Ensemble methods for multi-label learning of compositional data

Jan Verwaeren JAN.VERWAEREN@UGENT.BE
Willem Waegeman WILLEM.WAEGEMAN@UGENT.BE
Bernard De Baets BERNARD.DEBAETS@UGENT.BE
KERMIT, Department of Applied Mathematics, Biometrics and Process Control, Ghent University, Coupure links 653, B-9000 Ghent, Belgium

Abstract

Compositional data are a specific type of multivariate data, in which individual variables represent relative contributions to a whole, typically expressed as 1 or 100%. Driven by a lot of potential applications in many domains, this paper presents various supervised algorithms for learning compositional data from other data sources, leading to a novel multi-label learning setting that bears strong similarities with multivariate regression, multi-label classification and multi-class classification. Building further upon our previous work on this topic, we compare in this paper several ensemble methods that take into account specific properties about compositional data. We analyze three different types of kernel base learners for bivariate compositions, respectively based on maximum likelihood estimation, least-squares minimization and a label transformation. We also investigate two approaches for aggregating the bivariate predictions of these base learners into multivariate compositions, respectively based on a pairwise and a tree-based decomposition technique. The comparison of the algorithms is supported by empirical results on synthetic data and a real-world application in bioinformatics.

1. Introduction

Compositional data can be observed in many domains, like chemistry, geology, bioinformatics, to name just a few. As a simple introductory example of such data, let us consider the ingredients of lemonade. In a drink, several ingredients can occur, and some of these ingredients will have a higher contribution than others; a lemonade will mainly consist of water and lemon juice, but it will also contain minor percentages of sugar and minerals. Important here is that all ingredients sum up to a certain amount, in this case 100%, so that every ingredient is represented as a part of the whole. Instead of measuring only the presence or absence of ingredients as binary variables, we could also be interested in the degree of presence of ingredients in such an application.

Indeed, this is the main characteristic behind compositional data: more formally, every single variable in this source of multivariate data represents a relative contribution to a whole, usually expressed as a unit 1 or 100%. Moreover, compositional variables are assumed to be positive. So, they are constrained variables, and they are also dependent variables in a statistical sense: knowledge about the values of some of the variables changes the likelihood of observing particular values for the remaining unobserved variables.

This article introduces supervised algorithms for learning compositional data, as a specific type of multi-label learning. Such a learning problem is of course closely related to multivariate regression, see e.g. (Breiman and Friedman, 1997), which is often called multi-output regression in the machine learning community, but learning compositional data also bears similarities with multi-label classification and multi-class classification. The connection with multi-label classification follows from the need for modeling statistical dependence of output variables, a need that characterizes as well recent developments in multi-label classification, see e.g. (Cheng and Hüllermeier, 2009; Dembczyński et al., 2010). Moreover, if one discretizes the labels by putting a threshold on real label values, the setting simplifies to a multi-label learning problem. The connection with multi-class classification on the other hand follows from the fact that compositional data can

Appearing in *Proceedings of the 26th International Conference on Machine Learning*, Haifa, Israel, 2010.

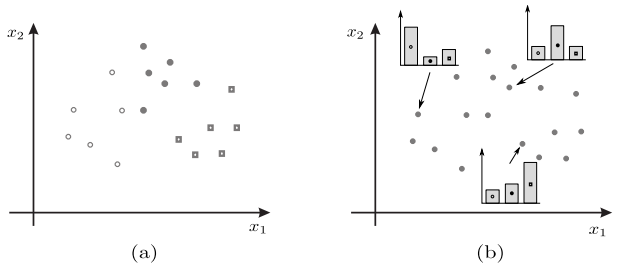

Figure 1. An example of a problem setting with two hypothetical features on the horizontal and vertical axis, illustrating the difference between crisp multi-class classification and learning partial class memberships. On the left, the symbols represent the crisp class labels in traditional multi-class classification, while on the right side, labels are represented by partial class memberships, as one unit of membership divided over all classes.

be interpreted as partial class memberships or prior class probabilities, since class memberships or class probabilities possess the property of summing up to 1. The difference between learning compositional data and multi-class classification is graphically explained in Figure 1.

Compositional data have been studied quite extensively in applied statistics, see e.g. (Aitchison, 1982; Billheimer et al., 2001; Pawlowski-Glahn and Egozcue, 2006), but research on this topic is almost non-existent in the machine learning literature, despite a large amount of potential applications in various disciplines. The few related methods presented in machine learning typically refer to the problem setting as learning *partial* or *mixed* class memberships, due to the abovementioned connection with multi-class classification. These methods are predominantly unsupervised, such as clustering algorithms where data instances can simultaneously exhibit a degree of membership to several clusters. The concept of partial class membership has for example been incorporated in mixed models (Gormley and Murphy, 2008), probabilistic graphical models (Airoldi et al., 2008) and Bayesian clustering techniques (Heller et al., 2008).

A limited number of related studies can also be found in fuzzy systems; compositional data are in this field known as fuzzy partitions or Ruspini partitions (Ruspini, 1969). Using the terminology *partial*, *mixed* or *fuzzy* memberships, applications of analyzing compositional data can be found in domains like text categorization (Erosheva et al., 2004), agriculture (Nisar-Ahamad et al., 2000), and bioinformatics (Marttinen et al., 2009).

We adopted a similar terminology in recent work (Waegeman and De Baets, 2009) and (Waegeman et al., submitted), where we introduced supervised algorithms for learning compositional data, using maximum likelihood estimation of logistic models. This article builds further upon these results, by investigating alternative algorithms and providing additional empirical results. We start in Section 2 with a formal description of the problem setting and a discussion of the aspects that make this setting different compared to more conventional learning paradigms. Subsequently, Section 3 discusses various basic methods for learning compositional data. These methods, which infer and postprocess the label vectors in a pairwise manner, are used as base learners in ensemble methods. Two different ensemble methods are proposed in Section 4. Finally, in Section 5 results are presented on synthetic data and a real-world application, demonstrating the usefulness of our approach.

2. Formal problem description

Let us start by introducing some notations. In a general multi-label learning problem, the goal is to learn a mapping from an input space $\mathbb{X}$ to a vectorial output space $\mathcal{Y}$ of dimension K. To this end, each object is represented by a D-dimensional feature representation $\mathbf{x} \in \mathbb{X}$ and a vector of labels $y \in \mathcal{Y}$. We have $\mathcal{Y} = \{0,1\}^K$ in multi-label classification and $\mathcal{Y} = \mathbb{R}^K$ in multivariate regression or multi-output regression. For learning K-dimensional compositional data, the vector of labels is a vector within the K-dimensional simplex $\mathcal{Y}$:

$$\mathcal{Y} = \Big\{ \mathbf{y} = (y_1, \ldots, y_K) \in \mathbb{R}^K \mid y_i \geq 0, \forall\, i \in \{1, \ldots, K\}; \sum_{i=1}^{K} y_i = 1 \Big\}. \quad (1)$$

So, every data object will be linked with a K-dimensional real-valued vector that will be called its label vector. Each label vector has one unit of membership divided over the K classes. As a result, each element of the label vector is positive and the sum of all elements of the vector equals one.

A training dataset $\mathbf{T}$ of N i.i.d. observations can then be denoted as a set of couples $\{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_N, \mathbf{y}_N)\}$. We will only focus on supervised learning, implying that both $\mathbf{x}$ and $\mathbf{y}$ are used to construct a predictive model. The i-th instance in a training set $\mathbf{T}$ will be denoted

$$(\mathbf{x}_i, \mathbf{y}_i) = \Big((x_{i,1}, \ldots, x_{i,D}), (y_{i,1}, \ldots, y_{i,K}) \Big).$$

The predictive model that we aim to fit to the data will be represented as $\mathbf{f} : \mathbb{X} \rightarrow \mathcal{Y}$, in which $\mathbf{f}(\mathbf{x}) =$

$(f_1(\mathbf{x}), \ldots, f_K(\mathbf{x}))$. Since it is assumed that label vectors lie in the simplex, the following additional constraint must be imposed on the model for all $\mathbf{x} \in \mathbb{X}$:

$$f_k(\mathbf{x}) \geq 0, \forall\, k \in \{1, \ldots, K\}\,, \tag{2}$$

$$\sum_{k=1}^{K} f_k(\mathbf{x}) = 1\,. \tag{3}$$

In a classical supervised machine learning setting, one aims to find a mapping or model $\mathbf{f} : \mathbb{X} \to \mathcal{Y}$ that minimizes the expected value of some regularized loss function, i.e.

$$\hat{\mathbf{f}}(\mathbf{x}) = \min_{\mathbf{f} \in \mathcal{H}} \mathcal{L}(\mathbf{f}, \mathbf{T}) + \lambda J(\mathbf{f})\,, \tag{4}$$

with $\mathcal{L}$ the loss function on the training dataset, $\mathcal{H}$ a hypothesis space of models, J a penalty term for the complexity of the model and λ a regularization parameter. We will further only consider instance-wise decomposable loss functions that can be written as

$$\mathcal{L}(\mathbf{f}, \mathbf{T}) = \sum_{i=1}^{N} L(\mathbf{f}(\mathbf{x}_i), y_i)\,, \tag{5}$$

with L the loss between the label and the model prediction of a single instance. When learning compositional data, one can simply adopt existing loss functions that are used for multivariate regression. We will consider the mean squared error between true and predicted label vectors in this study:

$$L_{\text{MSE}}(\mathbf{f}(\mathbf{x}), \mathbf{y}) = \sum_{k=1}^{K} (f_k(\mathbf{x}) - y_k)^2\,.$$

Similarly, one can argue to evaluate predicted compositional data using the mean absolute error instead of the mean squared error, resulting in a more realistic performance evaluation for heavily unbalanced label vectors.

3. Base learners for compositional data

In this section three basic approaches for learning compositional data are presented. The first approach performs a transformation of the label vector from the simplex to an unconstrained space, in which standard regression algorithms can be used to predict every component of the label vector independently. The second approach adopts kernel logistic regression models for compositional data, using maximum likelihood estimation techniques. The third approach further extends the second approach, using an alternative loss function.

3.1. Label transformation

The simplex constraints (1) that characterize compositional data imply that standard statistical tools cannot be applied on compositional data. From a learning perspective, one cannot use standard multivariate regression (also called multi-output regression) methods to predict compositional data. However, as suggested by Aitchison (1986), one can always transform the data to an Euclidean space where standard operations are valid. A common transformation is given by the following mapping:

$$\begin{aligned} U : \mathcal{Y} &\mapsto \mathbb{R}^{K-1} \\ \mathbf{y} &\to \left(\log \frac{y_1}{y_K}, \ldots, \log \frac{y_{K-1}}{y_K}\right). \end{aligned}$$

However, such a transformation cannot take zeroes in the original vectors into account. This should be considered as an important drawback, because zeroes frequently occur in applications of compositional data.

After transforming the label vectors to an Euclidean space of dimension $K-1$, existing multivariate regression algorithms can be applied. During the test phase, predicted label vectors have to be transformed again to the simplex, using the inverse operator U^{-1}. In the experiments in Section 5, kernel ridge regression will be applied as multivariate regression method, so that nonlinear relationships between features and label vectors can be modeled. Recall that kernel ridge regression will estimate the parameters of $K-1$ scoring functions, generally represented as

$$g_k(\mathbf{x}) = \mathbf{w}_k \cdot \phi(\mathbf{x}) + \theta_k\,, \tag{6}$$

with ϕ a feature mapping to a possibly high-dimensional feature space and $\mathbf{w}_1, ..., \mathbf{w}_{K-1}$ vectors of parameters that must be estimated based on training data. These vectors of parameters are usually different for every component of the label vector.

The representer theorem states that for a general class of models and loss functions the solution of optimization problem (4) can be written as a linear combination of the training vectors. In particular, kernel ridge regression and the other kernel methods presented below exhibit such characteristics, so that the kernelized scoring functions can be written as

$$g_k(\mathbf{x}) = \sum_{i=1}^{N} \alpha_{ik} K(\mathbf{x}_i, \mathbf{x}) + \theta_k\,, \tag{7}$$

with $\boldsymbol{\alpha}_k = (\alpha_{1k}, \ldots, \alpha_{Nk})$ the model parameters for the k-th label vector component. The regularization term in the loss function becomes

$$\sum_{k=1}^{K} ||\mathbf{w}_k||^2 = \sum_{k=1}^{K} \boldsymbol{\alpha}_k^T \mathbf{K} \boldsymbol{\alpha}_k\,,$$

with $\mathbf{K}$ the Gram matrix for the training points, i.e., $\mathbf{K}_{i,j} = K(\mathbf{x}_i, \mathbf{x}_j)$.

3.2. Kernel logistic regression

As an alternative approach that avoids label transformations, we propose to adopt a logistic type of model to represent label vectors in a one-versus-all way:

$$f_k(\mathbf{x}) = \frac{\exp(g_k(\mathbf{x}))}{\sum_{l=1}^{K} \exp(g_l(\mathbf{x}))}, \tag{8}$$

in which $g_1, ..., g_K : \mathbb{X} \to \mathbb{R}$ are scoring functions that assign values from the set of real numbers to data instances, similar to (6). These scoring functions are just linear models in traditional logistic regression models and kernel expansions of type (7) in kernel logistic regression.

A kernel logistic regression model is typically optimized by means of maximum likelihood estimation. As a specific form of (4), the regularized multinomial likelihood is given by:

$$\mathcal{L}(\mathbf{w}_1, ..., \mathbf{w}_K) = \prod_{i=1}^{N} \prod_{k=1}^{K} \left(f_k(\mathbf{x}_i)\right)^{y_{ik}} + \lambda \sum_{k=1}^{K} ||\mathbf{w}_k||^2,$$

with λ a regularization parameter. Remark that we allow that $\mathbf{y} \in \mathcal{Y}$ instead of $\mathbf{y} \in \{0,1\}^K$, unlike traditional logistic regression. Equivalently to maximizing the likelihood, one can minimize the regularized negative log-likelihood. The minimum is usually found with a simple gradient descent algorithm in the two-class case, but one arrives at a constrained optimization problem in the multi-class case, as constraint (2) must hold. To this end, variants of the sequential minimal optimization algorithm, found in implementations of support vector machines, have been proposed for the multi-class case (Keerthi et al., 2005).

3.3. Least-squares minimization

Remark that a logistic model of type (8) does not assume any probabilistic interpretation at all, which is a benefit, since label vectors should not be seen as class probabilities. Yet, when the likelihood is maximized, still a probabilistic interpretation must be adopted. This could be considered as a limitation of the previous method. As an alternative, the parameters of model (8) can be estimated by optimization of an (L2-)regularized least-squares criterion. Since this criterion directly optimizes the performance measure (L_{MSE}), this can be seen as an advantage. However, this option leads to a more difficult nonlinear optimization problem.

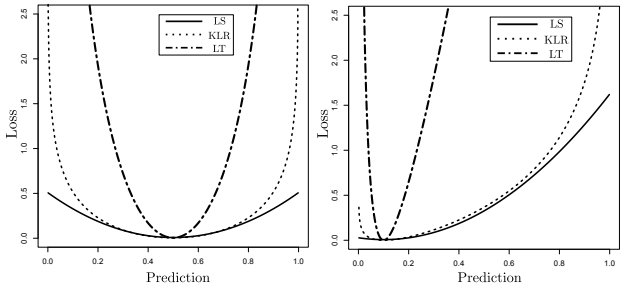

Figure 2. A visualization of the loss functions for the label transformation (LT), kernel logistic regression (KLR) and least-squares (LS) methods. The left and right figure give the loss when the true label vectors are $(0.5, 0.5)$ and $(0.1, 0.9)$, respectively.

The models fit by the methods discussed above are very similar to (8). The main difference lies in the criterion that is optimized in order to obtain estimates of the parameter vectors $\mathbf{w}_1, ..., \mathbf{w}_K$. To gain insight into the differences between these methods, we will now consider the case where $K = 2$. Figure 2 shows the loss functions for the label transformation (LT), kernel logistic regression (KLR) and least-squares (LS) methods. This figure illustrates the clear difference between the loss functions of LT, KLR and LS. Opposed to LS, both LT and KLR have unbounded loss functions that become strongly asymmetric as the value of the labels tend to 0 or 1. Moreover, the LT loss functions becomes very sharp at the boundaries of $[0, 1]$. This property makes the fitting procedure of the LT method sensitive to noise. Consequently, we could argue that KLR and LS result in more robust loss functions, which could result in more stable models.

4. Ensemble methods for combining base learners

In the previous section, three simple but computationally efficient base learners were presented. The first two of these base learners naturally generalize to situations where $K > 2$, but for the least-squares approach this is more difficult. However, in recent work (Waegeman et al., submitted), we showed that the performance of a model of type (8) can improve by using pairwise decomposition and voting techniques. In this paper these ideas are further extended for other base learners. Moreover, we also present a second ensemble method that overcomes the computational burden of pairwise decomposition methods.

4.1. Pairwise decomposition

Similar to the traditional one-versus-one ensemble for multi-class classification, we consider $K(K-1)/2$ base classifiers for learning compositional data of dimension K. As such, $K(K-1)/2$ new datasets are constructed as follows:

$$\mathbf{T}_{kl} = \{(\mathbf{x}_i, y_i^{kl}) \mid (\mathbf{x}_i, \mathbf{y}_i) \in \mathbf{T} \wedge y_i^{kl} = \frac{y_{ik}}{y_{ik} + y_{il}}\}\,, \quad (9)$$

with $1 \leq k < l \leq K$. Thus, by applying Eq. (9), a set of two-dimensional compositions is obtained for every instance, where y_i^{kl} can be interpreted too as compositional data. More specifically, it represents the value of the k-th label when the restriction is made that only the k-th and l-th label can be different from zero. The value of the l-th label can then be computed as $y_i^{lk} = 1 - y_i^{kl}$. We will use the notation $f_{kl} : \mathbb{X} \rightarrow [0,1]$ to denote the individual base learners, for which the outcome represents the prediction of the k-th component of the label vector. The predicted value for the l-th component can simply be computed as $f_{lk}(\mathbf{x}) = 1 - f_{kl}(\mathbf{x})$.

Once the parameters of all pairwise models have been estimated, the pairwise predictions must be further postprocessed to obtain predictions that make sense. More specifically, the predicted pairwise predictions $f_{kl}(\mathbf{x})$ must be transformed to a model of type $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), ..., f_K(\mathbf{x}))$, as considered in the previous section. Similar as Eq. (9) for the true label vectors, the Bradley-Terry model allows to establish a natural link for all $1 \leq k < l \leq K$:

$$\mu_{kl}(\mathbf{x}) = \frac{f_k(\mathbf{x})}{f_k(\mathbf{x}) + f_l(\mathbf{x})}\,, \quad (10)$$

where μ_{kl} denotes a theoretically assumed prediction with the base classifier trained on $\mathbf{T}_{kl}$. Unfortunately, as all base classifiers conduct a relatively independent optimization procedure, one will in practice notice that $f_{kl}(\mathbf{x}) \neq \mu_{kl}(\mathbf{x})$. If one replaces $\mu_{kl}(\mathbf{x})$ by $f_{kl}(\mathbf{x})$ in Eq. (10), then the resulting system of equations cannot be solved. Solving for $f_k(\mathbf{x})$ is much more complicated than solving for $\mu_{kl}(\mathbf{x})$, since $K(K-1)/2$ variables appear on the left side and only K variables appear on the right side. In total $K(K-1)/2$ equations must be satisfied, so that the system of equations counts more equations than free variables. A very similar situation occurs in probabilistic multi-class classification, when pairwise class probabilities have to be coupled to obtain posterior class probabilities per class. From this perspective, Wu et al. (2004) recently proposed two new algorithms for solving systems like $f_{kl}(\mathbf{x}) = \mu_{kl}(\mathbf{x})$ and compared their algorithms experimentally with previous approaches. As most of these methods do not make any probabilistic assumptions at all, it sounds reasonable to adopt them as well in our framework.

Various algorithms for combining the pairwise models f_{kl} into K-dimensional compositions have recently been empirically compared in (Waegeman et al., submitted) for learning compositional data, with kernel logistic regression as base learner. Since the focus of this article is a bit different, we will only report results for one of the approaches of (Wu et al., 2004), which turned out to be one of the best choices in our recent work. This method solves the following system for every $\mathbf{x}$:

$$f_k(\mathbf{x}) \;=\; \Big(\frac{f_k(\mathbf{x}) + f_l(\mathbf{x})}{K-1}\Big) f_{kl}(\mathbf{x})$$
$$\text{subject to} \quad \sum_{k=1}^{K} f_k(\mathbf{x}) = 1,\; f_k(\mathbf{x}) \geq 0,\; \forall k\,.$$

The solution of this system can be written as the unique global minimum of the following convex optimization problem:

$$\min_{\mathbf{f}} \;\; \sum_{k=1}^{K} \Big(\sum_{l:l\neq k} f_{lk}(\mathbf{x}) f_k(\mathbf{x}) - \sum_{l:l\neq k} f_{kl}(\mathbf{x}) f_l(\mathbf{x}) \Big)^2$$
$$\text{subject to} \;\; \sum_{k=1}^{K} f_k(\mathbf{x}) = 1,\; f_k(\mathbf{x}) \geq 0,\; \forall k\,.$$

Wu et al. (2004) show that the minimum can be found very efficiently with a simple iterative algorithm.

4.2. Tree-based aggregation

Contrary to multi-class classification, the above pairwise coupling approach cannot compete with one-versus-all type of models in terms of computational efficiency, since we need all training examples for every base classifier. Its main advantage should be rather found in an improved predictive performance, as shown in the experiments. In multi-class classification, only the training examples from the k-th and l-th class provide useful information for training model f_{kl}, so that the base classifiers are trained using only a small part of the entire dataset. In contrast, all training examples do matter for fitting base classifiers in our setting, even if both label vector components y_{ik} and y_{il} are zero. However, Eq. (9) cannot be used in such situations. We will therefore simply consider 0.5 as pairwise memberships when both y_{ik} and y_{il} equal zero.

As a computationally more efficient alternative to pairwise decomposition, we also test a second ensemble method that structures base classifiers as a tree. Since

compositional data represent relative contributions to a whole, it feels natural to order compositions in a hierarchical way. In this hierarchy, the root node represents the whole, and paths from the root to the leaves indicate successive splits of the whole into smaller parts. So, the two children of the root split the unit 1 into two compositions that sum up to one. These two contributions are then in subsequent children further split into smaller compositions, satisfying the property that the nodes of every level in the tree sum up 1. As such, the values in the leaves represents the original compositions.

Using a hierarchical structure for compositional data, we train a base classifier in all internal nodes. This requires $K-1$ base classifiers in total, delivering a substantial gain compared to pairwise decomposition in terms of computational complexity. In terms of predictive performance, one might theoretically expect that pairwise decomposition outperforms this tree-based approach, as the latter does not conduct any averaging strategy. One could of course improve predictive performance by averaging over many at random constructed trees, but this is beyond the purpose of this study. In the experiments we will only construct a single tree. The decision about the splits at every level in the tree is taken at random.

5. Experiments

The two ensemble methods and three base learners are in this section tested on synthetic data and a real-world case study in bioinformatics. Synthetic data allows to control some parameters in the data generation process, so that the different behaviour of the various algorithms can be better explained, while the real-world case study mainly aims to illustrate the practical need for supervised learning of compositional data. The base learners were implemented in the statistical package R. In particular, the parameters for the least squares learner were optimized with the BFGS algorithm (Broyden, 1970) which is a quasi-newton method. Furthermore, the Python libraries of Wu et al. (2004) were adopted for converting the outputs of the base learners to compositions. All experiments were carried out on a computer cluster in order to speed up computations and to allow a sufficient number of repetitions for observing statistically significant differences between algorithms.

5.1. Experiments on synthetic data

L_{MSE} is chosen as the performance measure in the experiments on synthetic data. In this setting, each data point consists of a feature vector $\mathbf{x} \in \mathbb{R}^{\not\in}$ and a label vector $\mathbf{y} \in \mathcal{Y}$ with $K = 4$. The feature vectors are obtained through sampling from a bivariate Gaussian distribution with parameters $\mu_1 = 15$, $\mu_2 = 15$, $\sigma_1 = 3$, $\sigma_2 = 5$ and $\rho = 0$. The label vectors are generated through sampling from a Dirichlet distribution with parameter vector $\boldsymbol{\alpha}(\mathbf{x}) \in \mathbb{R}^4_+$ which depends on the inputs as follows:

$$\alpha_i(\mathbf{x}) = 100 \times \frac{\Phi(\mathbf{x}; \mu_{1,i}, \mu_{2,i}, \sigma_{1,i}, \sigma_{2,i}, \rho_i)}{\sum_{j=1}^{4} \Phi(\mathbf{x}; \mu_{1,j}, \mu_{2,j}, \sigma_{1,j}, \sigma_{2,j}, \rho_j)} ,$$

for $i = 1, \ldots, 4$, where $\Phi(.; \mu_{1,i}, \mu_{2,i}, \sigma_{1,i}, \sigma_{2,i}, \rho_i)$ represents the density of a bivariate Gaussian. In this setting, the parameters were chosen as follows $\mu_{1,1} = 10$, $\mu_{2,1} = 10$, $\sigma_{1,1} = 5$, $\sigma_{2,1} = 5$, $\mu_{1,2} = 20$, $\mu_{2,2} = 10$, $\sigma_{1,2} = 5$, $\sigma_{2,2} = 5$, $\mu_{1,3} = 15$, $\mu_{2,3} = 10+5\sqrt{3}$, $\sigma_{1,3} = 3$, $\sigma_{2,3} = 3.5$, $\mu_{1,4} = 20$, $\mu_{2,4} = 30$, $\sigma_{1,4} = 8$, $\sigma_{2,4} = 7$ and $\rho_i = 0$. Remark that the non-uniformity of the standard deviations leads to a nonlinear Bayes-optimal model.

This scheme was used to create training and validation sets, each containing 20 instances. An independent test set containing 500 instances was created as well. To assess the statistical significance of differences in performance, this process was repeated 30 times.

In total, 7 learning strategies were applied to these datasets: 6 decomposition methods and MKLR, the direct kernel logistic regression approach for $K > 2$. The 6 decomposition methods differ in the decomposition technique (pairwise (PW) or tree-based (TREE)) and the base learners (LT, KLR and LS). To ensure enough flexibility for each method we opted to use an RBF-kernel for each learner. The hyper-parameters were optimized by means of cross-validation.

Figure 3 shows the results for each method in terms of the root mean squared error (RMSE) on the test sets. The results indicate that the influence of the base classifier is limited for the tree-based decomposition procedure. Furthermore, the influence of the type of decomposition technique was limited for the least-squares and the label transformation methods. The fact that KLR combined with the pairwise decomposition technique clearly outperforms all other methods is somewhat unexpected, but an interesting result.

5.2. An application in bioinformatics

The lack of publicly available benchmark datasets for learning compositional data implies that the proposed methods cannot be empirically compared on a compendium of datasets. In addition to synthetic data, we apply our algorithms to real-world data from the bioinformatics domain in order to illustrate the need for learning compositional data. More specifically, in

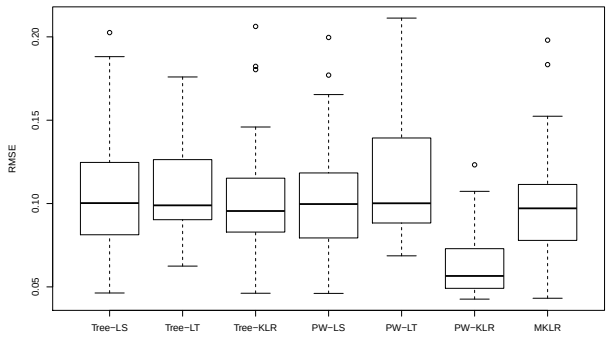

Figure 3. Performance results for each of the 7 learning strategies on the synthetic datasets. Boxplots of the RMSEs obtained through 30 repetitions of the data generation process.

microbiology, fatty acid methyl ester (FAME) profiles and 16S rRNA sequences offer two different descriptions of bacterial species. In the microbial literature, both sources of information are established as very useful characteristics for the discrimination of different bacterial species. For this reason, it makes sense to look for statistical relationships between both data sources. We will employ the methods presented above to find such relationships, in which FAME profiles serve as labels and 16S rRNA sequences as features. As discussed in (Marttinen et al., 2009), FAME profiles satisfy constraint (1), so they can be interpreted as partial class memberships for in total 71 classes ($K = 71$). We used the dataset of (Slabbinck et al., 2009), containing FAME profiles from 74 different bacterial species.

In addition to the FAME profiles, we collected for all 74 bacterial species one quality-controlled 16S rRNA sequence of their type strain. These sequences serve as features in our experiment. Since we are dealing with kernel methods, we subsequently computed a similarity or kernel matrix for the collected sequences, using the Dnadist program of the bioinformatics package PHYLIP (Felsenstein, 2004). This program calculates the similarity between two DNA sequences as the fraction of identical nucleotides.

A 4-fold cross validation was performed to asses the performance of the different learning strategies on this dataset. The regularization parameter, which is present in each optimization procedure, was chosen by means of a nested cross-validation loop. It should be noted that the compositional vectors contain a lot of zero values. To be able to use the LT procedure, a small constant was added to these values. The RMSE

Method	RMSE
Tree-LS	0.0245
Tree-LT	0.0266
Tree-KLR	0.0251
PW-LS	0.0275
PW-LT	0.0315
PW-KLR	0.0263
MKLR	0.0241

Table 1. Performance results for all learning strategies, in terms of the RMSE after 4-fold cross validation, on the FAME dataset.

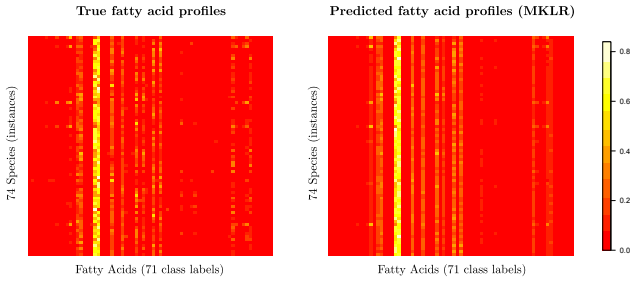

Figure 4. Left: heatmap of the 74 average FAME profiles that were used as compositional data ($K = 71$). Each row corresponds to the average FAME profile for one species. Right: heatmap of the 74 predicted average FAME profiles with MKLR.

of all methods is given in Table 1. For this dataset, tree-based decomposition clearly outperforms the pairwise decomposition methods, which is definitely a surprising result. The performance of the PW-LT strategy is notably worse than all others, which contradicts the findings in the synthetic datasets. This contrast might be explained by the type of noise present in the data. As stated above, the LT strategy is susceptible to the presence of noise at the boundary of $[0, 1]^K$. In the synthetic data the noise was introduced with a Dirichlet distribution that adds less noise to points situated at the boundary of the simplex. In the FAME dataset, this might not be the case. Finally, it can be seen that MKLR performs best in this setting, closely followed by the best tree-based method. Figure 4 visualizes the predictions for MKLR.

6. Discussion

In this paper we introduced the problem of learning compositional data, a novel multi-label learning problem that occurs in many real-world applications. To support this claim, a case study in bioinformatics was discussed. We proposed three methods and two aggregation techniques for learning compositional data and

we evaluated these methods on the case study dataset and synthetic data. The results indicate that both, decomposition techniques and MKLR can be used to learn from compositional data. Surprisingly, we found that tree-based methods were able to compete with pairwise decomposition techniques, not only in terms of computational complexity but also in terms of predictive power. We hope that this paper can motivate other researchers as well to start developing new algorithms in the challenging domain of learning compositional data.

Acknowledgments

W.W. is supported for this work by the Research Foundation of Flanders (FWO Vlaanderen).

References

E. Airoldi, D. Blei, S. Fienberg, and E. Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9:1981–2014, 2008.

J. Aitchison. The statistical analysis of compositional data (with discussion). *Journal of the Royal Statistical Society*, B44:139–177, 1982.

J. Aitchison. *The Statistical Analysis of Compositional Data*. Chapman &Hall, 1986.

D. Billheimer, P. Guttorp, and W. Fagan. Statistical interpretation of species composition. *Journal of the American Statistical Association*, 96:1205–1214, 2001.

L. Breiman and J. Friedman. Predicting multivariate responses in multiple linear regression. *Journal of the Royal Statistical Society: Series B*, 69:3–54, 1997.

C. G. Broyden. The convergence of a class of double-rank minimization algorithms. *Journal of the Institute of Mathematics and Its Applications*, 6:76–90, 1970.

W. Cheng and E. Hüllermeier. Combining instance-based learning and logistic regression for multilabel classification. *Machine Learning*, 76(2-3):211–225, 2009.

K. Dembczyński, W. Cheng, and E. Hüllermeier. Bayes optimal multilabel classification via probabilistic classifier chains. In *Proc. of the Twenty-Seventh International Conference on Machine Learning (ICML 2010)*, Haifa, Israel, 2010. to appear.

E. Erosheva, S. Fienberg, and J. Lafferty. Mixed-membership models of scientific publications. *PNAS*, 101:5220–5227, 2004.

J. Felsenstein. *Inferring Phylogenies*. Sinauer Associates, Inc, 2004.

I. Gormley and T. Murphy. A mixture of experts models for rank data with applications in election studies. *Annals of Applied Statistics*, 2:1452–1477, 2008.

K. Heller, S. Williamson, and Z. Ghahramani. Statistical models for partial membership. In *Proceedings of the 25th Conference on Machine Learning, Helsinki, Finland*, pages 392–399, 2008.

S. Keerthi, K. Duan, S. Shevade, and A. Poo. A fast dual algorithm for kernel logistic regression. *Machine Learning*, 61:151–165, 2005.

P. Marttinen, J. Tang, B. De Baets, P. Dawyndt, and J. Corander. Bayesian clustering of fuzzy feature vectors using a quasi-likelihood approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:74–85, 2009.

T. Nisar-Ahamad, K. Gropal-Rao, and J. Murthy. GIS-based fuzzy membership model for crop-land suitability analysis. *Agricultural Systems*, 63:75–95, 2000.

V. Pawlowski-Glahn and J. Egozcue. Compositional data and their analysis: an introduction. *Geological Society*, 264:1–10, 2006.

E.H. Ruspini. A new approach to clustering. *Information and Control*, 15:22–32, 1969.

B. Slabbinck, B. De Baets, P. Dawyndt, and P. De Vos. Towards large-scale FAME-based bacterial species identification using machine learning techniques. *Systematic and Applied Microbiology*, 32: 163–176, 2009.

W. Waegeman and B. De Baets. Learning partial class memberships in multi-clas classification problems: a probabilistic approach. In *Proceedings of the EUROFUSE Workshop on Preference Handling and Decision Support, Pamplona, Spain*, pages 47–54, 2009.

W. Waegeman, J. Verwaeren, B. Slabbinck, and B. De Baets. Supervised learning algorithms for classification problems with partial class memberhsip. *Fuzzy Sets and Systems*. submitted.

F. Wu, C. Lin, and R. Weng. Probability estimates for multi-class support vector machines by pairwise coupling. *Journal of Machine Learning Research*, 5: 975–1005, 2004.