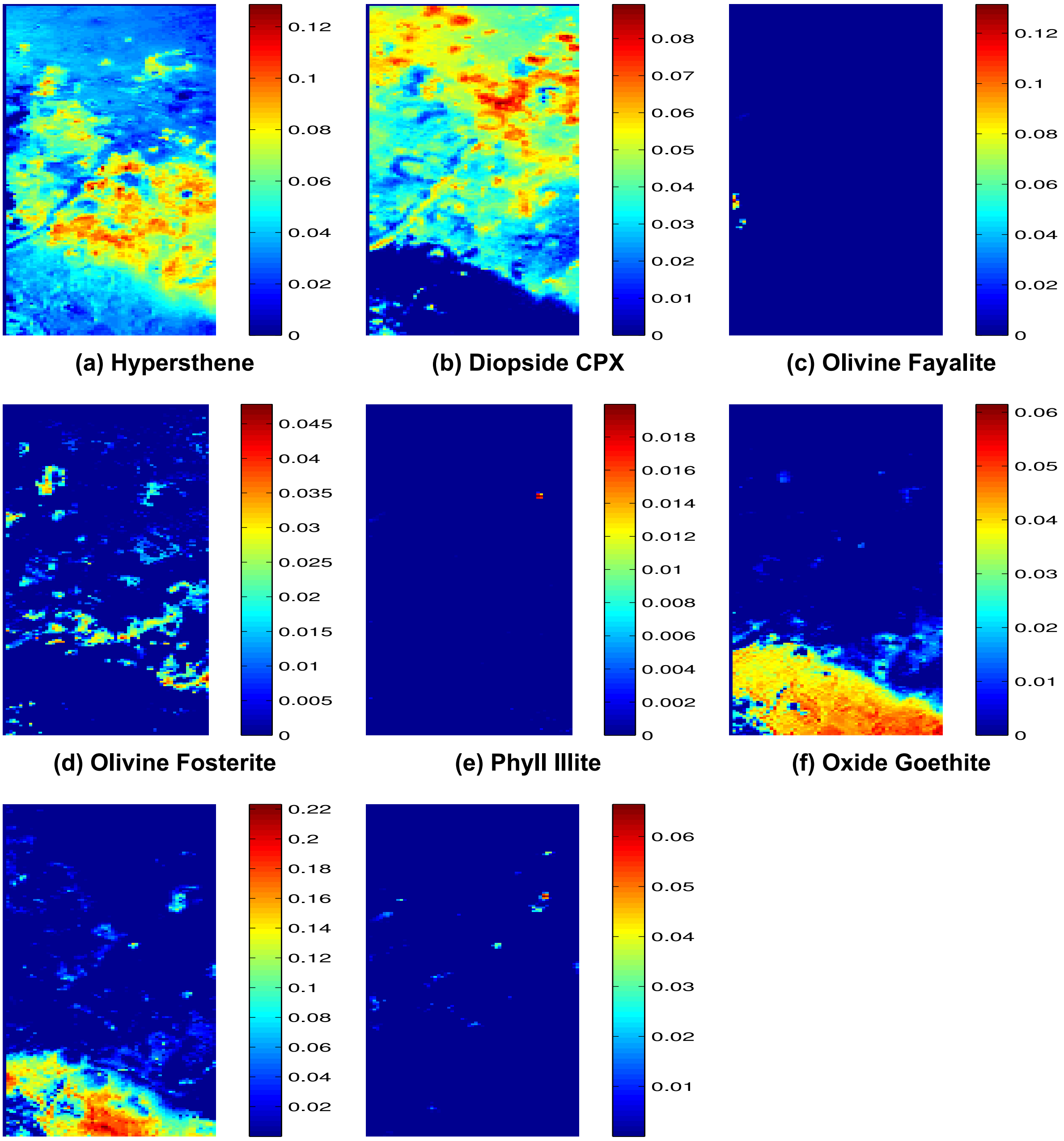


Figure 20: The 32 spectral signatures of minerals (ice and atmospheric gas included in the endmembers' dictionary).



(a) Hypersthene

(b) Diopside CPX

(c) Olivine Fayalite

(d) Olivine Fosterite

(e) Phyll Illite

(f) Oxide Goethite

(g) Maghemite

(h) Phyll Chlorite

Figure 21: Abundance maps of ADSpLRU for Syrtis Major region of Mars.

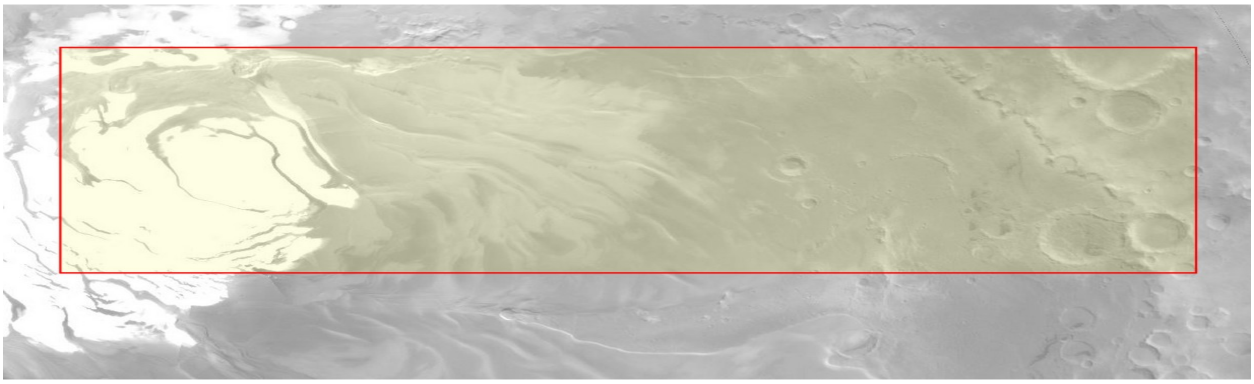


Figure 22: The South Polar cap scene of Mars.

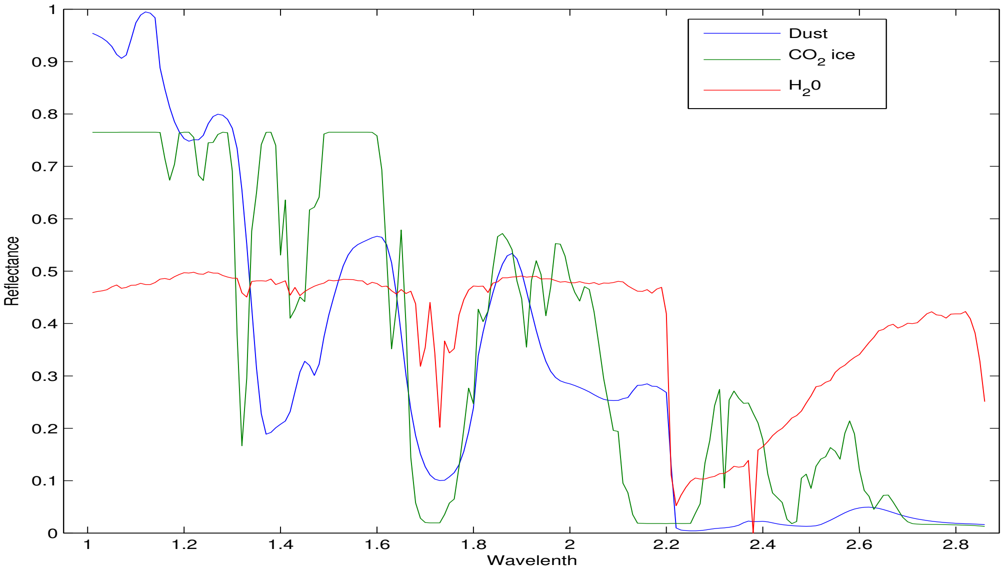


Figure 23: Reference spectra of the South Polar Cap hyperspectral datacube. The available endmembers are: (a) OMEGA typical dust materials with atmosphere absorption, (b) synthetic CO2 ice with grain size of 100 mm, (c) synthetic H2O ice with grain size of 100 mm.

agreement for both the tested algorithms. In addition, there is a differentiation concerning the H_2O's abundance estimates of ALMSpLRU, comparing to the maps presented in [144]. In particular, it seems that ALMSpLRU detects H_2O mostly in a specific region of the image, in sharp contrast to the maps of [144], where H_2O is diffused in a wide part of the datacube.

ORB898_1. ORB898_1 points to a relatively unexplored region of the surface of MARS (Fig. 25). The hyperspectral image captured by OMEGA instrument, consists of 257×112 pixels and $L = 110$ bands. In this case, we employed a dictionary composed of the 32 endmembers' spectral signatures that were used for the case of Syrtis Major (Fig. 20) and 12 additional spectra proposed in [128], in an attempt to enhance estimation accuracy. Figs 26 and 27 exhibit the abundance maps as retrieved by ADSpLRU and ALMSpLRU for 9 different endmembers, namely H_2O grain 1μm, H_2O grain 100μm,H_2O grain 1000μm, Epsomite, Sulfate Gypsum, CO_2 grain 100μm, Ferrihydrite and CO_2 grain

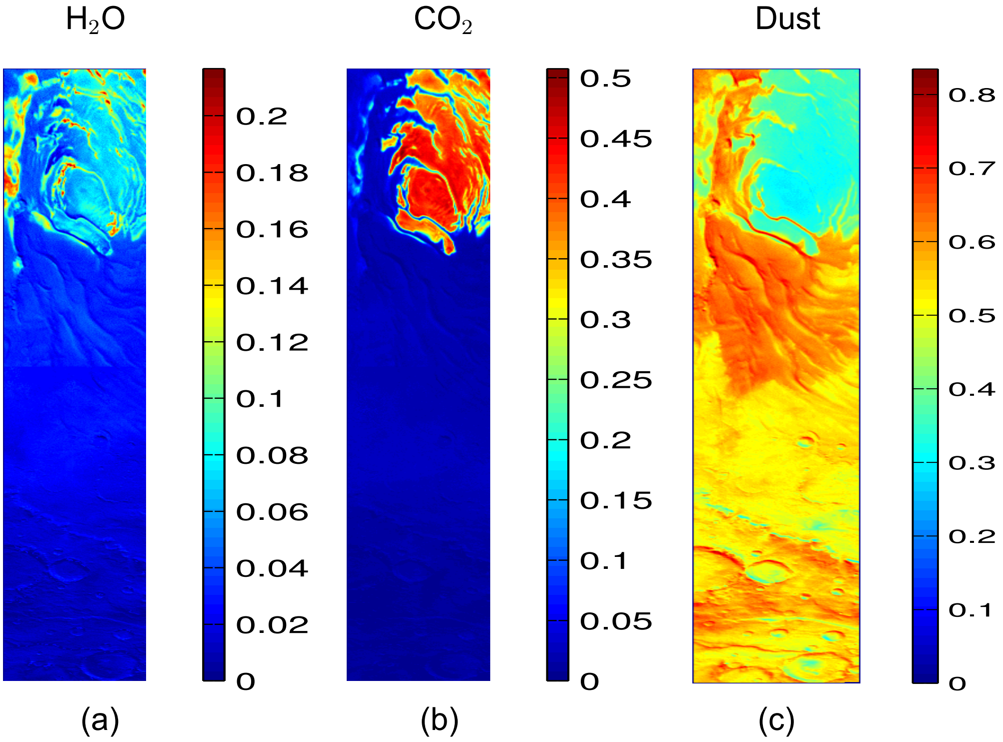


Figure 24: Abundance maps of South Polar Cap as estimated by ALMSpLRU.

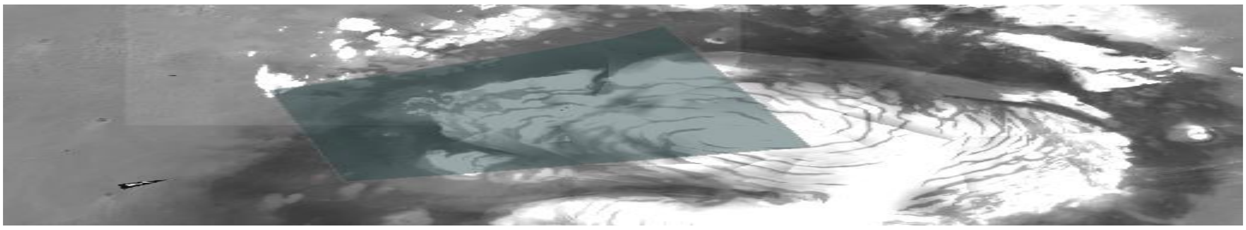
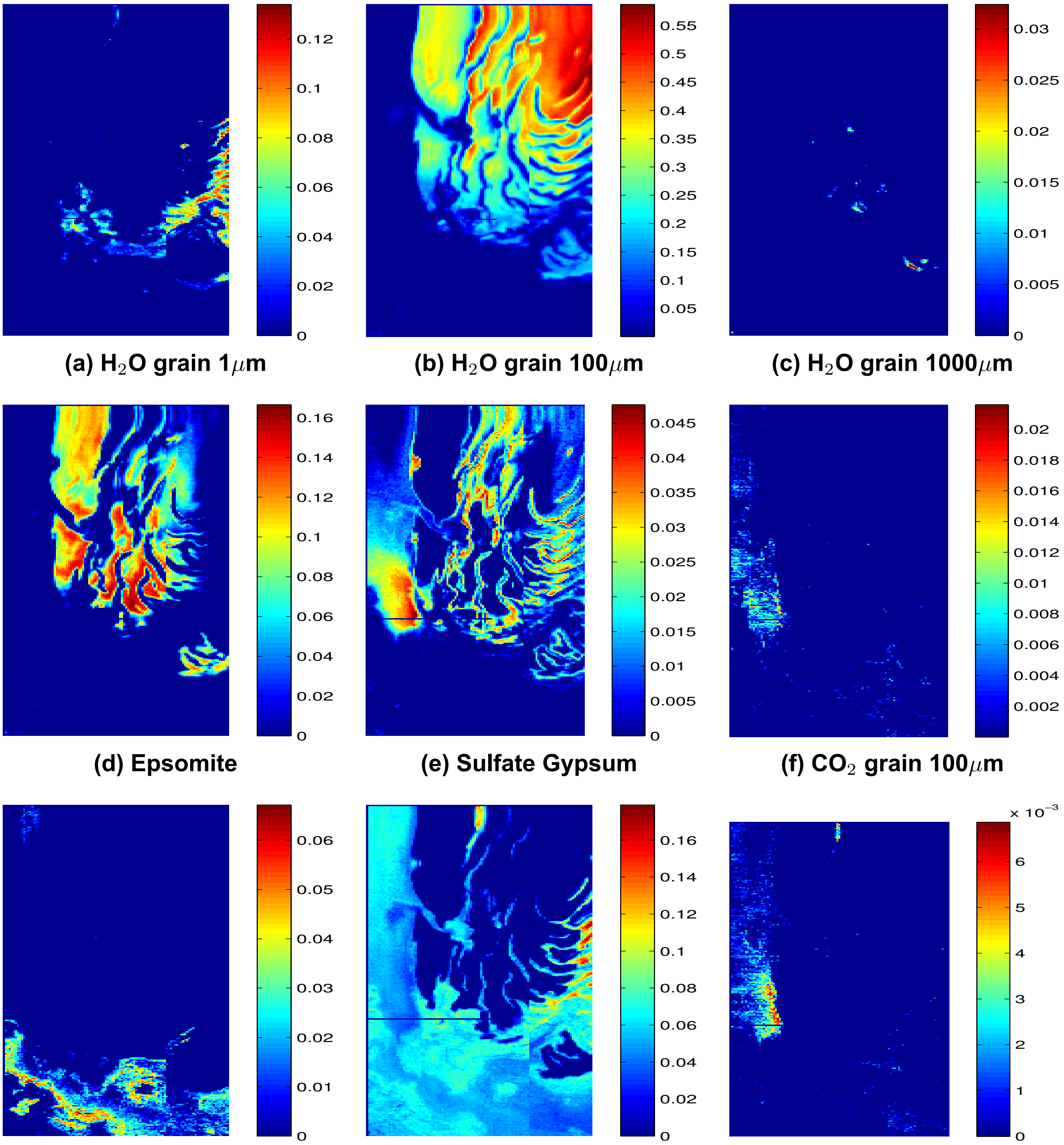


Figure 25: ORB898_1 (shadowed region).

10000μm. Abundance maps reveal a high degree of accordance in the estimates of the two tested algorithms for the majority of the endmembers. Nevertheless, it should be noted that there is remarkable discrepancy between the maps of ADSpLRU and ALMSpLRU corresponding to H_2O grain 1μm, CO_2 grain 100μm and CO_2 grain 1000μm. Additionally, it is derived, that there exists a significant presence of specific endmembers, e.g., H_2O grain 100μm in the examined scene, that form patterns of characteristic shapes.



(a) H_2O grain 1μm

(b) H_2O grain 100μm

(c) H_2O grain 1000μm

(d) Epsomite

(e) Sulfate Gypsum

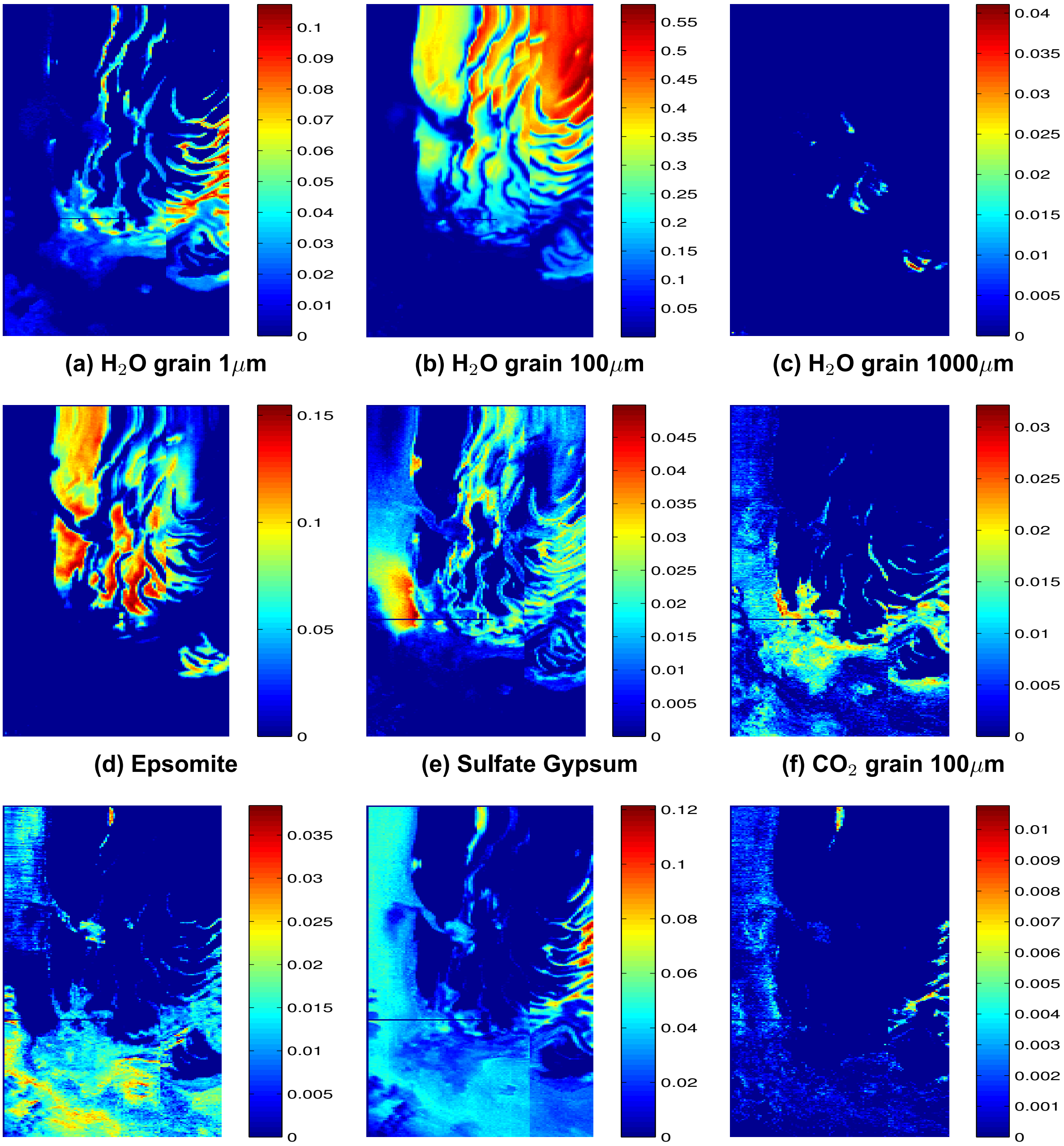
(f) CO_2 grain 100μm

(g) Olivine Forsterite

(h) Ferrihydrite

(i) CO_2 grain 10000μm

Figure 26: Abundance maps of ADSpLRU for ORB898_1.



(a) H_2O grain 1μm

(b) H_2O grain 100μm

(c) H_2O grain 1000μm

(d) Epsomite

(e) Sulfate Gypsum

(f) CO_2 grain 100μm

(g) Olivine Forsterite

(h) Ferrihydrite

(i) CO_2 grain 10000μm

Figure 27: Abundance maps of ALMSpLRU for ORB898_1.

4. ALTERNATING ITERATIVELY REWEIGHTED LEAST SQUARES MINIMIZATION FOR LOW-RANK MATRIX FACTORIZATION

In this chapter, a novel formulation for low-rank matrix factorization (LRMF) is proposed, that is suitable for denoising, matrix completion and nonnegative matrix factorization (NMF). Inspired by the merits of iterative reweighted schemes for sparse recovery and rank minimization, we come up with a generic low-rank promoting regularization scheme. Then, focusing on a specific instance of it, we propose a regularizer that imposes column-sparsity jointly on the two matrix factors that result from MF, thus promoting low-rankness on the optimization problem. The problems of denoising and matrix completion are redefined according to the new LRMF formulation and solved via efficient alternating iteratively reweighted least squares type algorithms. Theoretical analysis of these algorithms regarding the convergence and the rates of convergence to stationary points is provided. The proposed LRMF formulation is further extended by incorporating nonnegativity and sparsity constraints giving thus rise to a low-rank NMF scheme and to a low-rank and sparse NMF algorithm. The effectiveness of the proposed algorithms is verified on diverse simulated and real data experiments. More specifically, the derived algorithms are applied to the problems of hyperspectral image denoising and unsupervised unmixing, collaborative filtering for recommender systems and music signal decomposition showing their favorable properties over other relevant state-of-the-art algorithms.

4.1 MF based low-rank matrix estimation

Recently, low-rank matrix estimation has been effectively tackled using a *matrix factorization* approach. As also stated in Chapter 1, the crux of the MF based methods is that a low-rank matrix can be well represented by a product of two matrices $\mathbf{U} \in \mathcal{R}^{m \times r}$ and $\mathbf{V} \in \mathcal{R}^{n \times r}$, i.e., $\mathbf{X} = \mathbf{U}\mathbf{V}^T$, with the inner dimension r of the involved matrices being quite smaller than the outer dimensions, i.e., $r \ll \min(m, n)$. Recall that those ideas offer significant advantages when it comes to the processing of large-scale and high-dimensional datasets (where both m and n are huge) by reducing the size of the involved variables, thus decreasing both the storage space required from $\mathcal{O}(mn)$ to $\mathcal{O}\left((m+n)r\right)$ as well as the computational complexity of the algorithms used to solve the problem. However, a downside of this approach is that an additional variable is brought up, i.e., the inner dimension r of the factorization. The task of finding the actual r (which coincides with the rank of matrix $\mathbf{X}$) is relevant to the rank minimization problem and is referred in the literature also as dimensionality reduction, model order selection, etc.

The latter has given rise to methods that select r based on the minimization of various criteria, such as the Akaike information criterion (AIC), the Bayesian information criterion (BIC), the minimum distance length (MDL), [133], etc. However, these methods can be computationally expensive especially in large-scale datasets, since they require multiple runs using different values for r. Modern approaches that tackle this problem, termed low-rank matrix factorization (LRMF) techniques, [76], hinge on the following philosophy: a) overstate the rank r of the product with $d \geq r$ and then b) impose low-rankness thereof by utilizing appropriate norms. This rationale has given rise to LRMF techniques that solve the following,

$$\min \operatorname{rank}(\mathbf{UV}^T) \quad \text{subject to} \quad \mathcal{A}(\mathbf{UV}^T) = \mathbf{b}. \tag{4.1}$$

Problem (4.1) is NP-hard in general and thus different relaxation schemes have been put in the literature for addressing it. Among other approaches, the variational forms of the nuclear norm (Section 1.2.3.2), which induce tight upper bounds of the nuclear norm, i.e.,

$$\begin{aligned} \|\mathbf{X}\|_* &= \min_{\mathbf{U}\in\mathcal{R}^{m\times d}, \mathbf{V}\in\mathcal{R}^{n\times d}, \mathbf{X}=\mathbf{UV}^T} \|\mathbf{U}\|_F \|\mathbf{V}\|_F \\ &= \min_{\mathbf{U}\in\mathcal{R}^{m\times d}, \mathbf{V}\in\mathcal{R}^{n\times d}, \mathbf{X}=\mathbf{UV}^T} \frac{1}{2}\left(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2\right) \end{aligned} \tag{4.2}$$

are the most popular, [135]. In fact, minimization of either of the upper bounds defined in (4.2) favors low-rankness on **U** and **V** by inducing “smoothness” on these matrices. Moreover, in [131, 132], the authors generalize the above result by deriving tight upper bounds for all Schatten-p quasinorms[1] with $0 < p \leq 1$, (Theorem 1, [132]), i.e.,

$$\begin{aligned} \|\mathbf{X}\|_{\mathcal{S}_p}^p &= \min_{\mathbf{U}\in\mathcal{R}^{m\times d}, \mathbf{V}\in\mathcal{R}^{n\times d}, \mathbf{X}=\mathbf{UV}^T} \|\mathbf{U}\|_{\mathcal{S}_{2p}}^p \|\mathbf{V}\|_{\mathcal{S}_{2p}}^p \\ &= \min_{\mathbf{U}\in\mathcal{R}^{m\times d}, \mathbf{V}\in\mathcal{R}^{n\times d}, \mathbf{X}=\mathbf{UV}^T} \frac{1}{2}\left(\|\mathbf{U}\|_{\mathcal{S}_{2p}}^{2p} + \|\mathbf{V}\|_{\mathcal{S}_{2p}}^{2p}\right). \end{aligned} \tag{4.3}$$

4.2 The proposed LRMF formulation

In this thesis, we aspire to apply ideas stemming from *iterative reweighting methods for low-rank matrix recovery*, to this challenging low-rank matrix factorization scenario. Therefore, generalizing the above-described low-rank promoting norm upper bounds, we propose to minimize the sum of reweighted (as in (1.20)) Frobenius norms of the individual factors **U** and **V**. Hence, the newly introduced low-rank inducing function is defined as follows,

$$h(\mathbf{U}, \mathbf{V}) = \frac{1}{2}\left(\|\mathbf{U}\mathbf{W}_{\mathbf{U}}^{\frac{1}{2}}\|_F^2 + \|\mathbf{V}\mathbf{W}_{\mathbf{V}}^{\frac{1}{2}}\|_F^2\right), \tag{4.4}$$

where the weight matrices $\mathbf{W}_{\mathbf{U}}$ and $\mathbf{W}_{\mathbf{V}}$ are appropriately selected. The proposed low-rank promoting function defined in (4.4) is generic as it includes the previously mentioned MF based low-rank promoting terms as special cases. Indeed, according to (1.20), (1.21) and by setting $\mathbf{W}_{\mathbf{U}} = (\mathbf{U}^T\mathbf{U})^{p-1}$ and $\mathbf{W}_{\mathbf{V}} = (\mathbf{V}^T\mathbf{V})^{p-1}$ in (4.4), we get the upper bound of the Schatten-p quasinorm given in (4.3)[2], while for $p = 1$, i.e., $\mathbf{W}_{\mathbf{U}} = \mathbf{W}_{\mathbf{V}} = \mathbf{I}_d$, we get the variational form of the nuclear norm defined in (4.2).

Clearly, various choices of $\mathbf{W}_{\mathbf{U}}$ and $\mathbf{W}_{\mathbf{V}}$ give rise to different upper bounds for the Schatten-

[1] Recall Eq. 1.19.

[2] Recall here Eqs. (1.20) and (1.21).

p norms. In the rest of this chapter, we adhere to a specific instance of (4.4) which arises by setting $\mathbf{W_U} = \mathbf{W_V} = \mathbf{W}$ with

$$\mathbf{W} = \mathrm{diag}\Big(\left(\|\boldsymbol{u}_1\|_2^2 + \|\boldsymbol{v}_1\|_2^2\right)^{p/2-1}, \left(\|\boldsymbol{u}_2\|_2^2 + \|\boldsymbol{v}_2\|_2^2\right)^{p/2-1}, \dots, \left(\|\boldsymbol{u}_d\|_2^2 + \|\boldsymbol{v}_d\|_2^2\right)^{p/2-1} \Big), \quad (4.5)$$

where $0 < p \leq 1$ and $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ are the ith columns of **U** and **V**, respectively[3]. The selection of the common weight matrix of the factors as in (4.5) is not arbitrary. As we will see in Sections 4.3 and 4.4, this matrix leads to *iteratively reweighted least squares* (IRLS) schemes (see Section 1.2.1) for low-rank matrix factorization, generalizing the IRLS-p family of algorithms developed in [41] for sparse vector recovery. In addition by selecting a common **W** for **U** and **V**, matrices **U** and **V** are implicitly coupled w.r.t. their columns. If we now substitute (4.5) in (4.4) yields

$$h(\mathbf{U},\mathbf{V}) = \frac{1}{2}\sum_{i=1}^{d}(\|\boldsymbol{u}_i\|_2^2 + \|\boldsymbol{v}_i\|_2^2)^{p/2}. \quad (4.6)$$

Surprisingly, the resulting expression coincides with the (scaled by 1/2) group sparsity inducing $\ell_{p,2}^p$ norm ($0 < p \leq 1$) of the concatenated matrix $[\begin{smallmatrix}\mathbf{U}\\ \mathbf{V}\end{smallmatrix}]$, which for $p = 1$ reduces to the commonly used $\ell_{1,2}$ matrix norm. Intuitively, the low-rank inducing properties of the proposed in (4.6) joint column sparsity promoting term can be easily explained as follows. Let us consider the rank one decomposition of the matrix product $\mathbf{UV}^T$,

$$\mathbf{UV}^T = \sum_{i=1}^{d}\boldsymbol{u}_i\boldsymbol{v}_i^T. \quad (4.7)$$

Clearly, due to the subadditivity property of the rank, eliminating rank one terms of the summation on the right side of (4.7) results to a relevant decrease of the rank of the product $\mathbf{UV}^T$. Hence capitalizing on (4.6), we are led to LRMF optimization problems having the form,

$$\min_{\mathbf{U}\in\mathcal{R}^{m\times d},\mathbf{V}\in\mathcal{R}^{n\times d}} \sum_{i=1}^{d}(\|\boldsymbol{u}_i\|_2^2 + \|\boldsymbol{v}_i\|_2^2)^{p/2} \text{ subject to } \mathcal{A}(\mathbf{UV}^T) = \mathbf{b}. \quad (4.8)$$

It should be noted that the idea of imposing jointly column sparsity first appeared in [138], albeit in a Bayesian framework tailored to the NMF problem. In [139], the emerging via the maximum a posteriori probability (MAP) approach optimization problem boils down to the minimization of the column sparsity promoting concave logarithm function.

Next, the generic problem given in (4.8) is reformulated and solved for four important learning tasks namely a) denoising, b) matrix completion, c) low-rank NMF and d) low-rank and sparse NMF.

[3] If $\mathbf{U},\mathbf{V}$ had orthogonal columns, **W** in (4.5) would be equal to $(\mathbf{U}^T\mathbf{U} + \mathbf{V}^T\mathbf{V})^{p/2-1}$, whose resemblance to (1.21) is evident.

4.3 Denoising and matrix completion via the proposed LRMF approach

Next the problems of denoising and matrix completion are viewed through the lens of the proposed LRMF approach. The resulting problems are then addressed by novel alternating IRLS-type optimization algorithms which spring from the BSUM framework (see Section 2.1.4).

4.3.1 Denoising

By assuming that a) the linear operator $\mathcal{A}$ reduces to a diagonal matrix and b) our measurements $\mathbf{Y} \in \mathcal{R}^{m\times n}$ are corrupted by i.i.d. Gaussian noise, we come up with the following optimization problem,

$$\min_{\mathbf{U},\mathbf{V}} \sum_{i=1}^{d} (\|\boldsymbol{u}_i\|_2^2 + \|\boldsymbol{v}_i\|_2^2)^{p/2} \text{ subject to } \|\mathbf{Y} - \mathbf{U}\mathbf{V}^T\|_F^2 \leq \epsilon. \tag{4.9}$$

where ϵ is a small positive constant. By the Lagrange theorem we know that (4.9) can be equivalently written in the following form,

$$\{\hat{\mathbf{U}}, \hat{\mathbf{V}}\} = \underset{\mathbf{U},\mathbf{V}}{\text{argmin}} \frac{1}{2}\|\mathbf{Y} - \mathbf{U}\mathbf{V}^T\|_F^2 + \lambda \sum_{i=1}^{d} (\|\boldsymbol{u}_i\|_2^2 + \|\boldsymbol{v}_i\|_2^2)^{p/2} \tag{4.10}$$

where λ denotes the Lagrange multiplier.

4.3.2 Matrix completion

Another popular problem that follows the general model described by (4.8) is matrix completion, as it is widely addressed via low-rank minimization. As mentioned in Chapter 1, the main premise here lies in recovering missing entries of a matrix $\mathbf{Y}$ assuming high degree of correlation among its rows/columns, which gives rise to a low-rank structured matrix $\mathbf{X}$. Utilizing the proposed framework, the problem can be stated as,

$$\min_{\mathbf{U},\mathbf{V}} \sum_{i=1}^{d} (\|\boldsymbol{u}_i\|_2^2 + \|\boldsymbol{v}_i\|_2^2)^{p/2} \quad \text{subject to} \quad \mathcal{P}_\Omega(\mathbf{Y}) = \mathcal{P}_\Omega(\mathbf{U}\mathbf{V}^T). \tag{4.11}$$

where $\mathcal{P}_\Omega$ denotes the sampling operator on the set Ω of indexes of matrix $\mathbf{Y}$ where information is present. In the matrix factorization setting, the incomplete matrix $\mathbf{Y}$ is approximated by a matrix $\mathbf{X}$ expressed as $\mathbf{X} = \mathbf{U}\mathbf{V}^T$. As mentioned above, the rank r of the reconstructed matrix $\mathbf{X}$ is generally unknown and hence it is overstated with $d \geq r$.

Considering further the existence of additive i.i.d. Gaussian noise in $\mathbf{Y}$ we get,

$$\min_{\mathbf{U},\mathbf{V}} \sum_{i=1}^{d} (\|\boldsymbol{u}_i\|_2^2 + \|\boldsymbol{v}_i\|_2^2)^{p/2} \quad \text{subject to} \quad \|\mathcal{P}_\Omega(\mathbf{Y}) - \mathcal{P}_\Omega(\mathbf{U}\mathbf{V}^T)\|_F^2. \tag{4.12}$$

Utilizing the Lagrange theorem we end up with the following optimization problem

$$\{\hat{\mathbf{U}}, \hat{\mathbf{V}}\} = \underset{\mathbf{U},\mathbf{V}}{\text{argmin}} \frac{1}{2} \|\mathcal{P}_\Omega(\mathbf{Y}) - \mathcal{P}_\Omega(\mathbf{U}\mathbf{V}^T)\|_F^2 + \lambda \sum_{i=1}^{d} (\|\boldsymbol{u}_i\|_2^2 + \|\boldsymbol{v}_i\|_2^2)^{p/2}. \tag{4.13}$$

As it is shown later, the simplicity and tractability of the proposed regularizer facilitates the derivation of new and efficient in terms of computational complexity optimization algorithms, while the adoption of the minimization framework presented in the next section paves the way for the theoretical analysis of their convergence behavior.

4.3.3 Denoising and matrix completion algorithms

Having expressed the denoising and matrix completion problems utilizing the proposed framework, we present now two new efficient block coordinate minimization (BCM) algorithms for solving them. The alternating minimization, w.r.t. the 'blocks' **U** and **V**, of the proposed low-rank promoting function defined in (4.6) lies at the heart of those algorithms.

Remark 4.1. *The proposed low-rank promoting regularizer is a) nonsmooth and b) nonseparable w.r.t.* **U** *and* **V***.*

Both the above-mentioned properties, i.e., nonsmoothness and nonseparability induce severe difficulties in the optimization task that call for appropriate handling. More specifically, as it has been shown in [151], in BCM schemes the respective algorithms may lead to irregular points, i.e., coordinate-wise minima that are not necessarily stationary points of the minimized objective function (see Definition 2.2, Section 2.1.4.1). In light of this we follow a simple smoothing approach by including a small positive constant η^2 in the proposed regularizer, which now becomes,

$$\hat{h}(\mathbf{U}, \mathbf{V}) = \sum_{i=1}^{d} (\|\boldsymbol{u}_i\|_2^2 + \|\boldsymbol{v}_i\|_2^2 + \eta^2)^{p/2}. \tag{4.14}$$

This way we alleviate singular points, i.e., points where the gradient is not continuous, and the resulting optimization problems become smooth. On the other hand, nonseparability poses obstacles in getting closed-form expressions for the optimization variables **U** and **V**. For this reason, each of the associative optimization problems is reformulated using appropriate relaxation schemes. By working in an alternating fashion, each of these schemes results in closed form expressions. Next, the proposed algorithms that solve the denoising and matrix completion problems are analytically described.

4.3.3.1 Alternating IRLS denoising algorithm

In this section, we present a new algorithm designed for solving the denoising problem given in (4.10). To this end, let us first rewrite the respective objective function , including the term η^2 as,

$$f(\mathbf{U},\mathbf{V}) = \frac{1}{2}\|\mathbf{Y} - \mathbf{U}\mathbf{V}^T\|_F^2 + \lambda \sum_{i=1}^{d}(\|\boldsymbol{u}_i\|_2^2 + \|\boldsymbol{v}_i\|_2^2 + \eta^2)^{p/2}. \tag{4.15}$$

It is obvious that minimizing (4.15) alternatingly w.r.t. **U** and **V** is infeasible, since exact analytical expressions can not be obtained as a result of the nonseparable nature of the regularizing term. To this end, we tackle the problem in an iterative fashion. Specifically, at the $k+1$ iteration we solve two distinct subproblems, i.e. a) given the latest available update $\mathbf{V}_k$ of **V**, we minimize an approximate cost function w.r.t. **U** to get $\mathbf{U}_{k+1}$ and b) we use $\mathbf{U}_{k+1}$ in order to minimize another approximate cost function w.r.t. **V**. Following the block successive upper bound minimization (BSUM) approach that was briefly described in Section 2.1.4, we minimize at each iteration local tight upper bounds of the respective objective functions. That said, **U** is updated by minimizing an approximate second-order Taylor expansion of $f(\mathbf{U},\mathbf{V}_k)$ around the point $(\mathbf{U}_k,\mathbf{V}_k)$. Likewise, an approximate second-order Taylor expansion of $f(\mathbf{U}_{k+1},\mathbf{V})$ around $(\mathbf{U}_{k+1},\mathbf{V}_k)$ is utilized for obtaining $\mathbf{V}_{k+1}$. To be more specific $\mathbf{U}_{k+1}$ is computed by

$$\mathbf{U}_{k+1} = \underset{\mathbf{U}}{\operatorname{argmin}}\ \ l(\mathbf{U}|\mathbf{U}_k,\mathbf{V}_k), \tag{4.16}$$

where,

$$\begin{aligned} l(\mathbf{U}|\mathbf{U}_k,\mathbf{V}_k) = {} & f(\mathbf{U}_k,\mathbf{V}_k) + \operatorname{tr}\{(\mathbf{U}-\mathbf{U}_k)^T\nabla_{\mathbf{U}}f(\mathbf{U}_k,\mathbf{V}_k)\} \\ & + \frac{1}{2}\operatorname{vec}(\mathbf{U}-\mathbf{U}_k)^T\bar{\mathbf{H}}_{\mathbf{U}_k}\operatorname{vec}(\mathbf{U}-\mathbf{U}_k) \end{aligned} \tag{4.17}$$

and vec$(\cdot)$ denotes the row vectorization operator[4]. It is highlighted that $\bar{\mathbf{H}}_{\mathbf{U}_k}$ is not the true Hessian of $f(\mathbf{U},\mathbf{V}_k)$ at $\mathbf{U}_k$, denoted as $\mathbf{H}_{\mathbf{U}_k}$, but rather an approximation of it. Specifically, $\bar{\mathbf{H}}_{\mathbf{U}_k}$ is defined as an $md \times md$ positive-definite block diagonal matrix, expressed as

$$\bar{\mathbf{H}}_{\mathbf{U}_k} = \mathbf{I}_m \otimes \tilde{\mathbf{H}}_{\mathbf{U}_k}, \tag{4.18}$$

where $\otimes$ denotes the Kronecker product operation. For reasons that will be explained later, the $d \times d$ diagonal block $\tilde{\mathbf{H}}_{\mathbf{U}_k}$ is defined as

$$\tilde{\mathbf{H}}_{\mathbf{U}_k} = \mathbf{V}_k^T\mathbf{V}_k + \lambda\mathbf{D}_{(\mathbf{U}_k,\mathbf{V}_k)} \tag{4.19}$$

with

$$\begin{aligned} \mathbf{D}_{(\mathbf{U},\mathbf{V})} = p\operatorname{diag}\Big(& (\|\boldsymbol{u}_1\|_2^2 + \|\boldsymbol{v}_1\|_2^2 + \eta^2)^{p/2-1}, (\|\boldsymbol{u}_2\|_2^2 + \|\boldsymbol{v}_2\|_2^2 + \eta^2)^{p/2-1}, \\ & \ldots, (\|\boldsymbol{u}_d\|_2^2 + \|\boldsymbol{v}_d\|_2^2 + \eta^2)^{p/2-1}\Big). \end{aligned} \tag{4.20}$$

[4] Vectorization operation transforms an $m \times n$ matrix to a $mn \times 1$ vector.

As it is shown next (Lemma 4.1, Section 4.3.5), due to the form of $\bar{\mathbf{H}}_{\mathbf{U}_k}$ from (4.18) and (4.19) and its relation to the exact Hessian $\mathbf{H}_{\mathbf{U}_k}$ of $f(\mathbf{U}, \mathbf{V}_k)$ at $\mathbf{U}_k$, it turns out that $l(\mathbf{U}|\mathbf{U}_k, \mathbf{V}_k)$ bounds $f(\mathbf{U}, \mathbf{V}_k)$ from above and the conditions set by the BSUM framework are satisfied. Actually, the approximation of the exact Hessian by using (4.18) leads to a closed-from expression for updating **U** and a dramatic decrease of the required computational complexity, as it will be further explained below.

Following a similar path as above we come up with appropriate upper bound functions for updating **V** i.e,

$$\mathbf{V}_{k+1} = \underset{\mathbf{V}}{\text{argmin}} \ \ g(\mathbf{V}|\mathbf{U}_{k+1}, \mathbf{V}_k) \tag{4.21}$$

with

$$\begin{aligned} g(\mathbf{V}|\mathbf{U}_{k+1}, \mathbf{V}_k) = {} & f(\mathbf{U}_{k+1}, \mathbf{V}_k) + \text{tr}\{(\mathbf{V} - \mathbf{V}_k)^T \nabla_{\mathbf{V}} f(\mathbf{U}_{k+1}, \mathbf{V}_k)\} \\ & + \frac{1}{2}\text{vec}(\mathbf{V} - \mathbf{V}_k)^T \bar{\mathbf{H}}_{\mathbf{V}_k} \text{vec}(\mathbf{V} - \mathbf{V}_k) \end{aligned} \tag{4.22}$$

and $\bar{\mathbf{H}}_{\mathbf{V}_k}$ being a block diagonal $md \times md$ matrix[5] (similar to $\bar{\mathbf{H}}_{\mathbf{U}_k}$) whose $d \times d$ diagonal blocks $\tilde{\mathbf{H}}_{\mathbf{V}_k}$ are defined as

$$\tilde{\mathbf{H}}_{\mathbf{V}_k} = \mathbf{U}_{k+1}^T \mathbf{U}_{k+1} + \lambda \mathbf{D}_{(\mathbf{U}_{k+1}, \mathbf{V}_k)}. \tag{4.23}$$

By solving (4.16) and (4.21) we obtain analytical expressions for $\mathbf{U}_{k+1}$ and $\mathbf{V}_{k+1}$ that constitute the main steps of the proposed denoising algorithm given in Algorithm 4.1. As explained in Section 4.3.4, Algorithm 4.1 is an alternating IRLS (AIRLS) algorithm for low-rank matrix factorization applied to data denoising.

Algorithm 4.1: AIRLS denoising algorithm

```
Input: $\mathbf{Y}, \lambda > 0$
Initialize: $k = 0, \mathbf{V}_0, \mathbf{U}_0, \mathbf{D}_{(\mathbf{U}_0, \mathbf{V}_0)}$
repeat
    $\mathbf{U}_{k+1} = \mathbf{Y}\mathbf{V}_k \left(\mathbf{V}_k^T \mathbf{V}_k + \lambda \mathbf{D}_{(\mathbf{U}_k, \mathbf{V}_k)}\right)^{-1}$
    $\mathbf{V}_{k+1} = \mathbf{Y}^T \mathbf{U}_{k+1} \left(\mathbf{U}_{k+1}^T \mathbf{U}_{k+1} + \lambda \mathbf{D}_{(\mathbf{U}_{k+1}, \mathbf{V}_k)}\right)^{-1}$
    $k = k + 1$
until convergence
Output: $\hat{\mathbf{U}} = \mathbf{U}_{k+1}, \hat{\mathbf{V}} = \mathbf{V}_{k+1}$
```

[5] Note that $\bar{\mathbf{H}}_{\mathbf{V}_k}$ is also an approximation of the exact Hessian $\mathbf{H}_{\mathbf{V}_k}$ of $f(\mathbf{U}_{k+1}, \mathbf{V})$ at $\mathbf{V}_k$.

4.3.3.2 Alternating iteratively reweighed least squares matrix completion algorithm

Next the matrix completion problem, under the matrix factorization setting stated in (4.13), is addressed. As mentioned earlier, matrix factorization offers scalability making the derived algorithms amenable to processing big and high dimensional data. It should be emphasized that in the proposed formulation of the problem (4.13), the impediments arising by the low-rank promoting term (Remark 4.1) are now complemented by the difficulty to get computationally efficient *matrix-wise* updates for $\mathbf{U}$ and $\mathbf{V}$, due to the presence of the sampling operator $\mathcal{P}_\Omega$ in the data fitting term. That said, the objective function is now modified as

$$f(\mathbf{U},\mathbf{V}) = \frac{1}{2}\|\mathcal{P}_\Omega\left(\mathbf{Y} - \mathbf{U}\mathbf{V}^T\right)\|_F^2 + \lambda\sum_{i=1}^{d}(\|\boldsymbol{u}_i\|_2^2 + \|\boldsymbol{v}_i\|_2^2 + \eta^2)^{p/2}. \tag{4.24}$$

As in the denoising problem, we minimize quadratic upper bound functions based on approximate second-order Taylor expansions. In this respect, in order to get closed-form analytical expressions for $\mathbf{U}_{k+1}$ and $\mathbf{V}_{k+1}$ that involve exclusively matrix operations, we select again the upper bound functions $l(\mathbf{U}|\mathbf{U}_k,\mathbf{V}_k)$ and $g(\mathbf{V}|\mathbf{U}_{k+1},\mathbf{V}_k)$ defined in (4.17) and (4.22), with $\bar{\mathbf{H}}_{\mathbf{U}_k}$ and $\bar{\mathbf{H}}_{\mathbf{V}_k}$ as given before, with the difference that now $f(\mathbf{U},\mathbf{V})$ is defined as in (4.24). The resulting efficient matrix-wise update formulas are shown in Algorithm 4.2, where the new AIRLS matrix completion algorithm (AIRLS-MC) is presented.

Having presented the above two algorithms we give next some important remarks that apply for both of them and stem mainly from the fact that both of the lie in the BSUM framework.

Remark 4.2. *For $\lambda > 0$, approximation matrices $\bar{\mathbf{H}}_{\mathbf{U}_k}$ and $\bar{\mathbf{H}}_{\mathbf{V}_k}$ are always positive definite and hence invertible. As a consequence, both $l(\mathbf{U}|\mathbf{U}_k,\mathbf{V}_k)$ and $g(\mathbf{V}|\mathbf{U}_{k+1},\mathbf{V}_k)$ are strictly convex and hence they have unique minimizers. In addition, since approximations of the exact Hessians are used in the two block problems, we end up with quasi-Newton type update formulas for* $\mathbf{U}$ *and* $\mathbf{V}$.

Remark 4.3. *The gain of using matrices $\bar{\mathbf{H}}_{\mathbf{U}_k}$ and $\bar{\mathbf{H}}_{\mathbf{V}_k}$ in the approximation of the exact Hessians of $f(\mathbf{U},\mathbf{V})$ (given either by (4.15) or (4.24)) w.r.t.* $\mathbf{U}$ *and* $\mathbf{V}$ *is twofold. Not only (as proven Section 4.3.5) we remain in the BSUM framework, which offers favorable theoretical properties, but also we are able to update* $\mathbf{U}$ *and* $\mathbf{V}$ *at a very low computational cost. As it can be noticed in Algorithms 4.1 and 4.2, the inversions of $\bar{\mathbf{H}}_{\mathbf{U}_k}$ and $\bar{\mathbf{H}}_{\mathbf{V}_k}$ involved in the updates of* $\mathbf{U}$ *and* $\mathbf{V}$ *reduce to the inversions of the $d \times d$ matrices $\tilde{\mathbf{H}}_{\mathbf{U}_k}$ and $\tilde{\mathbf{H}}_{\mathbf{V}_k}$ thus inducing complexity in the order of $\mathcal{O}(d^3)$. Contrary, utilization of the exact Hessians w.r.t.* $\mathbf{U}$ *and* $\mathbf{V}$ *would have given rise to inversions with much higher computational complexity, i.e., $\mathcal{O}(\max(m,n) \times d^3)$.*

4.3.4 Relation to prior art

Both AIRLS and AIRLS-MC algorithms presented above belong to the family of iteratively reweighted least squares minimization algorithms, which date back to the 1930's [13].

Algorithm 4.2: AIRLS matrix completion (AIRLS-MC) algorithm

```
Input: $\mathbf{Y}, \lambda > 0$
Initialize: $k = 0, \mathbf{U}_0, \mathbf{V}_0, \mathbf{D}_{(\mathbf{U}_0,\mathbf{V}_0)}$
repeat
    $\mathbf{U}_{k+1} = \mathbf{U}_k - \left(\mathcal{P}_\Omega\left(\mathbf{U}_k\mathbf{V}_k^T - \mathbf{Y}\right)\mathbf{V}_k + \lambda\mathbf{U}_k\mathbf{D}_{(\mathbf{U}_k,\mathbf{V}_k)}\right)\left(\mathbf{V}_k^T\mathbf{V}_k + \lambda\mathbf{D}_{(\mathbf{U}_k,\mathbf{V}_k)}\right)^{-1}$
    $\mathbf{V}_{k+1} = \mathbf{V}_k - \left(\mathcal{P}_\Omega\left(\mathbf{V}_k\mathbf{U}_{k+1}^T - \mathbf{Y}^T\right)\mathbf{U}_{k+1} + \lambda\mathbf{V}_k\mathbf{D}_{(\mathbf{U}_{k+1},\mathbf{V}_k)}\right)$
                $\times \left(\mathbf{U}_{k+1}^T\mathbf{U}_{k+1} + \lambda\mathbf{D}_{(\mathbf{U}_{k+1},\mathbf{V}_k)}\right)^{-1}$
    $k = k + 1$
until convergence
Output: $\hat{\mathbf{U}} = \mathbf{U}_{k+1}, \hat{\mathbf{V}} = \mathbf{V}_{k+1}$
```

Recently, the IRLS method has been adopted for sparse vector recovery in [41], leading to an iterative algorithm that solves the following minimization problem at the $(k+1)$th iteration

$$\mathbf{x}_{k+1} = \arg\min_{\mathbf{x}} \sum_{i=1}^{m} w_i^k x_i^2 \quad s.t. \quad \mathcal{A}(\mathbf{x}) = \mathbf{b}, \tag{4.25}$$

where $\mathbf{x} = [x_1, x_2, \ldots, x_m]^T \in \mathcal{R}^{m\times 1}$ is the sparse vector to be recovered and $w_i^k = (|x_i^k|^2 + \eta^2)^{p/2-1}$. Theoretical guarantees for sparse signal recovery have been provided in [41] for $p = 1$. Generalizing, the minimization problem in (4.25) can be extended to promote structured (group) sparsity as follows

$$\mathbf{x}_{k+1} = \arg\min_{\mathbf{x}} \sum_{i=1}^{d} w_i^k ||\mathbf{x}_i||_2^2 \quad s.t. \quad \mathcal{A}(\mathbf{x}) = \mathbf{b}, \tag{4.26}$$

where now the m-dimensional vector $\mathbf{x}$ is structured in d groups, i.e., $\mathbf{x} = [\mathbf{x}_1^T, \mathbf{x}_2^T, \ldots, \mathbf{x}_d^T]^T$ and $w_i^k = (||\mathbf{x}_i^k||_2^2 + \eta^2)^{p/2-1}$.

More recently, the same idea has been applied for low-rank matrix recovery in [107]. In this vein the minimization problem is properly adjusted as,

$$\mathbf{X}_{k+1} = \arg\min_{\mathbf{X}} \text{tr}(\mathbf{W}_k\mathbf{X}^T\mathbf{X}) \quad \text{subject to} \quad \mathcal{A}(\mathbf{X}) = \mathbf{b}, \tag{4.27}$$

and $\mathbf{W}_k = (\mathbf{X}_k^T\mathbf{X}_k + \eta^2\mathbf{I}_n)^{p/2-1}$. As explained in Section 1.2.1 and Eq. (1.20), this problem is equivalent to minimizing the Schatten-p quasinorm of $\mathbf{X}$, thus promoting low-rank solutions.

To place our method in the above described framework, we rewrite our generic optimization problem, given in (4.8), as follows,

$$\min_{\mathbf{U},\mathbf{V}} \sum_{i=1}^{d} (\|\boldsymbol{u}_i\|_2^2 + \|\boldsymbol{v}_i\|_2^2)^{p/2-1}(\|\boldsymbol{u}_i\|_2^2 + \|\boldsymbol{v}_i\|_2^2) \quad \text{subject to} \quad \mathcal{A}(\mathbf{U}\mathbf{V}^T) = \mathbf{b}. \tag{4.28}$$

Then, from (4.28) we can define the following IRLS minimization scheme

$$\{\mathbf{U}_{k+1}, \mathbf{V}_{k+1}\} = \underset{\mathbf{U},\mathbf{V}}{\text{argmin}} \sum_{i=1}^{d} w_i^k (\|\boldsymbol{u}_i\|_2^2 + \|\boldsymbol{v}_i\|_2^2) \text{ subject to } \mathcal{A}(\mathbf{U}\mathbf{V}^T) = \mathbf{b}, \tag{4.29}$$

where $w_i^k = (\|\boldsymbol{u}_i^k\|_2^2 + \|\boldsymbol{v}_i^k\|_2^2 + \eta^2)^{p/2-1}$. This optimization task can be solved alternatingly with respect to **U** and **V** as follows,

$$\mathbf{U}_{k+1} = \underset{\mathbf{U}}{\text{argmin}} \sum_{i=1}^{d} w_i^{k,k} \|\boldsymbol{u}_i\|_2^2 \text{ subject to } \mathcal{A}(\mathbf{U}\mathbf{V}_k^T) = \mathbf{b}, \tag{4.30}$$

$$\mathbf{V}_{k+1} = \underset{\mathbf{V}}{\text{argmin}} \sum_{i=1}^{d} w_i^{k+1,k} \|\boldsymbol{v}_i\|_2^2 \text{ subject to } \mathcal{A}(\mathbf{U}_{k+1}\mathbf{V}^T) = \mathbf{b}, \tag{4.31}$$

where $w_i^{k,k} = (\|\boldsymbol{u}_i^k\|_2^2 + \|\boldsymbol{v}_i^k\|_2^2 + \eta^2)^{p/2-1}$ and $w_i^{k+1,k} = (\|\boldsymbol{v}_i^k\|_2^2 + \|\boldsymbol{u}_i^{k+1}\|_2^2 + \eta^2)^{p/2-1}$. It can be shown that if we consider a LS data fitting term in our objective function, the solution of the IRLS schemes (4.30) and (4.31) leads to the same exact expressions for $\mathbf{U}_{k+1}$ and $\mathbf{V}_{k+1}$ as those obtained for AIRLS in the previous section. Note that (4.30) and (4.31) hold close resemblance with the minimization problem (4.26) via the correspondence of the block vectors $\mathbf{x}_i$ with the column vectors $\mathbf{u}_i$ and $\mathbf{v}_i$ respectively. Hence, as (4.26) imposes group sparsity on a vector quantity, (4.30) and (4.31) are expected to induce column sparsity on the matrix $\left[\begin{smallmatrix}\mathbf{U}\\ \mathbf{V}\end{smallmatrix}\right]$ thus promoting low-rankness in a matrix factorization framework. This key feature of the proposed algorithms allows us to incorporate a *pruning procedure* which removes the columns that become zero as the algorithms evolve. By doing so, the per iteration computational complexity of the algorithms is gradually reduced, and this reduction may contribute significantly to the reduction of the total computational time required for convergence, as is also highlighted in Section 4.5, where empirical numerical results are presented.

4.3.5 Convergence analysis

In this subsection we analyze the convergence behavior of AIRLS and AIRLS-MC as presented above and without considering the above mentioned pruning procedure, which is basically an algorithmic mechanism to reduce complexity. The analysis is common for the two algorithms, since, as mentioned above, both minimize upper bound surrogate functions of the same form. We begin by first proving the following Lemma.

Lemma 4.1. *The surrogate functions $l(\mathbf{U}|\mathbf{U}_k, \mathbf{V}_k)$ and $g(\mathbf{V}|\mathbf{U}_{k+1}, \mathbf{V}_k)$ minimized at each iteration of AIRLS and AIRLS-MC are tight upper bounds of $f(\mathbf{U}, \mathbf{V}_k)$ and $f(\mathbf{U}_{k+1}, \mathbf{V})$, with $f(\mathbf{U}, \mathbf{V})$ being defined in (4.15) and (4.24) for the two algorithms, respectively.*

Proof: The surrogate functions $l(\mathbf{U}|\mathbf{U}_k, \mathbf{V}_k)$ and $g(\mathbf{V}|\mathbf{U}_{k+1}, \mathbf{V}_k)$ given in (4.17) and (4.22), are twice continuously differentiable and constitute approximations of the second-order Taylor expansions of the initial cost functions around $(\mathbf{U}_k, \mathbf{V}_k)$ and $(\mathbf{U}_{k+1}, \mathbf{V}_k)$ respectively. In (4.17), the true Hessian $\mathbf{H}_{\mathbf{U}_k}$ of $f(\mathbf{U}, \mathbf{V}_k)$ at $\mathbf{U}_k$ has been approximated by the $md \times md$

positive-definite block diagonal matrix $\bar{\mathbf{H}}_{\mathbf{U}_k}$ defined in (4.18). $\bar{\mathbf{H}}_{\mathbf{V}_k}$ is similarly defined. Our analysis is next focused on $l(\mathbf{U}|\mathbf{U}_k, \mathbf{V}_k)$. It can be easily shown that similar derivations can be made for $g(\mathbf{V}|\mathbf{U}_{k+1}, \mathbf{V}_k)$. As it can be seen by Eq.. (4.17), $l(\mathbf{U}|\mathbf{U}_k, \mathbf{V}_k)$ equals $f(\mathbf{U}, \mathbf{V}_k)$ at $(\mathbf{U}_k, \mathbf{V}_k)$. In order to show that it majorizes $f(\mathbf{U}, \mathbf{V}_k)$ for all other points closeby, it suffices to show that matrix $\mathbf{A} = \bar{\mathbf{H}}_{\mathbf{U}_k} - \mathbf{H}_{\mathbf{U}_k}$ is positive semidefinite [120]. Next we prove that for each of the two problems examined, the above-mentioned property holds for $\mathbf{A}$.

In denoising it is $\tilde{\mathbf{H}}_{\mathbf{U}_k} = \mathbf{V}_k^T \mathbf{V}_k + \lambda \mathbf{D}_{(\mathbf{U}_k, \mathbf{V}_k)}$, where $\mathbf{D}_{(\mathbf{U}_k, \mathbf{V}_k)}$ is defined in Eq. (4.20). Moreover, it can be shown that for the exact Hessian $\mathbf{H}_{\mathbf{U}_k}$ at $\mathbf{U}_k$ we get

$$\mathbf{H}_{\mathbf{U}_k} = \mathbf{I}_m \otimes (\mathbf{V}_k^T \mathbf{V}_k) + \lambda \mathbf{K}, \tag{4.32}$$

where $\mathbf{K} = [\mathbf{K}_{ij}], i, j = 1, 2, \ldots, m$ consists of $d \times d$ blocks $\mathbf{K}_{ij}$ defined as follows.

$$\mathbf{K}_{ij} = \begin{cases} p\text{diag}\left(\frac{\|\boldsymbol{u}_1^k\|_2^2 + \|\boldsymbol{v}_1^k\|_2^2 - (2-p)(u_{i1}^k)^2 + \eta^2}{\left(\|\boldsymbol{u}_1^k\|_2^2 + \|\boldsymbol{v}_1^k\|_2^2 + \eta^2\right)^{2-p/2}}, \cdots, \frac{\|\boldsymbol{u}_d^k\|_2^2 + \|\boldsymbol{v}_d^k\|_2^2 - (2-p)(u_{id}^k)^2 + \eta^2}{\left(\|\boldsymbol{u}_d^k\|_2^2 + \|\boldsymbol{v}_d^k\|_2^2 + \eta^2\right)^{2-p/2}} \right), \text{if } i = j \\ \\ p(2-p)\text{diag}\left(\frac{-u_{i1}^k u_{j1}^k}{\left(\|\boldsymbol{u}_1^k\|_2^2 + \|\boldsymbol{v}_1^k\|_2^2 + \eta^2\right)^{2-p/2}}, \cdots, \frac{-u_{id}^k u_{jd}^k}{\left(\|\boldsymbol{u}_d^k\|_2^2 + \|\boldsymbol{v}_d^k\|_2^2 + \eta^2\right)^{2-p/2}} \right), \text{if } i \neq j \end{cases} \tag{4.33}$$

Hence, after some algebraic manipulations, the matrix $\mathbf{A} = [\mathbf{A}_{ij}]$ is expressed as

$$\mathbf{A} = \mathbf{I}_m \otimes \mathbf{D}_{(\mathbf{U}_k, \mathbf{V}_k)} - \lambda \mathbf{K}. \tag{4.34}$$

Elaborating on $\mathbf{A}$ we get from (4.34), (4.33) and (4.20),

$$\mathbf{A}_{ij} = \lambda p(2-p)\text{diag}\Big(\frac{u_{i1}^k u_{j1}^k}{\left(\|\boldsymbol{u}_1^k\|_2^2 + \|\boldsymbol{v}_1^k\|_2^2 + \eta^2\right)^{2-p/2}}, \cdots, \frac{u_{id}^k u_{jd}^k}{\left(\|\boldsymbol{u}_d^k\|_2^2 + \|\boldsymbol{v}_d^k\|_2^2 + \eta^2\right)^{2-p/2}} \Big). \tag{4.35}$$

Notice that for

$$\mathbf{B}_i = \sqrt{\lambda p(2-p)}\text{diag}\Big(\frac{u_{i1}^k}{\left(\|\boldsymbol{u}_1^k\|_2^2 + \|\boldsymbol{v}_1^k\|_2^2 + \eta^2\right)^{1-p/4}}, \ldots, \frac{u_{id}^k}{\left(\|\boldsymbol{u}_d^k\|_2^2 + \|\boldsymbol{v}_d^k\|_2^2 + \eta^2\right)^{1-p/4}} \Big) \tag{4.36}$$

$\mathbf{A}_{ij} = \mathbf{B}_i^T \mathbf{B}_j$. So by defining $\mathbf{B} = [\mathbf{B}_1, \ldots, \mathbf{B}_m]$, it is straightforward that $\mathbf{A} = \mathbf{B}^T \mathbf{B}$, that is $\mathbf{A}$ is positive semidefinite.

In matrix completion, the approximate Hessian $\bar{\mathbf{H}}_{\mathbf{U}_k}$ is exactly the same with that in the denoising case, while the exact Hessian $\mathbf{H}_{\mathbf{U}_k}$ differs from its denoising counterpart given in (4.32) in the diagonal blocks only. More specifically, the ith diagonal block of $\mathbf{H}_{\mathbf{U}_k}$ takes now the form $\mathbf{V}^T \boldsymbol{\Omega}_i \mathbf{V} + \mathbf{K}_{ii}$, where $\boldsymbol{\Omega}_i$ is a $n \times n$ diagonal matrix containing ones on indexes included in the set Ω and related to the ith row of $\mathbf{Y}$ and zeros elsewhere. Since $\mathbf{V}^T \mathbf{V} - (\mathbf{V}^T \boldsymbol{\Omega}_i \mathbf{V}) \succeq 0$, we can use the same arguments as above for proving the semidefiniteness of the respective matrix $\mathbf{A}$. ■

Having shown that the proposed surrogate objective functions are upper bounds of the actual ones, in Proposition 4.1 given below the monotonic decrease of the initial objective functions defined in (4.15) and (4.24) of the respective algorithms is established.

Proposition 4.1. *The sequences of* $\{\mathbf{U}_k, \mathbf{V}_k\}$ *generated by AIRLS and AIRLS-MC decrease monotonically the respective objective functions, i.e.,*

$$f(\mathbf{U}_{k+1}, \mathbf{V}_{k+1}) \leq f(\mathbf{U}_{k+1}, \mathbf{V}_k) \leq f(\mathbf{U}_k, \mathbf{V}_k). \tag{4.37}$$

Proof: Since $\mathbf{U}_{k+1} = \underset{\mathbf{U}}{\text{argmin}} \quad l(\mathbf{U}|\mathbf{U}_k, \mathbf{V}_k)$ we get

$$l(\mathbf{U}_{k+1}|\mathbf{U}_k, \mathbf{V}_k) \leq l(\mathbf{U}_k|\mathbf{U}_k, \mathbf{V}_k) \equiv f(\mathbf{U}_k, \mathbf{V}_k). \tag{4.38}$$

From Lemma 4.1 we have,

$$f(\mathbf{U}, \mathbf{V}_k) \leq l(\mathbf{U}|\mathbf{U}_k, \mathbf{V}_k) \tag{4.39}$$

and, as a consequence,

$$f(\mathbf{U}_{k+1}, \mathbf{V}_k) \leq l(\mathbf{U}_{k+1}|\mathbf{U}_k, \mathbf{V}_k), \tag{4.40}$$

which leads to

$$f(\mathbf{U}_{k+1}, \mathbf{V}_k) \leq f(\mathbf{U}_k, \mathbf{V}_k). \tag{4.41}$$

Following the same reasoning, and since $\mathbf{V}_{k+1} = \underset{\mathbf{V}}{\text{argmin}} \quad g(\mathbf{V}|\mathbf{U}_{k+1}, \mathbf{V}_k)$ we get

$$f(\mathbf{U}_{k+1}, \mathbf{V}_{k+1}) \leq g(\mathbf{V}_{k+1}|\mathbf{U}_{k+1}, \mathbf{V}_k) \leq g(\mathbf{V}_k|\mathbf{U}_{k+1}, \mathbf{V}_k) \equiv f(\mathbf{U}_{k+1}, \mathbf{V}_k) \tag{4.42}$$

Combining (4.41) and (4.42) we get (4.37). ■

Corollary 4.1. *The sequence* $f(\mathbf{U}_k, \mathbf{V}_k)$ *converges to* $f^\infty \geq 0$*, as* $k \to \infty$*, for both AIRLS and AIRLS-MC.*

Proof: Since the objective functions for both algorithms are monotonically decreasing (Proposition 4.1) and bounded below by 0, the claim follows immediately. ■

4.3.5.1 Convergence to stationary points and rate of convergence

Having shown that the sequences $(\mathbf{U}_k, \mathbf{V}_k)|_{k=1}^{+\infty}$ generated by AIRLS and AIRLS-MC monotonically decrease the corresponding objective functions, we herein prove the convergence of the algorithms to the stationary points of their associated cost functions and derive the rates of convergence of the algorithms to these stationary points. The subsequent analysis is along the lines of the one presented in [77].

Given any pair $(\mathbf{U}, \mathbf{V})$ we define matrices $\mathbf{U}_*, \mathbf{V}_*$ resulting by the following minimization problems

$$\mathbf{U}_* = \underset{\mathbf{U}^+}{\text{argmin}} \quad l(\mathbf{U}^+|\mathbf{U}, \mathbf{V}) \quad \mathbf{V}_* = \underset{\mathbf{V}^+}{\text{argmin}} \quad g(\mathbf{V}^+|\mathbf{U}_*, \mathbf{V}).$$

Let us now denote as $\Delta((\mathbf{U},\mathbf{V}),(\mathbf{U}_*,\mathbf{V}_*))$ the following measure of proximity between $(\mathbf{U},\mathbf{V})$ and $(\mathbf{U}_*,\mathbf{V}_*)$,

$$\begin{aligned}\Delta((\mathbf{U},\mathbf{V}),(\mathbf{U}_*,\mathbf{V}_*)) = &\frac{1}{2}\Big(\|\mathbf{V}\left(\mathbf{U}-\mathbf{U}_*\right)^T\|_F^2 + \|\mathbf{U}_*\left(\mathbf{V}-\mathbf{V}_*\right)^T\|_F^2\Big)\\ &+\frac{\lambda}{2}\Big(\|\mathbf{D}^{\frac{1}{2}}_{(\mathbf{U},\mathbf{V})}\left(\mathbf{U}-\mathbf{U}_*\right)^T\|_F^2 + \|\mathbf{D}^{\frac{1}{2}}_{(\mathbf{U}_*,\mathbf{V})}\left(\mathbf{V}-\mathbf{V}_*\right)^T\|_F^2\Big). \end{aligned} \tag{4.43}$$

Lemma 4.2. *Successive differences in the values of cost functions $f(\mathbf{U},\mathbf{V})$ corresponding to AIRLS and AIRLS-MC are bounded below as follows*,

$$f(\mathbf{U}_k,\mathbf{V}_k) - f(\mathbf{U}_{k+1},\mathbf{V}_{k+1}) \geq \Delta((\mathbf{U}_k,\mathbf{V}_k),(\mathbf{U}_{k+1},\mathbf{V}_{k+1})). \tag{4.44}$$

Proof: Using Eqs. (4.38), (4.40) and (4.42), we have,

$$f(\mathbf{U}_k,\mathbf{V}_k) - f(\mathbf{U}_{k+1},\mathbf{V}_k) \geq l(\mathbf{U}_k|\mathbf{U}_k,\mathbf{V}_k) - l(\mathbf{U}_{k+1}|\mathbf{U}_k,\mathbf{V}_k) \quad \text{and} \tag{4.45}$$

$$f(\mathbf{U}_{k+1},\mathbf{V}_k) - f(\mathbf{U}_{k+1},\mathbf{V}_{k+1}) \geq g(\mathbf{V}_k|\mathbf{U}_{k+1},\mathbf{V}_k) - g(\mathbf{V}_{k+1}|\mathbf{U}_{k+1},\mathbf{V}_k) \tag{4.46}$$

Adding (4.45) and (4.46) we reach to the following inequality

$$\begin{aligned}f(\mathbf{U}_k,\mathbf{V}_k) - f(\mathbf{U}_{k+1},\mathbf{V}_{k+1}) \geq & l(\mathbf{U}_k|\mathbf{U}_k,\mathbf{V}_k) - l(\mathbf{U}_{k+1}|\mathbf{U}_k,\mathbf{V}_k)\\ &+ g(\mathbf{V}_k|\mathbf{U}_{k+1},\mathbf{V}_k) - g(\mathbf{V}_{k+1}|\mathbf{U}_{k+1},\mathbf{V}_k)\end{aligned} \tag{4.47}$$

Since $\mathbf{U}_{k+1}$ and $\mathbf{V}_{k+1}$ are stationary points of $l(\mathbf{U}|\mathbf{U}_k,\mathbf{V}_k)$ and $g(\mathbf{V}|\mathbf{U}_{k+1},\mathbf{V}_k)$ respectively ($\nabla_{\mathbf{U}} l(\mathbf{U}_{k+1}|\mathbf{U}_k,\mathbf{V}_k) = \mathbf{0}$ and $\nabla_{\mathbf{V}} g(\mathbf{V}_{k+1}|\mathbf{U}_{k+1},\mathbf{V}_k) = \mathbf{0}$) and by their second-order Taylor expansions around $(\mathbf{U}_{k+1},\mathbf{V}_k)$ and $(\mathbf{U}_{k+1},\mathbf{V}_{k+1})$ we have

$$\begin{aligned}l(\mathbf{U}_k|\mathbf{U}_k,\mathbf{V}_k) - l(\mathbf{U}_{k+1}|\mathbf{U}_k,\mathbf{V}_k) &= \frac{1}{2}\text{tr}\{\left(\mathbf{U}_k-\mathbf{U}_{k+1}\right)\left(\mathbf{V}_k^T\mathbf{V}_k + \lambda\mathbf{D}_{(\mathbf{U}_k,\mathbf{V}_k)}\right)\left(\mathbf{U}_k-\mathbf{U}_{k+1}\right)^T\}\\ &= \frac{1}{2}\|\mathbf{V}_k\left(\mathbf{U}_k-\mathbf{U}_{k+1}\right)^T\|_F^2 + \frac{\lambda}{2}\|\mathbf{D}^{\frac{1}{2}}_{(\mathbf{U}_k,\mathbf{V}_k)}\left(\mathbf{U}_k-\mathbf{U}_{k+1}\right)^T\|_F^2\end{aligned} \tag{4.48}$$

and

$$\begin{aligned}g(\mathbf{V}_k|\mathbf{U}_{k+1},\mathbf{V}_k) - g(\mathbf{V}_{k+1}|\mathbf{U}_{k+1},\mathbf{V}_k) = &\frac{1}{2}\text{tr}\{\left(\mathbf{V}_k-\mathbf{V}_{k+1}\right)\big(\mathbf{U}_{k+1}^T\mathbf{U}_{k+1}\\ &+\lambda\mathbf{D}_{(\mathbf{U}_{k+1},\mathbf{V}_k)}\big)\left(\mathbf{V}_{k+1}-\mathbf{V}_k\right)^T\}\\ = \frac{1}{2}\|\mathbf{U}_{k+1}\left(\mathbf{V}_k-\mathbf{V}_{k+1}\right)^T\|_F^2 &+ \frac{\lambda}{2}\|\mathbf{D}^{\frac{1}{2}}_{(\mathbf{U}_{k+1},\mathbf{V}_k)}\left(\mathbf{V}_k-\mathbf{V}_{k+1}\right)^T\|_F^2\end{aligned} \tag{4.49}$$

Combining (4.48), (4.49) and (4.47) we get inequality (4.44). ■

Lemma 4.3. $\Delta((\mathbf{U},\mathbf{V}),(\mathbf{U}_*,\mathbf{V}_*)) = 0$ *if and only if* $(\mathbf{U},\mathbf{V})$ *generated by AIRLS (AIRLS-MC) algorithm is a fixed point of AIRLS (AIRLS-MC).*

Proof: If $(\mathbf{U},\mathbf{V})$ is a fixed point, i.e. $\mathbf{U} = \mathbf{U}_*$ and $\mathbf{V} = \mathbf{V}_*$, then it is easily shown that $\Delta((\mathbf{U},\mathbf{V}),(\mathbf{U}_*,\mathbf{V}_*)) = 0$. Conversely, using (4.48) and (4.49) and since all the summands of $\Delta((\mathbf{U},\mathbf{V}),(\mathbf{U}_*,\mathbf{V}_*))$ are nonnegative, we have that if $\Delta((\mathbf{U},\mathbf{V}),(\mathbf{U}_*,\mathbf{V}_*))=0$ then

$$l(\mathbf{U}|\mathbf{U},\mathbf{V})-l(\mathbf{U}_*|\mathbf{U},\mathbf{V})=0 \tag{4.50}$$

$$\text{and }\; g(\mathbf{V}|\mathbf{U}_*,\mathbf{V})-g(\mathbf{V}_*|\mathbf{U}_*,\mathbf{V})=0. \tag{4.51}$$

Since both $l(\mathbf{U}|\mathbf{U},\mathbf{V})$ and $g(\mathbf{V}|\mathbf{U}_*,\mathbf{V})$ are strictly convex functions, $\mathbf{U}_*$ and $\mathbf{V}_*$ are unique. Hence the above equalities hold only if $(\mathbf{U},\mathbf{V})=(\mathbf{U}_*,\mathbf{V}_*)$, that is $(\mathbf{U},\mathbf{V})$ is a fixed point of AIRLS (AIRLS-MC). ■

As stated above, $\Delta((\mathbf{U}_k,\mathbf{V}_k),(\mathbf{U}_{k+1},\mathbf{V}_{k+1}))$ is actually used for quantifying the distance between $(\mathbf{U}_k,\mathbf{V}_k)$ and $(\mathbf{U}_{k+1},\mathbf{V}_{k+1})$ generated at successive iterations of the proposed algorithms. Thus, it is obvious that this measure will become equal to zero if a fixed point has been reached. For ease of notation, we will next denote this quantity as δ_k. That said, the main result of this section is summarized in the following proposition.

Proposition 4.2. *a) Any limit point of the sequences $(\mathbf{U}_k,\mathbf{V}_k)$ generated by AIRLS and AIRLS-MC is a stationary point of the respective objective function $f(\mathbf{U},\mathbf{V})$, for $\lambda>0$. b) AIRLS and AIRLS-MC converge sublinearly to stationary points with their rates of convergence expressed as*

$$\min_{1\leq k\leq K}\delta_k\leq\frac{f(\mathbf{U}_1,\mathbf{V}_1)-f^\infty}{K}. \tag{4.52}$$

Proof: a) We say that $(\mathbf{U}_*,\mathbf{V}_*)$ is a first-order stationary point of $f(\mathbf{U},\mathbf{V})$ (given either in (4.15) or (4.24)) if the following holds

$$\nabla_{\mathbf{U}}f(\mathbf{U}_*,\mathbf{V}_*)=\mathbf{0},\;\;\nabla_{\mathbf{V}}f(\mathbf{U}_*,\mathbf{V}_*)=\mathbf{0}. \tag{4.53}$$

Due to the adopted upper bound minimization approach, it is easily shown that (4.53) can be equivalently restated as [77],

$$\mathbf{U}_*=\arg\min_{\mathbf{U}}l(\mathbf{U}|\mathbf{U}_*,\mathbf{V}_*),\;\;\mathbf{V}_*=\arg\min_{\mathbf{V}}g(\mathbf{V}|\mathbf{U}_*,\mathbf{V}_*), \tag{4.54}$$

i.e., $(\mathbf{U}_*,\mathbf{V}_*)$, being a stationary point of $f(\mathbf{U},\mathbf{V})$, is also a fixed-point of the algorithm (AIRLS or AIRLS-MC) and vice-versa.

In the light of the above, it suffices to show that any limit point of the sequence generated by the algorithms is also a fixed point of them. To this end, for $\lambda>0$, the sequence $(\mathbf{U}_k,\mathbf{V}_k)_{k=1}^{\infty}$ generated by AIRLS (AIRLS-MC) remains bounded and thus contains convergent subsequences. Let $(\mathbf{U}_*,\mathbf{V}_*)$ be a limit point of AIRLS (AIRLS-MC). That said, there will be a subsequence $\{\mathbf{U}_k,\mathbf{V}_k\}$ that converges to $(\mathbf{U}_*,\mathbf{V}_*)$ hence $\Delta((\mathbf{U}_k,\mathbf{V}_k),(\mathbf{U}_*,\mathbf{V}_*))\to 0$. From Lemma 4.3, we know that $\Delta((\mathbf{U}_k,\mathbf{V}_k),(\mathbf{U}_*,\mathbf{V}_*))=0$ iff $(\mathbf{U}_*,\mathbf{V}_*)$ is a fixed point of the algorithms. Hence, due to the equivalence of (4.53) and (4.54), it can be easily conjectured that $(\mathbf{U}_*,\mathbf{V}_*)$ will also be a stationary point of the cost function.

b) Recall that $\delta_k=\Delta((\mathbf{U}_k,\mathbf{V}_k),(\mathbf{U}_{k+1},\mathbf{V}_{k+1}))$. Then from (4.44) by adding K successive terms we get,

$$\sum_{k=1}^{K} \delta_k \leq f(\mathbf{U}_1, \mathbf{V}_1) - f(\mathbf{U}_K, \mathbf{V}_K) \leq f(\mathbf{U}_1, \mathbf{V}_1) - f^{\infty} < \infty. \tag{4.55}$$

Note that all the terms of the sequence δ_k take nonnegative values. Let us now assume that there exists a (infinite) subsequence of δ_k that converges to a positive number. In such a case the sum $\sum_{k=1}^{K} \delta_k$ would not be bounded as $K \to \infty$, which contradicts (4.55). Therefore, all subsequences of δ_k converge to zero, i.e. the sequence δ_k also converges to zero. From Lemma 4.3, the zero limit point of δ_k corresponds in fact to a fixed point of AIRLS (AIRLS-MC) which as said above, is a stationary point of the respective objective function $f(\mathbf{U}, \mathbf{V})$.

By substituting the first part of inequality (4.55) by $K \min\limits_{1 \leq k \leq K} \delta_k \leq \sum_{k=1}^{K} \delta_k$ and solving for $\min\limits_{1 \leq k \leq K} \delta_k$ we get (4.52), which establishes a sublinear convergence rate for the proposed algorithms [77]. ■

Assumption 4.1. *The eigenvalues of $\mathbf{U}_k^T \mathbf{U}_k$ and $\mathbf{V}_k^T \mathbf{V}_k$ for $k \geq 1$ are uniformly bounded below and above by l_L and l_U respectively, i.e.,*

$$l_L \mathbf{I}_d \preceq \mathbf{U}_k^T \mathbf{U}_k \preceq l_U \mathbf{I}_d \quad \textit{and} \quad l_L \mathbf{I}_d \preceq \mathbf{V}_k^T \mathbf{V}_k \preceq l_U \mathbf{I}_d. \tag{4.56}$$

Using Assumption 4.1 we can provide more refined information with regard to the rates of convergence, bringing into play the curvature characteristics of the cost functions as well as the regularization parameter λ.

Corollary 4.2. *Under Assumption 4.1, we can derive the following convergence rate for Algorithms 4.1 and 4.2:*

$$\min_{1 \leq k \leq K} \{ \| \mathbf{U}_{k+1} - \mathbf{U}_k \|_F^2 + \| \mathbf{V}_{k+1} - \mathbf{V}_k \|_F^2 \} \leq \frac{4\tau}{2 l_L \tau + \lambda} \frac{f(\mathbf{U}_1, \mathbf{V}_1) - f^{\infty}}{K}, \tag{4.57}$$

where $\tau = \max\limits_{1 \leq i \leq d} (\|\boldsymbol{u}_i\|_2^2, \|\boldsymbol{v}_i\|_2^2)$.

Proof: It can be easily proved by suitably modifying δ_k using the inequalities $l_L \|\mathbf{U}_k - \mathbf{U}_{k+1}\|_F^2 \leq \|\mathbf{V}_k (\mathbf{U}_k - \mathbf{U}_{k+1})\|_F^2 \leq l_U \|\mathbf{U}_k - \mathbf{U}_{k+1}\|_F^2$ and $l_L \|\mathbf{V}_k - \mathbf{V}_{k+1}\|_F^2 \leq \|\mathbf{U}_{k+1} (\mathbf{V}_k - \mathbf{V}_{k+1})\|_F^2 \leq l_U \|\mathbf{V}_k - \mathbf{V}_{k+1}\|_F^2$. ■

4.4 Low-rank NMF and low-rank and sparse NMF

Herein, the LRMF presented earlier is extended so as to account for problems that involve nonnegative data. In this framework, two novel algorithms are presented, i.e., a) a low-rank NMF and b) a low-rank and sparse NMF algorithm.

4.4.1 Low-rank NMF

Low-rank NMF differs from the classical NMF in the inclusion of the low-rank constraint on the factors **U** and **V**, accounting thus for the unawareness of the true rank. As is shown in Section 4.5 this is very crucial in a class of applications such as music signal decomposition, blind source separation, etc. The emerging optimization problem is given below,

$$\{\hat{\mathbf{U}}, \hat{\mathbf{V}}\} = \underset{\mathbf{U}\geq\mathbf{0},\mathbf{V}\geq\mathbf{0}}{\text{argmin}}\frac{1}{2}\|\mathbf{Y} - \mathbf{U}\mathbf{V}^T\|_F^2 + \lambda \sum_{i=1}^{d} \left(\|\boldsymbol{u}_i\|_2^2 + \|\boldsymbol{v}_i\|_2^2 + \eta^2\right)^{\frac{p}{2}} \tag{4.58}$$

where $\mathbf{U} \geq \mathbf{0}$ and $\mathbf{V} \geq \mathbf{0}$ stand for elementwise nonnegativity of **U** and **V**, respectively. Problem (4.58) deviates from the denoising one of (4.10) in the incorporation of an additional constraint, i.e., the nonnegativity of $\mathbf{U}, \mathbf{V}$.

In what follows, we present a projected Newton-type method for efficiently addressing the problem defined in (4.58). It deserves to notice that we are now dealing with a constrained optimization problem since the solution set of the matrices **U** and **V** contains only elementwise nonnegative matrices. Following the same path presented above we aim at exploiting the curvature information of the formed cost function. However the constrained nature of the NMF problem induces some subtleties needed to be properly handled.

More specifically, the proposed alternating minimization algorithm shall now update matrices **U** and **V** so that they a) always belong to the feasibility set and b) guarantee the descent direction of the cost function at each iteration. The proposed scheme is along the lines of the NMF algorithm proposed in [73]. Each update of the factors takes place by making use of the projected Newton method introduced in [14]. Next, the minimization subproblems for updating the factors **U** and **V** are detailed.

As in the previous algorithms, surrogate quadratic functions of $f(\mathbf{U}, \mathbf{V}_k)$ and $f(\mathbf{U}_{k+1}, \mathbf{V})$ are required for updating matrices **U** and **V** with $f(\mathbf{U}, \mathbf{V})$ being the same as in Eq. (4.15), but now the entries of **U** and **V** belong to the set of nonnegative reals. Let us now consider the so-called set of *active constraints* defined w.r.t. each row $\mathbf{u}_i$ of **U** at iteration k as

$$\mathcal{I}_{\mathbf{u}_i}^k = \{j|0 \leq u_{ij}^k \leq \epsilon^k, [\nabla_{\mathbf{U}} f(\mathbf{U}_k, \mathbf{V}_k)]_{ij} > 0\}, \tag{4.59}$$

where $\epsilon^k = \min(\varepsilon, \|\mathbf{U}_k - \nabla_{\mathbf{U}} f(\mathbf{U}_k, \mathbf{V}_k)\|_F^2)$ (with ε a small positive constant) and u_{ij}s are the elements of matrix **U**. A similar set $\mathcal{I}_{\mathbf{v}_i}^k$ is defined based on the rows $\mathbf{v}_i$ of matrix **V** (the elements of **V** are denoted as v_{ij}s), i.e.,

$$\mathcal{I}_{\mathbf{v}_i}^k = \{j|0 \leq v_{ij}^k \leq \epsilon^k, [\nabla_{\mathbf{V}} f(\mathbf{U}_{k+1}, \mathbf{V}_k)]_{ij} > 0\}. \tag{4.60}$$

As is analytically explained in [73], these sets contain the coordinates of the row elements of matrices **U** and **V** that belong to the boundaries of the constrained sets, and at the same time are stationary at iteration k. To derive a projected Newton NMF algorithm, we replace the exact Hessian of each subproblem, with a positive definite matrix that has been partially diagonalized at each iteration w.r.t. the sets of active constraints defined above. The positive definite matrices utilized in this case, denoted as $\bar{\mathbf{H}}_{\mathbf{U}}^{\mathcal{I}_{\mathbf{U}}}$ and $\bar{\mathbf{H}}_{\mathbf{V}}^{\mathcal{I}_{\mathbf{V}}}$, in analogy to $\bar{\mathbf{H}}_{\mathbf{U}}$ and $\bar{\mathbf{H}}_{\mathbf{V}}$ used in the cases of denoising and matrix completion, are block diagonal, but consist of m and n, respectively, $d \times d$ *distinct* diagonal blocks. That is to say, the ith diagonal blocks of these matrices at iteration k, namely $\tilde{\mathbf{H}}_{\mathbf{U}}^{\mathcal{I}_{\mathbf{u}_i}^k}$ and $\tilde{\mathbf{H}}_{\mathbf{V}}^{\mathcal{I}_{\mathbf{v}_i}^k}$, are partially diagonalized versions of the $d \times d$ matrices $\tilde{\mathbf{H}}_{\mathbf{U}_k}$ and $\tilde{\mathbf{H}}_{\mathbf{V}_k}$ defined in (4.19) and (4.23). More specifically,

$$[\tilde{\mathbf{H}}_{\mathbf{U}}^{\mathcal{I}_{\mathbf{u}_i}^k}]_{pl} = \begin{cases} 0, \text{if } p \neq l, \text{and either } p \in \mathcal{I}_{\mathbf{u}_i}^k \text{ or } l \in \mathcal{I}_{\mathbf{u}_i}^k \\ [\tilde{\mathbf{H}}_{\mathbf{U}_k}]_{pl} \text{ otherwise} \end{cases}$$

and $\tilde{\mathbf{H}}_{\mathbf{V}}^{\mathcal{I}_{\mathbf{v}_i}^k}$ is defined similarly.

Based on the above, the quadratic surrogate functions $l(\mathbf{U}|\mathbf{U}_k, \mathbf{V}_k)$ and $g(\mathbf{V}|\mathbf{U}_{k+1}, \mathbf{V}_k)$ are now expressed as,

$$\begin{aligned} l(\mathbf{U}|\mathbf{U}_k, \mathbf{V}_k) &= f(\mathbf{U}_k, \mathbf{V}_k) + \text{tr}\{(\mathbf{U} - \mathbf{U}_k)^T \nabla_{\mathbf{U}} f(\mathbf{U}_k, \mathbf{V}_k)\} \\ &\quad + \frac{1}{2\alpha_{\mathbf{U}}^k} \text{vec}\,(\mathbf{U} - \mathbf{U}_k)^T \, \bar{\mathbf{H}}_{\mathbf{U}}^{\mathcal{I}_{\mathbf{U}}^k} \text{vec}\,(\mathbf{U} - \mathbf{U}_k) \end{aligned} \tag{4.61}$$

and

$$\begin{aligned} g(\mathbf{V}|\mathbf{U}_{k+1}, \mathbf{V}_k) &= f(\mathbf{U}_{k+1}, \mathbf{V}_k) + \text{tr}\{(\mathbf{V} - \mathbf{V}_k)^T \nabla_{\mathbf{V}} f(\mathbf{U}_{k+1}, \mathbf{V}_k)\} + \\ &\quad \frac{1}{2\alpha_{\mathbf{V}}^k} \text{vec}\,(\mathbf{V} - \mathbf{V}_k)^T \, \bar{\mathbf{H}}_{\mathbf{V}}^{\mathcal{I}_{\mathbf{V}}^k} \text{vec}\,(\mathbf{V} - \mathbf{V}_k)\,, \end{aligned} \tag{4.62}$$

where $\alpha_{\mathbf{U}}^k$ and $\alpha_{\mathbf{V}}^k$ denote step size parameters. Hence, **U** and **V** are updated by solving the following constrained minimization problems,

$$\mathbf{U}_{k+1} = \underset{\mathbf{U} \geq \mathbf{0}}{\text{argmin}} \; l(\mathbf{U}|\mathbf{U}_k, \mathbf{V}_k) \tag{4.63}$$

$$\mathbf{V}_{k+1} = \underset{\mathbf{V} \geq \mathbf{0}}{\text{argmin}} \; g(\mathbf{V}|\mathbf{U}_{k+1}, \mathbf{V}_k) \tag{4.64}$$

giving rise to feasible updates in the form

$$\text{vec}(\mathbf{U}_{k+1}(\alpha_{\mathbf{U}}^k)) = [\text{vec}(\mathbf{U}_k) - \alpha_{\mathbf{U}}^k \left(\bar{\mathbf{H}}_{\mathbf{U}}^{\mathcal{I}_{\mathbf{U}}^k}\right)^{-1} \text{vec}(\nabla_{\mathbf{U}} f(\mathbf{U}_k, \mathbf{V}_k))]_+ \tag{4.65}$$

$$\text{vec}(\mathbf{V}_{k+1}(\alpha_{\mathbf{V}}^k)) = [\text{vec}(\mathbf{V}_k) - \alpha_{\mathbf{V}}^k \left(\bar{\mathbf{H}}_{\mathbf{V}}^{\mathcal{I}_{\mathbf{V}}^k}\right)^{-1} \text{vec}(\nabla_{\mathbf{V}} f(\mathbf{U}_{k+1}, \mathbf{V}_k))]_+, \tag{4.66}$$

where $[x]_+ = \text{max}(x, 0)$. The step size parameters $\alpha_{\mathbf{U}}^k$ and $\alpha_{\mathbf{V}}^k$ are calculated based on the Armijo rule on the projection arc, [15], with the goal of achieving sufficient decrease of the initial cost function per iteration. Concretely, $\alpha_{\mathbf{U}}^k$ is set to $\alpha_{\mathbf{U}}^k = \beta_{\mathbf{U}}^{m_k}$ with $\beta_{\mathbf{U}} \in (0, 1)$ and m_k

is the first nonnegative integer such that

$$f(\mathbf{U}_k) - f(\mathbf{U}_{k+1}(\alpha_{\mathbf{U}}^k)) \geq \sigma \Bigg\{ \alpha_{\mathbf{U}}^k \sum_{i \notin \{\mathcal{I}_{\mathbf{u}_1}^k \cup \mathcal{I}_{\mathbf{u}_2}^k \cup \cdots \cup \mathcal{I}_{\mathbf{u}_m}^k\}} \frac{\partial f(\mathbf{U}_k, \mathbf{V}_k)}{\partial \mathrm{vec}(\mathbf{U})_i} \times$$
$$\left(\left(\bar{\mathbf{H}}_{\mathbf{U}}^{\mathcal{I}_{\mathbf{U}}^k} \right)^{-1} \mathrm{vec}(\nabla_{\mathbf{U}} f(\mathbf{U}_k, \mathbf{V}_k)) \right)_i + \sum_{i \in \{\mathcal{I}_{\mathbf{u}_1}^k \cup \mathcal{I}_{\mathbf{u}_2}^k \cup \cdots \cup \mathcal{I}_{\mathbf{u}_m}^k\}} \frac{\partial f(\mathbf{U}_k, \mathbf{V}_k)}{\partial \mathrm{vec}(\mathbf{U})_i} \times \mathrm{vec}(\mathbf{U}_k - \mathbf{U}_k(\alpha_{\mathbf{U}}^k))_i \Bigg\}. \tag{4.67}$$

where σ is a constant scalar. The same process described above for selecting $\alpha_{\mathbf{U}}^k$ and hence updating $\mathbf{U}$ is subsequently adopted for $\alpha_{\mathbf{V}}^k$ and $\mathbf{V}$. The resulting alternating projected Newton-type algorithm for low-rank NMF is given in Algorithm 4.3.

Remark 4.4. *The adopted Armijo-rule on the projection arc provides us guarantees regarding the monotonic decrease of the initial cost function per iteration. It should be noted that, contrary to the projected Newton NMF method of [73], in our case the diagonal matrices adopted are always positive definite and hence invertible offering stability to the derived algorithm. Finally, since the approximate Hessian matrices used are partially diagonal, efficient implementations can be adopted for reducing the computational cost.*

Algorithm 4.3: AIRLS nonnegative matrix factorization (AIRLS-NMF) algorithm

```
Input: $\mathbf{Y}, \lambda, \beta_{\mathbf{U}}, \beta_{\mathbf{V}}, \sigma$
Initialize: $k = 0, \mathbf{U}_0, \mathbf{V}_0, \mathbf{D}_{(\mathbf{U}_0, \mathbf{V}_0)}$
repeat
    Estimate the set of active constraints $\mathcal{I}_{\mathbf{U}}^k$
    $m_k = 0,\ \alpha_{\mathbf{U}}^k = 1$
    while (4.67) is not satisfied do
        $m_k = m_k + 1$, $\alpha_{\mathbf{U}}^k = \beta_{\mathbf{U}}^{m_k}$
    end
    $\mathrm{vec}(\mathbf{U}_{k+1}) = [\mathrm{vec}(\mathbf{U}_k) - \alpha_{\mathbf{U}}^k \left(\bar{\mathbf{H}}_{\mathbf{U}}^{\mathcal{I}_{\mathbf{U}}^k}\right)^{-1} \mathrm{vec}(\nabla_{\mathbf{U}} f(\mathbf{U}_k, \mathbf{V}_k))]_+$
    Estimate the set of active constraints $\mathcal{I}_{\mathbf{V}}^k$
    $m_k = 0,\ \alpha_{\mathbf{V}}^k = 1$
    while (4.67) is not satisfied do
        $m_k = m_k + 1$, $\alpha_{\mathbf{V}}^k = \beta_{\mathbf{V}}^{m_k}$
    end
    $\mathrm{vec}(\mathbf{V}_{k+1}) = [\mathrm{vec}(\hat{\mathbf{V}}_k) - \alpha_{\mathbf{V}}^k \left(\bar{\mathbf{H}}_{\mathbf{V}}^{\mathcal{I}_{\mathbf{V}}^k}\right)^{-1} \mathrm{vec}(\nabla_{\mathbf{V}} f(\mathbf{U}_{k+1}, \mathbf{V}_k))]_+$
    $k = k + 1$
until convergence
Output: $\hat{\mathbf{U}} = \mathbf{U}_{k+1}, \hat{\mathbf{V}} = \mathbf{V}_{k+1}$
```

4.4.2 Low-rank and sparse NMF

Low-rank and sparse NMF under the proposed LRMF framework presented above, arises by a simple extension of the low-rank NMF formulation given above, i.e.,

$$\min_{\mathbf{U}\geq 0,\mathbf{V}\geq 0} \|\mathbf{Y} - \mathbf{U}\mathbf{V}^T\|_F^2 + \lambda \sum_{i=1}^{d} \left(\|\boldsymbol{u}_i\|_2^2 + \|\boldsymbol{v}_i\|_2^2 + \eta^2 \right)^{\frac{p}{2}} + \lambda_1 \|\mathbf{V}\|_1. \tag{4.68}$$

where the ℓ_1 norm of the coefficient's matrix $\mathbf{V}$ has been included in the optimization problem. Note that although (4.68) seems to be a plain extension of the NMF problem (4.58), things are becoming much more trickier now. This is due to the non-smoothness of the ℓ_1 norm, which necessitates a different optimization strategy as detailed below.

More specifically, the proposed minimization algorithm utilizes the same *smooth* approximation of the low-rank promoting term of the cost function (4.68) incorporating an additional term $g(\mathbf{V}) = \lambda_1 \|\mathbf{V}\|_1$ which is the nonsmooth and separable part of the minimized cost function. Considering again the matrices $\mathbf{U}$ and $\mathbf{V}$ as blocks in our problem, each one of them may be updated as follows:

$$\mathbf{U}_{k+1} = \underset{\mathbf{U}\geq 0}{\text{argmin}} \ \text{tr}\{(\mathbf{U} - \mathbf{U}_k) \nabla_{\mathbf{U}} f(\mathbf{U}_k, \mathbf{V}_k)\} + \frac{1}{2\alpha^k} \text{vec}(\mathbf{U} - \mathbf{U}_k)^T \bar{\mathbf{H}}_{\mathbf{U}_k} \text{vec}(\mathbf{U} - \mathbf{U}_k) \tag{4.69}$$

and

$$\begin{aligned}\mathbf{V}_{k+1} = &\underset{\mathbf{V}\geq 0}{\text{argmin}} \ \text{tr}\{(\mathbf{V} - \mathbf{V}_k)^T \nabla_{\mathbf{V}} f(\mathbf{U}_{k+1}, \mathbf{V}_k)\} + \frac{1}{2\alpha^k} \text{vec}(\mathbf{V} - \mathbf{V}_k)^T \bar{\mathbf{H}}_{\mathbf{V}_k} \text{vec}(\mathbf{V} - \mathbf{V}_k) \\ &+ g(\mathbf{V})\end{aligned} \tag{4.70}$$

where $\bar{\mathbf{H}}_{\mathbf{U}}^k$ is a block diagonal matrix as defined in (4.18). Note that the cost functions involved in (4.69) and (4.70) are second-order approximations of $f(\mathbf{U}, \mathbf{V}_k)$ and $f(\mathbf{U}_{k+1},]\mathbf{V})$, around $(\mathbf{U}_k, \mathbf{V}_k)$ and $(\mathbf{U}_{k+1}, \mathbf{V}_k)$, respectively. The minimization of (4.70) as such gives rise to a scaled proximal operator, [92], in the form

$$\mathbf{V}_{k+1} = \text{prox}_{\|\cdot\|_1}^{\bar{\mathbf{H}}_{\mathbf{V}_k}} \left(\mathbf{V}_k - (\bar{\mathbf{H}}_{\mathbf{V}_k})^{-1} \nabla_{\mathbf{V}} f(\mathbf{U}_k, \mathbf{V}_k) \right). \tag{4.71}$$

The computation of (4.71) has given way to disparate approximate or iterative schemes (proximal Newton methods), [92], since there exists no closed form solution for non diagonal[6] $\bar{\mathbf{H}}_{\mathbf{V}_k}$ as in our case. Herein, we propose to apply an incremental strategy, [17], for approximately solving (4.70). More specifically, a gradient step is first applied on the smooth part of (4.70). The outcome of the gradient step is then provided as input to the proximal operator of the nonsmooth term, i.e., the ℓ_1 norm, whose output is finally projected to the feasible set. Regarding the update of $\mathbf{U}$, it is a much simpler task due to the absence of nonsmooth terms. In general, for the parameter α^k in Eqs. (4.70), (4.69) it

[6] Note that for $\bar{\mathbf{H}}_{\mathbf{V}_k}$ diagonal, the scaled proximal operator reduces to the known proximity operator of the ℓ_1 norm, i.e., the soft-thresholding operator.

holds $\alpha^k \in (0, 1]$. Overall, **U** and **V** are computed by using the following expressions[7],

$$\mathbf{U}_{k+1} = \mathcal{P}_{\mathcal{R}_+^{m\times d}} \left(\left(\hat{\mathbf{V}}_k^T \hat{\mathbf{V}}_k + \mathbf{D}_{(\hat{\mathbf{U}}_k, \hat{\mathbf{V}}_k)} \right)^{-1} \hat{\mathbf{V}}_k \mathbf{Y} \right) \tag{4.72}$$

$$\mathbf{V}_{k+1} = \mathcal{P}_{\mathcal{R}_+^{n\times d}} \left(\mathsf{SHR}_{\lambda_1} \left(\left(\hat{\mathbf{U}}_k^T \mathbf{U}_k + \mathbf{D}_{(\hat{\mathbf{U}}_{k+1}, \hat{\mathbf{V}}_k)} \right)^{-1} \hat{\mathbf{U}}_{k+1} \mathbf{Y} \right) \right) \tag{4.73}$$

where $\mathsf{SHR}_{\lambda_1}(\mathbf{X})$ is the soft-thresholding operator defined in (3.11)[8]. Since the difference matrix of $\bar{\mathbf{H}}_{\mathbf{V}_k}$ and $\bar{\mathbf{H}}_{\mathbf{U}_k}$ from their respective true Hessians is positive semidefinite, as shown above, the quadratic approximate functions are upper bounds of the original cost function. That said, the above-described scheme also resembles the block successive upper bound minimization framework of [79]. As stated earlier, and since we are dealing with a constrained minimization problem, the updates for **U** and **V** are projected to the feasible set thus accounting for the nonnegativity constraint. In addition, problem (4.70) is solved inexactly by the incremental strategy described earlier. In view of these, for ensuring that the cost function decreases at each step, an extrapolation step is next followed and the final estimates of **U** and **V** at the kth iteration are obtained as

$$\hat{\mathbf{U}}_{k+1} = \hat{\mathbf{U}}_k + \beta_{\mathbf{U}_k} \left(\mathbf{U}_{k+1} - \hat{\mathbf{U}}_k \right), \tag{4.74}$$

$$\hat{\mathbf{V}}_{k+1} = \hat{\mathbf{V}}_k + \beta_{\mathbf{V}_k} \left(\mathbf{V}_{k+1} - \hat{\mathbf{V}}_k \right). \tag{4.75}$$

Note that $\beta_{\mathbf{V}_k}$ and $\beta_{\mathbf{U}_k}$ are adjusted dynamically so that the cost function's sufficient decrease is guaranteed at each step. Along this line, numerous schemes, known as line search methods have come into play, e.g., backtracking, [15]. Those schemes affect the convergence and rate of convergence of the algorithms to stationary points. The resulting algorithm is given in Algorithm 4.4.

Remark 4.5. *The proposed AIRLS, AIRLS-MC, AIRLS-NMF and SpAIRLS-NMF algorithms annihilate jointly columns of the matrices* **U** *and* **V***, as a result of the column sparsity imposing nature of the introduced low-rank promoting term. This key feature of the proposed algorithms allows us to incorporate a mechanism which prunes the columns that become zero as the algorithms evolve, thus reducing the (column) dimension of the matrices. By doing so, the per iteration computational complexity of the algorithms is gradually reduced, and this reduction affects the total computational time required by the algorithms to converge, as is also highlighted in Section 4.5.*

4.4.2.1 Unsupervised HU as a low-rank and sparse NMF problem

In great many real cases, a first critical and indispensable step towards performing unmixing is to uncover the true number of endmembers that exist in a given hyperspectral scene. This challenging task (also known as *rank estimation* or *model order selection*), can be

[7] $\hat{\mathbf{U}}$ and $\hat{\mathbf{V}}$ used in these expressions are subsequently defined in (4.74) and (4.75) respectively.

[8] Note that in this case, the used notation implies that all elements of **X** are equally thresholded by λ_1.

Algorithm 4.4: Sparse AIRLS-NMF (SpAIRLS-NMF) algorithm

```
Input $\mathbf{Y}, \lambda > 0, \lambda_1 > 0$
Initialize $k = 0, \hat{\mathbf{V}}_0, \hat{\mathbf{U}}_0, \beta_{\mathbf{U}_0}, \beta_{\mathbf{V}_0}$
repeat
    $\mathbf{U}_{k+1} = \mathcal{P}_{\mathcal{R}_+^{m \times d}}\left(\left(\hat{\mathbf{U}}_k^T\hat{\mathbf{U}}_k + \hat{\mathbf{D}}_{(\hat{\mathbf{U}}_k,\hat{\mathbf{V}}_k)}\right)^{-1}\hat{\mathbf{V}}_k\mathbf{Y}\right)$
    $\hat{\mathbf{U}}_{k+1} = \hat{\mathbf{U}}_k + \beta_{\mathbf{U}_k}\left(\mathbf{U}_{k+1} - \hat{\mathbf{U}}_k\right)$
    $\mathbf{V}_{k+1} = \mathcal{P}_{\mathcal{R}_+^{n \times d}}\left(\mathsf{SHR}_{\lambda_1}\left(\left(\hat{\mathbf{U}}_{k+1}^T\hat{\mathbf{U}}_{k+1} + \mathbf{D}_{(\hat{\mathbf{U}}_{k+1},\hat{\mathbf{V}}_k)}\right)^{-1}\hat{\mathbf{U}}_{k+1}\mathbf{Y}\right)\right)$
    $\hat{\mathbf{V}}_{k+1} = \hat{\mathbf{V}}_k + \beta_{\mathbf{V}_k}\left(\mathbf{V}_{k+1} - \hat{\mathbf{V}}_k\right)$
    Update $\beta_{\mathbf{V}}^k, \beta_{\mathbf{U}}^k$
until convergence
Output: $\hat{\mathbf{U}} = \hat{\mathbf{U}}_{k+1}, \hat{\mathbf{V}} = \hat{\mathbf{V}}_{k+1}$
```

quite daunting in terms of the required computational burden. Several works have come into play in HU literature for attacking this problem which can be classified into two main categories: a) information theoretic criteria based approaches and b) eigenvalue thresholding methods. As far as the first class is concerned, various approaches have been proposed differing in the criterion used for penalizing an initially overestimated number of endmembers, e.g., Akaike's Information Criterion (AIC), Minimum Description Length (MDL), Bayesian Information Criterion (BIC), [54]. On the other hand, methods that belong to the second class include PCA based methods, Neyman-Peyrson detection theory based methods etc., [20].

The estimate of the number of endmembers by the above-mentioned algorithms is provided - at a second phase - as input to unsupervised unmixing algorithms, whose goal is to extract the endmembers' spectral signatures along with the abundance fractions of the pixels. A vast amount of works have been published in the literature dealing with unsupervised hyperspectral unmixing. As is the case with the vast majority of HU methods, most of the unsupervised HU methods hinge on the linear mixing model (LMM), which has been proven to be a reliable approximation, although it neglects nonlinear effects met in real situations. In the framework of the LMM, there exist both geometrical, [110], and statistical, [97], matrix factorization based approaches for performing unsupervised unmixing. Among the latter, nonnegative matrix factorization (NMF) based techniques have exhibited a robust behavior offering promising results.

Along these lines, the proposed low-rank and sparse NMF framework presented above can be utilized so as to *simultaneously* a) determine the number of endmembers, b) extract the endmembers' spectral signatures and c) estimate the abundance values of the pixels. The sophisticated low-rank promoting term penalizes both endmembers' and abundance matrices, i.e., **U** and **V** respectively, by enforcing joint sparsity on their columns. This way, we go one step beyond just revealing the rank, since we further encourage estimation of the true bases of the column spaces of these matrices. At the same time, sparsity is favored on the abundance matrices **V**, as it is physically meaningful. All in all, endmembers' number estimation and unmixing is yielded jointly by the introduced low-rank and sparse NMF approach. To the best of our knowledge, this is the first work that encapsulates those two problems simultaneously in a single task.

4.5 Experiments

In this section, simulated and real data experiments are provided for illustrating the key features of the proposed AIRLS, AIRLS-MC, AIRLS-NMF and SpAIRLS-NMF algorithms[9]. For comparison purposes, an alternating regularized least squares (noted here as ALS) algorithm corresponding to the full-observation version of the matrix completion softImpute-ALS algorithm proposed in [77] is utilized in the denoising type problems. In matrix completion experiments the softImpute-ALS algorithm, [77], and the iterative reweighted nuclear norm (IRNN) algorithm of [95] are employed. It should be noted that IRNN goes beyond the traditional nuclear norm minimization by adopting various sparsity imposing priors for the vector of singular values. This scheme gives rise to weighted nonconvex analogues of the traditional nuclear norm. In the sequel, we restrict our attention to IRNN which arises by applying the $\ell_{1,2}$ quasinorm on the vector of singular values. Note that IRNN, unlike AIRLS and softImpute-ALS, is not an MF based approach and thus involves computationally demanding SVD operations at each iteration. Moreover, the ARD-NMF algorithm, [139], which is based on the same philosophy to the one of our approach, yet through the lens of a Bayesian framework, is incorporated in the NMF type experiments. SpAIRLS-NMF is herein focused on the novel formulation of the unsupervised HU problem described in Section 4.4.2.1 and thus the vertex component analysis (VCA) endmembers' extraction algorithm, [110] is utilized for comparison purposes. It should be noted that for all the proposed algorithms the *column pruning mechanism is applied*. As a result, the per iteration complexity reduces during the execution of the algorithms. All experiments were conducted on an Intel Core i7-4790 CPU 3.60GHz x 8 CPU with 16GB RAM.

4.5.1 Simulated data experiments

Herein we endeavor to highlight the benefits of the proposed algorithms on simulated data. To this end, the algorithms are tested on two different experiments, i.e., a) for checking the performance of AIRLS and AIRLS-NMF in the presence of noise and b) for assessing the capacity of AIRLS-MC in dealing with different percentages of missing data. Moreover, SpAIRLS-NMF's performance is tested on a simulated hyperspectral dataset focusing on the unsupervised unmixing task, as well as in the competence of the algorithm in recovering the true number of endmembers.

[9] In all experiments provided next the norm parameter p is set to 1.

Table 5: Results obtained by ALS and AIRLS on the simulated denoising experiment.

SNR	10						20					
rank	5			10			5			10		
Algorithm	# Iter	time(s)	NRE	# Iter	time(s)	NRE	# Iter	time(s)	NRE	# Iter	time(s)	NRE
ALS	15	0,2774	0,1079	15	0,2853	0,1152	40,31	0,7739	0,0235	40,38	0,7666	0,0294
AIRLS	43,37	0,3949	0,0448	24,37	0,2426	0,0635	15,41	0,1571	0,0142	35,68	0,3421	0,02

Table 6: Results obtained by ALS and AIRLS on the simulated denoising experiment.

4.5.1.1 Performance of AIRLS and AIRLS-NMF in the presence of noise

In order to validate the performance of AIRLS and AIRLS-NMF in the presence of noise two different experimental settings are used. In both settings, a matrix $\mathbf{X}_0 \in \mathcal{R}^{m\times n}$ with $m = 500$, $n = 500$ and varying rank $r \in \{5, 10\}$ is randomly generated. Concretely, matrix $\mathbf{X}_0$ is produced by the product of two matrices, i.e., $\mathbf{U}_0 \in \mathcal{R}^{m\times r}$ and $\mathbf{V}_0 \in \mathcal{R}^{n\times r}$ with either a) zero-mean Gaussian entries of $\sigma = 1$ or b) uniformly distributed nonnegative values in the range 0 to 1. The latter is used for testing the NMF algorithms. In both cases additive Gaussian i.i.d. noise of different SNR $\in \{10, 20\}$ corrupts $\mathbf{X}_0$, thus resulting to the data matrix $\mathbf{Y}$ which is then provided as input to the tested algorithms. For the case of a), AIRLS is compared to the MMMF algorithm while in b), the ARD-NMF algorithm takes also part in the respective experiments. As a quantitative metric we utilize the normalized reconstruction error defined as NRE $= \frac{\|\mathbf{X}_0-\hat{\mathbf{U}}\hat{\mathbf{V}}^T\|_F}{\|\mathbf{X}_0\|_F}$. Since we emphasize in the recovery performance of the algorithms, the low-rank promoting parameter λ of the algorithms is selected from a set of values {0.1,1,5,10,50,80,100,200} via fine tuning in terms of the lowest achieved NRE. The algorithms stop when either the relative decrease of the reconstructed data between two successive iterations, i.e., $\frac{\|\hat{\mathbf{U}}_k\hat{\mathbf{V}}_k^T-\hat{\mathbf{U}}_{k+1}\hat{\mathbf{V}}_{k+1}^T\|_F}{\|\hat{\mathbf{U}}_k\hat{\mathbf{V}}_k^T\|_F}$ becomes less than 10^{-4} or 500 iterations are executed. The algorithms run for 100 instances of each experiment and the mean values of the various quantities (elapsed time, NRE, iterations executed and estimated rank) are provided in Tables 6 and 7.

In Table 6, the results of AIRLS and ALS are given. Therein, it is shown that AIRLS offers better estimation performance than ALS in all experiments. Interestingly, in most cases, this happens in less time than that spent by ALS, although AIRLS in some instances required more iterations. This favorable characteristic of AIRLS is due to its *column pruning capability, which reduces significantly the average time per iteration*. In the case of the NMF problem, it can be observed by Table 7 that the AIRLS-NMF achieved lower NRE that of ARD-NMF for the different levels of noise and rank of the sought matrices. Notably, AIRLS-NMF exhibited robustness in recovering the true rank in both cases examined, i.e., $r \in \{5, 10\}$, contrary to ARD-NMF which failed to estimate the true rank especially for $r = 10$.

Table 7: Results obtained by ARD-NMF and AIRLS on the simulated NMF experiment.

SNR	10				20			
rank	5		10		5		10	
Algorithm	est. rank	NRE	est. rank	NRE	est. rank	NRE	est. rank	NRE
ARD-NMF	4,36	0,0778	100	0,1023	4,66	0,0825	100	0,1008
AIRLS-NMF	5,14	0,048	10,25	0,0706	6,52	0,0181	10,23	0,0291

Table 8: Results of AIRLS-MC, softImpute-ALS and IRNN on the simulated matrix completion experiment.

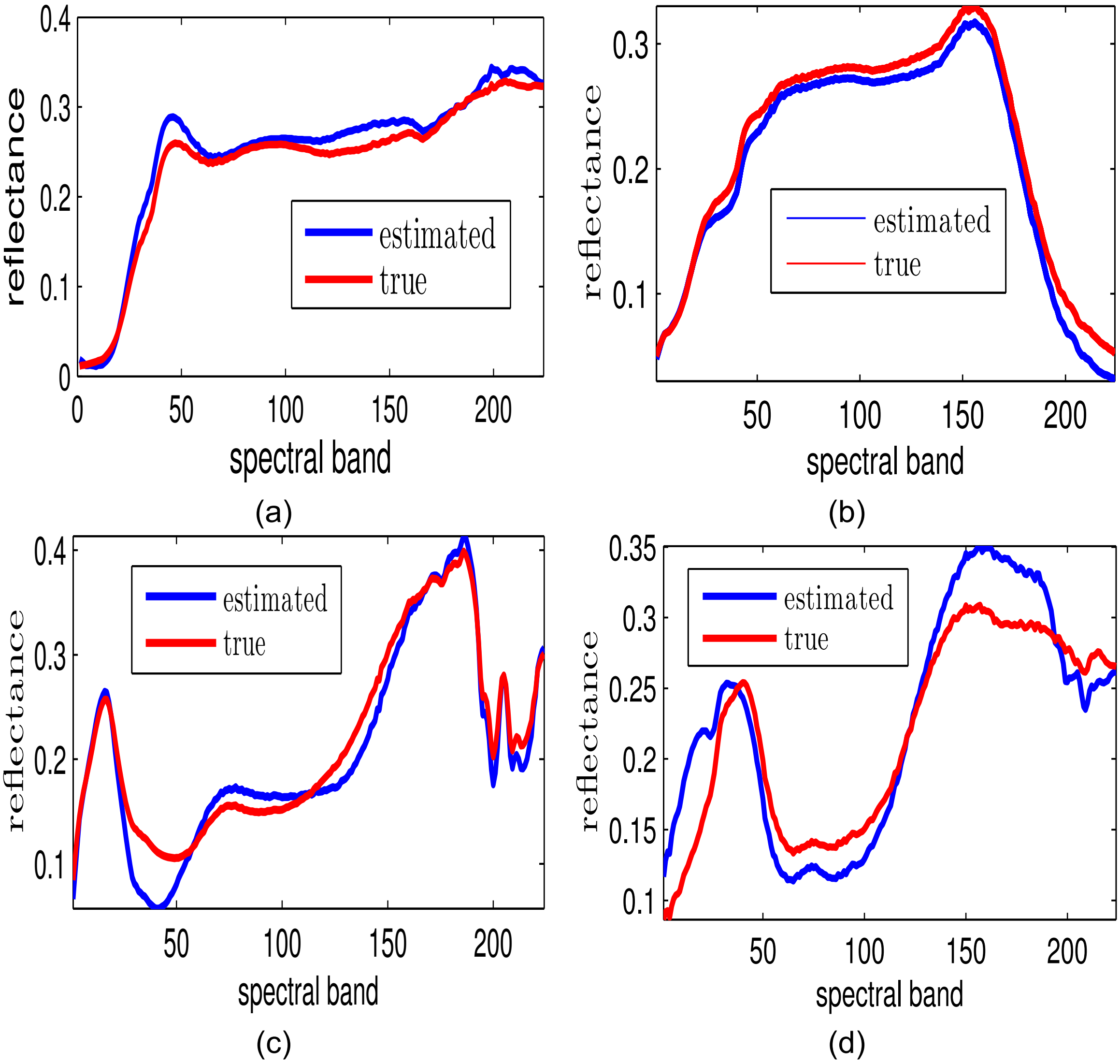
FR	0.4			0.6		
Algorithm	# Iter	time(s)	NRE	# Iter	time(s)	NRE
softImpute-ALS	244	59.5	0,1886	204	37.8	0,543
AIRLS-MC	349	42.6	0,069	457	27	0,277
IRNN	314	113.3	0,059	580	216.9	0,161

4.5.1.2 Performance of AIRLS-MC for different percentages of missing data

To evaluate the performance of AIRLS-MC in different scenarios, we classify the experimental settings of this subsection according to the degrees of freedom ratio (FR), [107], defined as FR $= r(2n - r)/\text{card}(\Omega)$, where r is the rank of **X**. Recovery becomes harsher as FR is close to 1, whereas easier problems arise when it takes values close to 0. AIRLS-MC is compared to softImpute-ALS and IRNN for FR equal to 0.4 and 0.6. In both cases a low-rank matrix $\mathbf{X}_0 \in \mathcal{R}^{m\times n}$ with $m = 1000$, $n = 1000$ and rank $r = 20$ is generated following the same setting as in the case of the denoising experiment described above. The NRE is used as the performance metric. For all algorithms, the parameter λ which is related to low-rank imposition, is fine tuned and the initial rank of the MF based methods is set to 100. The algorithms run for 20 instances of each experiment and the mean values of iterations, NRE and time to converge are given in Table 8. Moreover, the same stopping criteria mentioned previously are utilized. As is shown in Table 8, AIRLS-MC offers significantly higher accuracy than softImpute-ALS in less time in both experiments. On the other hand, IRNN outperforms both MF based methods in terms of NRE, at the cost of a much higher runtime. This shortcoming of IRNN is due to the computationally demanding SVDs executed at each iteration. Interestingly, AIRLS-MC converges in less time than softImpute-ALS, although it requires more iterations to converge. Actually, this happens due to the fact that AIRLS-MC estimates the true rank of the matrix after a few iterations. That is, the column pruning mechanism mentioned above reduces gradually its computational complexity.

4.5.1.3 Performance of SpAIRLS-NMF on simulated hyperspectral data

In this experiment we aim at corroborating the competence of SpAIRLS-NMF in uncovering the true number of endmembers along with estimating the spectral signatures of the endmembers. To this end, we generate a 500×4 abundance matrix whose elements fol-



(a)

(b)

(c)

(d)

Figure 28: Endmembers' spectral signatures obtained by SpAIRLS-NMF on the simulated data experiment.

low a uniform distribution in the interval [0,1]. This matrix is then sparsified by randomly keeping only 30% of its elements. From the USGS spectral library, we select randomly 4 endmembers' spectral signatures measured at $m = 224$ distinct spectral bands. Then, we linearly produce $n = 500$ simulated pixels' spectral signatures under the LMM framework. The pixel spectral signatures are then contaminated with additive i.i.d. Gaussian noise with standard deviation $\sigma = 10^{-3}$.

Since the main premise of our approach is the development of a blind unmixing method that exhibits robustness in the absence of knowledge of the true number of endmembers, we initialize the proposed algorithm with an overestimate $d = 10$ of the actual number of endmembers. Both endmembers' and abundance matrices are randomly initialized according to the uniform distribution. Interestingly, the proposed algorithm converges to abundance and endmembers' matrices consisting of 4 nonzero columns, which is the same as the actual number of endmembers that produced the data. Moreover, it can be easily observed from Fig. 28, that the estimated endmembers' spectral signatures present high degree of similarity to the real ones. Hence, we can conclude that the proposed algorithm is, in principle, capable of carrying out the challenging task of simultaneously estimating the number of endmembers and performing unsupervised hyperspectral unmixing in a linear mixing setting.

4.5.2 Real data experiments

Next, the effectiveness of all the proposed algorithms is corroborated in four different real data applications. More specifically, AIRLS is applied on a real HSI denoising application, AIRLS-MC is tested on two different recommender systems' (movie-lens 100K and 1M) datasets, while the performance of AIRLS-NMF is assessed through a music signal decomposition experiment. Lastly, SpAIRLS-NMF is evaluated on a real HSI unsupervised spectral unmixing experiment.

4.5.2.1 Hyperspectral image denoising

In this experiment we utilize the Washington DC Mall AVIRIS HSI captured at $m = 210$ contiguous spectral bands in the 0.4 to 2.4 μm region of the visible and infrared spectrum. The HSI consists of $n = 22500$ (150×150) pixels. As is widely known, [67], hyperspectral data are highly coherent both in the spectral and the spatial domains. Therefore, by organizing the tested image in a matrix, whereby each column corresponds to the spectral bands and each row to the pixels, it turns out that this matrix can be well approximated by a low-rank one. This fact motivates us to exploit the low-rank structure of the HSI under study for efficiently denoising a highly corrupted version thereof by Gaussian i.i.d. noise of SNR $= 6dB$.

In Fig. 29, false RGB images of the recovered HSIs by the proposed AIRLS algorithm and ALS are provided. In both algorithms, the number of columns of the initial factors $\mathbf{U}_0$ and $\mathbf{V}_0$ is overstated to $d = 100$ and the algorithms terminate when the relative decrease of the reconstructed HSI between two successive iterations reaches a value less than 10^{-4}. Moreover, their low-rank promoting parameter λ is selected so as to lead to solution matrices $\hat{\mathbf{U}}$ and $\hat{\mathbf{V}}$ of the same rank $r = 4$. As it can be noticed in Fig. 29, AIRLS reconstructs the HSI in a significantly improved accuracy as compared to ALS. This can be easily verified both by visually inspecting Figs. 29a-29d and quantitatively in terms of the estimated NRE (Fig. 29e). Notably, AIRLS converges in less iterations than those required by ALS (Fig. 29e), while at the same time less time per iteration is consumed, on average. The latter is achieved by virtue of the column pruning mechanism of AIRLS, which gradually reduces the size of matrix factors **U** and **V** from $m \times 100$ and $n \times 100$ to $m \times 4$ and $n \times 4$, respectively. This way, after only a few initial iterations, when the rank starts to decrease, the per iteration time complexity of AIRLS becomes much smaller than that required in its early iterations, as well as the one of ALS.

4.5.2.2 MC on Movielens 100K and 1M datasets

Herein, we focus on testing the performance of AIRLS-MC algorithm on a popular collaborative filtering application, i.e. a movie recommender system. To this end, we utilize two well-studied in literature large datasets: the Movielens 100K and the Movielens 1M datasets. Both datasets contain ratings by users collected over various periods of time, with integer values ranging from 1-5. Since most of the entries are missing, matrix completion algorithms can be utilized for predicting them. By assuming that there exists a high degree of correlation amongst the rating of different users, a low-rank structure is a rational choice for these datasets. For the case of the 100K dataset the “ub.base”[10] file which contains $\approx 90\%$ of the total ratings was splitted into two disjoint sets, i.e., a training set (consisting of $\approx 65\%$ of the total per user ratings) and a validation set ($\approx 25\%$). The “ub.test” file which contains $\approx 10\%$ of the ratings was utilized as the test set. For the case of the 1M Movielens dataset, the “ratings.dat” file was splitted into 3 disjoint sets, that is, a training set consisting of $\approx 50\%$ of the total ratings per user, a validation set $\approx 25\%$ and a test set ($\approx 25\%$). Note that the 100K dataset contains 100000 ratings of 943 users on 1682 movies with each user having rated at least 20 movies. That said, we need to address a quite challenging matrix completion problem, since 93% of the elements are missing. The situation is even harsher for the 1M dataset, which includes 1 million ratings from 6040 users on 3900 movies and 96% missing data. Finally, the normalized mean absolute value error (NMAE) defined as NMAE $= \frac{\sum_{(i,j)\in\Omega} |[\mathbf{U}\mathbf{V}^T]_{ij} - [\mathbf{Y}]_{ij}|}{4\mathrm{card}(\Omega)}$ is used as a performance metric.

First, we aim at illustrating the behavior of the proposed AIRLS-MC algorithm when it comes to the estimation performance and the speed of convergence. In this regard, for the case of the 100K dataset, the state-of-the-art IRNN and softImpute-ALS algorithms are utilized for comparison purposes. The low-rank promoting parameter λ of all competing algorithms is selected according to two different scenarios: A) we choose λ that achieves the minimum NMAE on the validation set after convergence and B) we select λ

[10] Movielens 100K and 1M datasets can be downloaded from https://grouplens.org/datasets/movielens/.

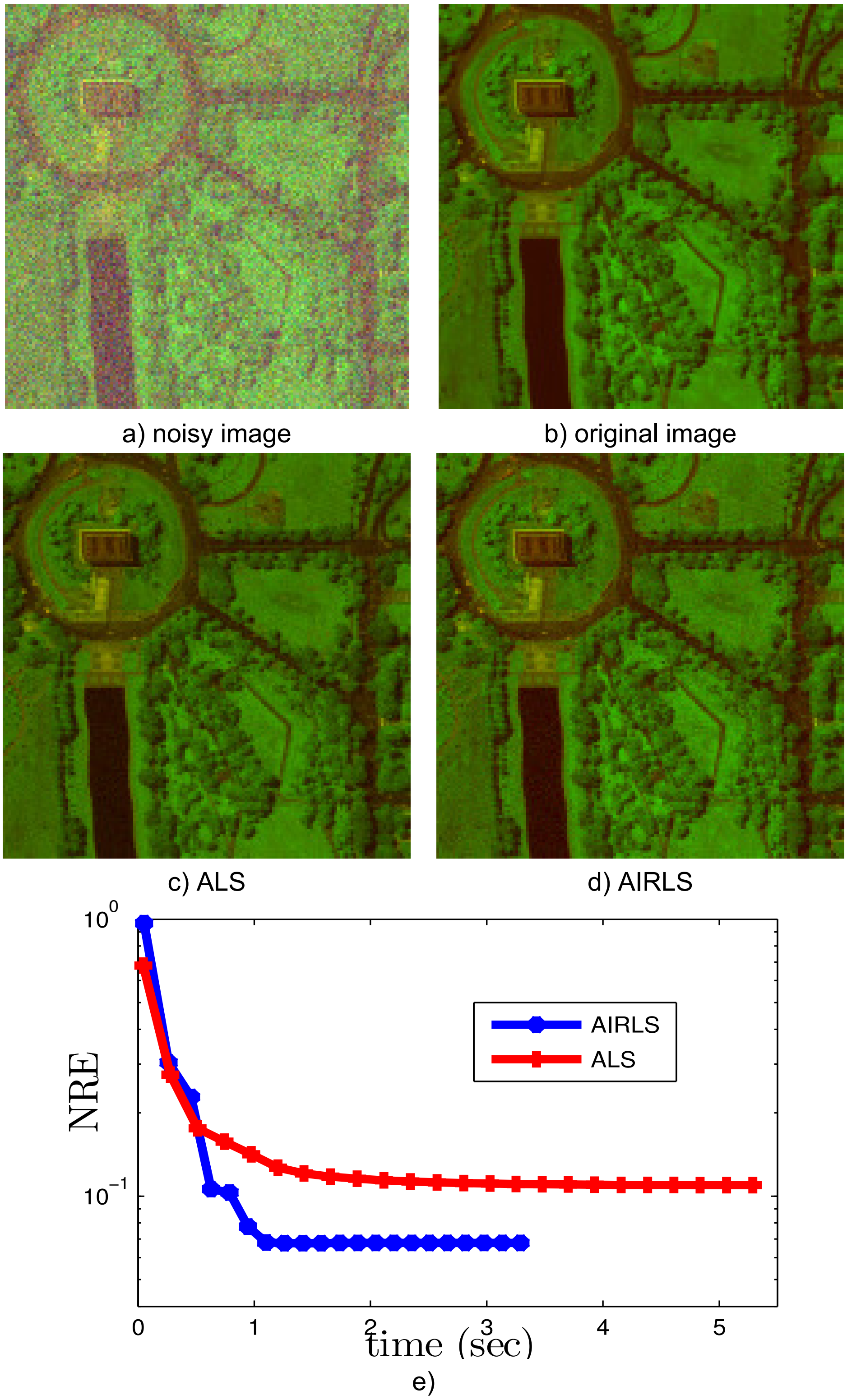


Figure 29: Evaluation of AIRLS and ALS on the Washington DC AVIRIS dataset.

Table 9: Results obtained by AIRLS-MC and softImpute-ALS on Movielens 100K dataset.

			# Iter	msec/iter	total time (sec)	NMAE
scenario	A	softImpute-ALS	247	300,2	74,3	0,2362
		IRNN	500	598,5	299,2	0,2036
		AIRLS-MC	591	21,7	12,82	0,2005
	B	softImpute-ALS	156	197,6	30.8	0,2968
		IRNN	500	740	370,18	0,2029
		AIRLS-MC	969	28,5	27,6	0,2010

so that the estimated matrices by both the tested algorithms are of the same rank, equal to 10. It should be noted that the same stopping criterion used in the previous experiment is adopted also here. As it can be seen in Fig. 30 and Table 9, the softImpute-ALS algorithm requires in general less iterations to converge than both AIRLS-MC and IRNN. However, the average per-iteration time complexity of AIRLS-MC is significantly less compared to its rivals. As is mentioned above, this is attributed to the column pruning scheme which decreases to a large degree the computational burden of the algorithm. This favorable property, results to a faster convergence of AIRLS-MC in both scenarios A and B as compared to both softImpute-ALS and IRNN, in terms of time. Among the three algorithms tested, IRNN is clearly the most demanding one in terms of average per-iteration time complexity as it can be observed from Fig. 30. As mentioned above, this is ascribed to the fact that IRNN entails “expensive” SVD operations, in contrast to the other two MF based algorithms. It should be noted that in scenario A, the estimated by AIRLS-MC and IRNN matrices $\hat{\mathbf{U}}$ and $\hat{\mathbf{V}}$ have rank equal to 6. On the other hand, in softImpute-ALS the solution matrices have rank equal to the one used at the initialization stage, i.e., 100.

When it comes to the generalization performance of the proposed algorithm, from Table 9 it can be observed that, in both scenarios A and B, AIRLS-MC achieved lower NMAE on the unseen test set than its MF counterpart softImpute-ALS and slightly lower NMAE than IRNN. This actually shows that the reduced computational complexity of AIRLS-MC does not come at a price of inferior performance in terms of the accuracy of the estimated matrices. Lastly, from Fig. 30 it can be noticed that the relative objective of AIRLS-MC presents abrupt increases at some iterations. It was experimentally verified that those changes (which imply large decreases of the successive values of the objective function) take place at iterations that coincide with zeroings of the columns of the matrix factors. This fact advocates that larger gains are obtained at iterations where the rank is reduced, as we are approaching at the low-rank solution matrices.

Fig. 31 and Table 10 show the performance of AIRLS-MC and softImpute-ALS on the 1M Movielens dataset[11]. The parameter λ of AIRLS-MC and softImpute-ALS is fined tuned, in the same way as in the 100K experiment, based on the best NMAE attained by the algorithms on the validation set. The rank is again initialized to $d = 100$ for both algorithms. Interestingly, AIRLS-MC reaches a more accurate solution in terms of the NMAE

[11] IRNN has not been included in this experiment owing to its higher computational requirements as compared to both the MF based algorithms.

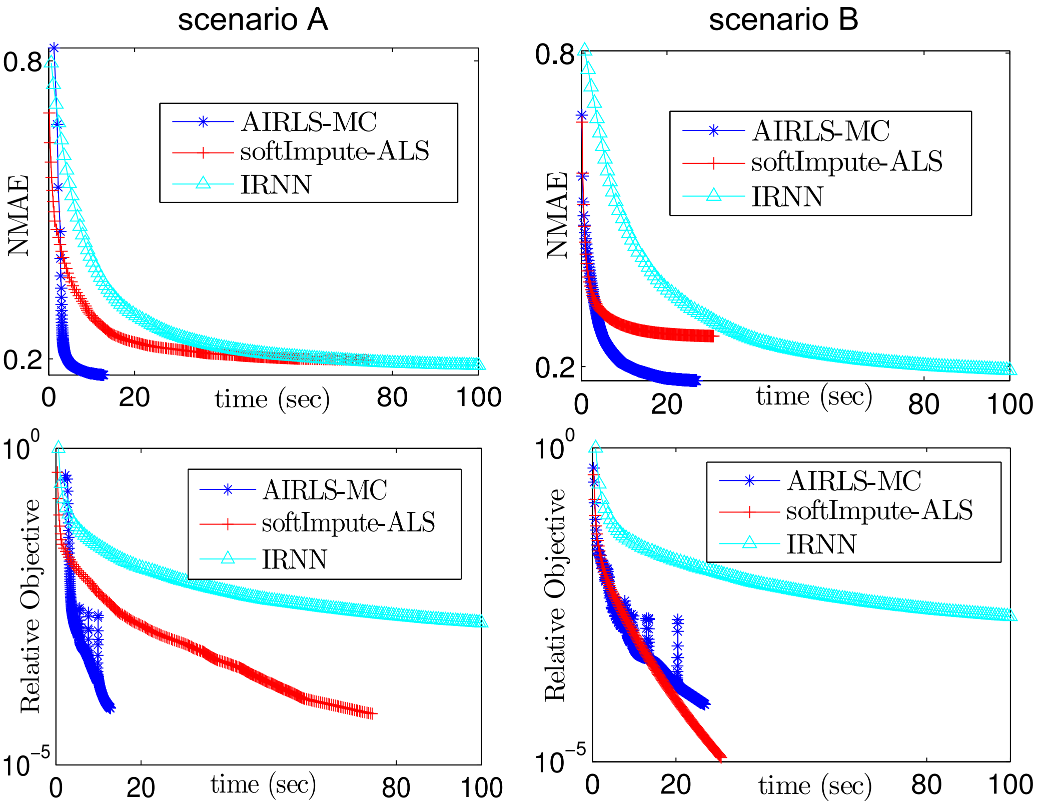


Figure 30: NMAE and relative objective vs time evolution (up to 100secs) of AIRLS-MC, softImpute-ALS and IRNN on the Movielens 100K validation dataset.

Table 10: Results obtained by AIRLS-MC and softImpute-ALS on Movielens 1M dataset.

	# Iter	msec/iter	total time (sec)	NMAE
softImpute-ALS	433	720	311.2	0.1862
AIRLS-MC	903	153	138.9	0.1760

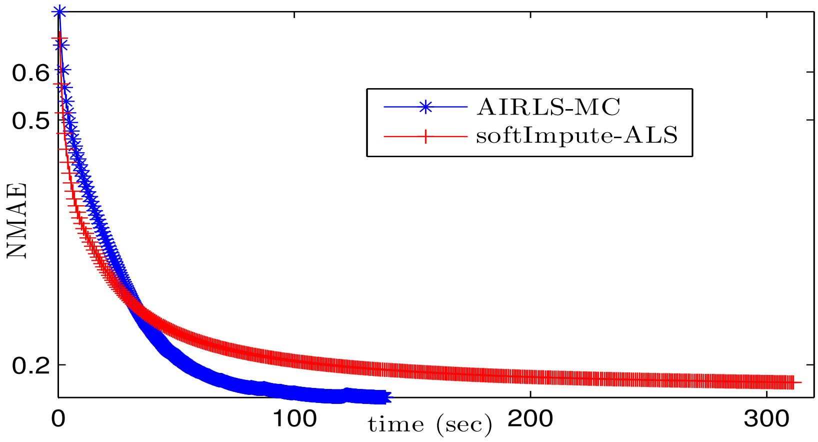


Figure 31: Evaluation of AIRLS-MC and softImpute-ALS on 1M Movielens dataset.

(as evaluated on the test set) in almost 40% of the time required by softImpute-ALS. Again, AIRLS-MC requires more iterations to converge as compared to its competitor. Nevertheless, as it can be also seen in Fig. 31, the column pruning mechanism which is activated in the initial iterations of AIRLS-MC results to a significant reduction of the average time spent per iteration.

4.5.2.3 Music signal decomposition

Herein, we test the competence of AIRLS-NMF algorithm in decomposing a real music signal. For this reason, AIRLS-NMF is compared to the most relevant state-of-the-art algorithm, i.e., ARD-NMF. In order to make as much fairer as possible the comparison between those two algorithms, the beta function of ARD-NMF algorithm of [139] was reduced to the square Frobenius norm, by appropriately setting the respective parameter. This way, ARD-NMF, likewise the proposed AIRLS-NMF, is based on Gaussian i.i.d noise assumptions. The music signal analyzed is a short piano sequence, i.e., a monophonic 15 seconds-long signal recorded in real conditions. As it can be noticed in Fig. 32, it is composed of four piano notes that overlap in all the duration thereof. Following the same process as in [139], the original signal is transformed into the frequency domain via the short-time Fourier transform (STFT). To this end, a Hamming window of size $L = 1024$ is utilized. By appropriately setting up the overlapping between the adjacent frames we are led to a spectrogram whereby the signal is represented by 673 frames in 513 frequency bins. The power of this spectrogram is then provided as input to the tested algorithms. The initial rank is set to 20 and the same stopping criterion as in the previous experiments

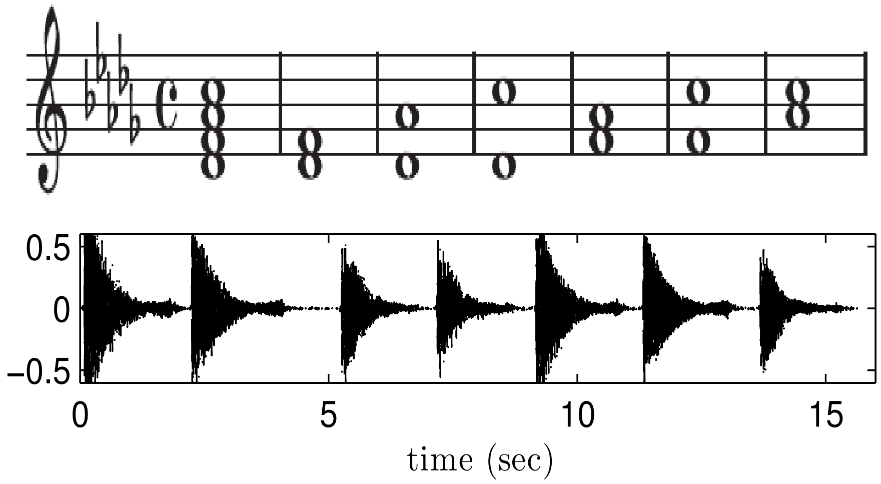


Figure 32: Music score (top) and original audio signal (bottom).

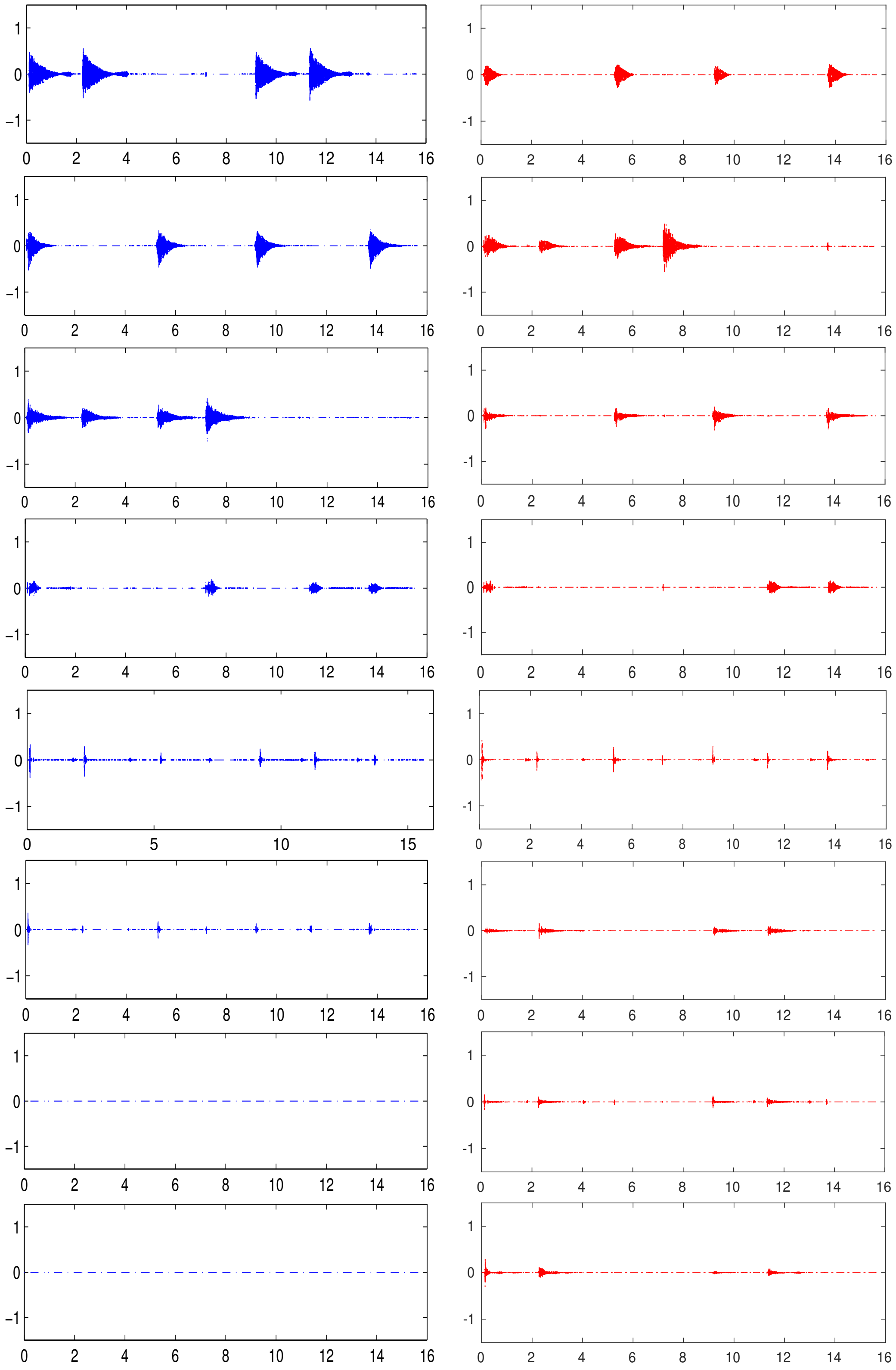
is utilized, with the threshold in this case set to 10^{-4}. Finally, the same process described in [139] was followed for reconstructing the music components, i.e., rank one terms of the product $\hat{\mathbf{U}}\hat{\mathbf{V}}^T$ in the time domain.

In Fig. 33, the first 8 components obtained by the two algorithms are ordered in decreasing values of the standard deviations of the time domain waveforms. As it can be noticed, AIRLS-NMF estimated the correct number of components, that is 6. Notably, the first four components of AIRLS-NMF correspond to the four notes while the rest two ones come from the sound of a hammer hitting the strings and the sound produced by the sustain pedal when it is released. On the contrary, ARD-NMF estimated 20 components, meaning that no rank minimization took place thus implying a data overfitting behavior. It should be emphasized that the favorable performance of AIRLS-NMF occurs though the noise is implicitly modelled as Gaussian i.i.d. Interestingly, as it can be seen in [139], AIRLS-NMF performed similarly to ARD IS-NMF, i.e., the version of ARD-NMF which makes more appropriate assumptions as to the noise statistics, by utilizing Itakura-Saito divergence.

4.5.3 Unsupervised hyperspectral unmixing

Herein our goal is to test and validate the SpAILRS-NMF algorithm on a real hyperspectral dataset. To this end we utilized the South Polar Cap HSI (see Section 3.4.3.2).

To evaluate the performance of SpAILRS-NMF on this real hyperspectral dataset, we suitably initialized the endmembers' matrix with the outcome of the VCA algorithm, [110], which was run with an overestimate $d = 8$ of endmembers. As is shown in Fig. 35(a), the resulting by VCA endmembers' matrix contains correlated spectra, which is expected from our prior knowledge i.e, that there exist 3 (instead of 8) pure endmembers in this specific image. Since there exist no reference spectra of the actual endmembers' spectral signatures, for comparison purposes, we use the ones returned by VCA which, this



(a) (b)

Figure 33: Music components obtained by AIRS-NMF (a) and ARD-NMF (b) on the short piano sequence.

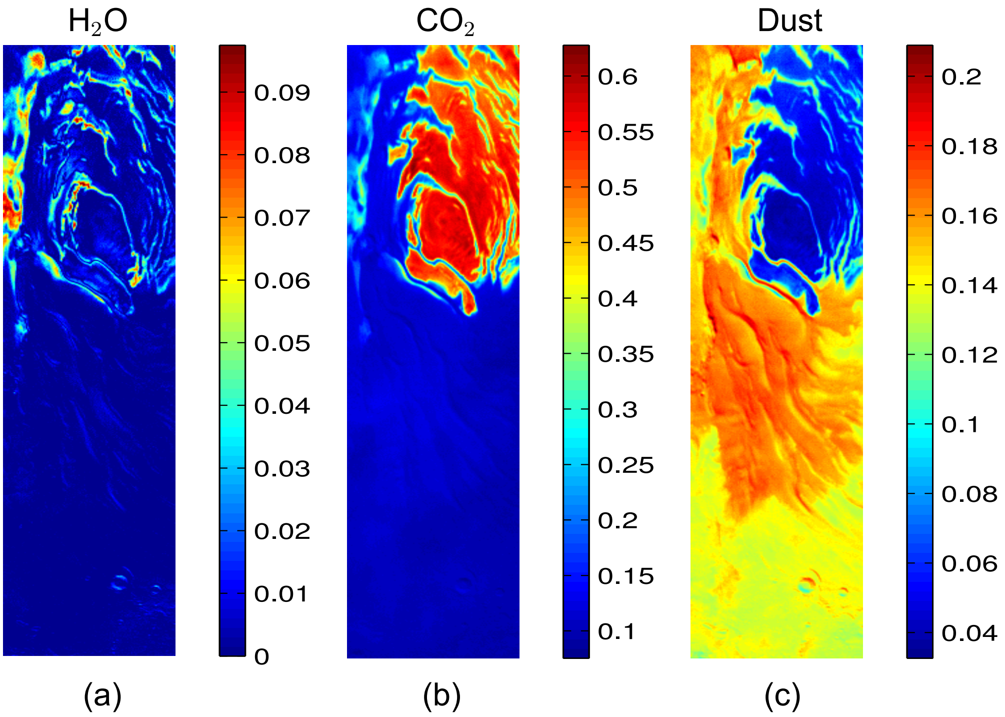
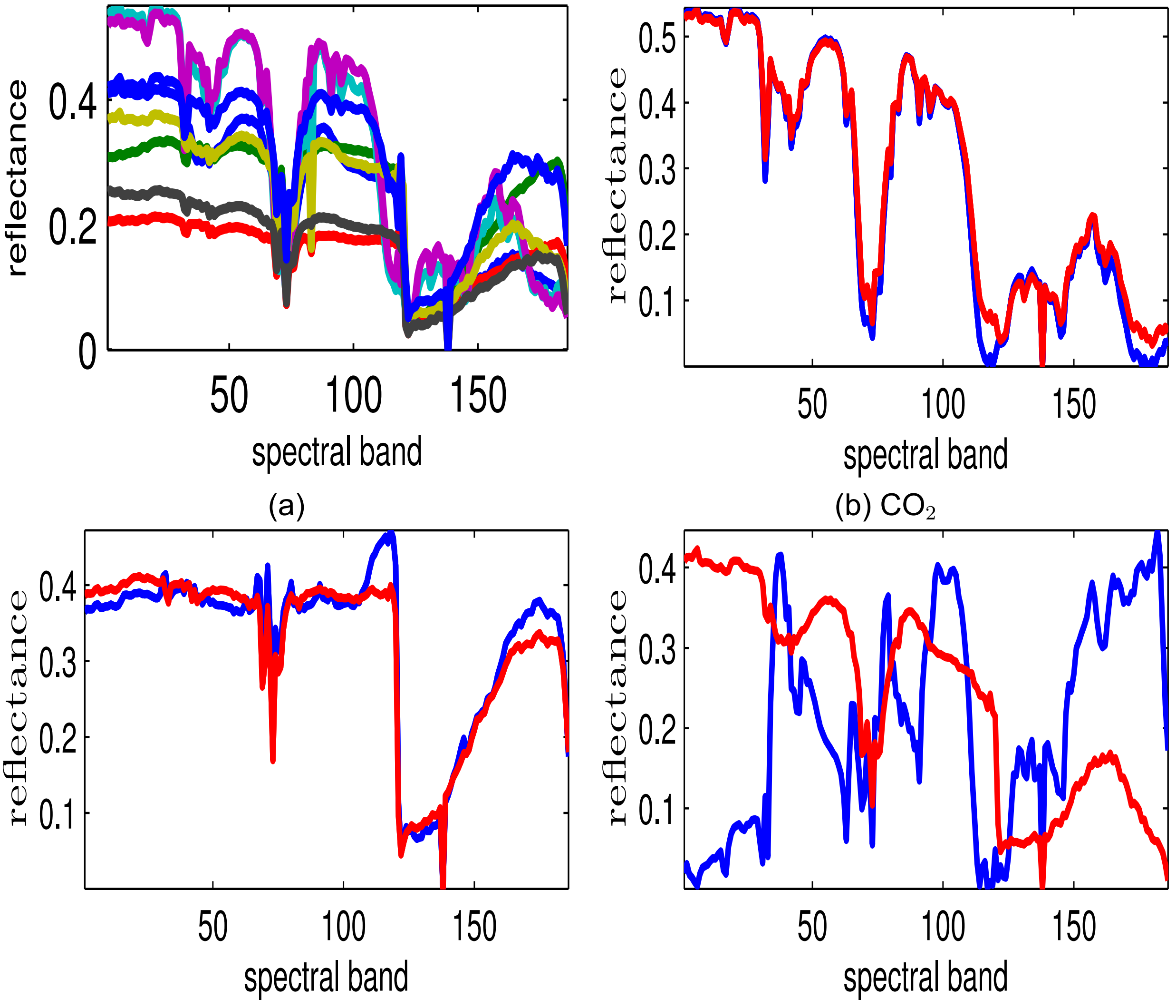


Figure 34: Abundance maps of South Polar Cap obtained by SpAIRLS-NMF algorithm.

time, was run for the correct number of endmembers. Notably, the proposed algorithm is proven to be capable of estimating the actual number of the endmembers. Moreover, the estimated endmembers' signatures (Fig. 35(b)-35(d)) for CO_2 and dust are quite close to those resulting by VCA, while the opposite holds for the respective spectral signature of the H_2O. This is probably due to the small abundance values of H_2O, as shown in Fig. 34a. Additionally, it is noted that the resulting abundance maps depicted in Fig. 34, are quite close to those that have been published in literature, [144].



(a)

(b) CO_2

(c) dust

(d) H_2O

Figure 35: (a) Eight endmembers' signatures obtained by VCA on South Polar Cap, (b)-(d) endmembers' signatures estimated by SpAIRLS-NMF (blue lines) and VCA (red lines).

5. ONLINE LOW-RANK SUBSPACE LEARNING FROM INCOMPLETE MEASUREMENTS

A common characteristic of the algorithms presented in the previous chapters is that all of them are batch-type algorithms, that is, each parameter updating takes into account all the available data. In the present chapter we turn to *online* algorithms, that is, algorithms where the parameter updating is based on a single data sample. The proposed algorithms have been developed to tackle the problem of low-rank subspace learning from incomplete measures, which is very closely related with the matrix factorization and matrix completion problems. An additional point of differentiation from the previous chapters is that one of the proposed algorithms stems from the variational Bayes framework. More specifically, two novel algorithms are introduced, namely, a) an online variational Bayes and b) an online deterministic cost function minimization based one. The main premise of both schemes is to leverage the low-rank promotion idea presented in Chapter 4 and adapt it to the *online processing* scenario. Since, the first algorithm hinges upon the Bayesian philosophy, an appropriate Bayesian model for the subspace learning problem is initially defined. Then, sticking to *batch* type processing, the variational Bayes method is utilized so as to approximately perform the posterior inference task. Finally, with a suitable extension and modification of the batch algorithm, an online algorithm is derived. Going one step further, sparsity constraints are imposed on the subspace matrix via appropriate Gaussian scale mixture priors, in order for the proposed scheme to be capable of addressing the sparse dictionary learning problem, [100], [161]. For the second - deterministic - algorithm, a novel low-rank regularized cost function is first introduced. Then, an alternating minimization process is followed for iteratively updating the subspace and the coefficients' matrix. Notably, the *nonseparability* of the adopted low-rank promoting term renders the cost function minimization and the subsequent derivation of the online deterministic algorithm a rather intriguing task. Extensive simulated and real data experiments corroborate the efficiency of the proposed schemes over other relevant state-of-the-art approaches.

5.1 Online subspace learning - A literature review

Detecting the underlying low-dimensional space (subspace) where high-dimensional data reside is at the heart of several signal processing and machine learning tasks, such as network anomalies detection,[102], image denoising, [49], [146], direction of arrival (DOA) estimation, [36], etc. Batch methods such as the celebrated PCA, which indubitably holds a prominent position in the family of this kind of algorithms, face considerable difficulties since a) their computational complexity scales with the size of the available measurement data and b) they require the storage of the whole bunch of data in memory. Therefore, its application is becoming practically prohibitive in the big data scenario under study.

In light of this, *online* subspace estimation (tracking) algorithms, that first came into the scene in the 1970s, [112, 26], have nowadays regain their popularity, [9, 36, 100]. These tools build upon the hypothesis that datums are sequentially arriving and thus the unknown subspace is *adaptively* estimated each time a new data sample becomes available. Interestingly, this premise, besides reducing the computational complexity, leads to schemes that do not require storage of the data in memory. Moreover, in a variety of applications dealing with large-scale datasets, the data to be processed are partially observed, i.e., a fraction of them might be missing. Depending on the case, incomplete datasets may result either from applying compressed sensing ideas in an effort to facilitate or account for failures in the data acquisition process, [29], [37] or from the inherent nature of signals met in disparate applications, e.g., collaborative filtering, [136], image reconstruction [75], etc. Consequently, algorithms that perform subspace tracking from (possibly highly) incomplete data have flourished notably in the last few years.

Focusing on the deterministic framework, the GROUSE algorithm, which brings forth an approach based on stochastic gradient descent on the Grassmanian manifold of subspaces, has been presented in [9]. Since stochastic approximation is at the core of GROUSE, its computational complexity classifies it to the low-complexity subspace tracking algorithms, [44]. Local and global convergence of GROUSE to the global minimum has been recently proved theoretically in [10] and [164], respectively. In [36], a second-order subspace tracking algorithm, of similar computational complexity to GROUSE, dubbed PETRELS, has been presented. PETRELS is an *unconstrained* alternating minimization recursive least squares (RLS)-type algorithm, building upon the seminal PASTd subspace tracking algorithm, [160], and extending it for handling missing data. A common characteristic of both the aforementioned algorithms is the rather strong assumption that the true rank of the sought subspace is known in advance. This shortcoming, which makes PETRELS exhibit an unstable behavior in case the assumption does not hold, is addressed in [103], where two different algorithms are described. Therein, the variational form of the nuclear norm is favorably employed for imposing low-rankness on the unknown subspace matrix, thus robustifying the algorithms in the challenging yet realistic scenario of lacking the knowledge of the subspace rank. In that vein, Algorithm 1 of [103] is introduced, deriving from an alternating minimization strategy on an exponentially weighted *regularized* cost function. In addition, a more efficient in terms of computational complexity Algorithm 2 is presented, based on a stochastic gradient descent approach.

In a Bayesian framework, low-rank subspace estimation from incomplete data has been recently dealt within [6]. Through an elegant joint column sparsity promoting mechanism, originally proposed in [138] in the context of nonnegative matrix factorization, the initially selected subspace rank is progressively reduced, tending to the true rank of the unknown subspace. In [6], group sparsity promoting Student-t type priors are employed and the variational Bayes method [152] is used for inference. In a similar vein, in [116], a Bayesian approach based on generalized approximate message passing for addressing the bilinear inference problem was presented. However, the subspace estimation algorithms developed in [6] and [116] are of a *batch* type and thus are not good candidates for processing high volumes of incomplete streaming data.

5.2 Problem formulation

In the online rationale, the concept of time is employed to describe the successive arrival of the data samples. To this end, let n be now the time-index and $\boldsymbol{y}(n)$ a sequence of high-dimensional $m \times 1$ vectors of observations that lie in a linear low-dimensional subspace of rank $r(n)$ with $r(n) \ll m$. Both the linear subspace and its rank may be time-varying. Accordingly, the observations at time n can be expressed as,

$$\boldsymbol{y}(n) = \boldsymbol{\mathcal{U}}(n)\mathbf{c}(n), \tag{5.1}$$

where $\boldsymbol{\mathcal{U}}(n)$ is a $m \times r(n)$ matrix whose columns span the underlying data subspace and vector $\mathbf{c}(n)$ contains the coefficients describing $\boldsymbol{y}(n)$ in this subspace. Since, in general, the true rank $r(n)$ of $\boldsymbol{\mathcal{U}}(n)$ is unknown and in order to account for noisy observations, we may assume that our data are produced based on the following linear regression model

$$\boldsymbol{y}(n) = \mathbf{U}(n)\mathbf{v}(n) + \boldsymbol{e}(n), \tag{5.2}$$

where $\mathbf{U}(n)$ is a $m \times d$ subspace matrix with $m \gg d \geq r(n)$ and span$(\boldsymbol{\mathcal{U}}(n)) \subseteq$ span$(\mathbf{U}(n))$. Moreover, in (5.2), the $d \times 1$ vector $\mathbf{v}(n)$ is the low-dimensional representation of $\boldsymbol{y}(n)$ in the subspace spanned by the columns of $\mathbf{U}(n)$ and $\boldsymbol{e}(n)$ is additive Gaussian noise. In other words, besides the noise, a reasonable *overestimate* of the true rank of the unknown data subspace is considered in our data generation model.

To generalize our model, we may further assume that a) the unknown subspace matrix $\mathbf{U}(n)$ may be sparse, a condition appearing in several applications[1] and b) part of the entries of $\boldsymbol{y}(n)$ are missing. The latter means that what we actually have is not $\boldsymbol{y}(n)$, but a sampled version of it, i.e., $\mathcal{P}_{\Omega_n}(\boldsymbol{y}(n))$ (where Ω_n is a set containing the indexes of $\boldsymbol{y}(n)$ where information is present), next denoted also as $\boldsymbol{z}(n)$,

$$\boldsymbol{z}(n) = \mathcal{P}_{\Omega_n}(\boldsymbol{y}(n)) \equiv \boldsymbol{\omega}(n) \odot \boldsymbol{y}(n) = \boldsymbol{\Omega}_n \boldsymbol{y}(n). \tag{5.3}$$

In (5.3), $\boldsymbol{\omega}(n)$ is a $\{0, 1\}$-binary $m \times 1$ vector having 0's at the positions where $\mathbf{y}(n)$ has missing entries and 1's elsewhere and $\boldsymbol{\Omega}_n = \text{diag}(\boldsymbol{\omega}(n))$. If we now stack together all the observation vectors (with possible missing elements) up to time n, as *columns* in a $m \times n$ matrix $\mathbf{Z}(n)$, yields

$$\mathbf{Z}(n) = \boldsymbol{\Omega}(n) \odot \mathbf{Y}(n) = \boldsymbol{\Omega}(n) \odot \big(\mathbf{U}(n)\mathbf{V}^T(n) + \mathbf{E}(n)\big), \tag{5.4}$$

where

$$\mathbf{Z}(n) = [\boldsymbol{z}(1), \boldsymbol{z}(2), \ldots, \boldsymbol{z}(n)] = [\mathbf{z}_1(n), \mathbf{z}_2(n), \ldots, \mathbf{z}_m(n)]^T, \tag{5.5}$$

$$\mathbf{Y}(n) = [\boldsymbol{y}(1), \boldsymbol{y}(2), \ldots, \boldsymbol{y}(n)] = [\mathbf{y}_1(n), \mathbf{y}_2(n), \ldots, \mathbf{y}_m(n)]^T, \tag{5.6}$$

$$\boldsymbol{\Omega}(n) = [\boldsymbol{\omega}(1), \boldsymbol{\omega}(2), \ldots, \boldsymbol{\omega}(n)] = [\boldsymbol{\varpi}_1(n), \boldsymbol{\varpi}_2(n), \ldots, \boldsymbol{\varpi}_m(n)]^T, \tag{5.7}$$

$$\mathbf{V}(n) = [\mathbf{v}(1), \mathbf{v}(2), \ldots, \mathbf{v}(n)]^T = [\boldsymbol{v}_1(n), \boldsymbol{v}_2(n), \ldots, \boldsymbol{v}_d(n)] \tag{5.8}$$

[1] This assumption is adopted only for the online variational Bayes algorithm.

and $\mathbf{E}(n) = [\boldsymbol{e}(1), \boldsymbol{e}(2), \ldots, \boldsymbol{e}(n)]$. In addition, we define the subspace matrix $\mathbf{U}(n)$ row and columnwise as[2]

$$\mathbf{U}(n) = [\mathbf{u}_1(n), \mathbf{u}_2(n), \ldots, \mathbf{u}_m(n)]^T = [\boldsymbol{u}_1(n), \boldsymbol{u}_2(n), \ldots, \boldsymbol{u}_d(n)] \,. \tag{5.9}$$

It can be noticed from Eqs. (5.5)-(5.9) that the column size of matrices $\mathbf{Z}(n), \mathbf{Y}(n), \boldsymbol{\Omega}(n)$ and $\mathbf{V}^T(n)$ increases with time, while $\mathbf{U}(n)$ is a fixed size $m \times d$ matrix.

The goals here are a) the estimation and tracking of the underlying low-dimensional subspace where measurement data reside, b) the estimation of the low-rank representation of data in this subspace in time and, as a by-product, c) the recovery of the complete measurement data matrix $\mathbf{Y}(n)$ via online matrix completion. In this context, given the batch of incomplete data $\mathbf{Z}(n)$, we aim at estimating the unknown low-rank subspace matrix $\mathbf{U}(n)$ and the latent matrix of projections $\mathbf{V}(n)$ in this subspace. However, in case of streamingly received data, the use of a batch iterative solver entails the processing of the whole bunch of data that are available up to every time instant, rendering the whole procedure computationally prohibitive and thus practically infeasible. A way to alleviate this impediment is by employing online data handling, whereby incomplete observation vectors $\mathbf{z}(n)$ are acquired and processed sequentially to learn and track $\mathbf{U}(n)$ and provide estimates of the vectors of coefficients $\mathbf{v}(n)$.

5.3 Online variational Bayes algorithm

In this section, we tackle the aforementioned problem using a Bayesian approach. First, an appropriate Bayesian model is defined that effectively promotes the low-rankness of the sought subspace through column sparsity inducing Laplace priors. As it will become clear below, the adopted modeling aims at revealing the true data subspace (spanned by the columns of $\boldsymbol{\mathcal{U}}(n)$) and its true rank $r(n)$, starting from an overestimate d of it. Based on the proposed Bayesian model, an iterative batch variational Bayes subspace estimation algorithm is developed, which after suitable adjustments, leads to an efficient online subspace learning scheme.

5.3.1 Proposed Bayesian model

To develop a Bayesian inference method, first a Bayesian model must be defined, whose basic structuring elements are a) the likelihood function of the data and b) suitable priors assigned to the parameters of the model. The likelihood function of the observed data depends on the statistical properties of the additive noise, which is commonly taken to be uncorrelated Gaussian with zero mean and constant variance. In this work, in order to place more importance on recent data and downgrade older measurements which is meaningful under time-varying conditions, we employ a so-called *forgetting factor* δ with

[2] Note that in (5.5)-(5.9), small boldface calligraphic letters have been used to denote columns of matrices and regular boldface letters to denote rows.

$0 \ll \delta < 1$ and define the noise distribution as[3],

$$\mathbf{E}(n) = \prod_{t=1}^{n} \mathcal{N}(\boldsymbol{e}(t)|\mathbf{0}, \beta^{-1}\delta^{t-n}\mathbf{I}_m), \tag{5.10}$$

where β is the noise precision parameter[4]. In the following, whenever not necessary, the time index n is omitted to simplify derivations. The time index is reestablished in Section 5.3.3, where the new online subspace estimation algorithm is presented. In this context, based on (5.4) and the noise distribution given in (5.10), the likelihood function of the measurement data is expressed as

$$p(\mathbf{Z} \mid \mathbf{V}, \mathbf{U}, \beta) = \prod_{t=1}^{n} p(\boldsymbol{z}(t) \mid \mathbf{v}(t), \mathbf{U}, \beta) = \prod_{t=1}^{n} \prod_{j \in \Omega_t} \mathcal{N}(z_j(t) \mid \mathbf{u}_j^T \mathbf{v}(t), \beta^{-1}\delta^{t-n}), \tag{5.11}$$

where Ω_t is the set of indices for which the corresponding entries of $\boldsymbol{\omega}(t)$ are 1 [5].

Now that the likelihood function has been defined, we proceed by presenting the prior distributions imposed on the subspace matrix **U** and the coefficients matrix **V**. These priors aim at simultaneously decreasing the rank and imposing sparsity on the unknown subspace matrix **U** by acting in the same way as the low-rank promoting term introduced in Chapter 4. In the Bayesian literature, a relevant scheme was proposed in [6] for reducing the rank by imposing column sparsity *jointly* on **U** and **V**. Herein, as in [6], this sparsity constraint is integrated in the modeling of the prior distributions of $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$, as explained below. At the same time, as stated earlier, in several applications (e.g. [27, 165]) the subspace matrix **U** is required to be sparse. That said, joint sparsity on $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ and the sparse structure on subspace matrix **U** are simultaneously incorporated in the modeling process of the corresponding prior distributions.

In light of this, three-level hierarchical priors[6] are assigned to the columns of **U** and **V**. At the first level of hierarchy the following Gaussian priors are defined,

$$p(\mathbf{U} \mid \mathbf{s}, \boldsymbol{\Gamma}, \beta) = \prod_{i=1}^{d} \mathcal{N}(\boldsymbol{u}_i \mid \mathbf{0}, \beta^{-1}s_i^{-1}\boldsymbol{\Gamma}_i^{-1}) \tag{5.12}$$

$$p(\mathbf{V} \mid \mathbf{s}, \beta) = \prod_{i=1}^{d} \mathcal{N}(\boldsymbol{v}_i \mid \mathbf{0}, \beta^{-1}s_i^{-1}\boldsymbol{\Delta}^{-1}), \tag{5.13}$$

where $\mathbf{s} = [s_1, s_2, \ldots, s_d]^T$, $\boldsymbol{\Gamma} = [\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \ldots, \boldsymbol{\gamma}_d]$, $\boldsymbol{\gamma}_i = [\gamma_{1i}, \gamma_{2i}, \ldots, \gamma_{mi}]^T$, $\boldsymbol{\Gamma}_i = \text{diag}(\boldsymbol{\gamma}_i)$ for $i = 1, 2, \ldots, d$ and $\boldsymbol{\Delta}(n) = \text{diag}([\delta^{n-1}, \delta^{n-2}, \ldots, \delta, 1]^T)$. It can be observed from (5.12) and (5.13) that the ith columns of **U** and **V** share the same *joint sparsity promoting parameters*

[3] From (5.10), it is evident that the role of δ is to increase the variance of the older measures, making them less reliable than the more recent measures.

[4] β is the inverse of the noise variance.

[5] The latter equality is due to (5.4), since, for given **U** and **V**, the distribution of **Z** is also Gaussian.

[6] Hierarchical priors are required in order to ensure *conjugacy* with respect to the likelihood as well as among them, which is a prerequisite for deriving a tractable posterior inference procedure, [146, 142].

s_i's. In particular, some of the s_i's take large values when Bayesian inference is performed and as a result, both the ith columns of **U** and **V** are driven to zero. At the same time, the diagonal matrix $\boldsymbol{\Gamma}_i$ which appears in the prior distribution of **U** is responsible for *independently* imposing sparsity on the entries of the ith column of the subspace matrix[7]. Notably, in cases where a parameter s_i does not enforce joint sparsity, the jth element of the ith column of **U**, u_{ji}, may be independently led to zero by the corresponding *subspace sparsity promoting* parameter γ_{ji} of $\boldsymbol{\Gamma}_i$. It should be also noted that the exponentially weighting matrix $\boldsymbol{\Delta}$ appears in the prior of **V**, but not in that of **U**. This is so because in a streaming data environment the size of **V** is time-increasing while the fixed-size subspace matrix **U** is estimated based not only on the most recent row $\mathbf{v}(n)$, but also on the previous rows of **U** with appropriate weighting. On the other hand, in such an online scenario, the current projection coefficients vector $\mathbf{v}(n)$ shall be estimated only from the more recent estimate of **U**, which, being fixed-size, does not have to be exponentially weighted.

The prior distribution of **V** in (5.13) can be written in an equivalent form with respect to the rows of **V** as follows

$$p(\mathbf{V} \mid \mathbf{s}, \beta) = \prod_{t=1}^{n} \mathcal{N}(\mathbf{v}(t) \mid \mathbf{0}, \beta^{-1}\delta^{t-n}\mathbf{S}^{-1}), \tag{5.14}$$

where $\mathbf{S} = \text{diag}(\mathbf{s})$. Note that it is the form of the prior in (5.14) that is mainly used in the analysis of the next sections, although (5.13) serves in this section to show how the rank is reduced by the proposed model. At the second level of the hierarchy we define the following conjugate[8] inverse Gamma distributions for **s** and $\boldsymbol{\Gamma}$,

$$p(\mathbf{s} \mid \boldsymbol{\lambda}) = \prod_{i=1}^{d} \mathcal{IG}(s_i \mid \frac{m+n+1}{2}, \frac{\lambda_i}{2}), \tag{5.15}$$

$$p(\boldsymbol{\Gamma} \mid \boldsymbol{\mathcal{P}}) = \prod_{j=1}^{m}\prod_{i=1}^{d} \mathcal{IG}(\gamma_{ji} \mid 1, \frac{\rho_{ji}}{2}). \tag{5.16}$$

where $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_d]^T$ and $\boldsymbol{\mathcal{P}}$ is the $m \times d$ matrix whose entries are the ρ_{ji}'s. Finally, at the third level of the hierarchy, conjugate Gamma distributions are defined for the scale parameters λ_i's and ρ_{ji}'s, i.e.

$$p(\lambda_i) = \mathcal{G}(\lambda_i; \mu, \nu) \tag{5.17}$$

$$p(\rho_{ji}) = \mathcal{G}(\rho_{ji}; \psi, \xi), \tag{5.18}$$

By integrating out **s** from (5.13) and (5.12) using (5.15) with $\boldsymbol{\Gamma}$ kept fixed, we are led to

[7] In case **U** is not sparse, we set $\boldsymbol{\Gamma}_i = \mathbf{I}_m$ in (5.12) and no prior applies to $\boldsymbol{\Gamma}$, i.e. Eqs (5.16) and (5.18) below are needless.

[8] A prior and a posterior are called *conjugate* distributions if they belong to the same family of distributions for a given likelihood.

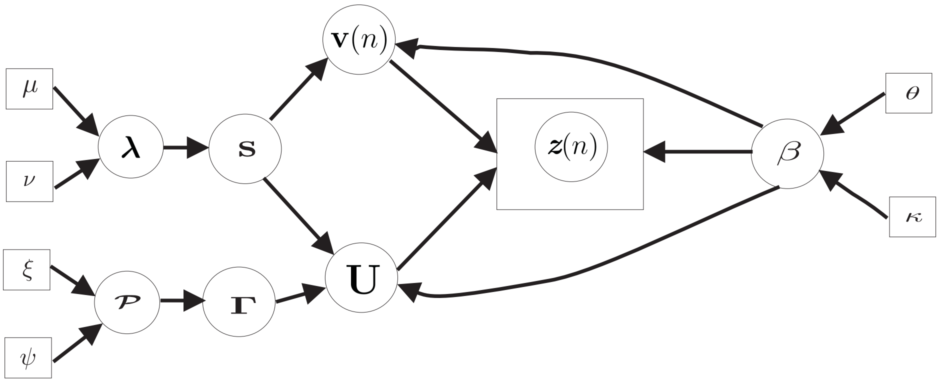


Figure 36: Directed acyclic graph of the proposed Bayesian model.

a *heavy-tailed multiparameter Laplace-type distribution* for the joint prior of **U** and **V** that promotes joint column sparsity, as is shown in Section 5.6. Similarly, by fixing **s**, from (5.12) and (5.16) we get a multiparameter Laplace prior that imposes sparsity on **U**.

The proposed Bayesian model is concluded by assigning a conjugate to the likelihood Gamma prior to model the precision of the noise β as follows,

$$p(\beta) = \mathcal{G}(\beta; \kappa, \theta). \tag{5.19}$$

It should be noted that the proposed Bayesian model, which is built upon the likelihood (5.11) and the priors (5.13)-(5.19), differs considerably and improves over the relevant model reported in [6]. The novelty of the new model comes from a) the promotion of sparsity on **U** aside from low-rank through the use of the parameter matrix Γ, b) the (necessary for online processing) exponential weighting of the data by incorporating a forgetting factor in the likelihood and the prior of **V** and c) the adoption of Laplace-type marginal priors for **U** and **V**, instead of Student-t used in [6], in order to promote sparsity and low-rankness. In the next section, based on the multi-hierarchical model introduced before and presented graphically in Fig. 36, an approximate Bayesian inference scheme is derived for low-rank sparse subspace learning from partial observations.

5.3.2 Batch variational Bayes inference

Inferring the joint posterior distribution of multiple variables given the data boils down to an intractable problem when it comes to composite Bayesian models, such as those springing from hierarchical dependences of the involved variables, which are modeled by suitable priors. This is also the case for the Bayesian model described in the previous section. Following the Bayes' theorem, the exact joint posterior of our variables given the observations is obtained by

$$p(\mathbf{U}, \mathbf{V}, \mathbf{s}, \Gamma, \boldsymbol{\lambda}, \boldsymbol{\mathcal{P}}, \beta \mid \mathbf{Z}) = \frac{p(\mathbf{Z}, \mathbf{U}, \mathbf{V}, \mathbf{s}, \Gamma, \boldsymbol{\lambda}, \boldsymbol{\mathcal{P}}, \beta)}{\int p(\mathbf{Z}, \mathbf{U}, \mathbf{V}, \mathbf{s}, \Gamma, \boldsymbol{\lambda}, \boldsymbol{\mathcal{P}}, \beta) d\mathbf{U} d\mathbf{V} d\mathbf{s} d\Gamma d\boldsymbol{\lambda} d\boldsymbol{\mathcal{P}} d\beta}. \tag{5.20}$$

Apparently, getting a closed form expression for the posterior given in (5.20) involves the daunting task of estimating the integral at the denominator. To obviate obstacles of this type, many approximate inference schemes have come to light in literature, [106, 31]. Herein, the ubiquitous variational Bayes inference approach is adopted, [152]. As also described in Section 2.1.5, the basic premise of this approach -inspired from the field of statistical physics- is the assumption that the posterior distribution can be approximately expressed in a factorized form. Based on this particular hypothesis, the exact joint posterior $p(\mathbf{U}, \mathbf{V}, \mathbf{s}, \boldsymbol{\Gamma}, \boldsymbol{\lambda}, \boldsymbol{\mathcal{P}}, \beta \mid \mathbf{Z})$ is approximated by $q(\mathbf{U}, \mathbf{V}, \mathbf{s}, \boldsymbol{\Gamma}, \boldsymbol{\lambda}, \boldsymbol{\mathcal{P}}, \beta)$, defined as

$$q(\mathbf{U}, \mathbf{V}, \mathbf{s}, \boldsymbol{\Gamma}, \boldsymbol{\lambda}, \boldsymbol{\mathcal{P}}, \beta) = q(\beta) \prod_{t=1}^{n} q(\mathbf{v}(t)) \prod_{j=1}^{m} \prod_{i=1}^{d} q(u_{ji}) \prod_{i=1}^{d} q(s_i) \prod_{i=1}^{d} q(\lambda_i) \prod_{j=1}^{m} \prod_{i=1}^{d} q(\gamma_{ji}) q(\rho_{ji}). \tag{5.21}$$

From (5.21) it is easily noticed that there has been considered full statistical *a posteriori* independence among the rows of $\mathbf{V}$, as well as among all the elements of the subspace matrix $\mathbf{U}$. As far as $\mathbf{v}(t)$'s are concerned, being statistical independent is something that is naturally brought up due to the presumed independence among the corresponding observation vectors $\mathbf{z}(t)$'s. On the other hand and in contrast to previous related works (e.g., [6]), posterior independence is imposed on the entries of $\mathbf{U}$ in (5.21). This gives rise to coordinate descent recursions for retrieving u_{ji}'s, which, as shown later, reduces significantly the computational complexity of the online subspace estimation task. Notably, as implied by (5.21), those explicit assumptions on the independence among the rows of $\mathbf{U}$ and the elements of $\mathbf{V}$ dictate relevant statistical independence on the variables of our model belonging to the second and the third level of hierarchy, namely $\mathbf{s}, \boldsymbol{\lambda}, \boldsymbol{\Gamma}$ and $\boldsymbol{\mathcal{P}}$.

In an attempt to bring to light the particular way that the posterior distributions $q(\cdot)$'s of all variables in (5.21) are recovered according to the variational Bayes scheme, we define the cell array $\boldsymbol{\theta} = \{\mathbf{v}(1), \dots, \mathbf{v}(n), u_{11}, \dots, u_{md}, s_1, \dots, s_d, \gamma_{11}, \dots, \gamma_{md}, \lambda_1, \dots, \lambda_d, p_{11}, \dots, p_{md}\}$[9]. The posterior distribution $q(\boldsymbol{\theta}_i)$ of each component $\boldsymbol{\theta}_i$ is then obtained by maximizing the evidence lower bound (ELBO)[10] (see Eq. 2.51). As detailed in Section 2.1.5.3, this minimization process leads to closed-form expressions, i.e.,

$$q(\boldsymbol{\theta}_i) \propto \exp\{\langle \ln p(\boldsymbol{\theta}_i | \boldsymbol{\theta}_{\neg i}, \mathbf{Z}) \rangle_{\neg i}\}. \tag{5.22}$$

In the last equation $\langle \cdot \rangle_{\neg i}$ denotes expectation taken with respect to $q(\boldsymbol{\theta}_{\neg i})$ (see Section 2.1.5). As explained in Section 2.1.5, by maximizing the evidence lower bound (ELBO) we resort to a coordinate ascent scheme whereby the parameters of each $q(\boldsymbol{\theta}_i)$ are computed based on the most recent estimates of the parameters of the rest $q(\boldsymbol{\theta}_j)$'s. This procedure is applied for our three-level hierarchical Bayesian model and the whole derivation is next provided.

[9] Note that for notational convenience, the entries of $\boldsymbol{\theta}$, i.e., the $\boldsymbol{\theta}_i$'s may represent either vectors or scalars.

[10] Recall that ELBO is a lower bound of the Kullback-Leibler divergence criterion between the the approximate posterior $q(\boldsymbol{\theta})$ and the true one, i.e., $p(\boldsymbol{\theta} \mid \mathbf{Z})$.

Due to the conjugacy of the respective prior distributions (5.13), (5.12) and the likelihood (5.11), the posterior distribution $q(\mathbf{v}(t))$ of the tth coefficient vector does turn out to be Gaussian, i.e.,

$$q(\mathbf{v}(t)) = \mathcal{N}\left(\mathbf{v}(t) \mid \langle \mathbf{v}(t) \rangle, \boldsymbol{\Sigma}_{\mathbf{v}(t)}\right), \tag{5.23}$$

with mean $\langle \mathbf{v}(t) \rangle$ and covariance matrix $\boldsymbol{\Sigma}_{\mathbf{v}(t)}$ given by,

$$\langle \mathbf{v}(t) \rangle = \langle \beta \rangle \boldsymbol{\Sigma}_{\mathbf{v}(t)} \langle \mathbf{U} \rangle^T \mathbf{z}(t), \tag{5.24}$$

$$\boldsymbol{\Sigma}_{\mathbf{v}(t)} = \langle \beta \rangle^{-1} \left(\langle \mathbf{U}^T \boldsymbol{\Omega}_t \mathbf{U} \rangle + \langle \mathbf{S} \rangle \right)^{-1}, \tag{5.25}$$

where we recall that $\boldsymbol{\Omega}_t = \text{diag}(\boldsymbol{\omega}(t))$. The expectation term $\langle \mathbf{U}^T \boldsymbol{\Omega}_t \mathbf{U} \rangle$ is expressed as,

$$\langle \mathbf{U}^T \boldsymbol{\Omega}_t \mathbf{U} \rangle = \langle \mathbf{U} \rangle^T \boldsymbol{\Omega}_t \langle \mathbf{U} \rangle + \sum_{j=1}^{m} \omega_{jt} \boldsymbol{\Sigma}_{\mathbf{u}_j} \tag{5.26}$$

where $\boldsymbol{\Sigma}_{\mathbf{u}_j} = \text{diag}([\sigma^2_{u_{j1}}, \sigma^2_{u_{j2}}, \ldots, \sigma^2_{u_{jd}}]^T)$ by virtue of the statistical independence assumed for the elements of $\mathbf{U}$. Note that $\sigma^2_{u_{ji}}$ is the variance of u_{ji} whose posterior turns out also to be Gaussian, i.e.,

$$q(u_{ji}) = \mathcal{N}(u_{ji} \mid \langle u_{ji} \rangle, \sigma^2_{u_{ji}}), \tag{5.27}$$

with

$$\langle u_{ji} \rangle = \langle \beta \rangle \sigma^2_{u_{ji}} \left(\langle \boldsymbol{v}_i \rangle^T \boldsymbol{\Delta} \mathbf{z}_j - \langle \boldsymbol{v}_i^T \boldsymbol{\Delta} \boldsymbol{\mathit{\Omega}}_t \mathbf{V}_{\neg i} \rangle \langle \mathbf{u}_{j\neg i} \rangle \right), \tag{5.28}$$

$$\sigma^2_{u_{ji}} = \langle \beta \rangle^{-1} \left(\langle \boldsymbol{v}_i^T \boldsymbol{\Delta} \boldsymbol{\mathit{\Omega}}_t \boldsymbol{v}_i \rangle + \langle \gamma_{ji} \rangle \langle s_i \rangle \right)^{-1}. \tag{5.29}$$

$\mathbf{V}_{\neg i}$ and $\mathbf{u}_{j\neg i}$ in (5.28) are the quantities arising after removing the ith column and the ith element of $\mathbf{V}$ and $\mathbf{u}_j$, respectively and $\boldsymbol{\mathit{\Omega}}_t = \text{diag}(\boldsymbol{\varpi}_t)$. As for the expectation terms appearing in (5.28) and (5.29), it holds,

$$\langle \boldsymbol{v}_i^T \boldsymbol{\Delta} \boldsymbol{\mathit{\Omega}}_t \mathbf{V}_{\neg i} \rangle = \langle \boldsymbol{v}_i \rangle^T \boldsymbol{\Delta} \boldsymbol{\mathit{\Omega}}_t \langle \mathbf{V}_{\neg i} \rangle + \sum_{t=1}^{n} \Delta^{n-t} \omega_{jt} \boldsymbol{\sigma}^T_{v(t)\neg i}, \tag{5.30}$$

$$\langle \boldsymbol{v}_i^T \boldsymbol{\Delta} \boldsymbol{\mathit{\Omega}}_t \boldsymbol{v}_i \rangle = \langle \boldsymbol{v}_i \rangle^T \boldsymbol{\Delta} \boldsymbol{\mathit{\Omega}}_t \langle \boldsymbol{v}_i \rangle + \sum_{t=1}^{n} \delta^{n-t} \omega_{jt} \sigma_{v_{ti}}, \tag{5.31}$$

with $\boldsymbol{\sigma}_{v(t)\neg i}$ standing for the ith column of $\boldsymbol{\Sigma}_{\mathbf{v}(t)}$ after removing its ith element $\sigma_{v_{ti}}$.

Next, the posterior distributions of the variables s_i's and γ_{ji}'s belonging to the second hierarchical level are unfolded. From (5.22) it can be shown that the column sparsity promoting parameters s_i's are *a posteriori* distributed according to the following generalized inverse Gaussian distribution,

$$q(s_i) = \mathcal{GIG}\left(s_i \mid -\frac{1}{2}, \langle\beta\rangle\left(\langle \boldsymbol{u}_i^T \boldsymbol{\Gamma}_i \boldsymbol{u}_i\rangle + \langle \mathbf{v}_i^T \boldsymbol{\Delta} \mathbf{v}_i\rangle\right), \langle\lambda_i\rangle\right). \tag{5.32}$$

For the mean $\langle s_i\rangle$ of the $\mathcal{GIG}$ distribution it holds,

$$\langle s_i\rangle = \sqrt{\frac{\langle\lambda_i\rangle}{\langle\beta\rangle\left(\langle \boldsymbol{u}_i^T \boldsymbol{\Gamma}_i \boldsymbol{u}_i\rangle + \langle \mathbf{v}_i^T \boldsymbol{\Delta} \mathbf{v}_i\rangle\right)}}. \tag{5.33}$$

Likewise, the posterior distribution of γ_{ji}'s that promote independently sparsity on the elements of the subspace matrix **U** is the generalized inverse Gaussian

$$q(\gamma_{ji}) = \mathcal{GIG}\left(\gamma_{ji} \mid -\frac{1}{2}, \langle\beta\rangle\langle s_i\rangle\langle u_{ji}^2\rangle, \langle\rho_{ji}\rangle\right), \tag{5.34}$$

with $\langle u_{ji}^2\rangle = \langle u_{ji}\rangle^2 + \sigma_{u_{ji}}^2$. Hence,

$$\langle\gamma_{ji}\rangle = \sqrt{\frac{\langle\rho_{ji}\rangle}{\langle\beta\rangle\langle s_i\rangle(\langle u_{ji}\rangle^2 + \sigma_{u_{ji}}^2)}}. \tag{5.35}$$

As far as the posteriors of hyperparameters λ_i and ρ_{ji}, associated with s_i and γ_{ji}, respectively, are concerned, both are Gamma distributions, i.e.,

$$q(\lambda_i) = \mathcal{G}\left(\lambda_i \mid \bar{\mu}, \bar{\nu}_i\right), \tag{5.36}$$

with $\bar{\mu} = \mu + \frac{n+m+1}{2}$ and $\bar{\nu}_i = \nu + \frac{1}{2}\langle\frac{1}{s_i}\rangle$, and

$$q(\rho_{ji}) = \mathcal{G}\left(\rho_{ji} \mid \bar{\psi}, \bar{\xi}_{ji}\right), \tag{5.37}$$

with $\bar{\psi} = \psi + 1$ and $\bar{\xi}_{ij} = \xi + \frac{1}{2}\langle\frac{1}{\gamma_{ij}}\rangle$. For the expected values of λ_i and ρ_{ji}, that is $\langle\lambda_i\rangle$ and $\langle\rho_{ji}\rangle$ we have,

$$\langle\lambda_i\rangle = \frac{\mu + \frac{n+m+1}{2}}{\nu + \frac{1}{2}\langle\frac{1}{s_i}\rangle}, \tag{5.38}$$

$$\langle\rho_{ji}\rangle = \frac{\psi + 1}{\xi + \frac{1}{2}\langle\frac{1}{\gamma_{ji}}\rangle}. \tag{5.39}$$

Using the form of the distributions in (5.32) and (5.34), the expectation terms $\langle\frac{1}{s_i}\rangle$ and $\langle\frac{1}{\gamma_{ji}}\rangle$

arising in (5.38) and (5.39) can be obtained as,

$$\left\langle \frac{1}{s_i} \right\rangle = \frac{1}{\langle s_i \rangle} + \frac{1}{\langle \lambda_i \rangle}, \tag{5.40}$$

$$\left\langle \frac{1}{\gamma_{ji}} \right\rangle = \frac{1}{\langle \gamma_{ji} \rangle} + \frac{1}{\langle \rho_{ji} \rangle}. \tag{5.41}$$

Concluding the posterior distributions of all the involved variables in our hierarchical model, it can be shown that the noise precision β is Gamma distributed as follows,

$$q(\beta) = \mathcal{G}\left(\beta \mid \bar{\kappa}, \bar{\theta}\right), \tag{5.42}$$

where $\bar{\kappa} = \kappa + \frac{n(m+d)+md}{2}$ and

$$\bar{\theta} = \theta + \sum_{j=1}^{m} \left(\langle \| \mathbf{\Delta}^{\frac{1}{2}} \left(\mathbf{z}_j - \boldsymbol{\Omega}_j \mathbf{V} \mathbf{u}_j \right) \|_2^2 \rangle + \langle \mathbf{u}_j^T \mathbf{S} \boldsymbol{\Gamma}_j \mathbf{u}_j \rangle \right) + \sum_{i=1}^{d} \langle s_i \rangle \langle \boldsymbol{v}_i^T \mathbf{\Delta} \boldsymbol{v}_i \rangle. \tag{5.43}$$

The expectation of β is given by $\langle \beta \rangle = \frac{\bar{\kappa}}{\bar{\theta}}$. As for the expectation terms arising in (5.43), it holds,

$$\begin{aligned} \langle \| \mathbf{\Delta}^{\frac{1}{2}} \left(\mathbf{z}_j - \boldsymbol{\Omega}_j \mathbf{V} \mathbf{u}_j \right) \|_2^2 \rangle = {} & \| \mathbf{\Delta}^{\frac{1}{2}} \left(\mathbf{z}_j - \boldsymbol{\Omega}_j \langle \mathbf{V} \rangle \langle \mathbf{u}_j \rangle \right) \|_2^2 + \mathrm{Tr}\left(\langle \mathbf{V} \rangle^T \mathbf{\Delta} \boldsymbol{\Omega}_j \langle \mathbf{V} \rangle \Sigma_{\mathbf{u}_j} \right) \\ & + \langle \mathbf{u}_j \rangle^T \sum_{t=1}^{n} \omega_{jt} \delta^{n-t} \Sigma_{\mathbf{v}(t)} \langle \mathbf{u}_j \rangle + \mathrm{Tr}\left(\Sigma_{\mathbf{u}_j} \sum_{t=1}^{n} \omega_{jt} \delta^{n-i} \Sigma_{\mathbf{v}(t)} \right) \end{aligned} \tag{5.44}$$

$$\langle \mathbf{u}_j^T \mathbf{S} \mathbf{\Gamma}_i \mathbf{u}_j \rangle = \langle \mathbf{u}_j \rangle^T \langle \mathbf{S} \rangle \langle \mathbf{\Gamma}_i \rangle \langle \mathbf{u}_j \rangle + \sum_{i=1}^{d} \langle s_i \rangle \langle \gamma_{ji} \rangle \sigma_{u_{ji}}^2 \tag{5.45}$$

Note that due to the novelty of the proposed Bayesian model and the assumed posterior independence of the entries of **U**, (5.27)-(5.45) are new. The mutual dependence among the moments of all the model parameters, that can be easily observed in the respective expressions, paves the way for an iterative scheme over the involved quantities. It should be emphasized though that since we aim at handling a massive amount of streaming data, the utilization of those expressions ends up to be a prohibitive task. More specifically, as the number n of the observations increases, calculations that involve quantities such as $\mathbf{Z}, \mathbf{V}$, become increasingly demanding in terms of both the memory storage and the computational effort. In light of this, an online scheme is presented in the next section, that favorably adjusts the above defined expressions to the streaming processing scenario.

5.3.3 Online variational Bayes subspace estimation

In this section we derive a new online variational Bayes algorithm for sparse and low-rank subspace estimation from incomplete data. As shown below, moving from the batch to the online scenario is not a trivial task. It requires a) the definition of appropriate fixed-size quantities that can be recursively updated and b) their combination with other formulas coming from the batch algorithm in a cohesive scheme. According to the online scenario, incomplete high dimensional datums $\mathbf{z}(n)$'s are streamingly received as n evolves. Then, the proposed algorithm proceeds by a) computing an estimate $\hat{\mathbf{v}}(n)$ of the coefficients vector of the observations on the subspace acquired in the previous iteration (i.e. $\hat{\mathbf{U}}(n-1)$) and next b) updating *elementwisely* the subspace matrix $\hat{\mathbf{U}}(n-1)$ to $\hat{\mathbf{U}}(n)$. In the sequel, for notational convenience, we disregard the expectation operator $\langle\cdot\rangle$. Then, with a slight but straightforward abuse of notation and by handling the time index appropriately, we get from (5.24), (5.25) and (5.26),

$$\hat{\mathbf{v}}(n) = \beta(n-1)\boldsymbol{\Sigma}_{\hat{\mathbf{v}}}(n)\hat{\mathbf{U}}^T(n-1)\mathbf{z}(n), \tag{5.46}$$

$$\boldsymbol{\Sigma}_{\hat{\mathbf{v}}}(n) = \beta^{-1}(n-1)\left(\hat{\mathbf{U}}^T(n-1)\boldsymbol{\Omega}_n\hat{\mathbf{U}}(n-1) + \sum_{j=1}^{m}\omega_j(n)\boldsymbol{\Sigma}_{\hat{\mathbf{u}}_j}(n-1) + \mathbf{S}(n-1)\right)^{-1}. \tag{5.47}$$

Next, we define the following *fixed-size with respect to time* quantities,

$$\mathbf{A}(n) = \hat{\mathbf{V}}^T(n)\boldsymbol{\Delta}(n)\mathbf{Z}^T(n), \tag{5.48}$$

$$\mathbf{Q}(n) = \hat{\mathbf{V}}^T(n)\boldsymbol{\Delta}(n)\hat{\mathbf{V}}(n) + \sum_{t=1}^{n}\delta^{n-t}\boldsymbol{\Sigma}_{\hat{\mathbf{v}}}(t), \tag{5.49}$$

and for $j = 1, 2, \ldots, m$,

$$\mathbf{P}_j(n) = \hat{\mathbf{V}}^T(n)\boldsymbol{\Delta}(n)\boldsymbol{\Omega}_j(n)\hat{\mathbf{V}}(n) + \sum_{t=1}^{n}\delta^{n-t}\omega_j(t)\boldsymbol{\Sigma}_{\hat{\mathbf{v}}}(t), \tag{5.50}$$

where $\boldsymbol{\Omega}_j(n) = \text{diag}(\boldsymbol{\varpi}_j(n))$, and

$$d_j(n) = \mathbf{z}_j^T(n)\boldsymbol{\Delta}(n)\mathbf{z}_j(n). \tag{5.51}$$

The basic idea in any online scheme is the formulation of the various quantities that carry the past knowledge of the relevant process in a time-recursive manner. Interestingly, Eqs. (5.48)-(5.51) can easily be written in time-recursive forms, i.e.,

$$\mathbf{A}(n) = \delta\mathbf{A}(n-1) + \hat{\mathbf{v}}(n)\mathbf{z}^T(n), \tag{5.52}$$

$$\mathbf{Q}(n) = \delta\mathbf{Q}(n-1) + \boldsymbol{\Sigma}_{\hat{\mathbf{v}}}(n) + \hat{\mathbf{v}}(n)\hat{\mathbf{v}}^T(n), \tag{5.53}$$

$$\mathbf{P}_j(n) = \delta\mathbf{P}_j(n-1) + \omega_j(n)\left(\boldsymbol{\Sigma}_{\hat{\mathbf{v}}}(n) + \hat{\mathbf{v}}(n)\hat{\mathbf{v}}^T(n)\right), \tag{5.54}$$

$$d_j(n) = \delta d_j(n-1) + z_j^2(n). \tag{5.55}$$

Moreover, for $j = 1, 2, \ldots, m$, we define the following matrices that stem from $\mathbf{P}_j(n)$'s with the addition of appropriate diagonal terms,

$$\mathbf{R}_j(n) = \mathbf{P}_j(n) + \boldsymbol{\Gamma}_j(n-1)\mathbf{S}(n-1). \tag{5.56}$$

where $\boldsymbol{\Gamma}_j = \text{diag}([\gamma_{j1}, \gamma_{j2}, \ldots, \gamma_{jd}]^T)$. Having aptly obtained the above computationally efficient formulas, we can now head for online processing. Towards this, the equations derived for the batch case are suitably modified by incorporating the previously defined recursively computed quantities. More specifically, by substituting (5.30), (5.31) in (5.28), (5.29) respectively and using (5.48), (5.50) and (5.56) we get the following time update expressions for the entries of the subspace matrix estimate $\hat{\mathbf{U}}$ at time n,

$$\hat{u}_{ji}(n) = \beta(n-1)\sigma^2_{\hat{u}_{ji}}(n-1)\left(a_{ij}(n) - \mathbf{r}^T_{j\neg i}(n)\hat{\mathbf{u}}_{j\neg i}(n)\right), \tag{5.57}$$

$$\sigma^2_{\hat{u}_{ji}}(n) = \beta^{-1}(n-1)r^{-1}_{j,ii}(n), \tag{5.58}$$

where $a_{ij}(n)$ is the ijth entry of the $d \times m$ matrix $\mathbf{A}(n)$, $\mathbf{r}^T_{j\neg i}(n)$ is the ith row of $d \times d$ autocorrelation matrix $\mathbf{R}_j(n)$ after neglecting its ith element, i.e., $r_{j,ii}$ and finally

$$\hat{\mathbf{u}}_{j\neg i}(n) = [\hat{u}_{j1}(n), \hat{u}_{j2}(n), \ldots, \hat{u}_{ji-1}(n), \hat{u}_{ji+1}(n-1), \ldots, \hat{u}_{jd}(n-1)]^T. \tag{5.59}$$

From (5.57) and (5.83) it is readily seen that each element of the jth row of $\mathbf{U}$ is updated at each time instance n, taking into account the most recent estimates of the remaining entries of the jth row in a cyclic manner. It is worthy to mention that this emerging iterative scheme, resulting from the espoused statistical independence among the elements of $\mathbf{U}$, can be viewed as a relevant to the cyclic coordinate descent strategy [15]. Following the same premise for the column sparsity promoting parameters we get from (5.33),

$$s_i(n) = \sqrt{\frac{\beta^{-1}(n-1)\lambda_i(n)}{\hat{\boldsymbol{u}}_i^T(n)\boldsymbol{\Gamma}_i(n)\hat{\boldsymbol{u}}_i(n) + \sum_{j=1}^m \gamma_{ji}(n)\sigma^2_{\hat{u}_{ji}}(n) + q_{ii}(n)}}, \tag{5.60}$$

where $q_{ii}(n)$ is the ith diagonal element of $\mathbf{Q}(n)$. As for the hyperparameters δ_i's of the s_i's we have from (5.38), (5.40) the following recursive equation

$$\lambda_i(n) = \frac{2\mu + (1-\delta)^{-1} + m + 1}{2\nu + s_i^{-1}(n-1) + \lambda_i^{-1}(n-1)}. \tag{5.61}$$

Note that in (5.61) the size of the effective time window, i.e., $(1-\delta)^{-1}$, is used in place of n, as in [142]. For γ_{ji}'s that independently favor sparsity on the entries of the subspace matrix $\mathbf{U}$, in an online scheme (5.35) takes the form,

$$\gamma_{ji}(n) = \sqrt{\frac{\rho_{ji}(n)}{\beta(n-1)s_i(n-1)\left(\hat{u}_{ji}^2(n) + \sigma_{\hat{u}_{ji}}^2(n)\right)}} \tag{5.62}$$

and for the hyperparameters ρ_{ji}'s, (5.39) and (5.41) yield

$$\rho_{ji}(n) = \frac{2(\psi+1)}{2\xi + \gamma_{ji}^{-1}(n-1) + \rho_{ji}^{-1}(n-1)}. \tag{5.63}$$

Finally, from (5.42)-(5.45) and applying some straightforward algebraic manipulations as in [142], we end up with the following efficient formula for computing the noise precision β, at each time iteration

$$\beta(n) = \frac{2\kappa + \frac{1}{1-\delta}\left(m+d\right) + md}{\left(2\theta + \sum_{j=1}^{m}\left(d_j(n) - \hat{\mathbf{u}}_j^T(n)\boldsymbol{a}_k(n) + \boldsymbol{\sigma}_{\hat{\mathbf{u}}_j}^T(n)\mathbf{r}_j(n)\right) + \sum_{i=1}^{d} s_i(n)q_{ii}(n)\right)} \tag{5.64}$$

where $\hat{\mathbf{u}}_j^T(n)$ is the jth row of $\hat{\mathbf{U}}(n)$, $\boldsymbol{a}_j(n)$ is the jth column of $\mathbf{A}(n)$, $\boldsymbol{\sigma}_{\hat{\mathbf{u}}_j}(n) = \text{diag}(\boldsymbol{\Sigma}_{\hat{\mathbf{u}}_j}(n))$ and $\mathbf{r}_j(n) = \text{diag}(\mathbf{R}_j(n))$.

As it can be seen, most of the above defined quantities resolve to efficient time-updating formulas. In doing so, the need for taking into consideration the whole bunch of data, which is computationally prohibitive in applications dealing with big data, is eliminated. By collecting and putting in a proper order the previously derived expressions, we are led to the new online variational Bayes sparse subspace learning (OVBSL) algorithm, which is summarized in Algorithm 5.1. The algorithm provides at each time iteration not only the sought estimates $\hat{\mathbf{v}}(n)$ and $\hat{\mathbf{U}}(n)$, but also estimates for all parameters of the model described in Section 5.3.1. Note also that all these parameters are directly linked to specific distributions through the posterior inference analysis of Section 5.3.1. By carefully inspecting OVBSL in Algorithm 5.1, it can be shown that its computational complexity is $\mathcal{O}(|\boldsymbol{\omega}(n)|d^2 + md)$, where $|\boldsymbol{\omega}(n)|$ is the number of observed entries at time n. It should be emphasized that a significant reduction in the computational complexity has been achieved (which would be otherwise $\mathcal{O}(|\boldsymbol{\omega}(n)|d^3)$) by adopting the element-by-element estimation of $\hat{\mathbf{U}}$ via a coordinate descent type procedure. As shown in Algorithm 5.1, all hyperparameters of OVBSL are set and fixed to very small values at the initialization stage of the algorithm, as is the custom in sparse Bayesian learning schemes [148]. This way, prior distributions become noninformative; in line with the fact that no information for the respective parameters is a priori available[11]. Hence parameter fine tuning or cross-validation is entirely avoided and all parameters of the model are inferred from the data, rendering the proposed algorithm ideally accustomed for use in a real-time setting. In the next section the proposed algorithm is set in a unified framework with other related state-of-the-art techniques and its advantages in terms of performance and complexity

[11] Actually, since those parameters are placed in the third and the fourth level of hierarchy, their values have no crucial role on the estimation of parameters of our interest, i.e., the first level ones.

are highlighted.

5.3.4 Relation with state-of-the-art

In this section we investigate and highlight the connection of the new Bayesian algorithm with two other closely related techniques that have recently appeared in the literature, namely the PETRELS algorithm presented in [36] and Algorithm 1 of [103]. All three algorithms under study are second-order online subspace learning schemes that deal with (possibly highly) incomplete data. Out of the three schemes, only the proposed algorithm has the provision to impose sparsity to the unknown subspace matrix. Hence, to make comparisons more clear, we relax this constraint, that is we set $\mathbf{\Gamma}_i = \mathbf{I}_m$ for $i = 1, 2, \ldots, d$ in our Bayesian model described in Section 5.3.1. As we shall see below, this Bayesian model can be considered as a unified framework from which all three schemes may originate. To be more specific, let us first recall the likelihood function of the model given in (5.11), which can be expressed as

$$p(\mathbf{Z} \mid \mathbf{U}, \mathbf{V}, \beta) \propto \exp\left(-\frac{\beta}{2}\left\|(\mathbf{Z} - \mathbf{\Omega} \odot (\mathbf{U}\mathbf{V}^T))\mathbf{\Delta}^{\frac{1}{2}}\right\|_F^2\right). \tag{5.65}$$

Based on (5.65), the maximum likelihood (ML) estimator is obtained by minimizing w.r.t. **U** and **V** the negative log-likelihood, resulting in the following minimization problem[12]

$$(\text{P1}) \qquad \min_{\mathbf{U},\mathbf{V}} \frac{\beta}{2}\left\|(\mathcal{P}_\Omega(\mathbf{Y}) - \mathcal{P}_\Omega(\mathbf{U}\mathbf{V}^T))\mathbf{\Delta}^{\frac{1}{2}}\right\|_F^2. \tag{5.66}$$

The so-termed PETRELS algorithm presented in [36] solves (P1) through an online alternating (between **U** and **V**) least squares (LS) technique, which provides both the estimates of the subspace matrix $\mathbf{U}(n)$ and the new vector of projection coefficients $\mathbf{v}(n)$ at each time iteration. However, by solving (P1) PETRELS does not take any special care for revealing the true rank of the sought subspace. The algorithm starts with an overestimate d of the rank (number of columns of **U**) and the estimates returned by the algorithm are related to a subspace of rank d, which may be far from the true rank.

Let us now consider the likelihood function given in (5.65) and the first level (Gaussian) priors of **U** and **V** in our model given by (5.13) and (5.12) for $s_i = s$, $i = 1, 2, \ldots, d$, where s is a constant parameter and not a random variable that can be determined from data. Then (5.13) and (5.12) are rewritten as,

$$p(\mathbf{U} \mid s, \beta) \propto \exp\left(-\frac{\beta}{2}s\left\|\mathbf{U}\right\|_F^2\right), \tag{5.67}$$

$$p(\mathbf{V} \mid s, \beta) \propto \exp\left(-\frac{\beta}{2}s\left\|\mathbf{\Delta}^{\frac{1}{2}}\mathbf{V}\right\|_F^2\right). \tag{5.68}$$

From the likelihood (5.65) and the priors (5.67) and (5.68) the maximum a-posteriori prob-

[12] In order to retain notational consistency with Chapter 4, the incomplete data matrix **Z** is next denoted as $\mathcal{P}_\Omega(\mathbf{Y})$ in the minimization problems.

Algorithm 5.1: The OVBSL algorithm

```
Initialize: $\hat{\mathbf{U}}(0), \mathbf{S}(0), \beta(0), \boldsymbol{\Gamma}_j(0), \boldsymbol{\Sigma}_{\hat{u}_j}(0), j = 1, 2, \ldots, m$
Set $\mathbf{A}(0) = \mathbf{0}, \mathbf{P}_j(0) = \mathbf{0}, d_j(0) = 0, j = 1, 2, \ldots, m$
Set $\mu = 10^{-6}, \nu = 10^{-6}, \psi = 10^{-6}, \xi = 10^{-6}, \kappa = 10^{-6}, \theta = 10^{-6}$
Set $\mathbf{Q}(0) = \mathbf{0}, \delta$
for $n = 1, 2, \ldots$
  Get $\mathbf{z}(n), \boldsymbol{\omega}(n)$
  $\boldsymbol{\Sigma}_{\hat{\mathbf{v}}}(n) = \beta^{-1}(n-1)\left(\hat{\mathbf{U}}^T(n-1)\boldsymbol{\Omega}_n\hat{\mathbf{U}}(n-1) + \sum_{j=1}^m \omega_j(n)\boldsymbol{\Sigma}_{\hat{\mathbf{u}}_j}(n-1) + \mathbf{S}(n-1)\right)^{-1}$
  $\hat{\mathbf{v}}(n) = \beta(n-1)\boldsymbol{\Sigma}_{\hat{\mathbf{v}}}(n)\hat{\mathbf{U}}^T(n-1)\mathbf{z}(n)$
  $\boldsymbol{\Sigma}(n) = \boldsymbol{\Sigma}_{\hat{\mathbf{v}}}(n) + \hat{\mathbf{v}}^T(n)\hat{\mathbf{v}}(n)$
  $\mathbf{Q}(n) = \delta\mathbf{Q}(n-1) + \boldsymbol{\Sigma}(n)$
  $\mathbf{A}(n) = \delta\mathbf{A}(n-1) + \hat{\mathbf{v}}(n)\mathbf{z}^T(n)$
  for $j = 1, 2, \ldots, m,$
   $\mathbf{P}_j(n) = \delta\mathbf{P}_j(n-1) + \omega_j(n)\boldsymbol{\Sigma}(n)$
   $\mathbf{R}_j(n) = \mathbf{P}_j(n) + \boldsymbol{\mathit{\Gamma}}_j(n-1)\mathbf{S}(n-1)$
   $d_j(n) = \delta d_j(n-1) + z_j^2(n)$
   for $i = 1, 2, \ldots, d,$
     $\hat{u}_{ji}(n) = \beta(n-1)\sigma_{\hat{u}_{ji}}(n-1)\left(a_{ij}(n) - \mathbf{r}_{j\neg i}^T(n)\hat{\mathbf{u}}_{j\neg i}(n)\right)$
     $\sigma_{\hat{u}_{ji}}^2(n) = \beta^{-1}(n-1)r_{j,ii}^{-1}(n)$
     $\rho_{ji}(n) = \frac{2(\psi+1)}{2\xi+\gamma_{ji}^{-1}(n-1)+\rho_{ji}^{-1}(n-1)}$
     $\gamma_{ji}(n) = \sqrt{\frac{\rho_{ji}(n)}{\beta(n-1)s_i(n-1)\left(\hat{u}_{ji}^2(n)+\sigma_{\hat{u}_{ji}}^2(n)\right)}}$
   end
   Set $\boldsymbol{\Sigma}_{\hat{\mathbf{u}}_j}(n) = \text{diag}\left([\sigma_{\hat{u}_{j1}}^2(n), \sigma_{\hat{u}_{j2}}^2(n), \ldots, \sigma_{\hat{u}_{jd}}^2(n)]^T\right)$
  end
  for $i = 1, 2, \ldots, d,$
   $\lambda_i(n) = \frac{2\mu+(1-\delta)^{-1}+m+1}{2\nu+s_i^{-1}(n-1)+\lambda_i^{-1}(n-1)}$
   $s_i(n) = \sqrt{\frac{\beta^{-1}(n-1)\lambda_i(n)}{\hat{\mathbf{u}}_i^T(n)\boldsymbol{\Gamma}_i(n)\hat{\mathbf{u}}_i(n)+\sum_{j=1}^m \gamma_{ji}(n)\sigma_{\hat{u}_{ji}}^2(n)+q_{ii}(n)}}$
   end
  Set $\mathbf{S}(n) = \text{diag}([s_1(n), s_2(n), \ldots, s_d(n)]^T)$
  $\beta(n) = \frac{2\kappa+\frac{1}{1-\delta}(m+d)+md}{\left(2\theta+\sum_{j=1}^d\left(d_j(n)-\hat{\mathbf{u}}_j^T(n)\mathbf{a}_j(n)+\boldsymbol{\sigma}_{\hat{\mathbf{u}}_j}^T(n)\mathbf{r}_j(n)\right)+\sum_{i=1}^d s_i(n)q_{ii}(n)\right)}$
end
```

ability (MAP) estimator of **U** and **V** defined by solving the following minimization problem,

$$\min_{\mathbf{U},\mathbf{V}} \{-\ln p(\mathbf{U},\mathbf{V} \mid \mathbf{Z})\} \equiv \min_{\mathbf{U},\mathbf{V}} \{-\ln [p(\mathbf{Z} \mid \mathbf{U},\mathbf{V},\beta) p(\mathbf{U} \mid s,\beta) p(\mathbf{V} \mid s,\beta)]\}, \tag{5.69}$$

is expressed as,

$$(\text{P2}) \qquad \min_{\mathbf{U},\mathbf{V}} \frac{\beta}{2}\left[\left\|(\mathcal{P}_\Omega(\mathbf{Y}) - \mathcal{P}_\Omega(\mathbf{U}\mathbf{V}^T))\boldsymbol{\Delta}^{\frac{1}{2}}\right\|_F^2 + s\,\|\mathbf{U}\|_F^2 + s\left\|\boldsymbol{\Delta}^{\frac{1}{2}}\mathbf{V}\right\|_F^2\right].$$

The minimization problem (P2) is at the heart of the analysis in [103]. Algorithm 1 of [103] is a second-order alternating ridge regression type scheme that solves (P2) sequentially and provides estimates of $\mathbf{U}(n)$ and $\mathbf{v}(n)$ at each time iteration. In [103], to promote the low-rank data representation, the minimization problem is originally formulated as

$$(\text{P2}') \qquad \min_{\mathbf{X}} \beta\left[\frac{1}{2}\left\|(\mathcal{P}_\Omega(\mathbf{Y}) - \mathcal{P}_\Omega(\mathbf{X}))\boldsymbol{\Delta}^{\frac{1}{2}}\right\|_F^2 + s\left\|\mathbf{X}\boldsymbol{\Delta}^{\frac{1}{2}}\right\|_*\right].$$

Then, in search for a nuclearhis-norm surrogate that would be amenable to online processing, the nuclear norm $||\mathbf{X}\boldsymbol{\Delta}^{\frac{1}{2}}||_*$ in $(\text{P2}')$ is replaced by its upper bound $(||\boldsymbol{\Delta}^{\frac{1}{2}}\mathbf{V}||_F^2 + ||\mathbf{U}||_F^2)/2$, with $\mathbf{X} = \mathbf{U}\mathbf{V}^T$, thus leading to (P2). Even though, compared to PETRELS, a more direct promotion of the low-rankness of the underlying subspace is employed in [103], again an overestimate d of the true rank is used and Algorithm 1 of [103] lacks a specific mechanism for imposing low-rankness explicitly by reducing the initial rank to the true rank as the algorithm evolves. In addition, special care should be taken for the parameter s that must be properly selected and updated in the framework of an online scheme.

Let us, finally, employ the complete Bayesian model of Section 5.3.1 (with the exception of the subspace matrix sparsity promoting parameters γ_{ji}'s which are set to 1). In such a case, as shown in Section 5.6, the joint prior of **U** and **V** can be expressed as

$$p(\mathbf{U},\mathbf{V} \mid \boldsymbol{\delta},\beta) \propto \exp\left(-\beta^{\frac{1}{2}}\sum_{i=1}^{d}\lambda_i^{\frac{1}{2}}\left(\|\mathbf{u}_i\|_2^2 + \|\mathbf{v}_i\|_{2,\boldsymbol{\Delta}}^2\right)^{\frac{1}{2}}\right). \tag{5.70}$$

From (5.65) and (5.70) the MAP estimator for **U** and **V** is now obtained from the solution of the following minimization problem,

$$(\text{P3}) \qquad \min_{\mathbf{U},\mathbf{V}}\left[\frac{\beta}{2}\left\|\left(\mathcal{P}_\Omega(\mathbf{Y}) - \mathcal{P}_\Omega(\mathbf{U}\mathbf{V}^T)\right)\boldsymbol{\Delta}^{\frac{1}{2}}\right\|_F^2 + \beta^{\frac{1}{2}}\sum_{i=1}^{d}\lambda_i^{\frac{1}{2}}\left(\|\boldsymbol{u}_i\|_2^2 + \|\boldsymbol{v}_i\|_{2,\boldsymbol{\Delta}}^2\right)^{\frac{1}{2}}\right]. \tag{5.71}$$

Note that the regularizing summation term in (P3) corresponds to the *weighted* $\ell_{1,2}$ norm of the matrix $[\mathbf{U}^T \;\; (\boldsymbol{\Delta}^{\frac{1}{2}}\mathbf{V})^T]^T$ [89], which is known to impose column sparsity [89] and thus explicitly reducing the rank of **U**, leading to more consistent estimates. Interestingly, this term is actually a generalized version of the low-rank imposing term introduced in Chapter 4. More specifically, assuming $\beta^{\frac{1}{2}}\lambda_i^{\frac{1}{2}} = \lambda, \forall\, i = 1,2,\ldots,d$, and $\boldsymbol{\Delta} = \mathbf{I}_n$ (5.71) boils down to (4.24) with $p = 1$.

Derived from the Bayesian model of Section 5.3.1, the minimization problem $(\mathrm{P3})$ is closely related to the analysis and the OVBSL algorithm presented above. It should be emphasized though that OVBSL is not a recursive alternating MAP estimation scheme, but a variational Bayes type technique that can be deemed as a generalization of the MAP approach. While a MAP procedure would provide the point estimates of the parameters of interest **U** and **V**, the proposed algorithm returns in addition the approximate distributions of all parameters involved in the model, including the weighting parameters λ_i's, which are now estimated directly from the data. Summarizing and compared to [36] and [103] the proposed algorithm a) is equipped with an inherent mechanism for inducing column sparsity and thus reducing the rank of the latent subspace matrix dynamically and b) is fully automatic as all parameters of the model are estimated from the data and thus any need for preselection (using heuristics) or fine tuning is entirely avoided.

5.4 Cost function minimization based algorithm

Departing from the Bayesian approach, a deterministic algorithm is next presented for addressing the low-rank subspace learning problem. The developed algorithm actually solves a simplified version of the minimization problem (P3) stated in (5.71).

More specifically, the proposed minimization problem consists of two terms: a) an exponentially weighted LS term which accounts for the fitting between the data and the model and b) a *smoothed* special instance of the weighted $\ell_{1,2}$ norm described above, arising by setting $\beta^{\frac{1}{2}}\lambda_i^{\frac{1}{2}} = \lambda, \forall\ i = 1, 2, \ldots, d$, in the second term of the cost function in Eq. (5.71), i.e.,

$$\begin{aligned}\{\hat{\mathbf{v}}(n), \hat{\mathbf{U}}(n)\} = \min_{\mathbf{v}(n), \mathbf{U}(n)} & \sum_{t=1}^{n} \delta^{n-t} \|\mathcal{P}_{\Omega_t}(\boldsymbol{y}(t)) - \mathcal{P}_{\Omega_t}\left(\mathbf{U}(n)\mathbf{v}(t)\right)\|_2^2 \\ & + \lambda \sum_{i=1}^{d} \sqrt{\boldsymbol{v}_i^T(n)\boldsymbol{\Delta}(n)\boldsymbol{v}_i(n) + \|\boldsymbol{u}_i(n)\|_2^2 + \eta^2}. \end{aligned} \tag{5.72}$$

Hence in (5.72) λ is a low-rank regularization parameter and η^2 is small positive constant which is used for smoothing purposes (see Section 4.3.3). It can be easily observed that (5.72) is actually an *online* version of the cost function (4.13) proposed in Chapter 4 for the *batch* matrix completion problem for $p = 1$.

It is worthy to underline that albeit the first term of (5.72) decouples over both the rows and the columns of $\mathbf{U}(n)$ and $\mathbf{V}(n)$, this is not the case with the second low-rank promoting term. Hence, getting closed form expressions for estimating $\mathbf{v}(n)$ and $\mathbf{U}(n)$ at time n is rendered infeasible, while also the lack of such a decoupling seems to hinder the derivation of an online scheme. However, as we show below by adopting an alternating minimization strategy that combines a regularized LS step for the updating of $\mathbf{v}(n)$ followed by cyclic coordinate descent type steps for the (columns of) $\mathbf{U}(n)$, an efficient online subspace learning algorithm can be obtained.

Defining the $d \times d$ diagonal matrix $\mathbf{D}(n)$ with diagonal entries

$$d_i(n) = \frac{\lambda}{\sqrt{\hat{\boldsymbol{v}}_i^T(n)\boldsymbol{\Delta}(n)\hat{\boldsymbol{v}}_i(n) + \|\hat{\boldsymbol{u}}_i(n)\|_2^2 + \eta^2}}, \tag{5.73}$$

we first minimize (5.72) w.r.t. $\mathbf{v}(n)$ and get an *approximate* closed-form solution[13] for $\hat{\mathbf{v}}(n)$, i.e.,

$$\hat{\mathbf{v}}(n) = \left(\hat{\mathbf{U}}^T(n-1)\boldsymbol{\Omega}_n\hat{\mathbf{U}}(n-1) + \mathbf{D}(n-1)\right)^{-1}\hat{\mathbf{U}}(n-1)^T\mathcal{P}_{\Omega_n}(\boldsymbol{y}(n)). \tag{5.74}$$

Next we adopt a block coordinate descent (BCD) type minimization of (5.72) w.r.t. the columns of $\mathbf{U}(n)$, [15], which results to the following expression for the estimate of its jith element[14],

$$\hat{u}_{ji}(n) = \left(\sum_{t=1}^{n}\delta^{n-t}\omega_j(t)\hat{v}_i^2(t) + d_i(n-1)\right)^{-1}\sum_{t=1}^{n}\delta^{n-t}\hat{v}_i(t)\times$$
$$\left(\omega_j(t)y_j(t) - \omega_j(t)\left(\sum_{t'<t}\hat{v}_{i'}(t)\hat{u}_{ji'}(n) + \sum_{t'>t}\hat{v}_{i'}(t)\hat{u}_{ji'}(n-1)\right)\right). \tag{5.75}$$

In (5.74), (5.75) there is a subtle point that should be accentuated: both $\hat{\mathbf{v}}(n)$ and $\hat{\mathbf{U}}(n)$ aside from their inherent interrelation shown in (5.74) and (5.75), involve the matrix $\mathbf{D}(n-1)$ which in turn includes quantities depending on $\hat{\boldsymbol{v}}_i(n-1)$ and $\hat{\boldsymbol{u}}_i(n-1)$. Hence, it arises that $\hat{\mathbf{v}}(n)$, estimated at time n, is also influenced by the projection coefficient vectors estimated in the previous time instants. A similar situation arises for the estimate $\hat{\mathbf{U}}(n)$ of the subspace matrix, which due to the presence of $\mathbf{D}(n-1)$ in (5.75) also relies on its estimate obtained in the previous time instant. It should be noted that this particular characteristic of our method is a consequence of the aforementioned *nondecoupling* nature of the low-rank regularizing term utilized.

Turning now to the online route, we need to express the previous updating equations, in terms of quantities whose size remains fixed as n increases. To this end, we observe that Eq. (5.75) can be written more compactly and avoid time-increasing summation terms by incorporating appropriate time-updating formulas in the same vein to the ones presented in Section 5.3.3. Specifically, we define the following fixed size w.r.t. time quantities

$$\mathbf{A}(n) = \hat{\mathbf{V}}^T(n)\boldsymbol{\Delta}(n)\mathbf{Z}^T(n), \tag{5.76}$$
$$\mathbf{P}_j(n) = \hat{\mathbf{V}}^T(n)\boldsymbol{\Delta}(n)\boldsymbol{\Omega}_j(n)\hat{\mathbf{V}}(n), \quad j = 1, 2, \ldots m, \tag{5.77}$$
$$q_i(n) = \hat{\boldsymbol{v}}_i^T(n)\boldsymbol{\Delta}(n)\hat{\boldsymbol{v}}_i(n). \quad i = 1, 2, \ldots d, \tag{5.78}$$

[13] The exact (yet not closed-form) expression for $\hat{\mathbf{v}}(n)$ contains $\mathbf{D}(n)$ in place of $\mathbf{D}(n-1)$ in Eq. (5.74).

[14] Note that in (5.75) we consider a single iteration of the BCD procedure and map BCD iterations to time iterations.

It is easy to verify that $\mathbf{A}(n)$, $\mathbf{P}_j(n)$ and $q_i(n)$ in (5.76), (5.77), (5.78), respectively, can be easily expressed time-recursively as

$$\begin{aligned} \mathbf{A}(n) &= \delta\mathbf{A}(n-1) + \hat{\mathbf{v}}(n)\boldsymbol{z}^T(n), && (5.79)\\ \mathbf{P}_j(n) &= \delta\mathbf{P}_j(n-1) + \omega_j(n)\hat{\mathbf{v}}(n)\hat{\mathbf{v}}^T(n), && (5.80)\\ q_i(n) &= \delta q_i(n-1) + \hat{v}_i^2(n). && (5.81) \end{aligned}$$

The term $\hat{\boldsymbol{v}}_i^T(n)\boldsymbol{\Delta}(n)\hat{\boldsymbol{v}}_i(n)$ which appears in the expression of $d_i(n)$ (5.73), coincides with $q_i(n)$ and thus it is efficiently computed via (5.81). Following the same path, (5.75) is rewritten by integrating the above-defined quantities yielding

$$\hat{u}_{ji}(n) = \frac{a_{ij}(n) - \mathbf{p}_{j\neg i}^T(n)\hat{\mathbf{u}}_{j\neg i}(n)}{p_{j,ii}(n) + d_i(n-1)}, \tag{5.82}$$

where $a_{ij}(n)$ denotes the ijth entry of the $d \times m$ matrix $\mathbf{A}(n)$, $\mathbf{p}_{j\neg i}^T(n)$ is the ith row of the $d \times d$ autocorrelation matrix $\mathbf{P}_j(n)$ after ignoring its ith entry $p_{j,ii}(n)$, and finally

$$\hat{\mathbf{u}}_{j\neg i}(n) = [\hat{u}_{j1}(n), \hat{u}_{j2}(n), \ldots, \hat{u}_{ji-1}(n), \hat{u}_{ji+1}(n-1), \ldots, \hat{u}_{jd}(n-1)]^T. \tag{5.83}$$

The proposed online column sparsity promoting subspace learning algorithm (OCSpSL) from incomplete data is summarized in Algorithm 5.2.

Algorithm 5.2: The OCSpSL algorithm

```
  Initialize $\mathbf{U}(0)$, $\mathbf{D}(0)$
 Set $\mathbf{A}(0) = \mathbf{0}$, $\mathbf{q}(0) = \mathbf{0}, \delta$
 for $n = 1, 2, \ldots$
   $\hat{\mathbf{v}}(n) = \left(\hat{\mathbf{U}}^T(n-1)\boldsymbol{\Omega}_n\hat{\mathbf{U}}(n-1) + \mathbf{D}(n-1)\right)^{-1}\hat{\mathbf{U}}(n-1)^T\mathcal{P}_{\Omega_n}(\boldsymbol{y}(n))$
   Update $\mathbf{q}(n), \mathbf{D}(n)$ from (5.81) and (5.73)
   Update $\mathbf{A}(n)$ from (5.79)
   for $j = 1, 2, \ldots, m$
     Update $\mathbf{P}_j(n)$ from (5.80)
     for $i = 1, 2, \ldots, d$
       Compute $\hat{u}_{ji}(n) = \frac{a_{ij}(n) - \mathbf{p}_{j\neg i}^T(n)\hat{\mathbf{u}}_{j\neg i}(n)}{p_{j,ii}(n) + d_i(n-1)}$
     end
   end
 end
```

In Table 11, OVBSL and OCSpSL are compared in terms of computational complexity and memory storage requirements with other related state-of-the-art algorithms. Besides PETRELS and Algorithm 1 of [103] mentioned above, two other algorithms are included, namely GROUSE reported in [9] and Algorithm 2 of [103], which is a first-order stochastic approximation type scheme. We see from Table 11 that the proposed algorithms require

Table 11: Computational complexity and memory storage requirements of online subspace learning algorithms.

Algorithm	GROUSE [9]	PETRELS [36]	Alg. 1 of [103]	Alg. 2 of [103]	OVBSL	OCSpSL
Comp. complexity	$\mathcal{O}(\lvert\omega(n)\rvert d^2 + md)$	$\mathcal{O}(\lvert\omega(n)\rvert d^2)$	$\mathcal{O}(\lvert\omega(n)\rvert d^3)$	$\mathcal{O}(\lvert\omega(n)\rvert d^2 + md)$	$\mathcal{O}(\lvert\omega(n)\rvert d^2 + md)$	$\mathcal{O}(\lvert\boldsymbol{\omega}(n)\rvert d^2)$
Memory requirements	$\mathcal{O}(md)$	$\mathcal{O}(md^2)$	$\mathcal{O}(md^2)$	$\mathcal{O}(md)$	$\mathcal{O}(md^2)$	$\mathcal{O}(md^2)$

less computations per iteration than Algorithm 1 of [103], while they have similar complexity with the remaining three algorithms. Note though that, as it will be also shown in the next section, PETRELS and GROUSE perform well under the condition that the true subspace rank $r(n)$ is known, while Algorithm 2 of [103], being a first-order algorithm is expected to have a much slower convergence rate compared to the remaining second-order schemes included in Table 11. With regard to memory requirements, both OVBSL and OCSpSL demand more storage space compared to the rest state-of-the art algorithms, yet at the same order of magnitude with PETRELS and Algorithm 1 of [103]. As expected, lower memory storage is required by the first-order methods namely GROUSE and Algorithm 2 of [103].

5.5 Experimental Results

In this section, the effectiveness of the proposed OVBSL and OCSpSL algorithms is evaluated in a variety of experiments carried out on synthetic and real data.

5.5.1 Synthetic data experiments

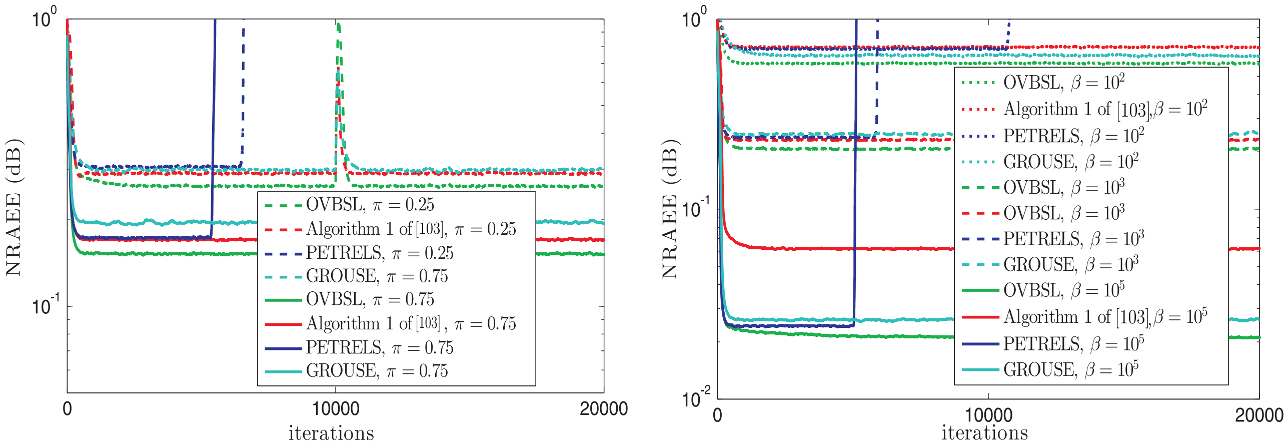
In the following, two different experiments are presented. Our first goal is to illustrate the efficiency of OVBSL and OCSpSL in tackling matrix completion. It should be noted that the sparsity imposition on the subspace matrix from OVBSL is purposely relaxed in this experiment, that is we set $\mathbf{\Gamma}_i = \mathbf{I}_m, \forall i = 1, 2, \ldots, d$. The performance of OVBSL in the challenging *sparse* subspace estimation problem is explored in the second experiment of this subsection. Therein, the aforementioned favorable characteristic of OVBSL algorithm, i.e., its potential to impose sparsity on the subspace matrix, is thoroughly investigated. To this end, the parameters γ_{kl}'s are then considered “active”, normally taking their values according to the full Bayesian model analytically described in Section 5.3.3.

5.5.1.1 Online matrix completion

In order to assess the performance of OVBSL algorithm in recovering missing data, we simulate a low dimensional subspace $\boldsymbol{\mathcal{U}} \in \mathcal{R}^{m \times r}$ with $m = 500$ and $r = 5$ and Gaussian i.i.d. entries $u_{ji} \sim \mathcal{N}(0, \frac{1}{m})$. Next, 20000 $r \times 1$ projection coefficient vectors $\mathbf{c}(n)$ are produced according to a Gaussian distribution $c_i(n) \sim \mathcal{N}(0, 1)$. The signal $\boldsymbol{y}(n)$ at time n is then generated by the product $\boldsymbol{\mathcal{U}}\mathbf{c}(n)$, it is normalized so that its power is equal to 1, and then contaminated by i.i.d. Gaussian noise $\mathbf{e}(n) \sim \mathcal{N}(\mathbf{0}, \beta^{-1}\mathbf{I}_m)$. To model the missing entries, we randomly select a fraction π of the entries from each datum $\boldsymbol{y}(n)$, which are assumed to be known, whereas the rest $(1-\pi) \times 100$% of the elements are considered to be missing. To show the merits of the proposed OVBSL algorithm, we compare it to three state-of-the-art techniques, namely GROUSE with greedy step-size, [164], PETRELS [36] and Algorithm 1 of [103]. It is worthy to mention that, as also previously mentioned, both GROUSE and PETRELS hinge on the assumption that the rank of the underlying subspace is known. Contrary, Algorithm 1 of [103] utilizes ℓ_2 norm regularization as described in the previous section, that robustifies the algorithm in the absence of this knowledge. Finally, the true standard deviation of the noise is provided as input for adaptively estimating the step-size of GROUSE while the low-rank regularization parameter of Algorithm 1 of [103] is set to 0.1 as is proposed in the relevant paper.

Moreover, to make things more interesting, we adhere to the challenging but realistic scenario whereby the true rank of the underlying subspace is unknown. Along this line, the rank of the subspace matrix is accordingly initialized in all tested algorithms to an overestimate of the true rank, namely $d = 10$. Our initial objective is to demonstrate the effectiveness of the proposed OVBSL algorithm when certain amounts of data are missing. To this end, we carry out two experiments corresponding to different fractions of the observed entries, i.e., $\pi = \{0.25, 0.75\}$, keeping the noise precision β fixed to 10^3. Since the competence of the subspace learning algorithms in tracking possible changes of the sought subspace is of crucial importance in many applications, an abrupt change of the subspace is induced at $n = 10000$ for $\pi = 0.25$. The performance of the tested algorithms is evaluated in terms of the normalized running average estimation error (NRAEE) defined as: NRAEE$(n) = \frac{1}{100}\sum_{t=n-99}^{n} \frac{\|\hat{\boldsymbol{y}}(t)-\boldsymbol{y}(t)\|_2}{\|\boldsymbol{y}(t)\|_2}$ where $\hat{\boldsymbol{y}}(t) = \hat{\mathbf{U}}(t)\hat{\mathbf{v}}(t)$. The average NRAEE of 10 independent runs of the experiment is shown in Fig. 37a. It is clear that the proposed OVBSL algorithm outperforms its rivals for both values of the fraction of the observed data π. At the same time, OVBSL is proven to be competent in tracking sudden changes of the latent subspace, since the transient deterioration of its performance caused by the deliberate change induced at $n = 10000$ is swiftly rectified in the subsequent iterations. Notably, in the lack of knowledge of the true rank of the subspace, PETRELS becomes unstable. Contrary, Algorithm 1 of [103] and GROUSE with the greedy step-size scheme present a robust behavior (note that GROUSE is given the true standard deviation of the noise for updating its step-size), though with clearly less reconstruction accuracy compared to the proposed OVBSL algorithm.

Based on the same experimental setting, OCSpSL is next compared with PETRELS [36] and Algorithm 1 of [103]. The low-rank regularization parameter of OCSpSL is set to 0.1. The algorithms are now tested for two different percentages of missing (at random) data, i.e., $\pi = \{0.5, 0.8\}$ and the initial rank of the subspace matrices is now set to $d = 20$. As is shown in Fig. 38, OCSpSL exhibits a robust behavior, outperforming its rivals in terms of NRAEE for both values of π, while PETRELS diverges after a number of iterations. Interestingly, OCSpSL besides its robustness and improved estimation performance, it is also able to retrieve the real subspace rank. After convergence most of the columns of $\hat{\mathbf{U}}(n)$ are zero and its rank is 5. On the contrary, Algorithm 1 of [103] ends up without



(a) (b)

Figure 37: Performance comparison among OVBSL, Algorithm 1 of [103], PETRELS and GROUSE, [164], for the matrix completion problem. (a) Robustness to different fractions of the observed entries (π) (b) Sensitivity to different levels of noise corruption.

decreasing the initially set rank value.

Next, we examine the robustness of OVBSL to noise corruption. To do so, we keep the fraction of the observed entries fixed to $\pi = 0.4$, focusing on the behavior of OVBSL and the competing schemes for three different values of the noise precision, i.e., $\beta = \{10^5, 10^3, 10^2\}$. Fig. 37b depicts the average NRAEE of 10 executions of the experiment obtained by the tested algorithms in the three different cases examined. It is easily noticed that herein as well, OVBSL achieves higher reconstruction accuracy than the competing schemes for all different β's, thus corroborating its robustness to various levels of noise corruption.

5.5.1.2 Online sparse subspace estimation

In the following, the compelling feature of OVBSL to favor *sparse* subspace estimates is thoroughly explored. To clearly demonstrate the merits of this key aspect of the algorithm, a sparse subspace matrix $\mathcal{U}$ of rank $r = 5$ is modeled. Then, the same above-described process is adopted for producing 20000 projection coefficient vectors $\mathbf{c}(n)$, that give rise to the corresponding signals $\mathcal{U}\mathbf{c}(n)$. Finally, Gaussian i.i.d. noise of precision $\beta = 10^3$ is assumed to contaminate the data. For now, focusing on the subspace matrix estimation problem, we depart from the matrix completion problem considering that data are fully observed (hence the fraction of the observed entries π equals to 1) and we test two versions of OVBSL, that is, when sparsity of the subspace a) is taken into account and b) is disregarded in the same way explained earlier and the greedy step-size version of GROUSE, [164]. The estimates of the subspace are assessed as time evolves by means of the normalized subspace reconstruction error (NSRE) defined as NSRE$(n) = \frac{\|\mathcal{P}_{\hat{\mathbf{U}}^{\perp}(n)}\mathcal{U}\|_F}{\|\mathcal{U}\|_F}$[15]. The

[15] $\mathcal{P}_{\hat{\mathbf{U}}^{\perp}(n)}\mathcal{U}$ denotes the projection of the true subspace matrix $\mathcal{U}$ to the orthogonal complement of the subspace spanned by the columns of the estimated subspace matrix $\hat{\mathbf{U}}^{\perp}(n)$.

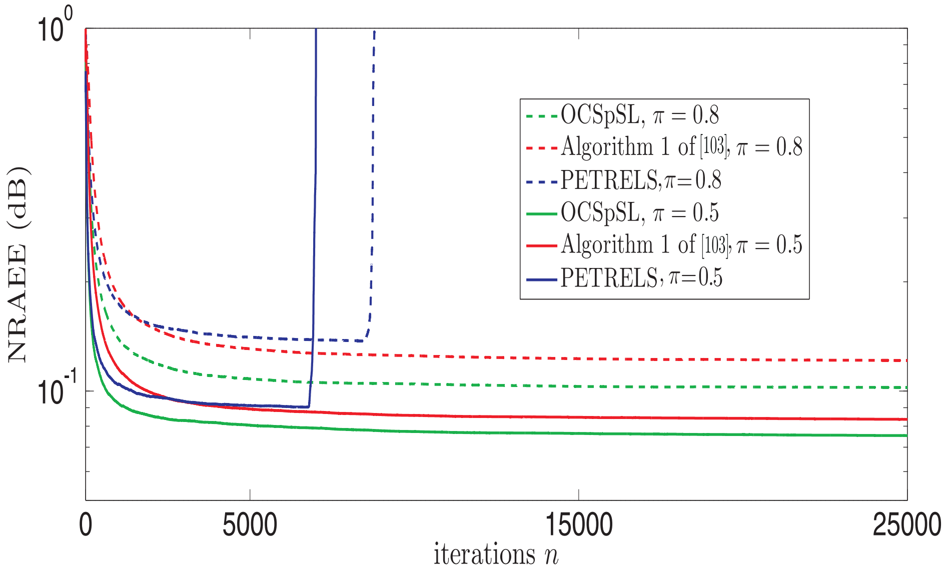


Figure 38: NRAEE obtained by OCSpSL, PETRELS and Algorithm 1 of [103] on the simulated matrix completion problem.

benefits emerging from taking into account the sparsity existing in the unknown subspace matrix, come to light by exploring OVBSL's performance for different levels of sparsity imposed on it, namely 0.7 and 0.9. In both cases, the subspace matrices are initialized to an overestimate of the rank, i.e., $d = 10$. Fig. 39a depicts the mean NSRE of 10 runs of the experiment obtained for the two versions of OVBSL and GROUSE as time evolves. As it can be readily seen, OVBSL achieves subspace estimates of higher accuracy compared to both its so to speak non-sparse version and GROUSE which, likewise, does not favor sparsity on the subspace matrix. It should be noted that the gains obtained by the sparse OVBSL are becoming abundantly clear as the sparsity level increases.

Next, OVBSL and GROUSE are probed in the challenging problem of sparse subspace estimation from partially observed data. Towards this, the same experimental setting described above is followed and two cases corresponding to two different combinations of sparsity level and fraction of observed entries are examined, namely a) sparsity-level=0.7 and $\pi = 0.75$ and b) sparsity-level=0.9 and $\pi = 0.5$. OVBSL is again evaluated for the two cases corresponding to its sparse and non-sparse version and GROUSE is also tested, initializing the rank d of subspace matrices to 5 and using NSRE as the performance metric. From Fig. 39b, it is verified that albeit data are incomplete, sparse OVBSL outperforms both its nonsparse version and GROUSE thus corroborating that taking advantage of the sparsity of the subspace matrix is still meaningful when the assumption of sparse subspace is valid.

5.5.2 Real data experiments

In this section we focus on the efficiency of OVBSL algorithm on real data experiments. More concretely, we conduct two different experiments corresponding a) to hyperspec-

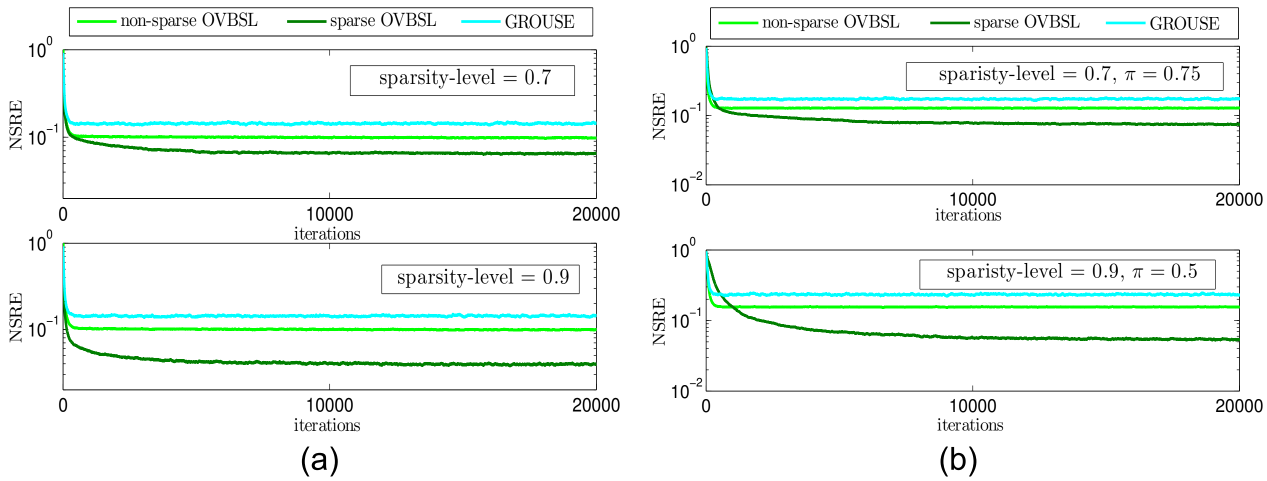


Figure 39: Performance comparison between sparse and nonsparse versions of OVBSL and GROUSE, [164]. (a) Robustness to different sparsity levels of the subspace matrix and $\pi = 1$ (b) Robustness to different percentages of missing entries and subspace sparsity levels.

tral image reconstruction out of partially observed measurements and b) to the eigenface learning problem. In both experiments OVBSL is compared with the state-of-the-art Algorithm 1 of [103] whose low-rank regularization parameter takes its value according to the heuristic rule that was also followed on the real data experiments considered in [103].

5.5.2.1 Pixel-by-pixel hyperspectral image recovery

As mentioned in Section 2.2 a key characteristic of HSIs is the high degree of correlation they present, both in the spectral and the spatial domains, [67]. Given a HSI, let us form a matrix with its columns corresponding to the pixels of the HSI, and its rows to the spectral bands. In doing so, it can be easily seen that the underlying high coherence appearing both in columns and rows leads to a matrix that may be of very low rank, as compared to its dimensions. Actually, this fact gives us good grounds for exploiting the low-rank structure in favor of recovering HSIs, in cases that data either are partly missing or have suffered by severe noise corruption. In the following, we test the performance of OVBSL and Algorithm 1 of [103] in recovering the Salinas Valley HSI, [67], out of a fraction $\pi = 0.2$ of its entries. Since both algorithms process data in an online fashion, we assume that the aforementioned time instances, hereafter, correspond to a sequence of all the pixels taken in a random order from the image. Put differently, the algorithms process the pixel spectral signatures (which are the columns of the formed matrix) one-by-one, as if they were becoming available in a streaming fashion. Notably, this type of processing aside from reducing the computational complexity, it alleviates the need for memory storage, thus paving the way for on-board processing. The rank of the subspace matrix is initialized to $d = 10$. To quantitatively assess the performance of the tested algorithms, we estimate the NRAEE (Fig. 40a), as the number of the processed pixels increases, and the structural similarity (SSIM) index values, [157], between the true and the reconstructed band images

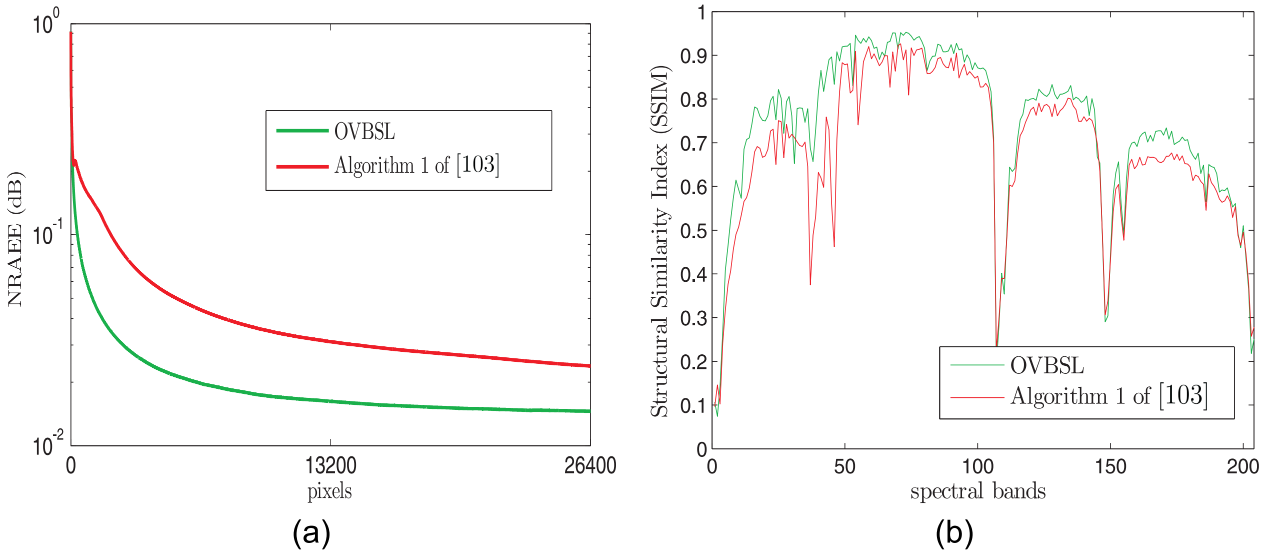
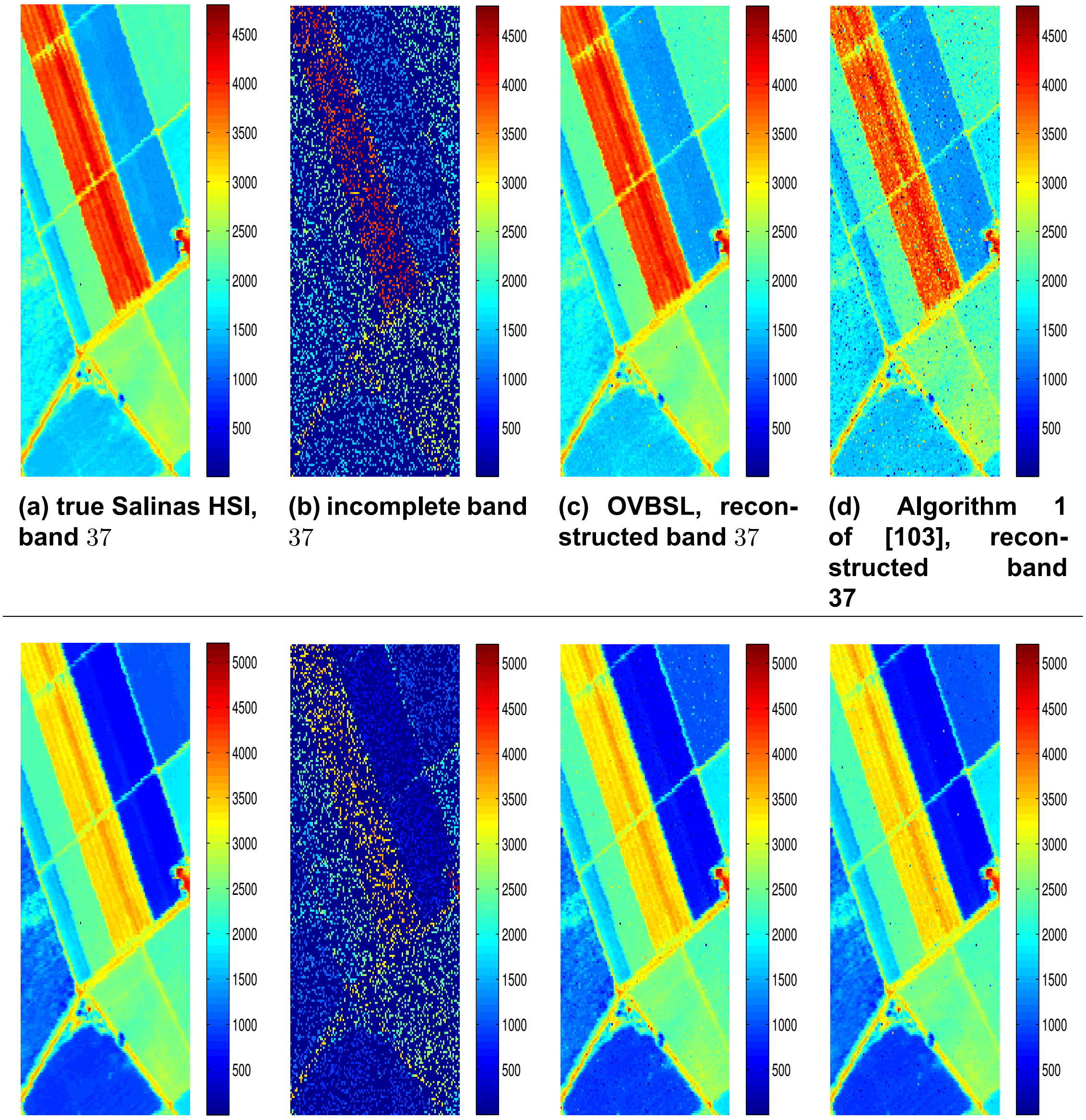


Figure 40: Performance comparison between OVBSL and Algorithm 1 of [103] in terms of NRAEE and SSIM. (a) NRAEE as the number of processed pixel increases (b) SSIM index per reconstructed band.

(Fig. 40b). It is clearly shown in Fig. 40a that OVBSL achieves higher reconstruction accuracy on average as compared to its rival in terms of NRAEE. Focusing on Fig. 40b, it can be noticed that OVBSL presents higher SSIMs in the majority of the spectral bands, with only few exceptions of bands, where the SSIM indexes obtained by OVBSL and Algorithm 1 of [103], either take close values or the latter gets slightly greater values, e.g., at band 31. In order to give further insight at the reconstructed bands, we provide in Figs. 41a, 41e, the true bands 31 and 37, respectively, accompanied by their incomplete versions (Figs. 41b, 41f) that were provided as inputs to the tested algorithms. From Figs 41c and 41d, it can be easily observed that, in good agreement with the SSIMs of the two algorithms at this band, OVBSL reconstructs the 37th band of Salinas HSI in remarkably higher accuracy than Algorithm 1 of [103]. As regards band 31 where the SSIM index of Algorithm 1 of [103] is slightly higher that that of OVBSL, Figs. 41g 41h show that the reconstructed images are quite similar for both algorithms. That said, OVBSL is favorably proven to be competent in processing this real HSI dataset, outperforming the state-of-the-art Algorithm 1 of [103].

5.5.2.2 Online eigenface learning

In this section, we qualitatively evaluate the performance of sparse OVBSL as compared to the nonsparse Algorithm 1 of [103] on another real dataset. Towards this, we use the MIT-CBCL face dataset [137], which contains $n = 2429$ face images of size 19×19 pixels. The tested algorithms process the images as m-dimensional vectors with $m = 361 (= 19^2)$, in an online fashion. The subspace matrix estimated by both algorithms can be deemed as a *learned dictionary* of faces. In doing so, each image can be reconstructed by a linear combination of the atoms (eigenfaces) contained in the subspace matrix. The rank of the



(a) true Salinas HSI, band 37

(b) incomplete band 37

(c) OVBSL, reconstructed band 37

(d) Algorithm 1 of [103], reconstructed band 37

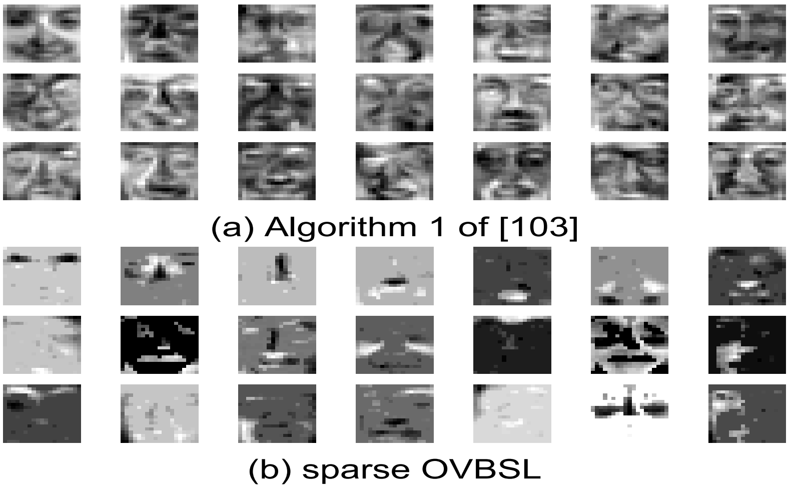
(e) true Salinas HSI, band 31

(f) incomplete band 31

(g) OVBSL, reconstructed band 31

(h) Algorithm 1 of [103], reconstructed band 31

Figure 41: Reconstruction of Salinas Valley HSI by OVBSL and Algorithm 1 of [103], for $\pi = 0.2$**.**



(a) Algorithm 1 of [103]

(b) sparse OVBSL

Figure 42: Eigenfaces obtained by Algorithm 1 of [103] and OVBSL on MIT-CBCL dataset.

subspace is initialized for both algorithms to 50. Fig. 42 shows the 21 more characteristic eigenfaces. Dark pixels correspond to negative values, while positive values are denoted with light colors. As it can be noticed, sparsity imposition from the sparse OVBSL leads to eigenfaces that present more localized features, contrary to those obtained by Algorithm 1 of [103], where features are spread out over the image. It should be also noted that OVBSL converged to a subspace matrix of low-rank. This fact resulted from the inherent advantageous characteristic of OVBSL to eliminate components presenting low variance, hence offering negligible information.

5.6 Appendix - MAP estimator of the proposed Bayesian model

The joint prior of **U** and **V** is expressed as

$$p(\mathbf{U}, \mathbf{V} \mid \boldsymbol{\lambda}, \beta, \boldsymbol{\Gamma}) = \prod_{i=1}^{d} p(\boldsymbol{v}_i, \boldsymbol{u}_i \mid \lambda_i, \beta, \boldsymbol{\Gamma}_i) \tag{5.84}$$

where

$$p(\boldsymbol{v}_i, \boldsymbol{u}_i \mid \lambda_i, \beta, \boldsymbol{\Gamma}_i) = \int_0^{\infty} p(\boldsymbol{v}_i \mid s_i, \beta) p(\boldsymbol{u}_i \mid s_i, \beta, \boldsymbol{\Gamma}_i) p(s_i \mid \lambda_i) ds_i \tag{5.85}$$

Using (5.13), (5.12) and (5.15) in (5.85) yields

$$p(\boldsymbol{v}_i, \boldsymbol{u}_i \mid \lambda_i, \beta, \boldsymbol{\Gamma}_i) = \int_0^{\infty} (2\pi)^{-\frac{n+m}{2}} \beta^{\frac{n+m}{2}} |\boldsymbol{\Delta}\boldsymbol{\Gamma}_i|^{\frac{1}{2}} s_i^{-\frac{3}{2}} \exp\left(-\frac{\beta s_i}{2} (\|\boldsymbol{u}_i\|_{2,\boldsymbol{\Gamma}_i}^2 + \|\boldsymbol{v}_i\|_{2,\boldsymbol{\Delta}}^2) - \frac{\lambda_i}{2 s_i} \right) ds_i, \tag{5.86}$$

where $\|\boldsymbol{v}_i\|^2_{2,\boldsymbol{\Delta}} = \boldsymbol{v}_i^T\boldsymbol{\Delta}\boldsymbol{v}_i$ and $\|\boldsymbol{u}_i\|^2_{2,\boldsymbol{\Gamma}_i} = \boldsymbol{u}_i^T\boldsymbol{\Gamma}_i\boldsymbol{u}_i$. Using in (5.86) the expression of the GIG distribution for s_i with parameters $\beta(\|\boldsymbol{u}_i\|^2_{2,\boldsymbol{\Gamma}_i} + \|\boldsymbol{v}_i\|^2_{2,\boldsymbol{\Delta}}), \lambda_i$ and $-1/2$, we easily get

$$\begin{aligned}p(\boldsymbol{v}_i, \boldsymbol{u}_i \mid \lambda_i, \beta, \boldsymbol{\Gamma}_i) =&(2\pi)^{-\frac{n+m}{2}}\beta^{\frac{n+m}{2}}|\boldsymbol{\Delta\Gamma}_i|^{\frac{1}{2}}2\mathcal{K}_{-\frac{1}{2}}\left(\beta^{\frac{1}{2}}\lambda_i^{\frac{1}{2}}(\|\boldsymbol{u}_i\|^2_{2,\boldsymbol{\Gamma}_{\boldsymbol{i}}} + \|\boldsymbol{v}_i\|^2_{2,\boldsymbol{\Delta}})^{\frac{1}{2}}\right)\\ &\times\left(\frac{\beta(\|\boldsymbol{u}_i\|^2_{2,\boldsymbol{\Gamma}_i} + \|\boldsymbol{v}_i\|^2_{2,\boldsymbol{\Delta}})}{\lambda_i}\right)^{\frac{1}{4}}\end{aligned} \tag{5.87}$$

where $\mathcal{K}_{-\frac{1}{2}}(\cdot)$ denotes the modified Bessel function of second kind with p degrees of freedom. By employing the identity,

$$\mathcal{K}_{-\frac{1}{2}}(x) = \left(\frac{\pi}{2x}\right)^{\frac{1}{2}}\text{exp}(-x)$$

in (5.87) and after some straightforward calculations, we end up with the following expression for the joint distribution of $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$,

$$p(\boldsymbol{v}_i, \boldsymbol{u}_i \mid \lambda_i, \beta, \boldsymbol{\Gamma}_i) = (2\pi)^{-\frac{n+m-1}{2}}\beta^{\frac{n+m}{2}}|\boldsymbol{\Delta\Gamma}_i|^{\frac{1}{2}}\lambda_i^{-\frac{1}{2}}\,\text{exp}\left(-\beta^{\frac{1}{2}}\lambda_i^{\frac{1}{2}}(\|\boldsymbol{u}_i\|^2_{2,\boldsymbol{\Gamma}_i} + \|\boldsymbol{v}_i\|^2_{2,\boldsymbol{\Delta}})^{\frac{1}{2}}\right). \tag{5.88}$$

Then, from (5.84)

$$\begin{aligned}p(\mathbf{U}, \mathbf{V} \mid \boldsymbol{\lambda}, \beta, \boldsymbol{\Gamma}) =&(2\pi)^{-\frac{(n+m-1)L}{2}}\beta^{\frac{(n+m)d}{2}}|\boldsymbol{\Delta}|^{\frac{d}{2}}\left(\prod_{i=1}^{d}\lambda_i^{\frac{1}{2}}|\boldsymbol{\Gamma}_i|^{\frac{1}{2}}\right)\\ &\times\text{exp}\left(-\beta^{\frac{1}{2}}\sum_{i=1}^{d}\lambda_i^{\frac{1}{2}}(\|\boldsymbol{u}_i\|^2_{2,\boldsymbol{\Gamma}_i} + \|\boldsymbol{v}_i\|^2_{2,\boldsymbol{\Delta}})^{\frac{1}{2}}\right)\end{aligned} \tag{5.89}$$

which is a multi-parameter (with respect to the λ_i's) Laplace-type distribution defined on the columns of the matrix $[(\boldsymbol{\Gamma}\odot\mathbf{U})^T \quad (\boldsymbol{\Delta}^{1/2}\mathbf{V})^T]^T$. Such a distribution is known to impose column sparsity and thus, due to the form of the matrix, joint column sparsity on **U** and **V**.

6. CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

In the present thesis we addressed three different, omnipresent in the literature structured matrix estimation problems. Common denominator of all the proposed techniques is the nonconvex nature of the formulated estimation tasks. No doubt, nonconvex methods, in contrast to convex ones, deprive theoretical guarantees as to their performance and convergence to global minima. However, empirical evidence has shown that they exhibit very promising results in many practical situations involving large-scale datasets. In the sequel, we summarize the concluding remarks of the thesis. First we recapitulate the contribution of the thesis for each of the three problems addressed by highlighting not only the specific characteristics of the introduced formulations but also the efficient algorithms that have been devised. Then, directions for further research relevant to the content of the present thesis are provided.

6.1 Summary

This thesis touched upon structured matrix estimation as is met in three different ubiquitous signal processing and machine learning tasks, i.e., a) simultaneously sparse, low-rank and nonnegative matrix estimation, b) sparse, low-rank and nonnegative matrix factorization and c) online low-rank subspace learning and matrix completion. The contribution of the thesis lies in both the novel mathematical formulation of these three problems that were introduced, as well as in the derivation of pioneering optimization and Bayesian inference algorithms for solving them. It should be noted that a certain degree of innovation also exists in expressing ubiquitous estimation tasks such as hyperspectral unmixing and denoising in terms of the aforementioned structured matrix representations.

Let us first focus on simultaneously sparse, low-rank and nonnegative matrix estimation. Broadly speaking, multiple structured estimation tasks are in their very infancy, yet. In the literature, the associated originally NP-hard problem has been seen via a convex relaxation perspective. However, theoretical studies have highlighted the existing gap in terms of sample complexity and recovery performance between convex and nonconvex approaches. Inspired by that, in the present thesis we introduced two different nonconvex formulations of the problem. First, capitalizing on the merits of a) the reweighted ℓ_1 norm when it comes to sparse recovery and b) the reweighted nuclear norm in low-rank matrix estimation, we proposed to combine these two norms and devise a new constrained optimization problem that promotes both sparsity and low-rankness. In addition, the nonnegativity constraint was also incorporated in the problem. The second approach, instead of the reweighted nuclear norm, it uses the variational form of the nuclear norm for imposing low-rankness. This way, the SVD operations required in the case of reweighted nuclear norm minimization are avoided, thus relieving us from a heavy computational load, especially in large-scale data. The resulting optimization problems were attacked by three different optimization solvers: a) an incremental proximal minimization algorithm, b) an alternating direction method of multipliers based algorithm and c) a block coordinate descent type algorithm. The proposed formulations and algorithms were then utilized for hyperspectral image unmixing. It is noted that the constraints of sparsity, low-rankness and nonnegativity can be all adopted as structures of the abundance matrices in hyperspectral images, which are characterized by high degree of spatial and spectral correlation. Extended simulated data experiments have shown the superior performance of the proposed algorithms as compared to solely sparse and solely low-rank approaches, as well as to a convex simultaneously sparse, low-rank and nonnegative matrix estimation scheme. Moreover, the algorithms have been applied on real hyperspectral imaging data in the framework of the hyperspectral unmixing problem. The relative experiments have exposed the favorable performance of the new algorithms over other state-of-the-art methods, thus verifying the effectiveness of the proposed approaches in dealing with large-scale data applications.

The second technical contribution of the thesis is related to low-rank matrix factorization (LRMF). LRMF has attracted considerable attention in recent years as it appears in several machine learning tasks such as low-rank matrix estimation, matrix completion, dictionary learning, etc. LRMF is an inherently nonconvex problem suffering from an intrinsic flaw: in most real data applications the dimension (rank) of the matrix factors is unknown. In the thesis we followed a commonly used practice to circumvent this shortcoming, i.e., we assumed an overestimate of the inner dimension of the factorization and then penalized the rank of the matrix factors. In this regard, we innovated by proposing a new low-rank promoting regularizer, which can be viewed as a weighted version of the variational form of the nuclear norm. Interestingly, all existing relevant approaches can be considered as special instances of the proposed low-rank promoting function, upon suitably selecting the weighting coefficients. Going one step further, we suggested the use of a common reweighting diagonal matrix on both matrix factors. In doing so, low-rankness is imposed by promoting jointly column-sparsity on both matrix factors. To show the broad utility of the proposed low-rank promoting penalty, we incorporated it to four different problems, i.e., denoising, matrix completion, nonnegative matrix factorization and sparse dictionary learning. The resulting optimization problems were addressed borrowing ideas from the block successive upper bound minimization framework that led us to quasi-Newton type algorithmic schemes. Interestingly, the adopted upper bounds result to matrixwise updates even in the matrix completion problem, thus offering significant computational savings. The derived alternating iteratively reweighted least squares-type algorithms can be viewed as extensions of the popular iteratively reweighted schemes (proposed for sparse recovery) to the low-rank minimization framework. An analysis showing the convergence of these algorithms to stationary points of the functions associated with the denoising and matrix completion formulations was also provided. The new algorithms were tested in a wealth of simulated experiments, as well as in disparate real data applications such as hyperspectral image denoising, matrix completion on recommender systems, music signal decomposition and unsupervised hyperspectral unmixing. Therein, it was shown that the proposed algorithms exhibit favorable results concerning both the estimation performance and the required runtime.

In the third part of the thesis we addressed an estimation problem which follows a different learning model, i.e., online low-rank subspace learning and matrix completion. Online learning has revived nowadays since it constitutes an indispensable tool in large-scale data processing. In this thesis, we presented two approaches to tackle this problem, one Bayesian and one deterministic. As far as the first is concerned, we proposed a novel Bayesian formulation of sparse and low-rank subspace learning, while also accounting for missing data. More specifically, three-level hierarchical prior distributions were assigned on the columns of both the subspace matrix and the coefficients matrix. The multi-hierarchical priors were suitably parameterized so as to promote jointly column-sparsity on the two matrices and independently promote sparsity on the subspace matrix. This way, problems such as online sparse PCA could be, for the first time in literature, addressed following a Bayesian perspective. In order to carry out the inference, the variational Bayes scheme was employed based on the mean-field approximation. In addition, by assuming statistical independence among all the elements of the subspace matrix, matrix inversions were avoided in the updates of the subspace matrix, thus leading to a significant decrease of computational complexity. Next, time-update formulas were defined and utilized for transforming the batch type updates to online ones giving thus rise to a new online variational Bayes subspace learning algorithm. Similar ideas described for the Bayesian algorithm were also utilized in the derivation of the deterministic cost function minimization based scheme. Concretely, joint-column sparsity in that case was promoted by suitably modifying the low-rank promoting term utilized for the low-rank MF problem to the online scenario. Again, elementwise updates were derived for the subspace matrix following a Gauss-Seidel approach. Both algorithms were tested on extensive simulated data experiments showing significant merits, as compared to relevant state-of-the-art competing algorithms, in terms of the estimation performance. Moreover, the application of the new online algorithms on hyperspectral image denoising and eigenface learning verified their competence in dealing with real and large-scale data.

6.2 Future research directions

By now the low-rank matrix factorization ideas presented in Chapter 4 have already been extended and applied for tensor completion, [150], and robust PCA, [62]. In both those cases, it was shown that the low-rank penalty introduced (in the frame of the thesis) offers significant gains in other cutting-edge machine learning problems. Next we provide an insight on other possible future directions of the research conducted in the thesis. These directions span all the three problems that have been studied, namely, simultaneously sparse, low-rank and nonnegative matrix estimation, low-rank matrix factorization and online low-rank subspace learning and matrix completion.

Concerning the problem of simultaneously sparse, low-rank and nonnegative matrix estimation, it is of high interest to further explore some theoretical aspects of the newly introduced reweighted ℓ_1 and nuclear norms as well the variational nuclear norm based formulation. As is shown, in both cases estimation performance is improved over convex approaches in regression type problems. However, the theoretical sample complexity of the introduced scheme under a specific deterministic linear mapping (i.e., the known endmembers' matrix which was used in hyperspectral unmixing), has not been established yet. Moreover, it deserves to investigate, both theoretically and experimentally, the gains in using the proposed nonconvex formulation in cases that either random mappings (e.g., subgaussian) or other deterministic ones (e.g., discrete wavelet transforms - DWT) are utilized. Finally, a rigorous convergence analysis of the derived algorithms would be very important, yet quite challenging since we deal with nonconvex problems.

The second direction concerns the generalized version of the variational form of the nuclear norm that was introduced for low-rank matrix factorization. A theoretical understanding of this low-rank promoting mechanism is of high priority in our future research. In particular, it would be very interesting to explore how this regularization term is related to either the Schatten-p norm or the weighted nuclear norm. Moreover, as analytically described in the thesis, low-rank matrix factorization has attracted great attention lately. Interestingly, a large part of the reported research concerns the theoretical characteristics of the problem. Focusing on the variational form of the nuclear norm, recent studies, [167, 18], have shown that despite nonconvexity of the problem, global convergence can be attained since there are no spurious local minima. It is thus quite intriguing to explore if similar premises hold for the low-rank promoting term proposed in the thesis. Finally, gains obtained by using this regularizer in terms of sample complexity and error bounds under various experimental settings shall also be studied in a future work.

Finally, a third future direction is related to the online variational Bayes low-rank subspace learning and matrix completion algorithm presented in Chapter 5. A very appealing topic of research is to generalize the introduced online variational Bayes scheme, by finding connections with the relevant online scheme of [125]. Following the natural gradient-type formulation of the latter, a second-order quasi-Newton and stochastic variational Bayes framework may be derived. In doing so, the online variational Bayes scheme introduced in the thesis, may be further extended to generic large-scale machine learning problems such as topic modelling, offering an efficient alternative to the existing first-order type scheme. Lastly, convergence of the online cost function minimization based algorithm might be established by leveraging ideas followed in [101].

ABBREVIATIONS

Abbreviation	Meaning
SVD	Singular Value Decomposition
LRMF	Low-Rank Matrix Factorization
BCD	Block Coordinate Descent
BSUM	Block Successive Upper Bound Minimization
MAP	Maximum A Posteriori
EM	Expectation Maximization
ELBO	Evidence Lower Bound
VB	Variational Bayes
IPSpLRU	Incremental Proximal Sparse and Low-Rank Unmixing
ADSpLRU	Alternating Direction Sparse and Low-Rank Unmixing
ALMSpLRU	ALternating Minimization Sparse and Low-Rank Unmixing
AIRLS	Alternating Iteratively Reweighted Least-Squares
NMF	Nonnegative Matrix Factorization
MC	Matrix Completion
OVBSL	Online VB Subspace Learning
OCSpSL	Online Column Sparse Subspace Learning
Bi-ICE	Bayesian Inference Iterative Conditional Expectations
HSI	HyperSpectral Image
HU	Hyperspectral Unmixing

NOTATION

Symbol	Meaning
x	Scalar
$\|x\|$	Absolute value of a scalar
$\mathbf{x}$	Vector
$\mathbf{0}$	Zero vector/matrix
$\mathbf{1}$	All ones vector/matrix
$\mathbf{X}$	Matrix
$\mathbf{X}^T$	Transpose of matrix X
$\mathbf{I}_l$	$l \times l$ identity matrix
$\|\|\mathbf{x}\|\|_p$	ℓ_p norm of vector $\mathbf{x}$
$\|\|\mathbf{X}\|\|_p$	ℓ_p norm of matrix $\mathbf{X}$
$\|\|\mathbf{X}\|\|_{\mathcal{S}_p}$	Schatten-p norm of matrix $\mathbf{X}$
supp($\mathbf{X}$)	Support set of matrix $\mathbf{X}$
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$
$\mathcal{GIG}(x; p, a, b)$	Generalized inverse Gaussian distribution defined for $x > 0$, with parameters $a > 0$, $b > 0$ and p a real number
$\mathcal{R}$	Field of real numbers
$\mathcal{R}^{m \times n}$	$m \times n$-dimensional space of real numbers
$\approx$	Approximately equal
$\odot$	Elementwise multiplication

REFERENCES

[1] https://bit.ly/2linkdo.

[2] https://bit.ly/2tr8jd6.

[3] H. Abou-Kandil. *Matrix Riccati equations: in control and systems theory*. Springer Science & Business Media, 2003.

[4] M. Aharon, M. Elad, and A. Bruckstein. rmK-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006.

[5] S-I. Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.

[6] S. D. Babacan, M. Luessi, R. Molina, and A. K. Katsaggelos. Sparse Bayesian methods for low-rank matrix estimation. *IEEE Transactions on Signal Processing*, 60(8):3964–3977, Aug. 2012.

[7] F. Bach, R. Jenatton, J. Mairal, G. Obozinski, et al. Convex optimization with sparsity-inducing norms. *Optimization for Machine Learning*, pages 19–53, 2011.

[8] F. Bach, J. Mairal, and J. Ponce. Convex sparse matrix factorizations. *arXiv preprint arXiv:0812.1869*, 2008.

[9] L. Balzano, R. Nowak, and B. Recht. Online identification and tracking of subspaces from highly incomplete information. In *48th Annual Allerton Conference on Communication, Control, and Computing*, Sept. 2010.

[10] L. Balzano and S. J. Wright. Local convergence of an algorithm for subspace identification from partial data. *Foundations of Computational Mathematics*, 15(5):1279–1314, 2015.

[11] D. Bannon. Hyperspectral imaging: Cubes and slices. *Nature Photonics*, 3(11):627, 2009.

[12] J. D. Bayliss, J. A. Gualtieri, and R. F. Cromp. Analyzing hyperspectral data with independent component analysis. In *26th AIPR Workshop: Exploiting New Image Sources and Sensors*, 1998.

[13] A. Beck. On the convergence of alternating minimization for convex programming with applications to iteratively reweighted least squares and decomposition schemes. *SIAM Journal on Optimization*, 25(1):185–209, 2015.

[14] D. P Bertsekas. Projected Newton methods for optimization problems with simple constraints. *SIAM Journal on Control and Optimization*, 20(2):221–246, 1982.

[15] D. P. Bertsekas. *Nonlinear Programming*. Athena scientific Belmont, 1999.

[16] D. P. Bertsekas. Incremental gradient, subgradient, and proximal methods for convex optimization: A survey. *Optimization for Machine Learning*, 2010(3):1–38, 2011.

[17] D. P. Bertsekas. Incremental proximal methods for large scale convex optimization. *Mathematical Programming*, 129(2):163–195, 2011.

[18] S. Bhojanapalli, B. Neyshabur, and N. Srebro. Global optimality of local search for low rank matrix recovery. In *Advances in Neural Information Processing Systems*, 2016.

[19] J. M. Bioucas-Dias and M. Figueiredo. Alternating direction algorithms for constrained sparse regression: application to hyperspectral unmixing. In *2nd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (IEEE WHISPERS)*, June 2010.

[20] J. M. Bioucas-Dias and J. MP. Nascimento. Hyperspectral subspace identification. *IEEE Transactions on Geoscience and Remote Sensing*, 46(8):2435–2445, 2008.

[21] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

[22] L. Bottou. Online learning and stochastic approximations. *Online Learning in Neural Networks*, 17(9):142, 1998.

[23] T. Bouwmans and E. H. Zahzah. Robust PCA via principal component pursuit: A review for a comparative evaluation in video surveillance. *Computer Vision and Image Understanding*, 122:22–34, 2014.

[24] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

[25] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[26] J. Bunch, C. Nielsen, and D. Sorensen. Rank-one modification of the symmetric eigenproblem. *Numerische Mathematik*, 31(1):31–48, 1978.

[27] D. Cai, X. He, and J. Han. Spectral regression: A unified approach for sparse subspace learning. In *IEEE International Conference on Data Mining (ICDM)*, 2007.

[28] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):11, 2011.

[29] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

[30] E. J. Candes, M. B. Wakin, and S. P Boyd. Enhancing sparsity by reweighted ℓ_1 minimization. *Journal of Fourier Analysis and Applications*, 14(5-6):877–905, 2008.

[31] A. T. Cemgil. *A Tutorial Introduction to Monte Carlo methods, Markov Chain Monte Carlo and Particle Filtering*. Elsevier Major Reference Works, 2012.

[32] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

[33] R. Chartrand. Exact reconstruction of sparse signals via nonconvex minimization. *IEEE Signal Processing Letters*, 14(10):707–710, 2007.

[34] R. Chartrand and W. Yin. Iteratively reweighted algorithms for compressive sensing. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2008.

[35] F. Chen and Y. Zhang. Sparse hyperspectral unmixing based on constrained $\ell_p - \ell_2$ optimization. *IEEE Geoscience and Remote Sensing Letters*, 10(5):1142–1146, 2013.

[36] Y. Chi, Y.C. Eldar, and R. Calderbank. PETRELS: Parallel subspace estimation and tracking by recursive least squares from partial observations. *IEEE Transactions on Signal Processing*, 61(23):5947–5959, 2013.

[37] S. Chouvardas, Y. Kopsinis, and S. Theodoridis. Robust subspace tracking with missing entries: The set-theoretic approach. *IEEE Transactions on Signal Processing*, 63(19):5060–5070, Oct. 2015.

[38] R. N. Clark, G. A. Swayze, R. Wise, K. E. Livo, T. M. Hoefen, R. F. Kokaly, and S. J. Sutley. USGS digital spectral library, 2007. `http://speclab.cr.usgs.gov/spectral.lib06/ds231/datatable.html`.

[39] P. L. Combettes and J-C. Pesquet. Proximal splitting methods in signal processing. In *Fixed-point algorithms for Inverse Problems in Science and Engineering*. Springer, 2011.

[40] A. d'Aspremont, L. E. Ghaoui, M. I. Jordan, and G. R. Lanckriet. A direct formulation for sparse PCA using semidefinite programming. In *Advances in Neural Information Processing Systems*, 2005.

[41] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk. Iteratively reweighted least squares minimization for sparse recovery. *Communications on Pure and Applied Mathematics*, 63(1):1–38, 2010.

[42] C. de Boor. *Elementary numerical analysis*. McGraw-Hill, 1972.

[43] W. Dong, G. Shi, X. Li, Y. Ma, and F. Huang. Compressive sensing via nonlocal low-rank regularization. *IEEE Transactions on Image Processing*, 23(8):3618–3632, Aug 2014.

[44] X. G. Doukopoulos and G. V. Moustakides. Fast and stable subspace tracking. *IEEE Transactions on Signal Processing*, 56(4):1452–1465, Apr. 2008.

[45] O. Eches, N. Dobigeon, and J. Tourneret. Enhancing hyperspectral image unmixing with spatial correlations. *IEEE Transactions on Geoscience and Remote Sensing*, 49(11):4239–4247, Nov 2011.

[46] J. Eckstein and D. P. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1-3):293–318, 1992.

[47] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, et al. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

[48] M. Elad. *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Springer, 2010.

[49] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, Dec. 2006.

[50] S. Erard and W. Calvin. New composite spectra of Mars, 0.4–5.7 μm. *Icarus*, 130(2):449–460, 1997.

[51] M. Fazel. *Matrix rank minimization with applications*. PhD thesis, Stanford University, 2002.

[52] M. Fornasier, H. Rauhut, and R. Ward. Low-rank matrix recovery via iteratively reweighted least squares minimization. *SIAM Journal on Optimization*, 21(4):1614–1640, 2011.

[53] S. Foucart and H. Rauhut. *A mathematical introduction to compressive sensing*. Birkhäuser Basel, 2013.

[54] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*. Springer, 2001.

[55] X. Fu, W-K. Ma, J. Bioucas-Dias, and T-H. Chan. Semiblind hyperspectral unmixing in the presence of spectral library mismatches. *arXiv preprint arXiv:1507.01661*, 2015.

[56] D. Ge, X. Jiang, and Y. Ye. A note on the complexity of lp minimization. *Mathematical Programming*, 129(2):285–299, 2011.

[57] R. Ge, C. Jin, and Y. Zheng. No spurious local minima in nonconvex low rank problems: A unified geometric analysis. In *34th International Conference on Machine Learning*, 2017.

[58] P. V. Giampouras, A. A. Rontogiannis, and K. D. Koutroumbas. Online low-rank subspace learning from incomplete data using rank revealing ℓ_2/ℓ_1 regularization. In *IEEE Statistical Signal Processing Workshop (SSP)*, Palma de Mallorca, June 2016.

[59] P. V. Giampouras, A. A. Rontogiannis, and K. D. Koutroumbas. ℓ_1/ℓ_2 regularized nonconvex low-rank matrix factorization. In *Signal Processing with Adaptive Sparse Structured Representations (SPARS)*, Lisbon, June 2017.

[60] P. V Giampouras, A. A Rontogiannis, and K. D Koutroumbas. Low-rank and sparse nmf for joint endmembers' number estimation and blind unmixing of hyperspectral images. In *25th European Signal Processing Conference (EUSIPCO)*, Kos Island, Sept. 2017.

[61] P. V. Giampouras, A. A. Rontogiannis, and K. D. Koutroumbas. Alternating iteratively reweighted minimization algorithms for low-rank matrix factorization. *(under review) IEEE Transactions on Signal Processing*, 2018.

[62] P. V. Giampouras, A. A. Rontogiannis, and K. D. Koutroumbas. Robust PCA via alternatingly iteratively reweighted low-rank matrix factorization. In *IEEE International Conference on Image Processing (ICIP)*, Athens, Oct. 2018.

[63] P. V. Giampouras, A. A. Rontogiannis, K. D. Koutroumbas, and K. E. Themelis. A sparse reduced-rank regression approach for hyperspectral image unmixing. In *3rd International Workshop on Compressed Sensing Theory and its Applications to Radar, Sonar and Remote Sensing (CoSeRa)*, Pisa, June 2015.

[64] P. V. Giampouras, A. A. Rontogiannis, K. E. Themelis, and K. D. Koutroumbas. Online Bayesian low-rank subspace learning from partial observations. In *23rd European Signal Processing Conference (EUSIPCO)*, Nice, Sept. 2015.

[65] P. V. Giampouras, A. A. Rontogiannis, K. E. Themelis, and K. D. Koutroumbas. Online sparse and low-rank subspace learning from incomplete data: A Bayesian view. *Signal Processing*, 137:199 – 212, 2017.

[66] P. V. Giampouras, K. E. Themelis, A. A. Rontogiannis, and K. D. Koutroumbas. Hyperspectral image unmixing via silultaneously sparse and low rank abundance matrix estimation. In *7th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (IEEE WHISPERS)*, Tokyo, June 2015.

[67] P. V. Giampouras, K. E. Themelis, A. A. Rontogiannis, and K. D. Koutroumbas. Simultaneously sparse and low-rank abundance matrix estimation for hyperspectral image unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 54(8):4775–4789, 2016.

[68] P. V. Giampouras, K. E. Themelis, A. A. Rontogiannis, and K. D. Koutroumbas. Structured abundance matrix estimation for land cover hyperspectral image unmixing. *Compressive Sensing of Earth Observations, CRC Press*, 2017.

[69] G. B. Giannakis, R. Cendrillon, V. Cevher, A. Swami, and Z. Tian. Introduction to the issue on signal processing for big data. *IEEE Journal of Selected Topics in Signal Processing*, 9(4):583–585, June 2015.

[70] N. Gillis et al. Nonnegative matrix factorization: Complexity, algorithms and applications. *Unpublished doctoral dissertation, Université catholique de Louvain. Louvain-La-Neuve: CORE*, 2011.

[71] G. Golub, S. Nash, and C. Van Loan. A Hessenberg-Schur method for the problem AX + XB= C. *IEEE Transactions on Automatic Control*, 24(6):909–913, Dec. 1979.

[72] G. Golub and C. Van Loan. *Matrix Computations*. JHU Press, 2012.

[73] P. Gong and C. Zhang. Efficient nonnegative matrix factorization via projected Newton method. *Pattern Recognition*, 45(9):3557–3565, 2012.

[74] S. Gu, L. Zhang, W. Zuo, and X. Feng. Weighted nuclear norm minimization with application to image denoising. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[75] S. F. Gull and G. J. Daniell. Image reconstruction from incomplete and noisy data. *Nature*, 272(20):686–690, 1978.

[76] B. Haeffele, E. Young, and R. Vidal. Structured low-rank matrix factorization: Optimality, algorithm, and applications to image processing. In *International Conference on Machine Learning*, 2014.

[77] T. Hastie, R. Mazumder, J. D. Lee, and R. Zadeh. Matrix completion and low-rank SVD via fast alternating least squares. *Journal of Machine Learning Research*, 16:3367–3402, 2015.

[78] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

[79] M. Hong, M. Razaviyayn, Z-Q. Luo, and J-S. Pang. A unified algorithmic framework for block-structured optimization involving big data: with applications in machine learning and signal processing. *IEEE Signal Processing Magazine*, 33(1):57–77, 2016.

[80] P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004.

[81] M-D. Iordache, J. M. Bioucas-Dias, and A. Plaza. Sparse unmixing of hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 49(6):2014–2039, 2011.

[82] M-D. Iordache, J. M. Bioucas-Dias, and A. Plaza. Collaborative sparse regression for hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 52(1):341–354, 2014.

[83] P. Jain, P. Netrapalli, and S. Sanghavi. Low-rank matrix completion using alternating minimization. In *45th annual ACM symposium on Theory of Computing*. ACM, 2013.

[84] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach. Proximal methods for sparse hierarchical dictionary learning. In *International Conference on Machine Learning*, 2010.

[85] R. Jenatton, G. Obozinski, and F. Bach. Structured sparse principal component analysis. In *International Conference on Artificial Intelligence and Statistics*, 2010.

[86] I. T. Jolliffe. Principal component analysis and factor analysis. In *Principal Component Analysis*. Springer, 1986.

[87] N. Keshava and J. F. Mustard. Spectral unmixing. *IEEE Signal Processing Magazine*, 19(1):44–57, 2002.

[88] Y-D. Kim and S. Choi. Variational Bayesian view of weighted trace norm regularization for matrix factorization. *IEEE Signal Processing Letters*, 20(3):261–264, March 2013.

[89] M. Kowalski. Sparse regression using mixed norms. *Applied and Computational Harmonic Analysis*, 27(3):303 – 324, 2009.

[90] Y. Langevin, F. Poulet, J. Bibring, and B. Gondet. Sulfates in the north polar region of Mars detected by OMEGA/Mars Express. *Science*, 307(5715):1584–1586, 2005.

[91] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*, 2001.

[92] J. D. Lee, Y. Sun, and M. A. Saunders. Proximal Newton-type methods for minimizing composite functions. *SIAM Journal on Optimization*, 24(3):1420–1443, 2014.

[93] J. Li and J.M. Bioucas-Dias. Minimum volume simplex analysis: a fast algorithm to unmix hyperspectral data. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, July 2008.

[94] Q. Li, Z. Zhu, and G. Tang. Geometry of factored nuclear norm regularization. *arXiv preprint arXiv:1704.01265*, 2017.

[95] C. Lu, J. Tang, S. Yan, and Z. Lin. Generalized nonconvex nonsmooth low-rank minimization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[96] C. Lu, J. Tang, S. Yan, and Z. Lin. Nonconvex nonsmooth low rank minimization via iteratively reweighted nuclear norm. *IEEE Transactions on Image Processing*, 25(2):829–839, 2016.

[97] X. Lu, H. Wu, and Y. Yuan. Double constrained NMF for hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 52(5):2746–2758, 2014.

[98] X. Lu, H. Wu, Y. Yuan, P. Yan, and X. Li. Manifold regularized sparse nmf for hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 51(5):2815–2826, 2013.

[99] W. K. Ma, J. M. Bioucas-Dias, T. H. Chan, N. Gillis, P. Gader, A. J. Plaza, A. Ambikapathi, and C. Y. Chi. A signal processing perspective on hyperspectral unmixing: Insights from remote sensing. *IEEE Signal Processing Magazine*, 31(1):67–81, 2014.

[100] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:19–60, 2010.

[101] J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. *IEEE Transactions on Image Processing*, 17(1):53–69, 2008.

[102] M. Mardani, G. Mateos, and G. B. Giannakis. Dynamic anomalography: Tracking network anomalies via sparsity and low rank. *IEEE Journal of Selected Topics in Signal Processing*, 7(1):50–66, 2013.

[103] M. Mardani, G. Mateos, and G. B. Giannakis. Subspace learning and imputation for streaming big data matrices and tensors. *IEEE Transactions on Signal Processing*, 63(10):2663–2677, 2015.

[104] S. Mei, B. Cao, and J. Sun. Encoding low-rank and sparse structures simultaneously in multi-task learning. *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[105] L. Miao and H. Qi. Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization. *IEEE Transactions on Geoscience and Remote Sensing*, 45(3):765–777, 2007.

[106] T. P. Minka. Expectation propagation for approximate Bayesian inference. In *17th Conference in Uncertainty in Artificial Intelligence*, San Francisco, CA, USA, 2001.

[107] K. Mohan and M. Fazel. Iterative reweighted algorithms for matrix rank minimization. *Journal of Machine Learning Research*, 13:3441–3473, 2012.

[108] E. A. Mylona, O. A. Sykioti, K. D. Koutroumbas, and A. A. Rontogiannis. Joint spectral unmixing and clustering for identifying homogeneous regions in hyperspectral images. In *IEEE International Conference Geoscience and Remote Sensing Symposium (IGARSS)*, July 2015.

[109] J. MP. Nascimento and J. M. Bioucas-Dias. Does independent component analysis play a role in unmixing hyperspectral data? *IEEE Transactions on Geoscience and Remote Sensing*, 43(1):175–187, 2005.

[110] J. MP. Nascimento and J. M. Bioucas-Dias. Vertex component analysis: a fast algorithm to unmix hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 43(4):898–910, 2005.

[111] F. Nie, H. Huang, and C. HQ. Ding. Low-rank matrix recovery via efficient Schatten-p norm minimization. In *AAAI Conference on Artificial Intelligence*, 2012.

[112] N. L. Owsley. Adaptive data orthogonalization. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP).*, 1978.

[113] S. Oymak, A. Jalali, M. Fazel, Y. C Eldar, and B. Hassibi. Simultaneously structured models with application to sparse and low-rank matrices. *IEEE Transactions on Information Theory*, 61(5):2886–2908, 2015.

[114] A. Parekh and I. W. Selesnick. Improved sparse low-rank matrix estimation. *Signal Processing*, 139:62–69, 2017.

[115] N. Parikh, S. Boyd, et al. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.

[116] J. T. Parker, P. Schniter, and V. Cevher. Bilinear generalized approximate message passing—part i: Derivation. *IEEE Transactions on Signal Processing*, 62(22):5839–5853, 2014.

[117] R. Peharz and F. Pernkopf. Sparse nonnegative matrix factorization with ℓ_0-constraints. *Neurocomputing*, 80:38–46, 2012.

[118] Y. Qian, S. Jia, J. Zhou, and A. Robles-Kelly. Hyperspectral unmixing via $\ell_{1/2}$ sparsity-constrained nonnegative matrix factorization. *IEEE Transactions on Geoscience and Remote Sensing*, 49(11):4282–4297, 2011.

[119] Q. Qu, N. M. Nasrabadi, and T. D. Tran. Abundance estimation for bilinear mixture models via joint sparse and low-rank representation. *IEEE Transactions on Geoscience and Remote Sensing*, 52(7):4404–4423, July 2014.

[120] M. Razaviyayn, M. Hong, and Z-Q. Luo. A unified convergence analysis of block successive minimization methods for nonsmooth optimization. *SIAM Journal on Optimization*, 23(2):1126–1153, 2013.

[121] B. Recht, M. Fazel, and P. A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, 2010.

[122] B. Recht, M. Fazel, and P. A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, 52(3):471–501, 2010.

[123] A. A. Rontogiannis, K. E. Themelis, and K. D. Koutroumbas. A fast variational Bayes algorithm for sparse semi-supervised unmixing of Omega/Mars Express data. In *5th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (IEEE WHISPERS)*, June 2013.

[124] R. Saab, R. Chartrand, and O. Yilmaz. Stable sparse approximations via nonconvex optimization. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2008.

[125] M-A. Sato. Online model selection based on the variational Bayes. *Neural Computation*, 13(7):1649–1681, 2001.

[126] P-A. Savalle, E. Richard, and N. Vayatis. Estimation of simultaneously sparse and low rank matrices. In *International Conference on Machine Learning (ICML)*, June 2012.

[127] F. Schmidt, S. Douté, and B. Schmitt. Wavanglet: an efficient supervised classifier for hyperspectral images. *IEEE Transactions on Geoscience and Remote Sensing*, 45(5):1374–1385, 2007.

[128] F. Schmidt, M. Legendre, and S. Le Mouëlic. Minerals detection for hyperspectral images using adapted linear unmixing: Linmin. *Icarus*, 237:61–74, 2014.

[129] S. L. Sclove. Application of model-selection criteria to some problems in multivariate analysis. *Psychometrika*, 52(3):333–343, 1987.

[130] S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[131] F. Shang, Y. Liu, and J. Cheng. Tractable and scalable Schatten quasi-norm approximations for rank minimization. In *Artificial Intelligence and Statistics (AISTATS)*, 2016.

[132] F. Shang, Y. Liu, and J. Cheng. Unified scalable equivalent formulations for schatten quasi-norms. *arXiv preprint arXiv:1606.00668*, 2016.

[133] S. Squires, A. Prügel-Bennett, and M. Niranjan. Rank selection in nonnegative matrix factorization using minimum description length. *Neural Computation*, 29(8):2164–2176, 2017.

[134] N. Srebro, J. Rennie, and T. Jaakkola. Maximum-margin matrix factorization. In *Advances in Neural Information Processing Systems*, pages 1329–1336, 2005.

[135] N. Srebro and A. Shraibman. Rank, trace-norm and max-norm. In *18th Annual Conference on Learning Theory*, 2005.

[136] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009.

[137] K-K. Sung. *Learning and example selection for object and pattern recognition*. PhD thesis, MIT, Artificial Intelligence Laboratory and Center for Biological and Computational Learning, Cambridge, MA, 1996.

[138] V. YF. Tan and C. Févotte. Automatic relevance determination in nonnegative matrix factorization. In *Signal Processing with Adaptive Sparse Structured Representations (SPARS)*, 2009.

[139] V. YF. Tan and C. Févotte. Automatic relevance determination in nonnegative matrix factorization with the β-divergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1592–1605, 2013.

[140] W. Tang, Z. Shi, Y. Wu, and C. Zhang. Sparse unmixing of hyperspectral data using spectral a priori information. *IEEE Transactions on Geoscience and Remote Sensing*, 53(2):770–783, 2015.

[141] K. E. Themelis, A. A. Rontogiannis, and K. D. Koutroumbas. Semi-supervised hyperspectral unmixing via the weighted lasso. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2010.

[142] K. E. Themelis, A. A. Rontogiannis, and K. D. Koutroumbas. A variational Bayes framework for sparse adaptive estimation. *IEEE Transactions on Signal Processing*, 62(18):4723–4736, 2014.

[143] K. E. Themelis, A. A. Rontogiannis, and K. D. Koutroumbas. Variational Bayes group sparse time-adaptive parameter estimation with either known or unknown sparsity pattern. *IEEE Transactions on Signal Processing*, 64(12):3194–3206, 2016.

[144] K. E. Themelis, F. Schmidt, O. Sykioti, A. A. Rontogiannis, K. D. Koutroumbas, and I. A. Daglis. On the unmixing of MEx/OMEGA hyperspectral data. *Planetary and Space Science*, 68(1):34–41, 2012.

[145] K.E. Themelis, A.A. Rontogiannis, and K.D. Koutroumbas. A novel hierarchical bayesian approach for sparse semisupervised hyperspectral unmixing. *IEEE Transactions on Signal Processing*, 60(2):585–599, 2012.

[146] S. Theodoridis. *Machine Learning: A Bayesian and Optimization Perspective*. Academic Press, 2015.

[147] S. Theodoridis and K. D. Koutroumbas. *Pattern Recognition (4th edition)*. Academic Press, 2008.

[148] M. E. Tipping. Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, 2001.

[149] I. Tosic and P. Frossard. Dictionary learning. *IEEE Signal Processing Magazine*, 28(2):27–38, 2011.

[150] I. C. Tsaknakis, P. V. Giampouras, A. A. Rontogiannis, and K. D. Koutroumbas. A computationally efficient tensor completion algorithm. *(to appear) IEEE Signal Processing Letters*, 2018.

[151] P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, 109(3):475–494, 2001.

[152] D. G Tzikas, CL Likas, and N. P. Galatsanos. The variational approximation for Bayesian inference. *IEEE Signal Processing Magazine*, 25(6):131–146, 2008.

[153] V. Vapnik. Statistical learning theory. *NY: Wiley*, 1998.

[154] S. A. Vavasis. On the complexity of nonnegative matrix factorization. *SIAM Journal on Optimization*, 20(3):1364–1377, 2009.

[155] Y-X. Wang, H. Xu, and C. Leng. Provable subspace clustering: When lrr meets ssc. In *Advances in Neural Information Processing Systems*, 2013.

[156] Y-X. Wang and Y-J. Zhang. Nonnegative matrix factorization: A comprehensive review. *IEEE Transactions on Knowledge and Data Engineering*, 25(6):1336–1353, 2013.

[157] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, Apr. 2004.

[158] M. E. Winter. N-FINDR: An algorithm for fast autonomous spectral end-member determination in hyperspectral data. In *Imaging Spectrometry V*, volume 3753, pages 266–276, 1999.

[159] S. J. Wright. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34, 2015.

[160] B. Yang. Projection approximation subspace tracking. *IEEE Transactions on Signal Processing*, 43(1):95–107, 1995.

[161] W. Yang and H. Xu. Streaming sparse principal component analysis. In *International Conference on Machine Learning (ICML)*, 2015.

[162] M. Yuan. Degrees of freedom in low rank matrix estimation. *Science China Mathematics*, 59(12):2485–2502, 2016.

[163] Q. Yuan, L. Zhang, and H. Shen. Hyperspectral image denoising employing a spectral–spatial adaptive total variation model. *IEEE Transactions on Geoscience and Remote Sensing*, 50(10):3660–3677, 2012.

[164] D. Zhang and L. Balzano. Global convergence of a grassmannian gradient descent algorithm for subspace estimation. In *19th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2016.

[165] Y. Zhang and L. E. Ghaoui. Large-scale sparse principal component analysis with application to text data. In *Advances in Neural Information Processing Systems*, 2011.

[166] Y-Q. Zhao and J. Yang. Hyperspectral image denoising via sparse representation and low-rank constraint. *IEEE Transactions on Geoscience and Remote Sensing*, 53(1):296–308, 2015.

[167] Z. Zhu, Q. Li, G. Tang, and M. B. Wakin. The global optimization geometry of low-rank matrix optimization. *arXiv preprint*, 2017.

[168] H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.

[169] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

[170] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286, 2006.