2013

Discriminative Dictionary Learning With Spatial Constraints

Muhammad Nazar Khan
University of Central Florida

DICTIONARY LEARNING FOR IMAGE ANALYSIS

by

NAZAR KHAN
B.Sc. Computer Science, Lahore University of Management Sciences, Pakistan, 2003
M.Sc. Computer Science, University of Saarland, Germany, 2007

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Spring Term
2013

Major Professor: Marshall F. Tappen

ABSTRACT

In this thesis, we investigate the use of dictionary learning for discriminative tasks on natural images. Our contributions can be summarized as follows:

- We introduce discriminative deviation based learning to achieve principled handling of the reconstruction-discrimination tradeoff that is inherent to discriminative dictionary learning.
- Since natural images obey a strong smoothness prior, we show how spatial smoothness constraints can be incorporated into the learning formulation by embedding dictionary learning into Conditional Random Field (CRF) learning. We demonstrate that such smoothness constraints can lead to state-of-the-art performance for pixel-classification tasks.
- Finally, we lay down the foundations of super-latent learning. By treating sparse codes on a CRF as latent variables, dictionary learning can also be performed via the Latent (Structural) SVM formulation for jointly learning a classifier over the sparse codes. The dictionary is treated as a *super-latent* variable that generates the latent variables.

To my parents for always being there for me.

To my wife and daughter for tolerating my absence (and presence).

To people struggling but not giving up.

ACKNOWLEDGMENTS

I would, first and foremost, like to thank God for letting me pass through graduate school with atleast a little bit of my sanity intact[1]. I would like to thank my advisor Dr. Marshall Tappen for all his help, guidance and patience. I also thank the committee members Dr. Hassan Foroosh, Dr. Kenneth Stanley and Dr. Xin Li for their insightful comments and suggestions during the preparation of this thesis. I must acknowledge the administrative assistance of all support staff at UCF. Also to be thanked are all the people at the Institute of International Education (IIE) and at the United States Educational Foundation in Pakistan (USEFP) for handling all funding arrangements and more for my stay at UCF.

I would especially like to thank Dr. Kazim Khan for considerable help and support during various phases of my graduate experience. His assistance truly went beyond the call of duty. I wish to thank colleagues at UCF for making a sometimes frustrating graduate experience worthwile.

Last but not least, I wish to thank my family for all their support and understanding.

[1] Or so I hope

TABLE OF CONTENTS

LIST OF FIGURES x

LIST OF TABLES xii

CHAPTER 1 INTRODUCTION 1

1.1 Introduction 1

1.2 Background 4

1.2.1 Sparse Coding 5

1.3 Dictionary Learning 6

1.3.1 Method of Optimal Directions 8

1.3.2 K-SVD 8

1.4 Discriminative Dictionary Learning 10

1.4.1 Extraction of Discriminative Atoms via Mutual Information 11

1.4.2 Construction of Discriminative Atoms 11

1.4.3 Regularized Dictionary Learning 14

1.4.4 Reconstruction-Discrimination Trade-off 16

1.5 Discriminative Sparse Coding 18

1.6 Comparison with Prior Work 19

CHAPTER 2 STABLE DISCRIMINATIVE DICTIONARY LEARNING 21

2.1 Introduction 21

2.2 Preliminaries 22

2.3 Discriminative Deviation Function 23

2.4 Stable Discriminative Dictionary Learning (SDDL) 26

2.5 Experiments and Results 27

2.6 Conclusion 30

CHAPTER 3 LEARNING WITH SMOOTHNESS PRIORS 34

3.1 Introduction 35

3.2 Related Work 36

3.3 Preliminaries 39

3.4 Discriminative Dictionary Learning with Spatial Neighborhood Constraints . . . 41

3.4.1 Stability 44

3.4.2 Inference and Parameter Learning 45

3.4.3 Initialization 49

3.5 Experiments and Results 49

3.5.1 Graz02 Bike Dataset 49

3.5.2 Weizmann Horses 55

3.5.3 VOC 2007 56

3.6 Conclusion 59

CHAPTER 4 SUPER-LATENT LEARNING FOR STRUCTURED PREDICTION . . . 60

4.1 Introduction 61

4.1.1 Notation 63

4.2 Learning with Latent and Super-Latent Variables 64

4.2.1 Learning The Classifier via Latent Variables 66

4.2.2 Learning the Super-latent Variable via Latent Variables 76

4.2.3 Convergence Analysis 78

4.3 Dealing with the Reconstruction-Discrimination Tradeoff 79

4.4 Discussion 83

CHAPTER 5 CONCLUSIONS 85

5.1 Is Sparsity Harmful? 87

LIST OF REFERENCES 89

LIST OF FIGURES

Figure 1.1 Reconstructive vs. Discriminative Dictionary Learning 2

Figure 1.2 Inherent smoothness in natural images 3

Figure 2.1 Discriminative deviation vs. discriminative softmax 26

Figure 2.2 Stability comparison 31

Figure 2.3 Comparison of precision-recall curves 32

Figure 2.4 Discriminatively learned dictionaries achieve better pixel-wise classification . 33

Figure 2.5 Localization on UCF Sports dataset 33

Figure 3.1 Benefit of training on Graz02 bike dataset 52

Figure 3.2 Pixel-wise classification results on the Graz02 bike dataset 54

Figure 3.3 Some sample results on the Weizmann Horse dataset and VOC 2007 dataset . 56

Figure 4.1 Optimization scheme with latent variables 64

Figure 4.2 Optimization scheme with latent and super-latent variables 65

Figure 4.3 Weights to enforce the discriminative manifold assumption 68

Figure 4.4 LASSO-Shooting step 73

Figure 4.5 Loss Augmented Inference 74

Figure 4.6 More results for Loss Augmented Inference 75

Figure 4.7 Convergence of Loss Augmented Inference 76

Figure 4.8 Optimization scheme for dealing with the reconstruction-discrimination tradeoff 80

Figure 4.9 Learning effect 82

LIST OF TABLES

Table 1.1 Comparison with related prior work 20

Table 2.1 Localization on UCF Sports 29

Table 3.1 Comparison with closely related approaches 38

Table 3.2 Comparison of pixel-level classfication on Graz02 bike test set 51

Table 3.3 Benefits of training on Graz02 bike dataset 53

Table 3.4 Benefit of training on the Weizmann Horse dataset 55

Table 3.5 Pixel-wise classification on VOC2007 dataset 58

CHAPTER 1
INTRODUCTION

1.1 Introduction

The coding of signals into a sparse representation has numerous benefits that have been exploited by computer vision researchers over the years. The new sparse representation requires an over-complete basis[1] called a dictionary. The basis can be constructed analytically from off-the-shelf parametric functions such as a Fourier basis or it can be learned from data. In this thesis, we investigate how dictionaries can be learned in a discriminative yet stable manner and introduce how smoothness priors can be incorporated into the learning framework. We also show how dictionaries can treated as super-latent variables and learned by exploiting a max-margin learning.

Traditionally, dictionaries have been learned in a reconstructive manner with recent successful attempts at discriminative learning. Figure 1.1 illustrates the difference between reconstructive and discriminative dictionary learning. With reconstructive learning, dictionary $\mathbf{D}_i$ is good at representing signals from its own class i, but nothing stops it from being good for some other class too. Such multiclass representability is not good when dictionaries are used for classification of signals. But with discriminative learning, $\mathbf{D}_i$ is encouraged to be representative of

[1] Number of basis vectors is greater than the signal dimension

class i and at the same time not representative of other classes. This leads to better classification of signals.

	Reconstructive				**Discriminative**		
	Class 1	Class 2	Class3		Class 1	Class 2	Class3
$\mathbf{D}_1$	✓	?	?	$\mathbf{D}_1$	✓	×	×
$\mathbf{D}_2$	?	✓	?	$\mathbf{D}_2$	×	✓	×
$\mathbf{D}_3$	?	?	✓	$\mathbf{D}_3$	×	×	✓

Figure 1.1: Reconstructive vs. Discriminative Dictionary Learning. With reconstructive learning, dictionary $\mathbf{D}_i$ can be good for class i, but nothing stops it from being good for some other class too. But with discriminative learning, $\mathbf{D}_i$ is encouraged to be good for class i and bad for other classes.

Discriminative dictionary learning, however, leads to an unstable formulation due to the so called *reconstruction-discrimination trade-off*. As the dictionaries become more discriminative, the curvature of the error surface for learning increases. This leads to slower convergence, or none at all. In this work, we investigate how dictionaries can be learned in a discriminative yet stable manner.

We limit our investigation to dictionaries for signals coming from natural images. As demonstrated in Figure 1.2, natural images exhibit an inherent smoothness in colors, patterns, textures and especially labels. We explore how these smoothness constraints can be incorporated into the discriminative dictionary learning framework.

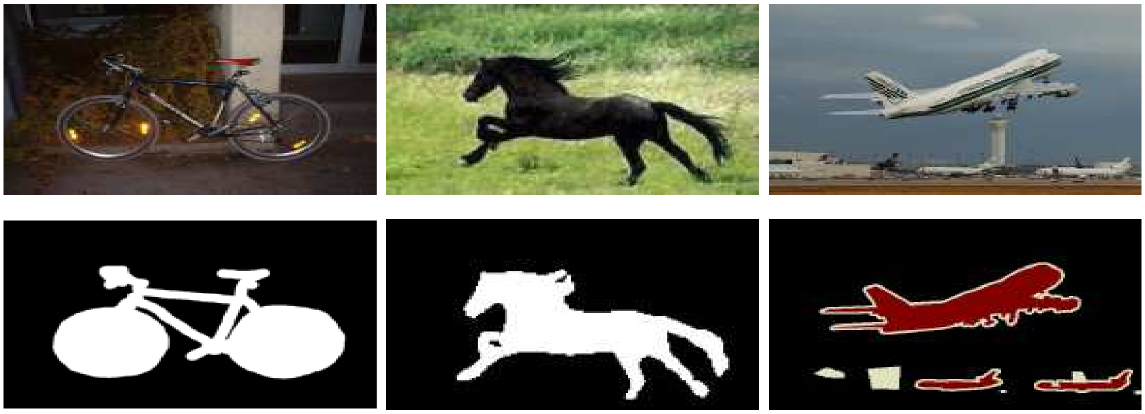


Figure 1.2: Natural images exhibit an inherent smoothness in colors, patterns, textures and especially labels. We explore how these smoothness and scale constraints can be incorporated into the discriminative dictionary learning framework.

Contributions: Specifically, the contributions of this thesis for the discriminative dictionary learning (DDL) problem are:

1. Stable discriminative dictionary learning (Chapter 2). We obtain a formulation that is a lower-bound on the formulation of Mairal *et al.* [1] but only needs one tuning parameter.

We transform this parameter into a true trade-off parameter and constrain its search space to mitigate the instability problem.

2. Incorporation of pairwise spatial smoothness constraints for dictionary learning by embedding dictionaries in a Conditional Random Field (CRF) (Chapter 3). The formulation from Chapter 2 is embedded into the node potentials and spatial pairwise constraints on sparse codes are used in the edge potentials.

3. A proper treatment of Max-Margin dictionary learning in a structured prediction framework (Chapter 4). We explain how the original formulation of Yang & Yang [2] lacks mathematical soundness and then present a more sound methodology. We also introduce a smoothness prior based on the discriminative manifold assumption.

The first contribution is general and applicable to any discriminative dictionary learning problem while the other contributions have only been investigated in the context of natural images to exploit their inherent neighborhood smoothness.

1.2 Background

In this section, we present an introduction to sparse coding and dictionary learning and review some of the existing approaches for dictionary learning in both the reconstructive and discriminative settings.

1.2.1 Sparse Coding

Sparse coding refers to the process of computing a representation of a signal in a new basis such that the representation contains mostly zeros. The so called “dictionary” is this newer basis in which the sparse code resides. While the basis vectors are required to be of unit norm, they need not be orthogonal. The basis vectors are also called the atoms of the dictionary and are represented by the columns of the dictionary matrix $\mathbf{D} \in \mathcal{D}_{n,k}$ where $\mathcal{D}_{n,k}$ is a Stiefel manifold. In other words, a dictionary resides in the subspace of matrices $\mathbb{R}^{n \times k}$ with columns having unit norms. That is, $\mathcal{D}_{n,k} = \{D \in \mathbb{R}^{n \times k} : \forall_{i=1,\ldots,k} ||\mathbf{d}_i||_2 = 1\}$. A signal $\mathbf{x} \in \mathbb{R}^n$ can be converted to its sparse representation $\boldsymbol{\alpha}^* \in \mathbb{R}^k$ under dictionary $\mathbf{D} \in \mathcal{D}_{n,k}$ via the optimization

$$\boldsymbol{\alpha}^* = \arg \min_{\boldsymbol{\alpha} \in \mathbb{R}^k} ||\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}||^2 \; s.t. \; ||\boldsymbol{\alpha}||_0 <= L \tag{1.1}$$

where $L \in \mathbb{Z}^+$ determines the maximum number of non-zero values allowed and is called the sparsity factor. With its ℓ_0 pseudo-norm constraint, sparse coding is an NP-hard problem whose solution can be approximated via a greedy approach known as othogonal matching pursuit (OMP) [3]. An alternative to true ℓ_0 sparse coding is the approximate ℓ_1 sparse coding

$$\boldsymbol{\alpha}^* = \arg \min_{\boldsymbol{\alpha} \in \mathbb{R}^k} ||\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}||^2 + \lambda||\boldsymbol{\alpha}||_1 \tag{1.2}$$

where the amount of sparsity is determined by the sparsity factor $\lambda \in \mathbb{R}^+$. The convexification from the ℓ_1 norm in place of the ℓ_0 norm leads to what is known as the basis pursuit (BP) approach

[4]. In what follows, we will use ℓ_1 sparse coding but the analysis equally applies to the case of ℓ_0 or other general forms of sparse coding[2].

For a set of N signals $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ the (ℓ_1) sparse coding problem can be written as

$$\mathbf{A}^* = \arg \min_{\mathbf{A} \in \mathbb{R}^{k \times N}} ||\mathbf{X} - \mathbf{D}\mathbf{A}||^2 + \lambda \sum_{j=1}^{N} ||\boldsymbol{\alpha}_j||_1 \tag{1.3}$$

where $\mathbf{A} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_N]$ is the set of sparse codes corresponding to signals in $\mathbf{X}$.

1.3 Dictionary Learning

Dictionary learning, as the name suggests, refers to learning the optimal basis for sparsely representing a set of input signals. The optimal dictionary for a set $\mathbf{X}$ is obtained via the optimization

$$\mathbf{D}^* = \arg \min_{\substack{\mathbf{D} \in \mathcal{D}_{n,k} \\ \mathbf{A} \in \mathbb{R}^{k \times N}}} ||\mathbf{X} - \mathbf{D}\mathbf{A}||_F^2 \tag{1.4}$$

where matrix $\mathbf{A}$ of sparse codes under dictionary $\mathbf{D}$ is itself obtained via optimization using, for example, (1.3).

Optimizing (1.4) jointly over dictionary $\mathbf{D}$ and sparse codes $\mathbf{A}$ is a non-convex problem but optimizing over either one alone is convex. Therefore, a standard approach for solving (1.4) is an iterative Lloyd's type algorithm whereby one parameter is fixed and the optimization is carried out over the other and then the roles are reversed. K-means is an example of a Lloyd's type

[2] Our goal with sparse coding is just to obtain a sparse representation. So the exact method is not crucial to our analysis.

algorithm[3] in which the optimization successively iterates between computing the cluster means and assigning the samples to clusters until convergence. What this means here is that one can fix the dictionary $\mathbf{D}$ and compute the optimal sparse codes $\mathbf{A}^*$ under $\mathbf{D}$. Then, the computed sparse codes can be fixed and the objective function in (1.4) can be optimized over $\mathbf{D}$ alone which is a convex problem. This process can be repeated until some convergence criterion (*e.g.* threshold on average reconstruction error) is met. Note, however, that the computed optima $\mathbf{D}^*$ and $\mathbf{A}^*$ are not guaranteed to be the global minimum solution because the joint non-convexity under both $\mathbf{D}, \mathbf{A}$ is broken into convexities under $\mathbf{D}$ and $\mathbf{A}$ alone. Algorithm 1 summarizes the process.

```
Algorithm 1: Dictionary Learning
Input: Set of signals $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$, Initial Dictionary $\mathbf{D} \in \mathbb{R}^{n \times k}$
Output: Optimal dictionary $\mathbf{D}^*$ and sparse codes $\mathbf{A}^* = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_N]$
$\mathbf{D}^* \leftarrow \mathbf{D}$;
while not converged do
    $\mathbf{A}^* \leftarrow \arg\min_{\mathbf{A}} ||\mathbf{X} - \mathbf{D}^*\mathbf{A}||_F^2 + \lambda \sum_{j=1}^{N} ||\boldsymbol{\alpha}_j||_1$ using (1.3);
    $\mathbf{D}^* \leftarrow \arg\min_{\mathbf{D}} ||\mathbf{X} - \mathbf{D}\mathbf{A}^*||_F^2$
end
```

Next, we introduce two representative Lloyd's type algorithms for dictionary learning.

[3] Lloyd is infact credited as the inventor of the K-means algorithm [5]

1.3.1 Method of Optimal Directions

In the method of optimal directions (MOD) [6], after computing the optimal sparse codes the whole dictionary is updated simultaneously by solving the least squares problem

$$\mathbf{D}^* = \arg \min_{\mathbf{D} \in \mathbb{R}^{n \times k}} ||\mathbf{X} - \mathbf{D}\mathbf{A}||_F^2 \tag{1.5}$$

in which no constraints are imposed on the column norms. This can be handled *a posteriori* by explicitly normalizing each atom (*i.e.* column) of $\mathbf{D}^*$.

1.3.2 K-SVD

In the K-SVD algorithm [7], the dictionary update step is changed to sequentially update one atom of the dictionary at a time. However, it additionally updates the sparse code coefficients associated with that atom simultaneously. This leads to faster convergence of the algorithm. As before, let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ be the set of N input signals. Let $\mathbf{A} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_N]$ be the sparse codes of $\mathbf{X}$ under dictionary $\mathbf{D}$. Let $\mathbf{d} = \mathbf{D}_k$ be the k-th column of dictionary $\mathbf{D}$ and let $\boldsymbol{\alpha} = \mathbf{A}^k$ be the k-th row of $\mathbf{A}$. The row-vector $\boldsymbol{\alpha}$ consists of the sparse coefficients in $\mathbf{A}$ that correspond to dictionary atom $\mathbf{d}$. While keeping everything else fixed, K-SVD updates $\mathbf{d}$ and $\boldsymbol{\alpha}$ as

$$\begin{aligned}\arg\min_{\mathbf{d},\boldsymbol{\alpha}} ||\mathbf{X}-\mathbf{D}\mathbf{A}||_F^2 &= \arg\min_{\mathbf{d},\boldsymbol{\alpha}} \left\| \mathbf{X} - \sum_{m=1}^{|\mathbf{D}_i|} \mathbf{D}_m \mathbf{A}^m \right\|_F^2 && (1.6)\\ &= \arg\min_{\mathbf{d},\boldsymbol{\alpha}} \left\| \underbrace{\left(\mathbf{X} - \sum_{m\neq k} \mathbf{D}_m \mathbf{A}^m \right)}_{\mathbf{E}_k} - \underbrace{\mathbf{D}_k}_{\mathbf{d}} \underbrace{\mathbf{A}^k}_{\boldsymbol{\alpha}} \right\|_F^2 && (1.7)\\ &= \arg\min_{\mathbf{d},\boldsymbol{\alpha}} ||\mathbf{E}_k - \mathbf{d}\boldsymbol{\alpha}||_F^2 && (1.8)\end{aligned}$$

In order to maintain sparsity of $\boldsymbol{\alpha}$, one can restrict the minimization to only those input signals that use $\mathbf{d}$, *i.e.* that have non-zeros values in $\boldsymbol{\alpha}$. Let $\boldsymbol{\alpha}^r$ be the vector restricted to contain only the non-zero entries of $\boldsymbol{\alpha}$ and $\mathbf{E}_k^r$ is a similarly restricted version of $\mathbf{E}_k$ that contains the residuals for only those input signals that have non-zero coefficients in $\boldsymbol{\alpha}$. The restricted minimization then becomes

$$\arg\min_{\mathbf{d},\boldsymbol{\alpha}^r} ||\mathbf{E}_k^r - \mathbf{d}\boldsymbol{\alpha}^r||_F^2 \qquad (1.9)$$

which is a rank-one matrix approximation problem and is solved by computing the singular value decomposition (SVD) of $\mathbf{E}_k^r$. Specifically, one needs to compute $\mathbf{U}\boldsymbol{\Delta}\mathbf{V}^T = svd(\mathbf{E}_k^r)$ and set $\mathbf{d} = \mathbf{U}_1$ and $\alpha = \boldsymbol{\Delta}(1,1)\mathbf{V}_1$ where $\mathbf{U}_1$ is the first column of $\mathbf{U}$ and $\mathbf{V}_1$ is the first column of $\mathbf{V}$. Due to the properties of SVD, the ℓ_2 norm of $\mathbf{d}$ is already 1 and therefore K-SVD enforces the unit-norm constraints on the dictionary columns.

Since the process is repeated for each of the k columns of the dictionary, the method is termed K-SVD. Convergence is guaranteed since every sequential update of a dictionary atom reduces a Frobenius norm associated to it without affecting other terms. However, this method is compatible with ℓ_0 sparse coding only (objective function (1.1)).

1.4 Discriminative Dictionary Learning

Traditionally, dictionaries have been learned in a reconstructive manner, *i.e.* via objective function (1.4). What this means is that a dictionary is optimized to accurately reconstruct the set of signals it was trained on. If this set of signals belongs to a particular class, then the dictionary can be expected to be representative of that class. Therefore, for classification purposes, one separate dictionary can be trained for each class and a test signal $\mathbf{x}$ can be classified as belonging to the class whose dictionary reconstructs $\mathbf{x}$ with the least reconstruction error.

While this reconstructive dictionary based classification is able to give good classification results, it has a fundamental weakness when applied to classification problems: *Nothing explicitly stops a dictionary from being representative of other classes too.* For instance, a dictionary trained to accurately reconstruct visual signals of class “bus”can also tend to reconstruct visual signals of class “train”with low reconstruction error. Therefore, there is a need to learn dictionaries in a discriminative manner. By this we specifically mean: *Learn dictionaries so that they reconstruct signals from their own class with low reconstruction error and those from other classes with high reconstruction error.*

1.4.1 Extraction of Discriminative Atoms via Mutual Information

For a given reconstructively learned dictionary $\mathbf{D}$, there can exist random subsets of columns of $\mathbf{D}$ that are more discriminative than $\mathbf{D}$ while being comparably reconstructive. But a random search for such subsets is not always successful. A more principled approach to searching for such *discriminatively reconstructive* subsets is based on maximization of mutual information between atoms. Representative works in this direction include [8, 9, 10, 11]. However, the initial reconstructive dictionary can place an upper limit on the discriminability of the extracted dictionary and therefore we do not pursue this line of research.

1.4.2 Construction of Discriminative Atoms

A more appealing option is to learn the dictionaries in a discriminative manner. Representative works in this direction include [1, 12, 13]. The basic idea here is to make the dictionary for class i representative for class i and explicitly not-so-representative of all other classes. Below we summarize some existing approaches for discriminative dictionary learning.

1.4.2.1 Discriminative Softmax

Let $\mathbf{D}_1, \ldots, \mathbf{D}_C$ be dictionaries learned for C classes of signals. For a signal $\mathbf{y}$, let $\mathbf{R} \in \mathbb{R}^C$ be the vector of reconstruction errors under each dictionary. If the signal $\mathbf{y}$ belongs to class i, then ideally the reconstruction error $\mathbf{R}_i$ should be less than $\mathbf{R}_j$ for all $j \neq i$. This can be achieved by the softmax function employed in [1, 12] that penalizes the reconstruction error $\mathbf{R}_i$ for the true class not being the minimum among all classes. The discriminative softmax function is given by

$$\mathcal{C}_i(\mathbf{R}) = \log \sum_{j=1}^{C} e^{-\lambda(\mathbf{R}_j - \mathbf{R}_i)} \tag{1.10}$$

where parameter λ is used as a discriminative parameter in [1, 12]. When $\mathbf{R}_i$ is the smallest among all classes, $\mathcal{C}_i(\mathbf{R})$ is close to 0 and asymptotically approaches a linear penalty otherwise. For the set of N signals $(\mathbf{y}_1, \ldots, \mathbf{y}_N)$ with labels $(x_1, \ldots, x_N)$, this allows the following discriminative dictionary learning formulation

$$\min_{\{\mathbf{D}_j\}_{j=1}^{C}} \sum_{i=1}^{N} \mathcal{C}_{x_i}(\mathbf{R}(\mathbf{y}_i)) + \gamma \mathbf{R}(\mathbf{y}_i) \tag{1.11}$$

where $\gamma > 0$ controls the reconstruction-discrimination trade-off.

It will be shown in Chapter 2 that parameter λ in fact only controls how good the softmax is at approximating the step function. It does not make it any more or less discriminative. This observation will be used in Chapter 2 to derive a lower bound on the discriminative softmax function that can be used to mitigate the instability of discriminative dictionary learning.

1.4.2.2 Inter-dictionary Incoherence

In [13], Ramirez *et al.* impose inter-dictionary incoherence constraints. This has the effect of implicitly making the dictionaries more discriminative. Their objective function is

$$\min_{\{\mathbf{D}_j\}_{j=1}^C} \sum_{i=1}^{N} \mathbf{R}_{ix_i} + \eta \sum_{i \neq j} ||\mathbf{D}_i^T \mathbf{D}_j||_F^2 \tag{1.12}$$

where $\mathbf{R}_{ix_i}$ is a shortened form of $\mathbf{R}(\mathbf{y}_i)_{x_i}$. The term $\sum_{i \neq j} ||\mathbf{D}_i^T \mathbf{D}_j||_F^2$ encourages all dictionaries to be incoherent from each other, thereby leading to improved discriminability. The discriminability constraint is implicit since the reconstruction errors $\mathbf{R}_i$ are not explicitly enforced to reflect their classes. Instead, the dictionaries that yield the reconstruction errors are forced to be indpendent form each other. As a result, the reconstruction errors are implicitly encouraged to be class specific.

1.4.2.3 Joint Classifier and Dictionary Learning

If a classifier is learned jointly with the dictionary, then the dictionary is encouraged to respect classification constraints as well. This leads to discriminative dictionaries and is the underlying idea of the following approaches.

Discriminative Extensions of KSVD Pham and Venkatesh [14] learned a linear predictive classifier alongwith the reconstructive dictionary that the KSVD algorithm (Section 1.3.2) learns. Dictionaries are encouraged to yield sparse codes that match the hypothesis of the learned classifier. As a result the dictionaries are discriminative. Extensions of this basic idea can be found in [15, 16].

Joint CRF+Dictionary Learning Yang & Yang [2] embedded dictionary learning in a Conditional Random Field (CRF) model by learning a linear predictive classifier on sparse codes. As the parameters of the linear predictor are learned to better classify the sparse codes, the underlying dictionary becomes more and more discriminative.

This is very similar to the approach in Chapter 2 where we also embed dictionaries in a CRF but instead of learning a linear predictive classifier, we use a simplification of the discriminative softmax function (Section 1.4.2.1) to induce discriminative sparse codes. In addition we impose smoothness constraints on neighboring sparse codes in the random field. While the CRF framework naturally allows using such smoothness constraints for dictionary learning, this provision is surprisingly not exploited by [2]. While our smoothness constraints are geared towards inducing discriminative sparse codes, they yield an added benefit of forcing the learning procedure towards stability.

In Chapter 3, we further show that the Max-Margin formulation for learning the linear predictor in [2] lacks mathematical soundness. We therefore present a more sound treatment of joint CRF+Dictionary learning via the Latent Structural SVM fomulation.

1.4.3 Regularized Dictionary Learning

Dictionary learning is an inherently ill-posed problem. To see this, consider the reconstruction error $||\mathbf{y} - \mathbf{D}\boldsymbol{\alpha}||^2$ for a signal $\mathbf{y}$ and its sparse decomposition $\boldsymbol{\alpha}$ under dictionary $\mathbf{D}$. The error remains unchanged if a column of $\mathbf{D}$ is scaled by a scaler s and the corresponding element in the sparse code $\boldsymbol{\alpha}$ is scaled by $\frac{1}{s}$. For a signal set $\mathbf{Y}$, since multiple configurations of $\mathbf{D}$ and those of the sparse codes $\mathbf{A}$ can yield the same reconstruction error performance $||\mathbf{Y} - \mathbf{DA}||_F^2$, the performance of a dictionary is invariant to its column norms. Therefore, in reconstructive dictionary learning, the unit vectors that generate the column vectors determine performance and not the columns themselves. Alternatively, dictionary learning is ill-posed unless it is performed as an optimization over manifolds of unit vectors.

The unit norm constraint on dictionary atoms is also important for sparse coding algorithms since they are affected, in accuracy, speed and stability, by the scales of the sparse codes. In [17], the stability of sparse coding has been linked with all singular values of submatrices of $\mathbf{D}$ being close to 1. The unit norm constraint on dictionary atoms makes the dictionaries well-conditioned and suitable for sparse coding.

Most dictionary learning approaches enforce the unit norm constraint a posteriori, *i.e.* the columns are explicitly normalized after learning. Though this makes the dictionaries suitable for subsequent sparse coding, the dictionary learning formulation itself can still suffer from instabilities. In [18], Yaghoobi *et al.* regularize the dictionary learning procedure by including norm constraints in the objective function. This leads to a more stable learning formulation. A somewhat different approach is introduced in [19] by Dai *et al.* whereby the sparse codes are regularized. The underlying idea is that since sparse codes with large magnitudes are indicative of ill-conditioned dictionaries, imposing a regularization penalty on the sparse codes will force the dictionary learning procedure to move towards well-conditioned dictionaries.

The work of Zheng *et al.* [20] uses the manifold assumption for sparse coding, i.e. points close in the data distribution should be close in the sparse code distribution. This constraint is incorporated into the dictionary learning objective function (1.4) via the use of a $k-$nearest neighbor (kNN) graph of the data. If the signals $\mathbf{y}_i$ and $\mathbf{y}_j$ show regularity on the data graph, then the corresponding sparse codes $\boldsymbol{\alpha}_i$ and $\boldsymbol{\alpha}_j$ should show regularity on the corresponding graph for sparse codes. This yields a graph regularized sparse coding and dictionary learning framework. Similar ideas are used in [21, 22].

Our work bears some similarity with this idea in the sense that we also enforce sparse code similarity constraints but we base them on spatial neighbourhoods of images instead of geometric neighbourhoods of signals.

Our analysis of the instability of dictionary learning has so far has been limited to the case of reconstructive dictionaries. The case of discriminative dictionary learning brings with itself another kind of inherent instability due to the so called *reconstruction-discrimination trade-off*. This is explained next.

1.4.4 Reconstruction-Discrimination Trade-off

Discriminative dictionary learning formulations tend to be inherently unstable. The source of this instability can be seen from the Hessian of the dictionary learning problem (1.4) which can be written as the outer-product $\mathbf{A}\mathbf{A}^T$ of the sparse codes. In an ideal discriminative setting, sparse codes belonging to all signals of class i will be identical and therefore rank$(\mathbf{A}_i\mathbf{A}_i^T) = 1$. This is a manifestation of the so called *reconstruction-discrimination trade-off*:

- If the learning formulation favors reconstructive dictionaries, limited discriminability will be achieved.
- If the learning formulation favors discriminative dictionaries, the optimization becomes unstable.

The trade-off dictates that *while the Hessian needs to be well-conditioned in order to compute a dictionary that is discriminative, a discriminative dictionary in turn makes the resulting Hessian tend towards being ill-conditioned*. So a discriminative objective function will approach singularities as the learned dictionaries become increasingly discriminative. This can lead to very slow convergence or worse – numerical inaccuracies. As a result, while dictionary learning is ill-posed, discriminative dictionary learning can become ill-conditioned and this problem is inherent to the discriminative formulation.

To this end, Mairal *et al.* [1, 12] use a continuation strategy whereby they start-off from a stable reconstructive dictionary learning formulation and gradually move towards the less stable but more discriminative formulation. In this way, they attempt to mitigate the effects of instability. In Chapter 2, we show how the reconstruction-discrimination trade-off can be handled in a slightly more principled manner.

Interestingly, while learned dictionaries are often eventually used for analyzing natural images which are characterized by atleast a local smoothness prior, no such local neighborhood context is used in the dictionary learning process. The main contribution of this thesis is the incorporation of neighborhood smoothness priors for discriminative dictionary learning. Our motivation for imposing smoothness constraints is two-fold:

1. Imposing smoothness constraints on sparse codes can lead to regularized dictionary learning in the manner of [19] and mitigate the effects of the reconstruction-discrimination trade-off.

2. Since we are interested in the analysis of natural images, which exhibit natural smoothness patterns, dictionaries learned over natural images should also respect such smoothness constraints.

In Chapter 3, we show how to discriminatively learn dictionaries while enforcing smoothness constraints from the local spatial neighborhoods. This is done by embedding the dictionary learning framework in a Conditional Random Field (CRF). In Chapter 4, we do the same but in a Max-Margin setting. The basic idea is to embed the sparse codes as latent variables in the CRF and treat the underlying dictionary as a *super-latent* variable.

1.5 Discriminative Sparse Coding

While we focus on learning discriminative dictionaries, alternative approaches exist that focus on making the sparse coding step discriminative. For instance, Huang and Aviyente [23] add Fishers discriminant to the sparse coding objection function (1.1) to encourage high interclass and low intra-class variation in the the resulting sparse codes. However, their formulation simultaneously codes all signals in a sparse and discriminative, supervised manner. In Chapter 4, we show how discriminability can be added to both the sparse coding and dictionary update steps in a joint framework.

1.6 Comparison with Prior Work

Table 1.1 presents a comparison of the contributions of this thesis with related prior works. It can be seen from the table that prior work is surprisingly lacking in enforcing spatial neighborhood smoothness constraints on the dictionary learning framework. Learning in and for the structured prediction setting has also not received much attention.

Table 1.1: Comparison with related prior work.

	DL	DDL	Reg.	Spatial. Reg.	Str. Pred.	Max-Margin
[6]	✓	×	×	×	×	×
[7]	✓	×	×	×	×	×
[18]	✓	×	✓	×	×	×
[19]	✓	×	✓	×	×	×
[1]		✓	×	×	×	×
[12]		✓	×	×	×	×
[13]		✓	×	×	×	×
[14]		✓	×	×	×	×
[15]		✓	×	×	×	×
[16]		✓	×	×	×	×
[2]		✓	×	×	✓	partial
[21]		✓	✓	✓	×	×
[22]		✓	✓	partial	×	×
Ours		✓	✓	✓	✓	✓

CHAPTER 2
STABLE DISCRIMINATIVE DICTIONARY LEARNING

Discriminative learning of sparse-code based dictionaries tends to be inherently unstable. We show that using a discriminative version of the deviation function to learn such dictionaries leads to a more stable formulation that can handle the reconstruction/discrimination trade-off in a principled manner. Results on Graz02 and UCF Sports datasets validate the proposed formulation.

2.1 Introduction

Sparse coding offers a generalization of vocabulary[1] based bag-of-words approaches to recognition of objects. Whereas a standard bag-of-words approach represents an input signal as an optimally sparse vector based on the closest vocabulary word, sparse coding allows representing signals using a linear combination of a few dictionary items. In order to improve upon the ultimate goal of better recognition/classification, multiple approaches attempt to compute dictionaries in a discriminative manner.

[1] Alternative terms in literature are codebooks, dictionaries.

One approach for obtaining discriminative dictionaries is to compute a large overcomplete dictionary in a reconstructive manner and then to extract the more discriminative items from it using mutual information between dictionary items and class labels [9, 10, 11, 24]. But the fundamental weakness of this approach is that the initial reconstructive dictionary places a ceiling on the discriminability of the extracted dictionary.

A better alternative is to incorporate discriminability into the reconstructive dictionary learning framework [1, 12, 13]. However, these approaches suffer from the instability of the discriminative term and require careful tuning of the reconstructive and discriminative parameters in order to avoid instability.

In this work we follow this second approach and introduce a discriminative version of the deviation function that yields a more stable learning formulation by allowing the trade-off between reconstruction and discrimination to be handled in a more principled manner via constraining the search-space for the tuning parameter.

2.2 Preliminaries

An input signal $\mathbf{x} \in \mathbb{R}^n$ can be represented using a sparse code vector $\boldsymbol{\alpha}_j \in \mathbb{R}^k$ under an overcomplete ($n < k$) dictionary $\mathbf{D}_j \in \mathbb{R}^{n \times k}$ obtained as the solution to the sparse coding problem

$$\boldsymbol{\alpha}_j = \arg\min_{\boldsymbol{\alpha} \in \mathbb{R}^k} ||\mathbf{x} - \mathbf{D}_j \boldsymbol{\alpha}||_F^2 \; s.t \; ||\boldsymbol{\alpha}||_0 \leq L \tag{2.1}$$

where L is the sparsity factor (maximum number of non-zero coefficients in $\boldsymbol{\alpha}$)[2]. This can be thought of as a generalization of standard vocabulary based bag-of-words approaches where an input signal is represented as an optimally sparse vector consisting of only one non-zero coefficient corresponding to the closest vocabulary word. The reconstruction error $\mathcal{R}_j$ for signal $\mathbf{x}$ under dictionary $\mathbf{D}_j$ can be computed as

$$\mathcal{R}_j = ||\mathbf{x} - \mathbf{D}_j \boldsymbol{\alpha}_j||_F^2 \tag{2.2}$$

For a set of M signals $\mathbf{x}_1 \ldots \mathbf{x}_M$, the optimal reconstructive dictionary $\mathbf{D}$ and sparse codes $\boldsymbol{\alpha}$ can be computed via

$$\mathbf{D}, \boldsymbol{\alpha} = \arg\min_{\mathbf{D},\boldsymbol{\alpha}} \sum_{i=1}^{M} \mathcal{R}(\mathbf{x}_i) \tag{2.3}$$

which can be solved via the KSVD [7] or MOD [6] algorithms.

For N class classification, per-class dictionaries $\mathbf{D}_1 \ldots \mathbf{D}_N$ can be learned and a test signal $\mathbf{x}$ can be classified via $\arg\min_{j=1\ldots N} \mathcal{R}_j$. In order to make the dictionaries more discriminative we incorporate a discriminative deviation function into the learning framework and this is explained next.

2.3 Discriminative Deviation Function

For a set of values $x_1, \ldots, x_N$ deviation is defined as the difference between an observed value x_i and the mean $\overline{x}$. For a signal belonging to class i we define reconstruction error based

[2] In the rest of this chapter, sparsity factor L is implied on every sparse code $\boldsymbol{\alpha}$.

discriminative deviation as

$$\mathcal{D}_i = \mathcal{R}_i - \frac{\sum_{j=1}^{N} \mathcal{R}_j}{N} \tag{2.4}$$

which is positive if $\mathcal{R}_i$ is above the mean $\frac{\sum_{j=1}^{N} \mathcal{R}_j}{N}$ and negative if $\mathcal{R}_i$ is below the mean. Minimizing $\mathcal{D}_i$ for a signal from class i encourages the reconstruction error $\mathcal{R}_i$ to be lowest among $\mathcal{R}_1, \ldots, \mathcal{R}_N$. This leads to more discriminability and allows us to obtain the following discriminative dictionary learning formulation

$$C(\{\mathbf{D}\}_{j=1}^{N}) = \min_{\{\mathbf{D}\}_{j=1}^{N}} \sum_{i=1}^{N} \sum_{l \in S_i} (\mathcal{D}_{li} + \gamma \mathcal{R}_{li}) \tag{2.5}$$

where S_i is the set of input signals belonging to class i and $\mathcal{D}_{li}$ is the discriminative deviation $\mathcal{D}_i(\mathbf{x}_l)$ of signal $\mathbf{x}_l$ for class i and $\mathcal{R}_{li}$ is the reconstruction error $\mathcal{R}_i(\mathbf{x}_l)$. The reconstructive weight $\gamma > 0$ controls the trade-off between discrimination and reconstruction.

One can show via Jensen's inequality that discriminative deviation $\mathcal{D}_i$ in (2.4) is a lower-bound on the discriminative softmax function used by Mairal *et al.* [1]. Therefore, objective function (2.5) is also a lower-bound on the discriminative cost function found in [1] with very similar behavior as demonstrated in Figure 2.1. It is important to note that this behavior is achieved without the discriminative parameter λ from [1].

To show that deviation is a lower bound on the softmax, we write the discriminative softmax function from Mairal *et al.* [1] as

$$\mathcal{C}_i^{\lambda}(\mathcal{R}_1, \ldots, \mathcal{R}_N) = \log \left(\frac{N}{N} \sum_{j=1}^{N} e^{-\lambda(\mathcal{R}_j - \mathcal{R}_i)} \right) \tag{2.6}$$

$$= \log \left(\sum_{j=1}^{N} \frac{1}{N} e^{-\lambda(\mathcal{R}_j - \mathcal{R}_i)} \right) + \log N \tag{2.7}$$

we can use Jensen's inequality to write

$$\mathcal{C}_i^\lambda(\mathcal{R}_1, \ldots, \mathcal{R}_N) \geq \sum_{j=1}^{N} \frac{1}{N} \left(\log e^{-\lambda(\mathcal{R}_j - \mathcal{R}_i)}\right) + \log N \tag{2.8}$$

$$= \sum_{j=1}^{N} \frac{1}{N} \left(-\lambda(\mathcal{R}_j - \mathcal{R}_i)\right) + \log N \tag{2.9}$$

$$= \frac{-\lambda}{N} \left(\sum_{j=1}^{N} \mathcal{R}_j - N\mathcal{R}_i\right) + \log N \tag{2.10}$$

This gives us the following lower-bound on the discriminative dictionary learning cost function from [1]

$$\hat{C}(\{\mathbf{D}\}_{j=1}^N) = \min_{\{\mathbf{D}\}_{j=1}^N} \sum_{i=1}^{N} \sum_{l \in S_i} \frac{-\lambda}{N} \left(\sum_{j=1}^{N} \mathcal{R}_{lj} - N\mathcal{R}_{li}\right) + \lambda\gamma\mathcal{R}_{li} \tag{2.11}$$

$$= \min_{\{\mathbf{D}\}_{j=1}^N} \sum_{i=1}^{N} \sum_{l \in S_i} \left(\mathcal{D}_{li} + \gamma\mathcal{R}_{li}\right) \tag{2.12}$$

where we have dropped the constant $\log N$ and the scaler multiple λ to obtain the same objective function as (2.5). This lower-bound can be solved for $\mathbf{D}_1 \ldots \mathbf{D}_N$ without the linear approximations proposed in [1].

In [1], a continuation strategy is proposed for stable iterative minimization whereby parameter values are initially set to values corresponding to stable reconstructive optimization and gradually changed to move towards the more discriminative but less stable optimization. However, the search space for the parameters remains unclear. We show in the next section how cost function (2.5) can be made more stable by constraining the search space of the reconstructive parameter γ and using it as a true trade-off parameter.

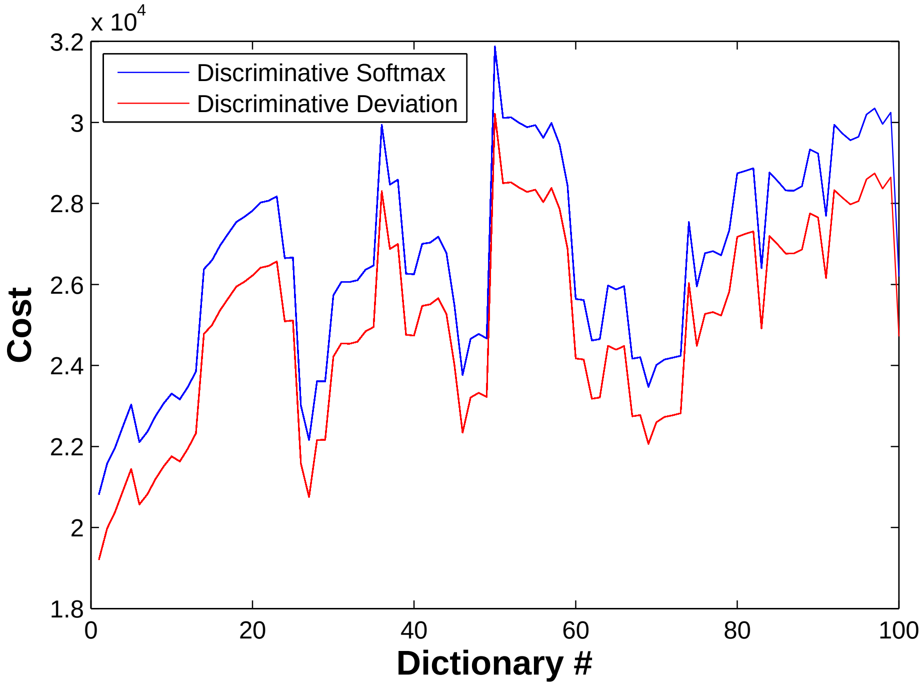


Figure 2.1: Comparison of the discriminative deviation based objective function (2.5) with the discriminative softmax based objective function from [1] for 100 different dictionary configurations. Function (2.5) exhibits similar behavior without the need for a discriminative parameter as in [1].

2.4 Stable Discriminative Dictionary Learning (SDDL)

By constraining γ to lie between 0 and 1, the following more balanced objective function can be obtained

$$C(\{\mathbf{D}\}_{j=1}^{N}) = \min_{\{\mathbf{D}\}_{j=1}^{N}} \sum_{i=1}^{N} \sum_{l \in S_i} (1 - f(\gamma))\mathcal{D}_{li} + \gamma \mathcal{R}_{li} \tag{2.13}$$

where γ is used as a true trade-off parameter. The function $f(\cdot)$ introduces a non-linearity that allows a larger range of values of γ to be considered before running into instability issues. We choose $f(\gamma) = \sqrt{\gamma}$. As a result, the weight $1 - \sqrt{\gamma}$ of the less stable discriminative term remains small for a larger range of γ values while allowing the weight γ of the more stable reconstructive term to drop more drastically.

Cost function (2.13) can be optimized via Newton iterations, MOD [6], or KSVD [7]. We optimize by employing the MOD algorithm.

2.5 Experiments and Results

To validate our formulation, we perform pixel-wise classification on the Graz02 bikes dataset and on the UCF Sports action dataset.

Graz02 We select the first 300 images of the bike category from the Graz02 dataset and use odd numbered images for training and even numbered images for testing. For each training image, dense SIFT features are computed from overlapping patches of size 32×32 with a grid spacing of 12 pixels. For testing images the grid spacing is set to 4.

We run 30 iterations of KSVD[3] to train 2 separate reconstructive dictionaries $\mathbf{D}_f$ and $\mathbf{D}_b$ for foreground and background respectively using the training images and the provided ground-truth shape masks. Each dictionary has 256 items and the sparsity factor L is set to 8. To demonstrate the improvement of our discriminative approach over reconstructive approaches, these dictionaries are used as initial solution for the iterative optimization of (2.13). For each SIFT feature in a test

[3] http://www.cs.technion.ac.il/∼ronrubin/software.html

image I, we compute the reconstruction errors $\mathcal{R}_f$ and $\mathcal{R}_b$ under both dictionaries and classify as foreground if $\mathcal{R}_f < \tau\mathcal{R}_b$ where the optimal value of $0 < \tau \leq 1$ is learned from the training data via cross-validation. Alternatively, τ can be set adaptively for each test image based on the first and second moments of the reconstruction errors. Interpolation is carried out for missing pixel values and the result is smoothed to obtain the final pixel-wise classification confidence that is used in all subsequent precision-recall curve calculations.

Figure 2.2 demonstrates that, compared to [1], our stable formulation (2.13) offers more control over the optimization due to one less parameter to search over and also due to constraining its only parameter to lie between 0 and 1. On the other hand, in [1], there is a lack of clarity as regards to what range of values to consider for the discriminative parameter λ as well as the reconstructive parameter γ.

Figure 2.3 compares precision-recall curves on the Graz02 bikes dataset using reconstructively learned dictionaries via KSVD (dashed curves) and discriminatively trained dictionaries via SDDL (solid curves). Blue curves represent adaptive setting of the classification parameter τ for each test image. Red curves represent τ optimally learned from the training set. It can be observed that discriminative dictionaries yield better classification performance. The benefit of learning an optimal τ from the training set can also be observed. The best achieved EER (Equal Error Rate where precision=recall) is 69.5% which is better than that achieved by [13].

UCF Sports Similar to our setup for the Graz02 dataset, we learn foreground and background dictionaries on dense STIP descriptors[4] [25] for the Diving and Gym (beam) categories from the UCF Sports actions dataset [26]. We replicate the evaluation setup of Yao *et al.* [27] who consider these two classes to be difficult. We compare against their action localization performance in Table 2.1. Considering that we neither do tracking nor ground-truth based initialization for test videos as in [27], our pixel classification based localization is comparable. Figure 2.5 demonstrates localization results on two selected frames.

Table 2.1: Localization on UCF Sports. Percentage of frames with localized bounding boxes having intersection over union with ground-truth $> \frac{1}{2}$. Consider ing that we neither do tracking nor ground-truth based initialization for test videos as in [27], our pixel classification based localization is comparable.

	Gym (beam)	Diving
[27]	62%	68%
SDDL	52%	55%

[4] http://www.irisa.fr/vista/Equipe/People/Laptev/download/stip-2.0-linux.zip

2.6 Conclusion

We have introduced a new discriminative deviation based formulation for dictionary learning that is more stable than previous work while requiring only one tuning parameter and handling the reconstruction-discrimination trade-off in a more principled manner. Its applicability has been shown on two real-world datasets.

However, while natural images have an inherent smoothness prior, this prior is not utilized in the dictionary learning framework. In Chapter 3 we describe how such a smoothness prior can be incorporated into the discriminative dictionary learning framework. This allows discriminative dictionary learning while respecting smoothness constraints.

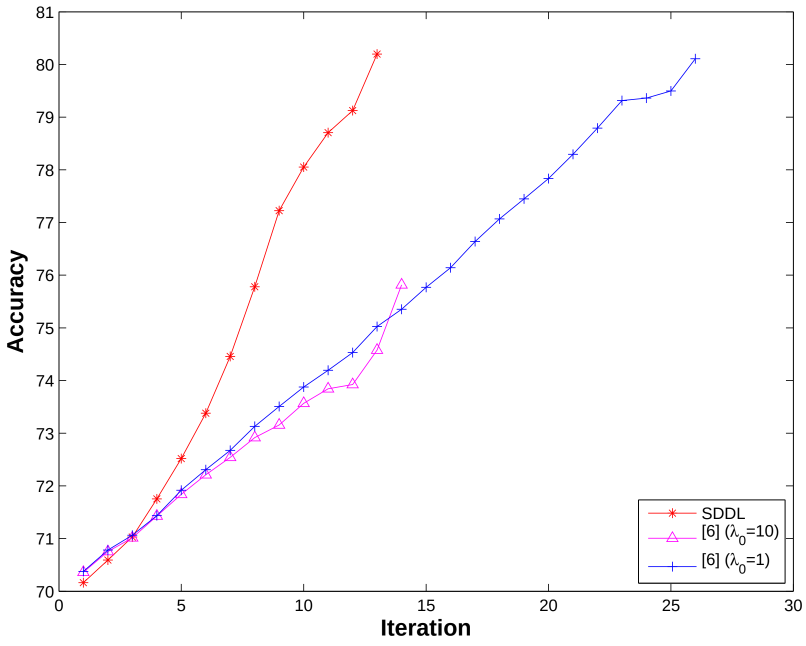


Figure 2.2: Stability comparison of SDDL with the formulation of [1] with high ($\lambda_0 = 10$) and low ($\lambda_0 = 1$) initializations of their discriminative parameter λ and reconstructive parameter γ initialized to 100. λ and γ were gradually updated as proposed in [1]. All three optimizations were continued until instability. For [1], learning with high discriminability leads to instability quickly while not achieving high accuracy while learning with low discriminability takes longer to achieve high accuracy. In contrast, SDDL achieves faster learning and only requires a single tuning parameter constrained between 0 and 1.

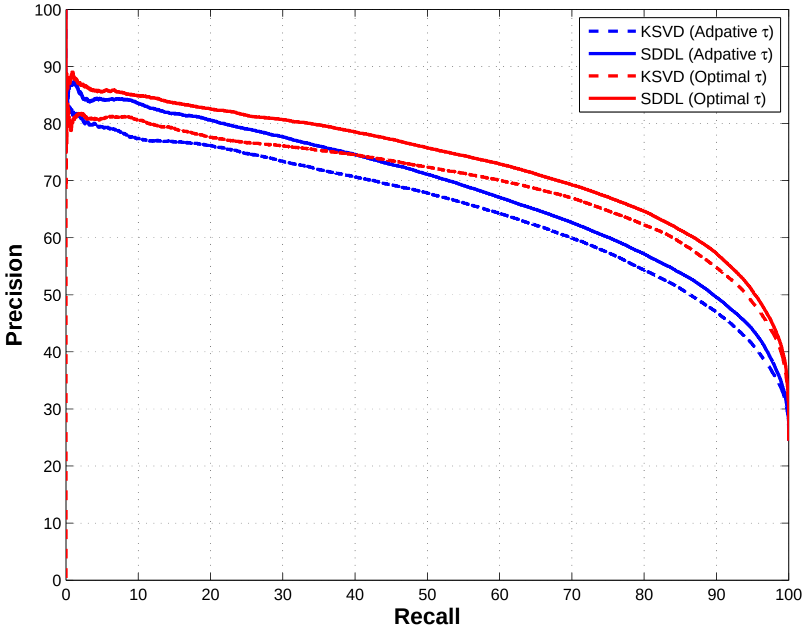


Figure 2.3: Comparison of precision-recall curves on the testing set of Graz02 bikes dataset using reconstructively learned dictionaries via KSVD (dashed curves) and discriminatively trained dictionaries via SDDL (solid curves). See text for details.

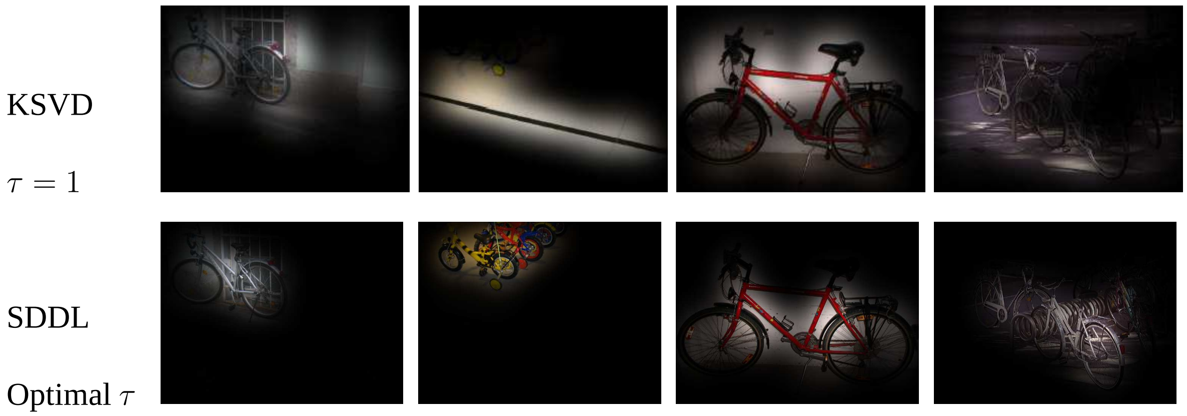


Figure 2.4: **Row 1:** Reconstructive dictionaries (KSVD) with $\mathcal{R}_f < \mathcal{R}_b$ based pixel-wise classification shows a greater tendency to classify background as foreground while **Row 2:** Our discriminatively learned dictionaries (SDDL) with $\mathcal{R}_f < \tau\mathcal{R}_b$ and optimal τ are able to achieve much better pixel-wise classification.

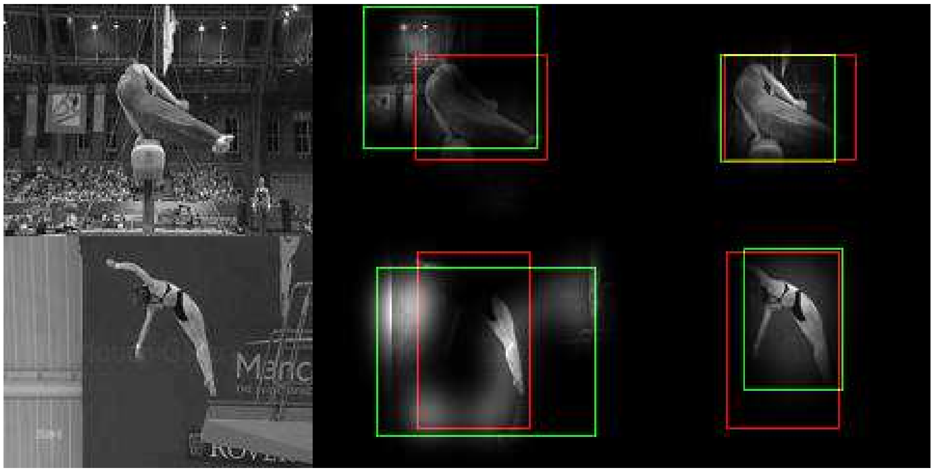


Figure 2.5: Dictionary based (green) and ground-truth (red) localization on UCF Sports dataset. Left: Original frame. Middle: Untrained. Right: SDDL Trained.

CHAPTER 3
LEARNING WITH SMOOTHNESS PRIORS

*Natural images are characterized by a smoothness prior. While discriminatively learned dictionaries have successfuly been employed to classify image pixels, the learning process has, traditionally, exploited no such prior. We present a novel approach to discriminative dictionary learning with neighborhood constraints. This is done by embedding dictionaries in a Conditional Random Field (CRF) and imposing label-dependent smoothness constraints on the resulting sparse codes at adjacent sites. This way, a smoothness prior is used while learning the dictionaries and not just to make inference. This is in contrast with all competing approaches that learn dictionaries without such a prior. Pixel-level classification results on the Graz02 bikes dataset demonstrate that dictionaries learned in our discriminative setting with neighborhood smoothness constraints can equal the state-of-the-art performance of bottom-up (*i.e. *superpixel-based) segmentation approaches.*

Furthermore, we isolate the benefits of our learning formulation and CRF inference to show that our dictionaries are more discriminative than dictionaries learned without such constraints even without CRF inference. An additional benefit of our smooth-

ness constraints is more stable learning which is a known problem for discriminative dictionaries.

3.1 Introduction

Discriminative learning of sparse-coding based dictionaries has been shown to improve performance on various computer vision tasks. Interestingly, while these dictionaries are often eventually used for analyzing natural images which are characterized by a local smoothness prior, no such local neighborhood context is used in the dictionary learning process. We show how to discriminatively learn dictionaries while enforcing smoothness constraints from the local spatial neighborhoods. This is done by embedding the dictionary learning framework in a Conditional Random Field (CRF).

Dictionary learning has successfully been used for various signal classification tasks such as pixel-level classification of images [1, 12, 13, 28], object localization [9], image classification [16], face recognition [29] and video classification [10, 11]. Standard approaches learn dictionaries either reconstructively [7] or discriminatively [1, 12, 13, 16, 28] but do not attempt to exploit neighborhood context in the learning process.

Images of real world objects in real world settings exhibit strongly smooth labels. Generally, object pixels lie adjacent to each other and background pixels lie adjacent to each other. This calls for a smoothness prior in the energy formulation and it allows us to enforce smoothness constraints on neighboring sparse code pairs for a dictionary.

But since boundaries of objects do not share this smoothness prior, there is a need for a discontinuity preserving prior too. This discontinuity preserving prior is what allows us to enforce (non-)smoothness constraints between dictionaries from different classes. *To the best of our knowledge, this is the first attempt at learning discriminative dictionaries with intra-class sparse code smoothness as well as inter-class sparse code (non-)smoothness constraints*.

Besides increased discriminability, an additional benefit of such smoothness constraints is the mitigation of numerical instability which is inherent to discriminative dictionary learning [1, 28]. Interestingly, a recent stability analysis [19] for reconstructive dictionaries also concluded that sparse code smoothness plays an important role in stable learning.

3.2 Related Work

Dictionary Learning: One approach for obtaining discriminative dictionaries is to compute a large overcomplete dictionary in a reconstructive manner and then to extract the more discriminative items from it using mutual information between dictionary items and class labels [9, 10, 11, 24]. But the fundamental weakness of this approach is that the initial reconstructive dictionary places a ceiling on the discriminability of the extracted dictionary.

A better alternative is to incorporate discriminability into the reconstructive dictionary learning framework [1, 12, 13]. However, these approaches suffer from the inherent instability of the reconstructive/discriminative trade-off and require careful tuning of the reconstructive and discriminative parameters in order to avoid instability. Khan and Tappen [28] introduce a discriminative version of the deviation function that yields a more stable learning formulation by allowing the trade-off between reconstruction and discrimination to be handled in a more principled manner via constraining the search-space for the tuning parameter. However, instability is still not totally avoided.

While smoothness priors are ubiquitous in analysis of visual information, dictionary learning for image analysis has relied on local evidences only. Yang & Yang [2] introduced joint dictionary and CRF parameter learning but the dictionary is used to compute unary node potentials only and therefore neighborhood smoothness constraints are not exploited in the learning of the dictionary. Mairal *et al.* [30] introduce simultaneous sparse coding whereby similar image patches are encouraged to have similar sparse codes. We use the same intuition but for learning dictionaries instead of sparse code computation and we use a neighborhood structure instead of patch similarity. The closest related work in terms of smoothness constraints is that of Guo *et al.* [22] which uses sparse code smoothness constraints for image classification. The key difference from their work is that we operate on the pixel level and therefore ours is a structured prediction problem while theirs is a standard classification problem. For a given image, they infer a single label while we infer the pixel labelling structure. Table 3.1 summarizes the relationships between our work and its closest counterparts.

Learning with inter-dictionary constraints is used by Yang *et al.* [31] for an image super-resolution application by learning coupled dictionaries that enforce the sparse code of a low-resolution image patch to accurately reconstruct the underlying high-resolution image patch. Our formulation uses inter-dictionary constraints for the purposes of discriminative learning and is not constrained to two dictionaries.

In this work, we merge the discriminative dictionary learning framework of [1, 28] with the Discriminative Random Field (DRF) framework of [32] and use the dictionaries to compute pairwise edge potentials in addition to node potentials. This allows local neighborhood information to be used when learning dictionaries over the random field.

Table 3.1: Comparison with closely related approaches.

	[22]	[2]	Ours
Structured Prediction	×	✓	✓
Smoothness Constraints	✓	×	✓
Per-class Dictionaries	×	×	✓
Linear Classifier	✓	✓	×

Semantic Segmentation: Most (and state of the art) class segmentation approaches merge bottom-up and top-down cues. The idea is to use an initial (over-)segmentation to choose appropriate segments from. Representative works include [33] which constructs a CRF over single scale superpixels and [34] which use multi-scale superpixels. Using an initial over-segmentation yields an adpative domain for feature computation instead of fixed sub-windows. This can alleviate the scale selection problem for small/large instances of objects. Another benefit is that these initial segments naturally tend to preserve object boundaries. In contrast, pure pixel-level top-down class segmentation approaches (which includes the afore-mentioned dictionary-based approaches) need to rely upon post-processing techniques such as Gaussian smoothing or inference on a CRF to enforce spatial coherency. Our work moves enforcement of spatial consistency from the post-processing step to the dictionary learning step. We show in Section 3.5 that such a neighborhood constrained dictionary learning mechanism can lead to top-down pixel-level classification performance that matches the bottom-up super-pixel segmentation approaches.

3.3 Preliminaries

For an image $\mathbf{y}$ with ground-truth labeling $\mathbf{x}$, let $\mathcal{V}$ be a uniformly spaced grid of image locations or 'sites'and $\mathbf{y}_i \in \mathbb{R}^n$ be an n dimensional feature vector extracted at site $i \in \mathcal{V}$. For each site i, $\mathcal{N}_i$ denotes the neighboring sites of i and $x_i \in \{1 \dots C\}$ denotes the true label.

For each feature vector $\mathbf{y}_i \in \mathbb{R}^n$, let $\mathbf{s}_{ic} \in \mathbb{R}^k$ be its sparse code vector under a dictionary $\mathbf{D}_c \in \mathbb{R}^{n \times k}$ for class $c \in \{1 \dots C\}$. The sparse code vector $\mathbf{s}_{ic}$ is obtained as a solution to the ℓ_1 sparse coding problem

$$\mathbf{s}_{ic}(\mathbf{y}_i, \mathbf{D}_c) = \arg\min_{\mathbf{s} \in \mathbb{R}^k} \frac{1}{2} ||\mathbf{y}_i - \mathbf{D}_c \mathbf{s}||_F^2 + \lambda ||\mathbf{s}||_1 \qquad (3.1)$$

which can be solved using the LASSO algorithm[1]. The reconstruction error $\mathbf{R}_{ic}$ for a signal $\mathbf{y}_i$ under a dictionary $\mathbf{D}_c$ is computed using the optimal sparse code vector $\mathbf{s}_{ic}$ obtained via (3.1)

$$\mathbf{R}_{ic}(\mathbf{y}_i, \mathbf{D}_c) = \frac{1}{2}||\mathbf{y}_i - \mathbf{D}_c\mathbf{s}_{ic}||_F^2 + \lambda||\mathbf{s}_{ic}||_1 \tag{3.2}$$

$\mathbf{R}_i \in \mathbb{R}^C$ denotes the vector of per-class reconstruction errors for signal $\mathbf{y}_i$. Both (3.1) and (3.2) are rendered non-differentiable with respect to dictionary $\mathbf{D}_c$ due to the presence of the ℓ_1 norm. The derivatives are therefore computed using implicit differentiation as explained in Section 3.4.2.1.

Discriminative Deviation: For discriminative learning, our energy function makes use of the discriminative deviation function intriduced in Chapter 2. For the c_{th} entry in a vector $\mathbf{v}$, deviation is defined as the difference from the mean

$$\mathcal{D}_c^{\mathbf{v}} = \mathbf{v}_c - \bar{\mathbf{v}}. \tag{3.3}$$

For a signal $\mathbf{y}_i$ belonging to class x_i with reconstruction error vector $\mathbf{R}_i \in \mathbb{R}^C$, reconstruction error based discriminative deviation is

$$\mathcal{D}_{x_i}^{\mathbf{R}_i} = \mathbf{R}_{ix_i} - \bar{\mathbf{R}}_i \tag{3.4}$$

which is positive if $\mathbf{R}_{ix_i}$ is above the mean and negative if $\mathbf{R}_{ix_i}$ is below the mean. Minimizing $\mathcal{D}_{x_i}^{\mathbf{R}_i}$ encourages the reconstruction error $\mathbf{R}_{ix_i}$ to be lowest among $\mathbf{R}_{i1}, \ldots, \mathbf{R}_{iC}$. This leads to more discriminability and allows us to obtain the following discriminative dictionary learning formulation

$$\min_{\{\mathbf{D}\}_{j=1}^N} \sum_{i=1}^{M} \left(\mathcal{D}_{x_i}^{\mathbf{R}_i} + \gamma\mathbf{R}_{ix_i}\right) \tag{3.5}$$

[1] http://www.di.ens.fr/willow/SPAMS/ and http://www.di.ens.fr/~mschmidt/Software/UGM.html

where M is the number of training signals. This encourages dictionaries to be good at reconstructing signals from their own class while also being bad for signals from other classes. The reconstructive weight $\gamma > 0$ controls the trade-off between discrimination and reconstruction.

3.4 Discriminative Dictionary Learning with Spatial Neighborhood Constraints

Let $\mathbf{Y} = [\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(N)}]$ be N training images with the corresponding labelings denoted by $\mathbf{X} = [\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}]$. Without loss of generality, let L be the set of all possible labelings on any given grid of sites. Clearly, L is an exponentially large set. Then the probability of image labeling $\mathbf{x}^{(t)}$ conditioned on the observed image $\mathbf{y}^{(t)}$ can be written as a Gibbs field

$$P(\mathbf{x}^{(t)}|\mathbf{y}^{(t)}, \{\mathbf{D}\}_1^C, \boldsymbol{\kappa}) = \frac{1}{Z} e^{-E(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}, \{\mathbf{D}\}_1^C, \boldsymbol{\kappa})} \tag{3.6}$$

where $Z = \sum_{\mathbf{x} \in \mathrm{L}} e^{-E(\mathbf{x}, \mathbf{y}^{(t)}, \{\mathbf{D}\}_1^C, \boldsymbol{\kappa})}$ is the so-called partition function and

$$\begin{aligned} E(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}, \{\mathbf{D}\}_1^C, \boldsymbol{\kappa}) &= \sum_{i \in \mathcal{V}^{(t)}} E_i(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}, \{\mathbf{D}\}_1^C, \boldsymbol{\kappa}) \\ = \sum_{i \in \mathcal{V}^{(t)}} e^{-\kappa_d} \underbrace{\left(\mathcal{D}_{x_i}^{\mathbf{R}_i} + e^{-\kappa_d^{\text{rec}}} \mathbf{R}_{i x_i}\right)}_{\text{data term}} \\ + e^{-\kappa_s} \underbrace{\sum_{j \in \mathcal{N}_i} e^{-\kappa_s^{\text{ind}}} \bar{\delta}_{x_i x_j} + e^{-\kappa_s^{\text{dep}}} s_{\text{dep}}}_{\text{smoothness term}} \end{aligned} \tag{3.7}$$

where

$$s_{\text{dep}} = -\, \delta_{x_i x_j} \left(\mathcal{D}_{x_i}^{\mathbf{p}} + \mu \mathbf{p}_{x_i}\right) + \bar{\delta}_{x_i x_j} \mathbf{p}_{x_i} \tag{3.8}$$

is the data-dependent smoothness term, δ is the Dirac delta function, $\bar{\delta}$ is its complement and

$$\mathbf{p} = \mathbf{s}_{ix_i}^T \mathbf{s}_j \tag{3.9}$$

is a C dimensional vector of the similarity of sparse code $\mathbf{s}_{ix_i}$ with all sparse codes of the adjacent site j. The weights of the data term and the smoothness term are determined by parameters κ_d and κ_s respectively. Negative exponentials of all weights are used to ensure positive weightings and unconstrained optimization.

Data term: Encourages $\mathbf{R}_{ix_i}$ to be low and $\mathbf{R}_{ic}$ to be high for all $c \neq x_i$. Value of κ_d^{rec} determines the weightage given to the reconstructive term relative to the discriminative deviation term.

Data-independent smoothness: The data-independent smoothness term $\bar{\delta}_{x_i x_j}$ penalizes dissimilar labels on adjacent sites and rewards similar labels.

Data-dependent smoothness: The goal is to encourage signals with the same label to have similar sparse code vectors and those with different labels to have dissimilar sparse code vectors. During learning, this encourages dictionaries to be more sensitive to object boundaries. During inference, this allows smoothing to be reduced at edges (in feature space) and results in sharper segmentations. For adjacent pixels i, j with the same label $x_i = x_j$, the data-dependent smoothness term encourages sparse code vectors $\mathbf{s}_{ix_i}$ and $\mathbf{s}_{jx_i}$ under dictionary $\mathbf{D}_{x_i}$ to be most similar among all classes. This is achieved by once again employing the discriminative deviation function as used in the data term. *The advantage of using discriminative deviation is dictionary learning with label-dependent smoothness constraints on adjacent sparse codes*. If only the term $\mathbf{s}_{ix_i}^T \mathbf{s}_{jx_i}$ is used instead, then only dictionary $\mathbf{D}_{x_i}$ is affected. Parameter $\mu \geq 0$ determines the trade-off between discriminative deviation and the similarity of the sparse code vectors. For adjacent pixels with different labels, the sparse code vectors only under dictionary $\mathbf{D}_{x_i}$ are encouraged to be different. Since our graphical model contains loops, this eventually implies sparse code dissimilarity under both classes x_i and x_j. However, no inter-dictionary constraint is enforced in this case.

Energy function (3.7) makes our formulation a Discriminative Random Field (DRF) [32] which is a variant of a Conditional Random Field (CRF) [35]. Instead of learning linear CRF parameter vectors, we learn non-linear dictionaries. It has similarities with [2] but has a richer representational model since it is multiclass, learns discriminative dictionaries, and includes data-dependent smoothness. Our formulation tries to explain all classes instead of just foreground. *More importantly, the data-dependent smoothness term includes, in addition to the data, the dictionaries as well. During learning, this encourages dictionaries to have responses for neighboring pixels that reflect their labels. Therefore, energy function* (3.7) *imposes neighborhood constraints on the discriminative dictionary learning frameworks from [1, 28]. It can also be viewed as the structured prediction counterpart of [22].*

Weights: We use 5 weights to handle 4 terms in (3.7). This can make our formulation somewhat susceptible to local minima. However, this redundancy helps in dealing with over-smoothed MAP inference which affects especially pseudolikelihood based minimization [36]. While weights κ_d and κ_s handle the general trade-off between local evidence and spatial consistency of labels, the rest of the weights handle more refined aspects of the energy functional: κ_d^{rec} handles the trade-off between discrimination and reconstruction, while κ_s^{ind} and κ_s^{dep} handle the trade-off between classical data-independent Potts potentials and data-dependent potentials. A suitable value of κ_s

can control over-smoothness while allowing the more subtle play between data-independent and data-dependent smoothness terms to be explored more finely.

3.4.1 Stability

Our smoothness constraints can alternatively be considered as *pseudo-regularization* of dictionaries based on the regularity of pixel labels in natural images.

It is well-known that

1. Sparse coding is sensitive to incoherence among a dictionary's atoms [7], and
2. Discriminability is increased by having mutually incoherent dictionaries [13].

Therefore, it is beneficial to increase both intra- and inter-dictionary incoherence. Intra-dictionary incoherence is enforced by $\bar{\delta}_{x_i x_j}\mathbf{p}_{x_i}$ in Equation (3.8). The discriminative deviation term $\mathcal{D}^{\mathbf{p}}_{x_i}$ enforces inter-dictionary incoherence and also leads to well-conditioned dictionaries by requiring adjacent same-class sparse codes to be similar[2]. So our formulation contains the well-known sources of stability. In contrast, despite embedding dictionary learning in a CRF framework, Yang & Yang [2] do not impose such dictionary-related smoothness constraints.

[2] Relation between smooth sparse codes and dictionary conditioning is explained in [19]

3.4.2 Inference and Parameter Learning

Computation of (3.6) and its corresponding likelihood function require computation of the partition function Z which is intractable due to the exponentially large size of the set L of all labelings. Therefore, for inference we use approximate techniques such as Mean Field Inference or Loopy Belief Propagation[3]. For learning parameters, we use the pseudolikelihood approximation defined as

$$\tilde{P}(\mathbf{x}^{(t)}|\mathbf{y}^{(t)}, \{\mathbf{D}\}_1^C, \boldsymbol{\kappa}) = \prod_{i \in \mathcal{V}} \frac{1}{z_i} e^{-E_i} \tag{3.10}$$

where

$$z_i = \sum_{x_i=1}^{C} e^{-E_i} \tag{3.11}$$

The advantage of using this approximate pseudolikelihood is that the intractable computation of the true partition function Z is replaced by the tractable computation of the local normalization functions z_i. Negative log-pseudolikelihood is then written as

$$-\log \tilde{P}(\mathbf{x}^{(t)}|\mathbf{y}^{(t)}, \{\mathbf{D}\}_1^C, \boldsymbol{\kappa}) = \sum_{i \in \mathcal{V}} E_i + \log z_i \tag{3.12}$$

The gradient of the negative log-pseudolikelihood with respect to any arbitrary parameter $\theta \in \{\{\mathbf{D}\}_1^C, \boldsymbol{\kappa}\}$ is

$$\sum_{i \in \mathcal{V}} \frac{dE_i}{d\theta} + \frac{1}{z_i} \sum_{x_i=1}^{C} e^{-E_i} \frac{-dE_i}{d\theta} \tag{3.13}$$

[3] Available at http://www.di.ens.fr/~mschmidt/Software/UGM.html

The only non-trivial gradient in (3.13) is $\frac{dE_i}{d\mathbf{D}_c}$ for the dictionary of an arbitrary class c. It requires the computation of the intermediate gradients $\frac{d\mathbf{R}_{ic}}{d\mathbf{D}_c}$, $\frac{d\mathbf{s}_{ic}}{d\mathbf{D}_c}$ and $\frac{d\mathbf{s}_{jc}}{d\mathbf{D}_c}$ for $j \in \mathcal{N}_i$. As explained earlier, these are non-trivial computations and can be performed using implicit differentiation as explained next.

3.4.2.1 Reconstruction Error Gradient via Implicit Differentiation

This section presents a method for computing the gradients of the non-differentiable sparse coding procedure (3.1) and hence for the reconstruction error (3.2) also. Our explanation follows [37, 38, 2].

In order to compute $\frac{d\mathbf{R}_{ic}}{d\mathbf{D}_c}$, it is beneficial to rewrite the ℓ_1 sparse coding problem in its complete form

$$\mathbf{R}_{ic}(\mathbf{y}_i, \mathbf{D}_c) = \frac{1}{2}||\mathbf{y}_i - \mathbf{D}_c\mathbf{s}_{ic}(\mathbf{y}_i, \mathbf{D}_c)||_F^2 + \lambda||\mathbf{s}_{ic}(\mathbf{y}_i, \mathbf{D}_c)||_1 \tag{3.14}$$

where sparse code $\mathbf{s}_{ic}(\mathbf{y}_i, \mathbf{D}_c)$ is fixed and computed via (3.1). Therefore dictionary $\mathbf{D}_c$ affects reconstruction error $\mathbf{R}_{ic}$ directly as well as indirectly through the fixed sparse code $\mathbf{s}_{ic}(\mathbf{y}_i, \mathbf{D}_c)$. To make the notation clearer, we will drop the subscripts i and c without loss of generality.

We first compute $\frac{d\mathbf{s}^*}{d\mathbf{D}}$ representing the gradient of the optimal sparse code vector $\mathbf{s}^* \in \mathbb{R}^k$ corresponding to an arbitrary signal $\mathbf{y} \in \mathbb{R}^n$ under an arbitrary dictionary $\mathbf{D} \in \mathbb{R}^{n \times k}$. Since $\mathbf{s}^*$ is a minimizer of the reconstruction error $\mathbf{R}(\mathbf{s}) = \frac{1}{2}||\mathbf{y} - \mathbf{D}\mathbf{s}||_F^2 + \lambda||\mathbf{s}||_1$

$$\nabla_{\mathbf{s}}\mathbf{R}(\mathbf{s})|_{\mathbf{s}=\mathbf{s}^*} = \mathbf{0} \tag{3.15}$$

$$\mathbf{D}^T(\mathbf{D}\mathbf{s} - \mathbf{y})|_{\mathbf{s}=\mathbf{s}^*} = -\lambda sign(\mathbf{s})|_{\mathbf{s}=\mathbf{s}^*} \tag{3.16}$$

$$\mathbf{D}^T(\mathbf{D}\mathbf{s}^* - \mathbf{y}) = -\lambda sign(\mathbf{s}^*) \tag{3.17}$$

where $sign(\cdot)$ is an elementwise sign operator and $sign(0) = 0$. Taking the derivative with respect to $\mathbf{D}_{ij}$ on both sides

$$\frac{d}{d\mathbf{D}_{ij}}\mathbf{D}^T(\mathbf{D}\mathbf{s}^* - \mathbf{y}) = -\frac{d}{d\mathbf{D}_{ij}}\lambda sign(\mathbf{s}^*) \tag{3.18}$$

For non-zero values the $sign(\cdot)$ function has 0 gradient and it has a discontinuity at 0. However, since the the left hand side cannot be infinite, we set the gradient at 0 to be 0 which makes the right hand side $\mathbf{0}_{k\times 1}$. Let $\wedge$ be the set of indices of the active set (*i.e.* non-zero coefficients) of s* and let $\bar{\wedge}$ be its complement. Since $\frac{d\mathbf{s}_m}{d\mathbf{D}_{ij}}$ is not well-defined for $\mathrm{s}_m = 0$, we set $\frac{d\mathbf{s}^*_{\bar{\wedge}}}{d\mathbf{D}_{ij}} = \mathbf{0}_{|\bar{\wedge}|\times 1}$. Accordingly, we can write

$$\frac{d}{d\mathbf{D}_{ij}}\mathbf{D}^T(\mathbf{D}\mathbf{s}^* - \mathbf{y}) = \mathbf{0}_{k\times 1} \tag{3.19}$$

$$\frac{d}{d\mathbf{D}_{i\wedge_j}}\mathbf{D}^T_{\wedge}(\mathbf{D}_{\wedge}\mathbf{s}^*_{\wedge} - \mathbf{y}) = \mathbf{0}_{|\wedge|\times 1} \tag{3.20}$$

$$\mathbf{D}^T_{\wedge}\mathbf{D}_{\wedge}\frac{d\mathbf{s}^*_{\wedge}}{d\mathbf{D}_{i\wedge_j}} + \frac{d\mathbf{D}^T_{\wedge}\mathbf{D}_{\wedge}}{d\mathbf{D}_{i\wedge_j}}\mathbf{s}^*_{\wedge} - \frac{d\mathbf{D}^T_{\wedge}\mathbf{y}}{d\mathbf{D}_{i\wedge_j}} = \mathbf{0}_{|\wedge|\times 1} \tag{3.21}$$

and finally

$$\underbrace{\frac{d\mathbf{s}^*_\wedge}{d\mathbf{D}_{i\wedge_j}}}_{|\wedge|\times 1} = (\mathbf{D}_\wedge^T\mathbf{D}_\wedge)^{-1}\begin{bmatrix} -\mathbf{D}_{i\wedge_1}\mathbf{s}^*_{\wedge_j} \\ \vdots \\ -\mathbf{D}_{i\wedge_{j-1}}\mathbf{s}^*_{\wedge_j} \\ y_i - \mathbf{D}_{i\wedge}\mathbf{s}^*_\wedge - \mathbf{D}_{i\wedge_j}\mathbf{s}^*_{\wedge_j} \\ -\mathbf{D}_{i\wedge_{j+1}}\mathbf{s}^*_{\wedge_j} \\ \vdots \\ -\mathbf{D}_{i|\wedge|}\mathbf{s}^*_{\wedge_j} \end{bmatrix} \tag{3.22}$$

Values in $\frac{d\mathbf{s}^*_\wedge}{d\mathbf{D}_{i\wedge_j}}$ can be placed at corresponding locations to form $\frac{d\mathbf{s}^*}{d\mathbf{D}_{ij}}$ which will be a k dimensional vector with at most $|\wedge|$ non-zero entries. This allows us to write the gradient with respect to the whole dictionary as

$$\underbrace{\frac{d\mathbf{s}^*}{d\mathbf{D}}}_{k\times nk} = \begin{bmatrix} \frac{d\mathbf{s}^*}{d\mathbf{D}_{11}} & \ldots & \frac{d\mathbf{s}^*}{d\mathbf{D}_{n1}} & \ldots & \frac{d\mathbf{s}^*}{d\mathbf{D}_{nk}} \end{bmatrix} \tag{3.23}$$

in which $n|\wedge|$ columns will each contain at most $|\wedge|$ non-zero entries for a maximum of $n|\wedge|^2$ non-zero entries out of a total of nk^2 entries. Finally, the derivative of the reconstruction error can be computed as

$$\underbrace{\frac{d\mathbf{R}(\mathbf{s}^*,\mathbf{D})}{d\mathbf{D}}}_{1\times nk} = \frac{\partial\mathbf{R}}{\partial\mathbf{s}^*}^T\frac{d\mathbf{s}^*}{d\mathbf{D}} + \frac{\partial\mathbf{R}}{\partial\mathbf{D}} \tag{3.24}$$

$$= \underbrace{\left[-\mathbf{D}^T(\mathbf{y}-\mathbf{D}\mathbf{s}^*) + \lambda sign(\mathbf{s}^*)\right]^T}_{1\times k}\underbrace{\frac{d\mathbf{s}^*}{d\mathbf{D}}}_{k\times nk} + \underbrace{(\mathbf{y}-\mathbf{D}\mathbf{s}^*)^T}_{1\times n}\underbrace{\left(-\frac{d\mathbf{D}\mathbf{s}^*}{d\mathbf{D}}\right)}_{n\times nk} \tag{3.25}$$

3.4.3 Initialization

$\mathbf{D}$: Dictionaries can be initialized to be random or obtained through K-means, K-SVD or any other reconstructive or even discriminative dictionary learning technique. In order to allow a fairer comparison with [2], we initialize via K-means.

$\boldsymbol{\kappa}$: Inference on the random field in (3.7) is very sensitive[4] to the smoothness weights κ_s, κ_s^{dep}, and κ_s^{ind}. Therefore, before learning, it is important to properly initialize them. Initializing $\boldsymbol{\kappa} = \{\kappa_d$, $\kappa_d^{\text{rec}}, \kappa_s, \kappa_s^{\text{ind}}, \kappa_s^{\text{dep}}\}$ to $\{-2, -3, -1, 3, 10\}$ was emiprically found to be a good starting point.

3.5 Experiments and Results

3.5.1 Graz02 Bike Dataset

To validate our formulation, we perform pixel-wise classification on the Graz02 bikes dataset [39]. We select the first 300 images and use odd numbered images for training and even numbered images for testing. For each image, dense SIFT features are computed from overlapping patches of size 32×32 with a grid spacing of 20 pixels. Beliefs for missing pixels are interpolated from their neighborhoods.

A note on smoothing of raw classification results. As noted in [13], the ground-truth masks for the bikes category in the Graz02 dataset include significant background pixels due to the wheel

[4] [2], for instance, do not attempt to learn their smoothness weight $\mathbf{w}_2$ for this reason.

interior being labelled as bike. As a result, it is possible to obtain ‘improved’[5] precision-recall curves by smoothing the pixel-wise classification *e.g.* by a Gaussian filter. This allows the classifications to be closer to the ground-truth even when the system has ‘correctly’[6] learned to classify the wheel interior as background. We therefore show our results (Figure 3.1) with and without this additional smoothing step. It should be noted that this additional smoothing step has been employed by [1, 13, 28].

Table 3.2 shows that our formulation achieves a better EER than the state-of-the-art in dictionary learning based approaches. Prior-segmentation based approaches [33, 34] are the state-of-the-art in such semantic segmentation tasks since they rely on superpixels which naturally lead to adaptive feature domains and boundary preservation. Our results match the superpixel based method of [33] which, like our approach, uses a single scale[7]. The state-of-the-art is achieved by Lempitsky *et al.* [34] who use a multi-scale superpixel approach and a much richer feature set that includes geometric information as well. This obviously suggests future research efforts to employ superpixels instead of fixed grids. But it should not take anything away from the demonstrated benefits of learning dictionaries with neighborhood smoothness priors. Superpixels can be integrated into the CRF framework almost seamlessly (*e.g.* [40, 41]). We have shown that a dictionary learning based approach can yield similar results to state-of-the-art via an appropriate boundary preservation term that leads to learning of dictionaries with neighborhood constraints.

[5] Such quantitative vs. qualitative anamolies have been alluded to in [40].

[6] The VOC dataset, for instance, marks wheel interiors as background.

[7] Even single scale superpixels offer more scale information compared to fixed size patches on fixed grids

CRF+Dictionary		Dictionary		Shape Mask
Ours	[2]	[28]	[13]	[42]
72.1	62.4	69.5	68	61.8

Table 3.2: Comparison of EER (%) of precision-recall curves for pixel-level classfication of Graz02 bike test set. Our results exceed the state-of-the-art in top-down dictionary learning based approaches. See text for comparison with bottom-up super-pixel based segmentation approaches.

Benefit of Training Figure 3.1 demonstrates the benefit of training iterations on the equal error rate (EER) of the precision-recall curve of the Graz02 bike test data. Iteration 0 corresponds to initial dictionaries computed using K-means. Standard dictionary based approaches like [1, 13, 28] use an additional manual Gaussian smoothing step to impose spatial coherence on the pixel labels. For comparison, we perform the same smoothing step after CRF inference. Our learning procedure without additional smoothing was able to learn CRF parameters that out-perform manual smoothing after 8 iterations.

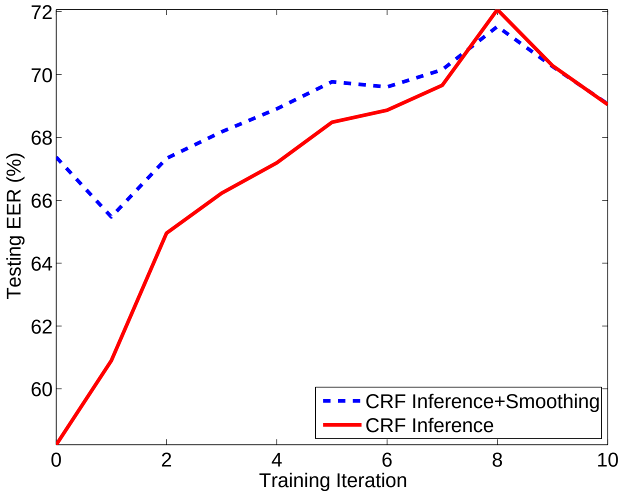


Figure 3.1: Benefit of training iterations on the equal error rate (EER) of the precision-recall curve of the test data for Graz02 bike category. Our learning procedure without additional smoothing was able to learn CRF parameters that out-perform manual smoothing after 8 iterations.

Benefit of Neighborhood Constraints Table 3.3 demonstrates in isolation the benefits of training CRF weight parameters and neighborhood constrained learning of dictionaries. Column 1, for instance, shows that dictionaries learned with neighborhood constraints perform better even when inference is carried out without spatial propagation of labels and row 2 generally shows that our learning formulation gives around 6% improvement over the initial dictionaries. Similarly, column 3 shows that learning of CRF weights results in around 10% improvement.

Table 3.3: EER values on Graz02 bike test set from using **left to right**: no CRF inference, initial CRF weight paramters $\boldsymbol{\kappa}_\mathbf{0}$, learned CRF weight parameters $\boldsymbol{\kappa}^*$ and **top to bottom**: initial K-means dictionaries $\mathbf{D_0}$ and dictionaries learned with neighborhood constraints $\mathbf{D}^*$. The benefit of training CRF weight parameters and the use of neighborhood constraints can be seen in isolation. See text for details.

	No CRF	$\boldsymbol{\kappa}_\mathbf{0}$	$\boldsymbol{\kappa}^*$
$\mathbf{D_0}$	55.1	58.2	66.7
$\mathbf{D}^*$	62.3	63.2	72.1

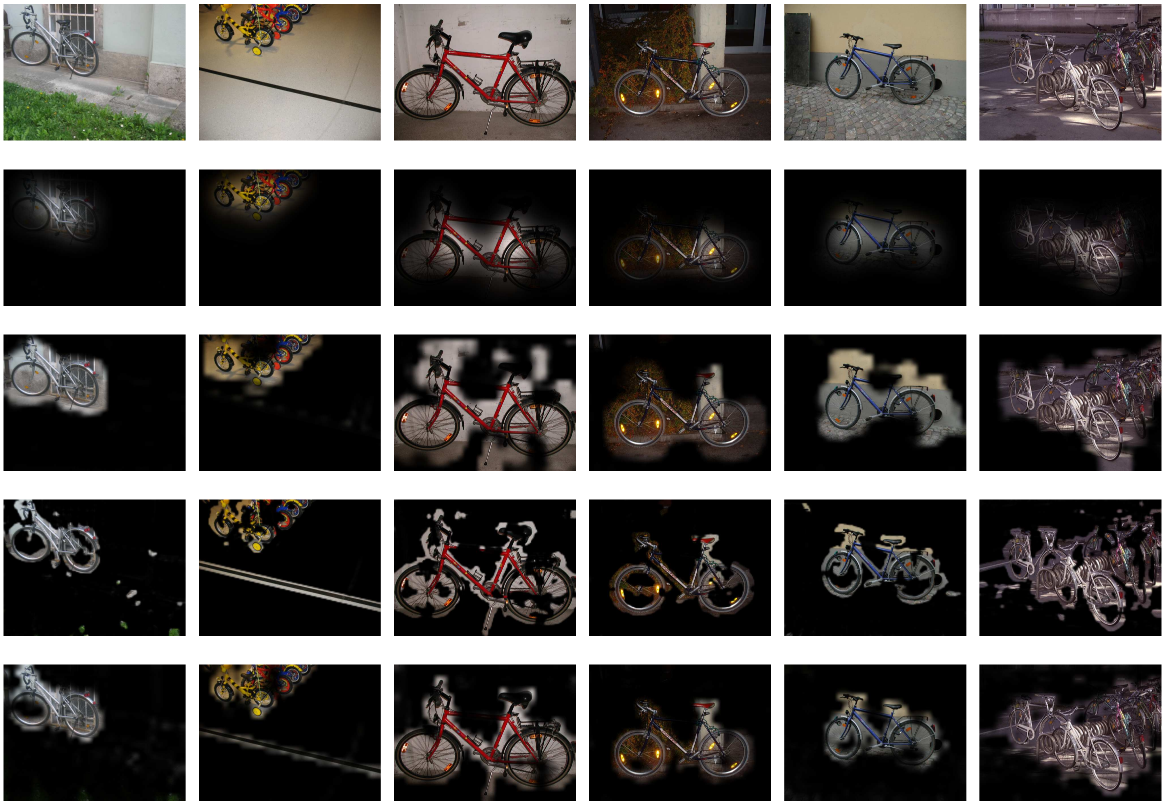


Figure 3.2: Pixel-wise classification results for some test images from the Graz02 bike dataset. **1st Row:** Original. **2nd Row:** Results from the technique in Chapter 2 (Khan & Tappen [28] with vanilla Gaussian smoothing on raw classification without spatial constraints). **3rd Row:** Yang & Yang [2] (CRF with Potts model). The advantages of using boundary-preserving smoothness can be clearly observed in **4th Row:** Our CRF inference on a grid with spacing of 4 pixels followed by interpolation. **5th Row:** Our CRF inference with classification on a grid with spacing of 20 pixels followed by interpolation. The labellings of [28] and [2] appear to be over-smoothed and can tend to cross over object boundaries. Such over-smoothing can lead to an inflated EER value as explained in the text. Implementation of [2] was made available by the original authors.

3.5.2 Weizmann Horses

We compare the benefit of our formulation on the Weizmann Horses dataset trained on the first 25 even-numbered images and tested on the first 25 odd-numbered images. Accuracy criterion is percentage of correctly classified pixels without enforcing spatial consistancy of labels (*i.e.* no CRF inference) for any of the compared methods. Competing dictionary based approaches that are trained without neighborhood smoothness constraints are therefore also tested without neighborhood information to make the comparison favor those approaches. Table 3.4 shows that on this dataset too, our dictionaries perform better than competingapproaches even without CRF inference. Some sample results are shown in Figure 3.3.

Table 3.4: Benefit of learning with spatial smoothness constraints on the Weizmann Horse dataset. Even without CRF inference, our dictionaries have a better pixel classification percentage on the test set compared to dictionaries learned without smoothness constraints.

Method	**Accuracy (%)**
KSVD [7]	72
Disc. Deviation [28]	77
Disc. Softmax [1]	77
Ours	**80**

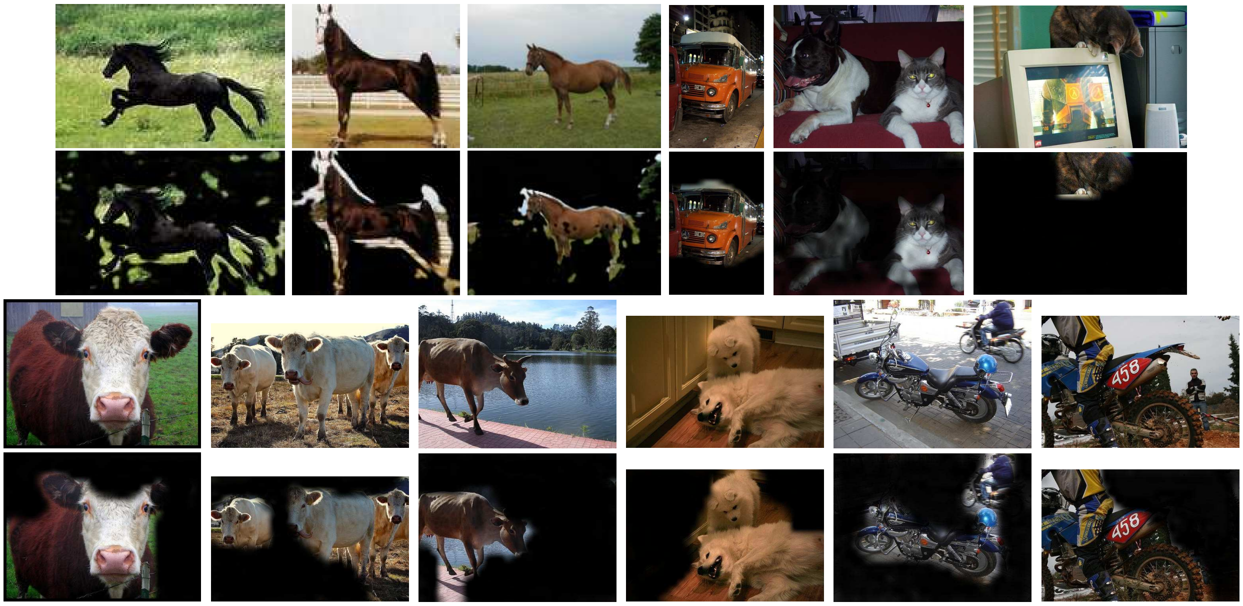


Figure 3.3: Some sample results on the Weizmann Horse dataset and VOC 2007 dataset. The advantage of using neighborhood information can be seen for cat segmentation on the cat and dog image in which large patches on both animals are similar and yet inference using our dictionaries was able to extract the cat with rather crisp boundaries.

3.5.3 VOC 2007

Table 3.5 presents the EER values for figure-ground segmentation on the 20 categories of the Pascal VOC 2007 dataset [43]. Training and testing is performed on the images containing the relevant category. Figure 3.3 shows some sample results. The advantage of using neighborhood information can be seen for cat segmentation on the cat and dog image in which large patches on both animals are similar and yet inference using our dictionaries was able to extract the cat with rather crisp boundaries.

For classification against all other categories in the manner of Yang & Yang [2], we trained a dictionary for the cow category on the 422 training images and tested on all 210 test images. We obtain 8.5% EER on the pixel level compared to the 8% on patches reported in [2]. It should be noted that in [2], going from patch to pixel level was seen to decrease performance by around 10%.

Table 3.5: EER values for figure-ground segmentation on the VOC 2007 dataset.

Class	KSVD [7]	Ours
aeroplane	35.2	**43.7**
bicycle	28.3	**41.2**
bird	35.3	**42.3**
boat	26.3	**35.5**
bottle	16.1	**30.2**
bus	43.7	**69.0**
car	29.1	**43.2**
cat	39.9	**63.3**
chair	9.1	**10.6**
cow	46.0	**70.0**
dining table	38.8	**52.7**
dog	33.3	**51.5**
horse	36.6	**42.0**
motorbike	47.2	**62.9**
person	28.3	**43.0**
potted plant	23.0	**31.4**
sheep	47.5	**54.3**
sofa	21.8	**28.0**
train	54.3	**74.0**
tv/monitor	16.3	**29.1**

3.6 Conclusion

We have introduced a novel discriminative dictionary learning procedure that imposes neighborhood contraints during the learning process. This is motivated by the smoothness and boundary-preserving priors on natural images and achieved by embedding dictionary learning in a CRF framework. As an additional benefit, such smoothness constraints lead to stable dictionary learning which is inherent to the problem of discriminative dictionary learning. Detailed analysis on the Graz02 bike dataset demonstrates a distinct quantitative as well as qualitative advantage over competing dictionary-based approaches.

While results are shown for the 2-class case only, the formulation applies to the general N-class case. However, this can potentially lead to a significant increase in sparse coding computation. An alternative is an N-class learning formulation that performs discriminative sparse coding on a single dictionary for all classes.

An interesting extension is the use of sparse long range random fields [44] for dictionary learning via multiscale information.

CHAPTER 4
SUPER-LATENT LEARNING FOR STRUCTURED PREDICTION

By treating sparse codes as latent variables, we show how a discriminative dictionary can be learned as a super-latent *variable. The formulation is applicable to both unstructured and structured prediction tasks. For structured outputs, we show how dictionaries can be embedded as super-latent variables into conditional random fields (CRF) and optimized for discrimination using the latent structural SVM formulation. As in Chapter 3, the CRF embedding allows us to perform discriminative dictionary learning with spatial smoothness constraints which (i) leads to a more stable minimization and (ii) respects natural smoothness priors. Our smoothness prior takes the form of a* discriminative manifold assumption.

While the approach in Chapter 3 explicitly modelled the reconstruction errors and sparse codes for discrimination, in this chapter we jointly learn a linear classifier with the dictionary. This leads to a two-stage optimization formulation whereby the classifier encourages the dictionary to yield discriminative sparse codes and the dictionary encourages the classifier to perform well on the sparse codes that it yields.

The super-latent formulation is particularly affected by the reconstruction-discrimination tradeoff. Therefore, it must be handled carefully.

4.1 Introduction

Learning with latent variables (also known as missing data or hidden variables) is a well-established area in machine learning. The basic idea is to find optimal values of the latent variables and then learn the optimization variables for these latent values. The two stages are alternated until convergence. Typical examples of this idea include the EM algorithm, the K-means algorithm and latent SVM learning. For dictionary learning too this is a standard technique, exemplified by the KSVD algorithm, for instance.

If the latent variables are generated by an underlying process whose parameters need to be searched over, then the underlying parameters can be termed as super-latent variables. In this chapter, we present an extension of learning with latent variables to learning with latent and super-latent variables. The formulation is applicable to both unstructured as well as structured prediction tasks.

While discriminative dictionary learning has been shown to improve performance on a number of computer vision tasks [1, 14], the learning formulation has traditionaly been restricted to non-structured outputs even when the task is structured prediction. Dictionary embedding in a CRF as in Chapter 3 means that the learning takes place in a structured prediction setting. In this chapter, we learn discriminative dictionaries in a structured prediction setting *via a structured prediction formulation*. We embed the dictionaries in a Conditional Random Field (CRF) and treat the resulting sparse codes as latent variables on the CRF. Optimal CRF parameters and optimal sparse codes can be learned by utilizing the latent Structural SVM formulation [45]. Finally, the optimal dictionaries can be learned from the optimal sparse codes. Since the dictionary determines the latent variables, we treat the dictionary as a super-latent variable.

Conditional Random Fields are a useful tool for structured prediction tasks since they allow dependencies between output variables to be respected. As a result, the output of inference on a CRF is a structured entity. For problems with missing data, CRFs allow inclusion of latent variables. However, parameter learning on CRFs requires computation of the so-called partition function which tends to be intractable. One alternative for learning CRF parameters without computing the partition function is to replace the probabilities by energies and perform energy minimization via a regularized risk minimization approach. This leads to the so-called Structural SVM formulation [46] which can be extended to handle latent variables to yield the Latent Structural SVM formulation [45].

The task that we handle in this work is pixel-level classification of images. That is, for a given image, we find the class label at each pixel. Natural images obey a certain smoothness prior whereby neighboring pixels have similar features and similar labels. In other words, a pixel-wise labeling of an image is a structured entity with dependencies among the pixel labels as opposed to an unstructured entity with pixel labels being independent of each other. The dependencies are determined by the particular neighborhood structure imposed by the random field. Therefore, our task can be formulated as a structured prediction task.

While the approach in Chapter 3 explicitly modelled the reconstruction errors and sparse codes for discrimination, in this chapter we jointly learn a linear classifier with the dictionary. This leads to a two-stage optimization formulation whereby the dictionary is encouraged to yield discriminative sparse codes and the classifier is encouraged to perform well on the sparse codes that the dictionary yields.

We show that the standard formulations of classifier learning with latent variables presented in [47] for unstructured prediction and in [45] for structured prediction cannot be extended in a straight-forward manner for learning super-latent varaibles. For discriminative dictionary learning, this is due, in part, to the reconstruction-discrimination tradeoff. Therefore, we present a modification that is applicable for the discriminative dictionary learning task.

4.1.1 Notation

Let $(x, y, h) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{H}$ denote an image x, its per-pixel labeling y and its per-pixel latent variable vectors h, respectively. Let $\mathcal{S} = \{(x_1, y_1), \ldots, (x_N, y_N)\} \in (\mathcal{X} \times \mathcal{Y})^N$ be the set of input-output pairs. In the following, subscript $_k$ will denote the k_{th} training sample and subscript $_i$ will denote i_{th} pixel location. So, for instance, x_k is the k_{th} training image and x_i is the local feature descriptor around the i_{th} pixel in any arbitrary image x. Similarly, h_k are the latent variables corresponding to image x_k and h_i is the latent variable vector for the i_{th} pixel. Similarly, y_k is an image labeling and y_i is the label at the i_{th} pixel.

4.2 Learning with Latent and Super-Latent Variables

In this section, we first present the standard formulation for learning a classifier with latent variables and then show how this can be extended to optimize a super-latent variable via the latent variables. For classifier learning, the goal is to learn the following linear prediction rule

$$f_{\mathbf{w}}(x) = \arg \max_{(y,h)\in\mathcal{Y}\times\mathcal{H}} \mathbf{w}^T\Phi(x, y, h) \tag{4.1}$$

where $\mathbf{w}$ is the linear predictor and $\Phi(x, y, h)$ is a joint feature vector describing the relationship between input x, output y and latent variables h. For unstructured output y the formulation resembles [47] and for structured output it resembles [45]. The basic idea for learning classifier $\mathbf{w}$ is to alternate between optimal latent variable computation and optimal classifier computation until convergence. This is illustrated in Figure 4.1. The intuition is to learn a classifier on optimal latent variables and then to refine the optimal latent variables based on the learned classfier and so on until convergence.

$$\boxed{\mathbf{w}^* \longleftrightarrow h^*}$$

until convergence

Figure 4.1: Alternating optimization scheme for learning optimal linear classifier $\mathbf{w}^*$ via latent variables h^*.

The latent variables in our case are sparse codes generated from an underlying dictionary. If this dictionary is also to be learned, then it can be treated as a super-latent variable and the standard formulation shown in Figure 4.1 can be extended to include an additional optimization step over the super-latent variable h'. This is illustrated in Figure 4.2. The intuition is to learn a classifier that performs well on optimal latent variables and also to learn a super-latent variable that generates optimal latent variables.

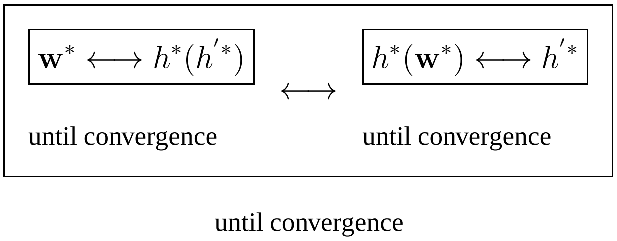

$\mathbf{w}^* \longleftrightarrow h^*(h^{'*})$
$\longleftrightarrow$
$h^*(\mathbf{w}^*) \longleftrightarrow h^{'*}$
until convergence
until convergence
until convergence

Figure 4.2: Alternating optimization scheme for learning optimal linear classifier $\mathbf{w}^*$ with latent variables h^* and super-latent variable $h^{'*}$.

Since the case of unstructured outputs is simpler, we present in the following sections a treatment for the more complex case of structured outputs using a Latent Structural SVM.

4.2.1 Learning The Classifier via Latent Variables

In this section we present the standard method for learning classifier $\mathbf{w}$ via (4.1) for the case of structured outputs.

4.2.1.1 Latent Structural SVM

We first show how to perform the optimization $\mathbf{w}^* \longleftrightarrow h^*(h^{'*})$ in Figures 4.1 and 4.2. Following [45], the linear predictor $\mathbf{w}$ can be learned by the following regularized empirical risk optimization

$$\begin{aligned} \min_{\mathbf{w}} \frac{||\mathbf{w}||^2}{2} &+ C \sum_{k=1}^{N} \left(\max_{(\hat{y},\hat{h})} [\mathbf{w}^T \Phi(x_k, \hat{y}, \hat{h}) + \Delta(y_k, \hat{y}, \hat{h})] \right) \\ &- C \sum_{k=1}^{N} \left(\max_{h} \mathbf{w}^T \Phi(x_k, y_k, h) \right) \end{aligned} \tag{4.2}$$

This minimization can be performed using the method of subgradients. The approach can be outlined as

1. Latent Variable Completion

   (a) Compute $h_k^* = \arg\max_{h \in \mathcal{H}} \mathbf{w}^T \Phi(x_k, y_k, h)$.

2. Loss-Augmented Inference

   (a) Compute $\hat{y}_k = \arg\max_{y \in \mathcal{Y}} \mathbf{w}^T \Phi(x_k, y, h_k) + \Delta(y_k, y, h_k)$ where h_k are some initial values for the latent variables.

(b) Compute $\hat{h}_k = \arg\max_{h \in \mathcal{H}} \mathbf{w}^T \Phi(x_k, \hat{y}_k, h)$ which is the same latent variable completion step as above.

(c) Compute $\hat{y}_k = \arg\max_{y \in \mathcal{H}} \mathbf{w}^T \Phi(x_k, y, \hat{h}_k) + \Delta(y_k, y, \hat{h}_k)$.

3. Sub-gradient Descent

(a) $\mathbf{w}^{\text{new}} = \mathbf{w} - \eta(\mathbf{w} + C \sum_{k=1}^{N} \Phi(x_k, \hat{y}_k, \hat{h}_k) - \Phi(x_k, y_k, h_k^*))$

Next, we show equivalence of Latent Structural SVM with the Conditional Random Field (CRF) formulation containing latent variables. The CRF viewpoint makes the dependency structure more apparent and gives concrete methods for solving the Latent Variable Completion and Loss-Augmented Inference steps required for solving the Latent Structural SVM formulation.

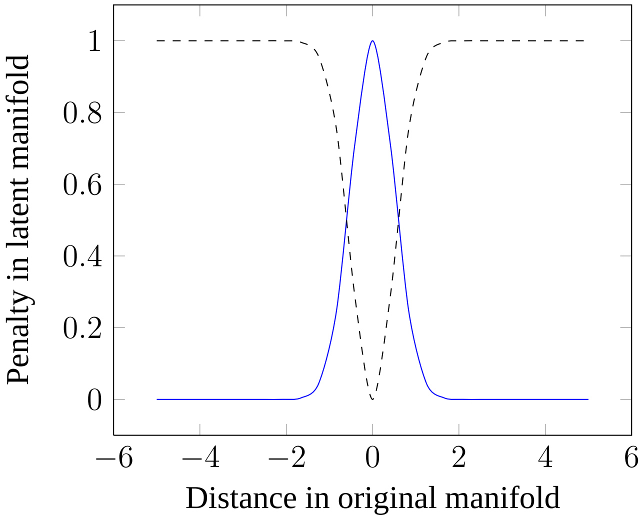


Figure 4.3: Weights to enforce the *discriminative manifold assumption* from Equation (4.4). For similar labels, Euclidean distance in sparse space is penalized according to inverse distance in feature space (solid curve). For different labels, the relationship is proportional (dashed curve).

4.2.1.2 Conditional Random Field

Structural dependencies between output variables in a Structural SVM are represented by the joint feature vector Φ. Such inter-dependencies between output variables can also be represented using a Conditional Random Field (CRF). The basic idea is to have a Markovian dependency of output labels at the pixel level. The energy of a particular labeling y can be computed as

$$E_{\mathbf{w}}(y|x,h) = \sum_{i\in\mathcal{V}} y_i \mathbf{w}_1^T h_i - \sum_{j\in\mathcal{N}_i} \mathbf{w}_2 \frac{m_{ij}}{2} ||h_i - h_j||^2 \tag{4.3}$$

where $\mathcal{V}$ is a structured grid of pixel locations, $\mathcal{N}_i$ is the set of neighboring grid locations of the i_{th} pixel and m_{ij} is a weight for enforcing the manifold assumption – similarity in the latent manifold should reflect similarity in the original manifold if labels are similar, otherwise the relationship should be inverted. The weights can be defined as

$$m_{ij} = \begin{cases} e^{-\eta||x_i - x_j||^2} & \text{if } y_i = y_j \\ \\ 1 - e^{-\eta||x_i - x_j||^2} & \text{if } y_i \neq y_j \end{cases} \tag{4.4}$$

in order to penalize non-smoothness of latent variables according to the smoothness of the input features and the class labels (see Figure 4.3). For instance, if adjacent input signals are similar and belong to the same class, then the latent variables should be similar too. Such a weighting allows our formulation to respect the *discriminative manifold assumption* via class dependent, spatial constraints on the input signals and latent variables. When inferring labels on the random field, these constraints lead to boundary-preservation. It should be noted that the structural dependency of the output labeling y is determined by the definition of the neighborhood $\mathcal{N}_i$ which in our case consists of simple pairwise neighbors.

Denoting $\mathbf{w} = [\mathbf{w}_1; \mathbf{w}_2]$ and $\Phi_i(x, y, h) = [y_i h_i; -\sum_{j \in \mathcal{N}_i} \frac{m_{ij}}{2} ||h_i - h_j||^2]$, we can express our joint feature vector as $\Phi(x, y, h) = \sum_{i \in \mathcal{V}} \Phi_i(x, y, h)$ and the CRF energy as $E_{\mathbf{w}}(y|x, h) = \mathbf{w}^T \Phi(x, y, h)$. Therefore, the linear prediction rule of the Latent Structural SVM from Section 4.2.1.1 can be represented in terms of a CRF and vice versa. The view in terms of a CRF makes the neighborhood structure more apparent.

Furthermore, for the case of separable loss functions, the Loss-Augmented Inference step required for solving the Latent Structural SVM formulation reduces to standard MAP-inference on the CRF. An appropriate choice of the loss function for measuring goodness of a labeling is the Hamming loss given by[1] $\Delta(y_k, y, h_k) = \sum_{i \in \mathcal{V}} \bar{\delta}_{y_{ki} y_i}$ which simply counts the number of locations where labeling y differs from the ground-truth labeling y_k. Since it can be decomposed into per-node loss terms, the unary potentials of the CRF can be appropriately modified. Standard MAP-inference on the CRF will now yield the solution to the Loss-Augmented Inference step.

We show in the next section that, for our task, the Latent Variable Completion step for solving the Latent Structural SVM formulation amounts to discriminative sparse coding on the CRF.

Latent Variable Completion In the language of Latent SVMs, latent variable completion requires finding the latent variables that maximize the score for the linear prediction rule. Since the latent variables in our case are sparse codes under a dictionary $\mathbf{D}$ and since they reside on a random field with an underlying neighborhood structure, latent variable completion amounts to *discriminative sparse coding on a random field*. Specifically, the problem that needs to be solved is

$$h^* = \arg\max_{h \in \mathcal{H}} \mathbf{w}^T \Phi(x, y, h) \tag{4.5}$$

$$= \arg\min_{h} \frac{1}{2} ||x - \mathbf{D}h||_F^2 - \alpha \mathbf{w}^T \Phi(x, y, h) + \lambda \sum_i ||h_i||_1 \tag{4.6}$$

where $\alpha > 0$ handles the reconstruction-discrimination tradeoff. Because of the spatial smoothness term $||h_i - h_j||^2$ in $\Phi_i(x, y, h)$, we avoid joint optimization of the h_i's by optimizing over each sparse code iteratively. This is an instance of the so-called Cyclic Coordinate Descent approach.

[1] Even though Hamming loss only requires the labelings, the expression $\Delta(y_k, y, h_k)$ reflects the fact that the latent variables h_k determine the labeling y.

Specifically, we optimize over each sparse code by fixing all other sparse codes

$$h_i^* = \arg\min_s \frac{1}{2}||x_i - \mathbf{D}s||_2^2 - \alpha \mathbf{w}^T \Phi_i(x, y, s) + \lambda ||s||_1 \tag{4.7}$$

Expressed as $\min_s L(s) + \lambda||s||_1$ where $L(s)$ is smooth, this optimization is exactly what is solved by the LASSO [48]. However, the inclusion of the ℓ_2 norm of sparse code s via Φ_i reduces the sparsity inducing effect of the ℓ_1 norm in a manner similar to the elastic net formulation of [49]. Therefore, the value of parameter $\mathbf{w}_2$ within the linear predictor controls how much sparsity is retained. A smaller value retains sparsity. We use a modification of LASSO-shooting [50] to solve (4.7). LASSO-shooting is also an example of Cyclic Coordinate Descent. Next we give brief derivation of the algorithm.

Noting that the ℓ_1 norm $||s||_1$ has a constant gradient of $sign(s)$ for non-zero entries in s, a necessary and sufficient condition on the non-zero entries of s for s to be a minimizer of $L(s) + \lambda||s||_1$ is

$$\frac{\partial L(s)}{\partial s_p} + sign(s_p) = 0 \qquad \text{for } \{p : s_p \neq 0\} \tag{4.8}$$

which can be written as

$$d_p^T d_p s_p + \sum_{l \neq p} d_p^T d_l s_l - d_p^T x_i - \alpha y_i \mathbf{w}_{1_p} + \alpha \mathbf{w}_2 \sum_{j \in \mathcal{N}_i} m_{ij}(s - h_j)_p = -\lambda sign(s_p) \tag{4.9}$$

The left-hand side is a linear function of s_p with positive slope $g = d_p^T d_p + \alpha \mathbf{w}_2 \sum_{j \in \mathcal{N}_i} m_{ij}$ and intercept $c = \sum_{l \neq p} d_p^T d_l s_l - d_p^T x_i - \alpha y_i \mathbf{w}_{1_p} - \alpha \mathbf{w}_2 \sum_{j \in \mathcal{N}_i} m_{ij} h_{j_p}$. The right-hand side is an inverted step function with a step of -2λ at $s_p = 0$. As can be seen from Figure 4.4, if intercept c of the right-hand side (dashed line) is less than or equal to $|\lambda|$, then Equation (4.9) has no solution and therefore s_p cannot be part of the active set of s (*i.e.* s_p must be zero). Otherwise, a solution for s_p exists and can be found using elementary operations. This gives us the following mechanism for computing s_p

$$s_p = \begin{cases} \frac{\lambda - c}{g} & \text{if } c > \lambda \\ \frac{-\lambda - c}{g} & \text{if } c < \lambda \\ 0 & \text{if } c \leq |\lambda| \end{cases} \tag{4.10}$$

The process can be cycled through all entries of s until convergence to yield a discriminative sparse code that maximizes the score on the linear predictor $\mathbf{w}$ while respecting neighborhood constraints imposed by the CRF grid structure. Therefore, this step amounts to *discriminative sparse coding on a random field*. The converged sparse code s is the solution h_i^* to (4.7) for the i_{th} location. This iteratively leads to the solution h_k^* for the whole of training sample k (Objective (4.6)). In terms of the Latent Structural SVM formulation, computing optimal sparse codes h_k^* for all training samples completes the Latent Variable Completion step.

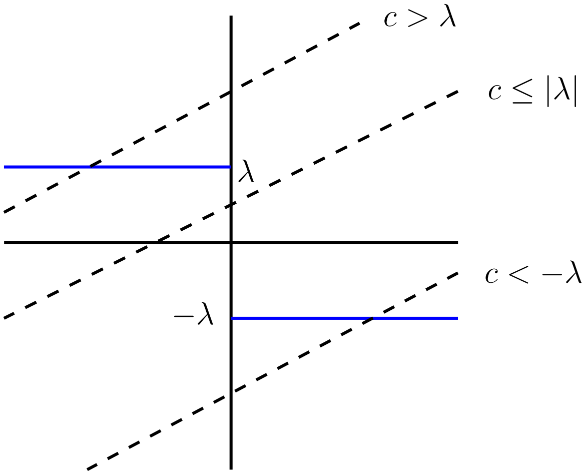
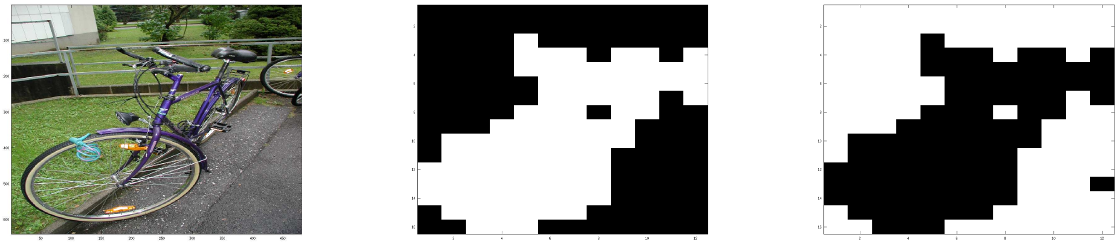


Figure 4.4: LASSO-Shooting step. Dashed lines represent the left-hand side of Equation (4.9) for intercepts $\{> \lambda, \leq |\lambda|, < -\lambda\}$ while the solid step-function in blue is the right-hand side. It can be seen that both sides will never be equal (*i.e.* no solution) iff the intercept of the left-hand side is less than or equal to $|\lambda|$.

Since Latent Variable Completion is required for Loss Augmented Inference, we present some results for Loss Augmented Inference only to illustrate the applicability of the methods in the pixel-wise calssification task for a few images from the Graz02 bike dataset. By weighting the Hamming loss appropriately, the inference step should yield most violating labelings that are close to the inverse of the ground-truth labeling. This behavior is illustrated in Figures 4.5 and 4.6. Convergence behavior can be seen in Figure 4.7.



Original Ground-truth Most violating labeling

Figure 4.5: Loss Augmented Inference to find the most violating labeling. Weighted Hamming loss was used as the loss function Δ to encourage the most violated labeling to be close to the inverse of the ground-truth labeling. Result shown is after convergence.

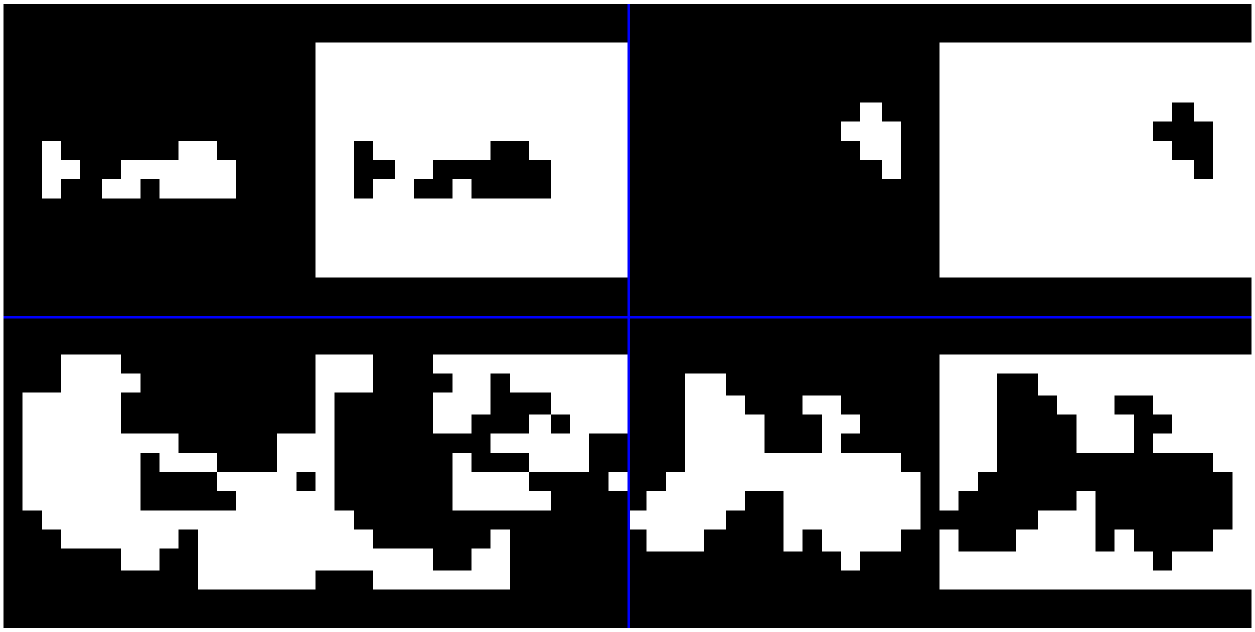


Figure 4.6: Some more results for Loss Augmented Inference to find most violating labelings. Each block shows the ground-truth labeling on the left and the most violating labeling on the right.

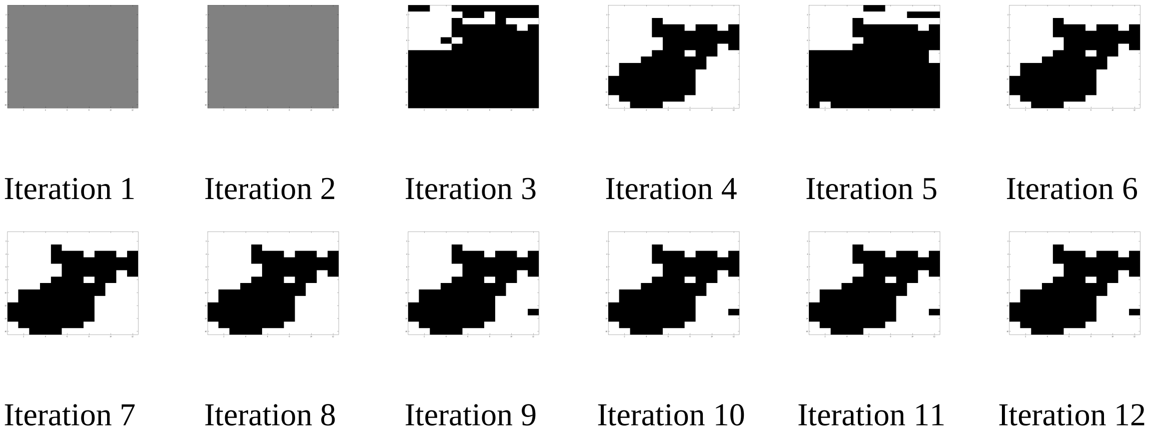

Iteration 1
Iteration 2
Iteration 3
Iteration 4
Iteration 5
Iteration 6
Iteration 7
Iteration 8
Iteration 9
Iteration 10
Iteration 11
Iteration 12

Figure 4.7: Convergence behavior of Loss Augmented Inference to find the most violated labeling shown in Figure 4.5.

4.2.2 Learning the Super-latent Variable via Latent Variables

In this section we show how to perform the optimization $h^* \longleftrightarrow h^{'*}$ from Figure 4.2. To do so, we first give a concrete meaning to the latent and super-latent variables. As mentioned earlier, we treat the sparse codes under a dictionary $\mathbf{D} \in \mathcal{D}$ as our latent variables. Therefore, the dictionary $\mathbf{D}$ is our super-latent variable. It is easy to see that the Latent Variable Completion step in Section 4.2.1.1 yields sparse codes that maximize the score of our linear prediction rule. Let X be the set of all input signals and H^* be the corresponding set of optimal sparse codes. The goal is to find a dictionary $\mathbf{D}^*$ that yields sparse codes H^* when presented with input signals X. If such a dictionary can be found, then, by construction, it will yield sparse codes that score highly on the linear prediction rule. In other words, the dictionary would be discriminative.

After independent derivation of this work, it was found that atleast two approaches [21, 22] already exist in the computer vision literature that construct discriminative dictionaries in similar fashion. The idea is straight-forward – find dictionary $\mathbf{D}^*$ that minimizes the reconstruction error between the input signals X and their reconstructions using the given optimal sparse codes H^*. That is

$$\mathbf{D}^*(X, H^*) = \arg\min_{\mathbf{D}\in\mathcal{D}} \frac{1}{2}||X - \mathbf{D}H^*||_F^2 \tag{4.11}$$

This can be solved using either the Lagrangian dual formulation from [51] or, more simply, using the method of optimal directions (MOD) [6] as

$$\mathbf{D}^*(X, H^*) = \pi(XH^{*^T}(H^*H^{*^T})^{-1}) \tag{4.12}$$

where operator π is a projection of the dictionary atoms onto the ℓ_2-ball.

Our overall 2-level optimization scheme is somewhat similar to standard techniques for solving models with latent variables. Other examples include EM, K-means and for dictionary learning, K-SVD. The difference in our approach is that at each level we perform another alternating optimization. The hierarchy of levels is due to the presence of a super-latent variable. The role of the latent variables can be understood in each optimization level as follows:

1. $\mathbf{w}^* \longleftrightarrow \mathbf{H}^*$: Learn classifier that performs well on *discriminative* sparse codes.

2. $\mathbf{H}^* \longleftrightarrow \mathbf{D}^*$: Learn dictionary that yields *discriminative* sparse codes.

So the latent variable completion step encourages the dictionary to become discriminative by yielding sparse codes that are discriminative and they encourage the classifier to improve its performance on such discriminative sparse codes.

4.2.3 Convergence Analysis

Standard dictionary learning solves

$$\arg \min_{(\mathbf{D},H)\in\mathcal{D},\mathcal{H}} = \frac{1}{2}||X - \mathbf{D}H||_F^2 \tag{4.13}$$

by alternating between a *sparse coding step*

$$H^* = \arg\min_{H\in\mathcal{H}} \frac{1}{2}||X - \mathbf{D}^*H||_F^2 + \lambda \sum_i ||h_i||_1 \tag{4.14}$$

and a *dictionary update step*

$$\mathbf{D}^*(X, H^*) = \arg\min_{\mathbf{D}\in\mathcal{D}} \frac{1}{2}||X - \mathbf{D}H^*||_F^2 \tag{4.15}$$

until convergence of the reconstruction error $\frac{1}{2}||X - \mathbf{D}^*H^*||_F^2$. Since both steps minimize the same objective function and the value of the objective function reduces at each update, convergence is guaranteed.

For our problem of super-latent dictionary learning, however, we modify the sparse coding step above by adding a discriminative term $-\alpha\mathbf{w}^T\Phi$ to objective (4.14). As a result, convergence properties of standard dictionary learning do not apply to our formulation. The discriminative sparse coding step has an objective function different from the dictionary update step. This, once again, is a manifestation of the reconstruction-discrimination tradeoff. Parameter α plays an important role by regularizing the impact of the discriminative term. Therefore, a continuation strategy can be employed whereby α is gradually increased from an initially small value that favors stable reconstruction to a larger value that favors less stable discrimination.

4.3 Dealing with the Reconstruction-Discrimination Tradeoff

A potential problem with the optimization scheme shown in Figure 4.2 is that in practical applications the reconstruction-discrimination tradeoff limits the ability of the super-latent variable $h^{'*}$ to generate optimal latent variables $h^*(h^{'*})$. As a result, a disconnect appears between the two optimization stages. The classifier learns to perform well on optimal latent variables h^* but the super-latent variable is never able to produce such optimal latent variables. To force a connection between these two stages, it is better to train the classifier on the latent variables that the super-latent variable is able to yield, *i.e.* $h(h^{'*})$ instead of h^*. This is illustrated in Figure 4.8. For dictionary learning, the role of the latent variables can be understood in each optimization level as follows:

1. $\mathbf{w}^* \longleftrightarrow \mathbf{H}(\mathbf{D}^*)$: Learn classifier that performs well on latent sparse codes under super-latent dictionary $\mathbf{D}^*$.

2. $\mathbf{H}^* \longleftrightarrow \mathbf{D}^*$: Learn super-latent dictionary that yields *discriminative* latent sparse codes.

So the latent variable completion step encourages the dictionary to become discriminative by yielding sparse codes that are discriminative. The classifier is then encouraged to improve its performance on sparse codes that dictionary $\mathbf{D}^*$ is able to yield.

$$\boxed{\begin{array}{ccc} \boxed{\mathbf{w}^* \longleftrightarrow h(h^{'*})} & \longleftrightarrow & \boxed{h^* \longleftrightarrow h^{'*}} \\ \text{until convergence} & & \text{until convergence} \end{array}}$$

until convergence

Figure 4.8: Overall scheme for alternating optimization of the optimal linear classifier $\mathbf{w}^*$ and latent sparse codes h^* and super-latent dictionary $h^{'*}$. Since the super-latent variable can yield sub-optimal latent variables because of the reconstruction-discrimination tradeoff, the classifier is forced to perform well on the latent variables that the super-latent variable yields, *i.e.* $h(h^{'*})$ instead of h^*.

For optimization $\mathbf{w}^* \longleftrightarrow h(h^{'*})$ in Figure 4.8, Latent Variable Completion corresponds to computing $h(h^{'*})$. For our task, computing $h(h^{'*})$ merely corresponds to a standard sparse coding step. This is in contrast to the Latent Variable Completion step in 4.2.1.1 where computation of h^* corresponds to a discriminative sparse coding step based on the classifier $\mathbf{w}$ for our task. The optimization for learning $\mathbf{w}$ can therefore be simplified as

$$\min_{\mathbf{w}} \frac{||\mathbf{w}||^2}{2} + C \sum_{k=1}^{N} \left(\max_{(\hat{y})} [\mathbf{w}^T \Phi(x_k, \hat{y}, h(h^{'*})) + \Delta(y_k, \hat{y}, h(h^{'*}))] \right) \\ - C \sum_{k=1}^{N} \left(\mathbf{w}^T \Phi(x_k, y_k, h(h^{'*})) \right) \tag{4.16}$$

The corresponding subgradient method can be outlined as

1. Latent Variable Completion

   (a) Compute $h_k(h^{'*})$.

2. Loss-Augmented Inference

   (a) Compute $\hat{y}_k = \arg\max_{y \in \mathcal{H}} \mathbf{w}^T \Phi(x_k, y, h_k(h^{'*})) + \Delta(y_k, y, h_k(h^{'*}))$.

3. Sub-gradient Descent

   (a) $\mathbf{w}^{\text{new}} = \mathbf{w} - \eta(\mathbf{w} + C \sum_{k=1}^{N} \Phi(x_k, \hat{y}_k, h_k(h^{'*})) - \Phi(x_k, y_k, h_k(h^{'*})))$

To isolate the effect of dictionary learning using discriminative sparse codes, we performed learning on a subset of the Graz02 bikes dataset using no spatial constraints (*i.e.* unstructured output). Figure 4.9 demonstrates the learning effect of our 2 level optimization scheme when initialized using a dictionary computed via K-means and a classifier computed via a linear SVM. Training set accuracy improved from around 68% to around 92% before the reconstruction-discrimination tradeoff made the learning unstable (as can be seen from the drastic changes in accuracy towards the end).

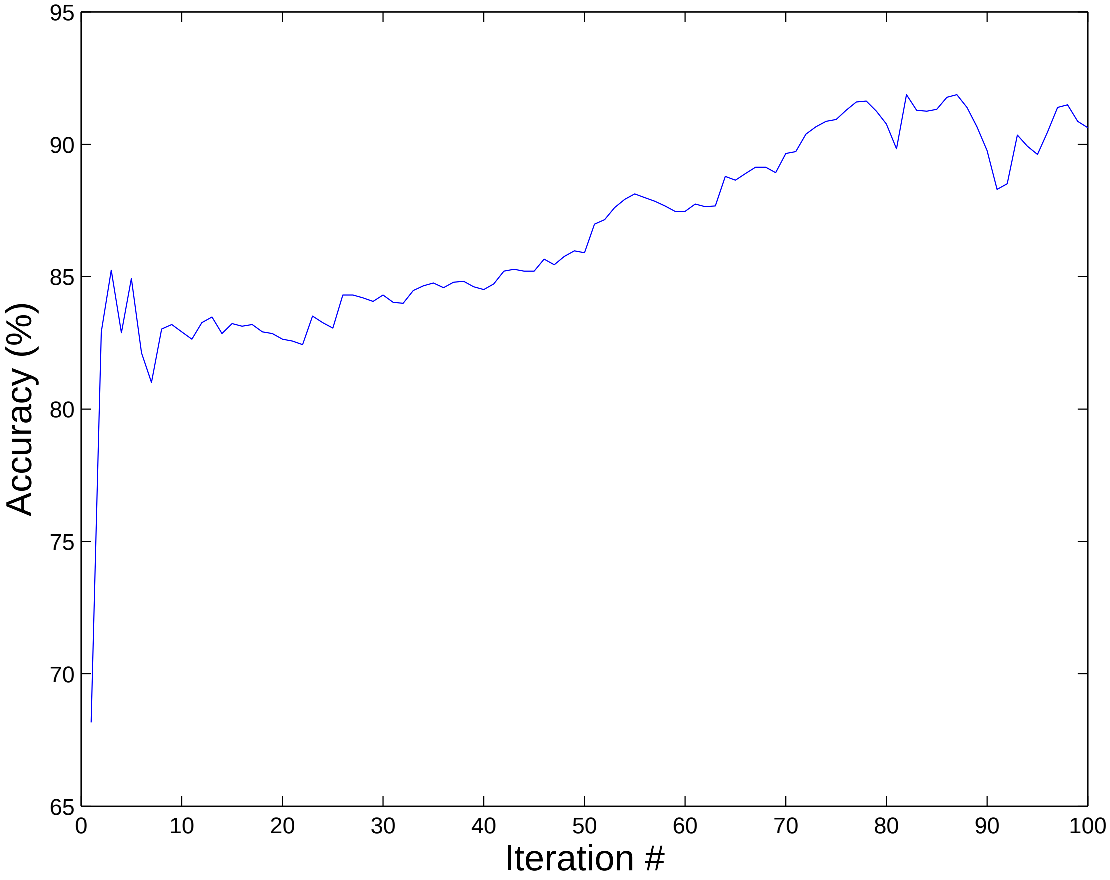


Figure 4.9: Learning effect of our 2 level optimization scheme when initialized using a dictionary computed via K-means and a classifier computed via a linear SVM. Training set accuracy improved from around 68% to around 92% before the reconstruction-discrimination tradeoff made the learning unstable (as can be seen from the drastic changes in accuracy towards the end).

4.4 Discussion

In this chapter, we have laid the foundations of super-latent dictionary learning. The super-latent learning formulation applies to both structured and unstructured prediction tasks.

For structured prediction, the formulation corresponds to a Latent Structural SVM whose equivalence to a CRF has been explained. The CRF view-point makes the dependency structure of the output more apparent and allows standard CRF inference algorithms to be used for solving the SVM formulation. Structured output allows us to impose spatial constraints in the learning formulation. For tasks without structured outputs and/or not requiring spatial constraints, our super-latent formulation corresponds to a Latent SVM which leads to simpler optimization.

A key step in super-latent dictionary learning involves computing a dictionary given the set of signals and their corresponding discriminative sparse codes. Since the sparse codes are discriminative, the Hessian for the dictioanry learning problem approaches singularities – a manifestation of the reconstruction-discrimination tradeoff. Therefore, it is important to use a continuation strategy to start from a stable reconstructive setting and gradually move towards the more unstable discriminaitve setting.

Another potential problem with a straight-forward extension of the latent learning frameworks from SVM literature [47, 45] to the super-latent learning framework is the possibility of a disconnect between the two learning stages. A straight-forward extension assumes that the super-latent dictioanry is able to yield optimal latent sparse codes on which a classifier can then be trained. But because of the reconstruction-discrimination tradeoff once again, the super-latent dictionary is only able to yield sub-optimal latent sparse codes. As a result when classifier performance is optimized over the optimal latent sparse codes, it is optimized over something that the super-latent dictionary is not able to yield. Hence there can be a disconnect between the two optimization stages. We have therefore proposed to optimize the classifier over latent sparse codes that a discriminative dictionary actually yields instead of optimally discriminative sparse codes that the dictionary might never be able to yield. This way the two optimization stages can better influence each other. That is, the classfier will encourage the dictionary to yield more discriminative sparse codes while the dictionary will encourage the classifier to perform well on the sparse codes that it yields.

CHAPTER 5
CONCLUSIONS

We have investigated the use of dictionary learning for discriminative tasks on natural images. We have presented that the reconstruction-discrimination trade-off is a fundamental, inherent issue when it comes to discriminative dictionary learning. Discriminative learning will necessarily lead to ill-conditioned dictionaries. To this end, we have presented, in Chapter 2, a method for mitigating the ill-conditioning problem. Specifically, we have introduced the *discriminative deviation* function to yield a more principled formulation for handling the reconstruction-discrimination tradeoff. We have shown that discriminative deviation can be seen as a lower-bound on the discriminative softmax function function and hence our formulation is a faithful lower-bound on the formulation of Mairal *et al.* [1].

Moreover, since natural images obey a strong smoothness prior, we have shown in Chapter 3 that inclusion of spatial smoothness constraints in the learning formulation benefits dictionary learning for natural image analysis. Such smoothness constraints can be incorporated by embedding dictionaries in a CRF. We have introduced a novel discriminative dictionary learning procedure that imposes neighborhood contraints during the learning process in addition to the inference process. We have also incorporated a boundary-preserving prior on natural images. As an additional benefit, such smoothness constraints lead to more stable dictionary learning which is inherent to the problem of discriminative dictionary learning. Detailed analysis on the Graz02 bike dataset demonstrates a distinct quantitative as well as qualitative advantage over competing dictionary-based approaches.

Finally, we have laid the foundations of *super-latent* dictionary learning in Chapter 4. The super-latent learning formulation applies to both structured and unstructured prediction tasks. In fact, it does not have to be limited to dictionary learning only. It can be applied to problems involving latent variables whose generating super-latent variables also need to be optimized for. For the discriminative dictionary learning task, by treating sparse codes as latent variables embedded in a CRF, we have shown that dictionary learning can also be performed via the Latent Structural SVM formulation. The sparse code yielding dictionary is treated as a super-latent variable in this case. We have shown that a key component of the solution rests on solving a novel problem of *discriminative sparse coding on a random field*. Not surprisingly, the reconstruction-discrimination tradeoff introduces particular challenges for discriminative super-latent learning and need to be handled carefuly.

Given that the reconstruction-discrimination tradeoff is a fundamental hurdle for discriminative dictionary learning, it will be worthwhile for future research efforts to be expended at intelligent ways of avoiding/handling it.

We conclude with a brief take on the justification for sparsity and discuss whether sparsity can be harmful.

5.1 Is Sparsity Harmful?

In light of the reconstruction-discrimination trade-off, an important question to ask would be whether sparsity is beneficial at all? For classficiation tasks, the final goal is discriminability. Does sparsity have any benefit in terms of learning discriminative representations – or worse, does sparsity lead to the reconstruction-discrimination tradeoff? Such questions have been addressed in works such as [52, 53, 54] based on earlier claims that sparse representations lead to better classification. These works have justifiably concluded[1] that the quality of being sparse does not necessarily lead to better classification accuracy. However, we contend that the goal of sparsity in discriminative settings should not be better classfication in the first place. The goal of sparsity in discriminative settings should be just that – sparsity. That is, given a discriminative model, does there exist a simpler representation?

We first discuss the justification of sparsity in the pure reconstructive setting. In a reconstructive setting, sparsity can be justified via Occam's razor – *'it is vain to do with more what can be done with fewer'*. That is, a signal should be represented with the least possible amount of complexity. There is no point retaining redundant information in a representation. Indeed, standard vector quantization is the *sparsest possible representation* of a signal. In relation to vector quantization, sparse coding is, in fact, a less sparse representation. This can be viewed as a sort of anti-razor[2]. So sparse coding in fact tries to find a model that is simple but not over-simplified.

[1] It is worth noting that [53] concludes sparsity to be beneficial for dictionary learning but not for the classification task.

[2] 'It is vain to try to do with fewer what requires more.'[55]

For the discriminative setting, we contend that Occam's razor should still apply – we search for the simplest *discriminative* representation. Applied to the dictionary learning task in particular, the goal is to make dictionaries discriminative by enforcing constraints on intermediate sparse codes. Requiring these intermediate representations to be sparse is just a manifestation of Occam's razor.

It must be noted that the reconstruction-discrimination trade-off is not caused by the sparsity requirement. It is caused by the discriminability requirement and will persist regardless of whether the intermediate representations are sparse codes or dense codes. As explained in Section 1.4.4, the trade-off can be understood in terms of the Hessian of the dictionary learning problem. For a given class of signals, the Hessian can be understood as the outer-product of the sparse codes. Discriminability leads to class-specific sparse codes that resemble each other and hence the Hessian will approach a singular matrix. *Crucially, this will be true even when the codes are not sparse anymore and therefore sparsity is not the real culprit.*

LIST OF REFERENCES

[1] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, “Discriminative learned dictionaries for local image analysis.,” in *CVPR*, 2008.

[2] J. Yang and M.-H. Yang, “Top-down visual saliency via joint CRF and dictionary learning.,” in *CVPR*, 2012.

[3] S. Mallat and Z. Zhang, “Matching pursuit with time-frequency dictionaries,” *IEEE Transactions on Signal Processing*, vol. 41, pp. 3397–3415, 1993.

[4] S. S. Chen, D. L. Donoho, Michael, and A. Saunders, “Atomic decomposition by basis pursuit,” *SIAM Journal on Scientific Computing*, vol. 20, pp. 33–61, 1998.

[5] S. P. Lloyd, “Least square quantization in pcm,” tech. rep., Bell Telephone Laboratories, 1957.

[6] K. Engan, S. O. Aase, and J. H. Husøy, “Frame based signal compression using method of optimal directions (mod).,” in *ISCAS (4)*, 1999.

[7] M. Aharon, M. Elad, and A. Bruckstein, “K-svd: Design of dictionaries for sparse representation,” in *SPARS*, 2005.

[8] N. Slonim and N. Tishby, “Agglomerative information bottleneck,” in *NIPS*, pp. 617–623, MIT Press, 1999.

[9] B. Fulkerson, A. Vedaldi, and S. Soatto, “Localizing objects with smart dictionaries,” in *ECCV*, 2008.

[10] J. Liu and M. Shah, “Learning human actions via information maximization,” in *CVPR*, 2008.

[11] J. Liu, Y. Yang, and M. Shah, “Learning semantic visual vocabularies using diffusion distance,” in *CVPR*, 2009.

[12] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, “Supervised dictionary learning,” in *NIPS*, 2008.

[13] I. Ramirez, P. Sprechmann, and G. Sapiro, “Classification and clustering via dictionary learning with structured incoherence and shared features.,” in *CVPR*, 2010.

[14] D.-S. Pham and S. Venkatesh, “Joint learning and dictionary construction for pattern recognition,” in *CVPR*, IEEE Computer Society, 2008.

[15] Q. Zhang and B. Li, “Discriminative k-svd for dictionary learning in face recognition,” in *CVPR*, pp. 2691–2698, IEEE, 2010.

[16] Z. Jiang, Z. Lin, and L. S. Davis, “Learning a discriminative dictionary for sparse coding via label consistent k-svd,” in *CVPR*, pp. 1697–1704, 2011.

[17] E. J. Candès and T. Tao, “Decoding by linear programming,” *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, 2005.

[18] M. Yaghoobi, T. Blumensath, and M. Davies, “Regularized dictionary learning for sparse approximation,” in *European Signal Processing Conference (EUSIPCO)*, 2008.

[19] W. Dai, T. Xu, and W. Wang, “Simultaneous codeword optimization (simco) for dictionary update and learning,” *CoRR*, vol. abs/1109.5302, 2011.

[20] M. Zheng, J. Bu, C. Chen, C. Wang, L. Zhang, G. Qiu, and D. Cai, “Graph regularized sparse coding for image representation,” *IEEE Transactions on Image Processing*, vol. 20, no. 5, pp. 1327–1336, 2011.

[21] S. Gao, I. W.-H. Tsang, L.-T. Chia, and P. Zhao, “Local features are not lonely–laplacian sparse coding for image classification,” in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 3555–3561, IEEE, 2010.

[22] H. Guo, Z. Jiang, and L. S. Davis, “Discriminative dictionary learning with pairwise constraints,” in *ACCV*, 2012.

[23] K. Huang and S. Aviyente, “Sparse representation for signal classification,” in *NIPS*, pp. 609–616, 2006.

[24] Q. Qiu, J. Zhoulin, and R. Chellappa, “Sparse dictionary-based representation and recognition of action attributes,” in *ICCV*, 2011.

[25] H. Wang, M. M. Ullah, A. Kläser, I. Laptev, and C. Schmid, “Evaluation of local spatio-temporal features for action recognition,” in *BMVC*, 2009.

[26] M. D. Rodriguez, J. Ahmed, and M. Shah, “Action mach a spatio-temporal maximum average correlation height filter for action recognition.,” in *CVPR*, 2008.

[27] A. Yao, D. Uebersax, J. Gall, and L. J. V. Gool, “Tracking people in broadcast sports,” in *DAGM*, 2010.

[28] N. Khan and M. Tappen, “Stable discriminative dictionary learning via discriminative deviation,” in *ICPR*, 2012.

[29] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, “Robust face recognition via sparse representation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, pp. 210–227, Feb. 2009.

[30] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, “Non-local sparse models for image restoration,” in *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 2272–2279, IEEE, 2009.

[31] J. Yang, Z. Wang, Z. Lin, S. Cohen, and T. Huang, “Couple Dictionary Training for Image Super-resolution,” *IEEE Transactions on Image Processing*, 2012.

[32] S. Kumar and M. Herbert, “Discriminative random fields: A discriminative framework for contextual interaction in classification.,” in *ICCV*, 2003.

[33] B. Fulkerson, A. Vedaldi, and S. Soatto, “Class segmentation and object localization with superpixel neighborhoods,” in *Proc. ICCV*, 2009.

[34] V. Lempitsky, A. Vedaldi, and A. Zisserman, “A pylon model for semantic segmentation,” in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2011.

[35] J. Lafferty, M. A, and P. F, “Conditional random fields: Probabilistic models for segmenting and labeling sequence data.,” in *ICML*, 2001.

[36] S. V. N. Vishwanathan, N. N. Schraudolph, M. W. Schmidt, and K. P. Murphy, “Accelerated training of conditional random fields with stochastic gradient methods,” in *In ICML*, pp. 969–976, 2006.

[37] J. A. Bagnell and D. M. Bradley, “Differentiable sparse coding,” in *NIPS*, pp. 113–120, 2008.

[38] J. Yang, K. Yu, and T. S. Huang, “Supervised translation-invariant sparse coding,” in *CVPR*, pp. 3517–3524, 2010.

[39] A. Opelt, A. Pinz, M. Fussenegger, and P. Auer, “Generic object recognition with boosting,” *PAMI*, vol. 28, 2004.

[40] P. Kohli, L. Ladicky, and P. H. S. Torr, “Robust higher order potentials for enforcing label consistency,” in *CVPR*, 2008.

[41] M. Tappen, K. G. Samuel, C. Dean, and D. Lyle, “The logistic random field – a convenient graphical model for learning parameters for mrf-based labeling,” in *CVPR*, 2008.

[42] M. Marszalek and C. Schmid, “Accurate object recognition with shape masks,” *International Journal of Computer Vision*, pp. 191–209, 2012.

[43] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results.” http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[44] Y. Li and D. P. Huttenlocher, “Sparse long-range random field and its application to image denoising.,” in *ECCV (3)*, pp. 344–357, 2008.

[45] C. Yu and T. Joachims, “Learning structural svms with latent variables,” in *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 1169–1176, ACM, 2009.

[46] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun, “Support vector machine learning for interdependent and structured output spaces,” in *Proceedings of the twenty-first international conference on Machine learning*, p. 104, ACM, 2004.

[47] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part based models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[48] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *Journal of the Royal Statistical Society (Series B)*, vol. 58, pp. 267–288, 1996.

[49] H. Zou and T. Hastie, “Regularization and variable selection via the elastic net,” *Journal of the Royal Statistical Society, Series B*, vol. 67, pp. 301–320, 2005.

[50] W. J. Fu, “Penalized Regressions: The Bridge versus the Lasso,” *Journal of Computational and Graphical Statistics*, vol. 7, no. 3, pp. 397–416, 1998.

[51] H. Lee, A. Battle, R. Raina, and A. Y. Ng, “Efficient sparse coding algorithms,” in *Advances in Neural Information Processing Systems 19* (B. Schölkopf, J. Platt, and T. Hoffman, eds.), pp. 801–808, Cambridge, MA: MIT Press, 2007.

[52] Q. Shi, A. Eriksson, A. van den Hengel, and C. Shen, “Is face recognition really a compressive sensing problem?,” in *CVPR*, pp. 553–560, 2011.

[53] R. Rigamonti, M. Brown, and V. Lepetit, “Are sparse representations really relevant for image classification?,” in *CVPR*, pp. 1545–1552, 2011.

[54] D. Zhang, M. Yang, and X. Feng, “Sparse representation or collaborative representation: Which helps face recognition?,” in *ICCV*, pp. 471–478, 2011.

[55] K. Menger, “A counterpart of occam’s razor in pure and applied mathematics ontological uses,” *Synthese*, vol. 12, no. 4, pp. 415–428, 1960.