Pattern Recognition Letters
journal homepage: www.elsevier.com

Hybrid Generative-Discriminative Training of Gaussian Mixture Models

Wolfgang Roth[a,**], Robert Peharz[b], Sebastian Tschiatschek[c], Franz Pernkopf[a]

[a] *Graz University of Technology, Inffeldgasse 16c/EG, 8010 Graz, Austria*
[b] *University of Cambridge, Trumpington Street, Cambridge CB2 1PZ, United Kingdom*
[c] *ETH Zürich, Universitätstrasse 6, 8092 Zürich, Switzerland*

ABSTRACT

Recent work has shown substantial performance improvements of discriminative probabilistic models over their generative counterparts. However, since discriminative models do not capture the input distribution of the data, their use in missing data scenarios is limited. To utilize the advantages of both paradigms, we present an approach to train Gaussian mixture models (GMMs) in a hybrid generative-discriminative way. This is accomplished by optimizing an objective that trades off between a generative likelihood term and either a discriminative conditional likelihood term or a large margin term using stochastic optimization. Our model substantially improves the performance of classical maximum likelihood optimized GMMs while at the same time allowing for both a consistent treatment of missing features by marginalization, and the use of additional unlabeled data in a semi-supervised setting. For the covariance matrices, we employ a diagonal plus low-rank matrix structure to model important correlations while keeping the number of parameters small. We show that a non-diagonal matrix structure is crucial to achieve good performance and that the proposed structure can be utilized to considerably reduce classification time in case of missing features. The capabilities of our model are demonstrated in extensive experiments on real-world data.

© 2018 Elsevier Ltd. All rights reserved.

1. Introduction

Many systems involve decision making procedures where for some given input $\boldsymbol{x} \in \mathbb{R}^D$ a categorical output $c \in \{1, \ldots, C\}$ needs to be computed. Supervised learning provides methods to derive a classifier based on a set of N input-output samples $\{(\boldsymbol{x}_1, c_1), \ldots, (\boldsymbol{x}_N, c_N)\}$. A common way to solve this task is to learn a generative probabilistic model by estimating the parameters $\boldsymbol{\theta}$ of a joint distribution $p(\boldsymbol{x}, c|\boldsymbol{\theta})$ and then predicting class $\hat{c}$ of sample $\boldsymbol{x}$ as the class with the highest posterior probability using Bayes' rule, i.e.,

$$\hat{c} = \arg\max_{c} p(c|\boldsymbol{x}, \boldsymbol{\theta}) = \arg\max_{c} \frac{p(\boldsymbol{x}|c, \boldsymbol{\theta})\, p(c|\boldsymbol{\theta})}{p(\boldsymbol{x}|\boldsymbol{\theta})}. \tag{1}$$

Generative models are typically trained according to the *maximum likelihood (ML)* principle where the parameters $\boldsymbol{\theta}_{ML}$ are estimated so as to maximize the joint likelihood when the samples are assumed to be independent and identically distributed (i.i.d), i.e.,

$$\boldsymbol{\theta}_{ML} = \arg\max_{\boldsymbol{\theta}} \prod_{n=1}^{N} p(\boldsymbol{x}_n, c_n|\boldsymbol{\theta}). \tag{2}$$

An advantage of the generative approach is its consistent treatment of missing features, given that the *missing at random (MAR)* assumption holds (Marlin, 2008). Classification with missing features is highly relevant in practice. In health care, data values of a medical examination might be missing, however, a doctor can often make a plausible diagnosis even if not all measurements are available. Another example are sensor networks where some of the sensors may fail to produce measurements.

Another advantage of the generative approach is that it can be naturally used in a semi-supervised setting. Semi-supervised learning exploits, in addition to a set of N_l *labeled* data samples $\{(\boldsymbol{x}^l_1, c_1), \ldots, (\boldsymbol{x}^l_{N_l}, c_{N_l})\}$, a set of N_u *unlabeled* data samples $\{\boldsymbol{x}^u_1, \ldots, \boldsymbol{x}^u_{N_u}\}$ to improve the classification performance. It is straightforward to extend the ML principle to include the ad-

[**] Corresponding author
e-mail: roth@tugraz.at (Wolfgang Roth)

ditional unlabeled data via their marginal likelihood, i.e.,

$$\boldsymbol{\theta}_{ML} = \arg\max_{\boldsymbol{\theta}} \prod_{n=1}^{N_l} p(\boldsymbol{x}_n^l, c_n|\boldsymbol{\theta}) \prod_{n=1}^{N_u} p(\boldsymbol{x}_n^u|\boldsymbol{\theta}). \tag{3}$$

Semi-supervised learning is especially interesting since obtaining the class labels is typically expensive and time-consuming whereas unlabeled data is abundant.

On the downside, the classification performance of generative models is often inferior to the performance of discriminatively trained models that estimate the parameters $\boldsymbol{\theta}$ to model the class posterior probability $p(c|\boldsymbol{x}, \boldsymbol{\theta})$ directly (Pernkopf et al., 2012). This holds especially true if the chosen model provides only a poor approximation of the true underlying distribution (Lasserre et al., 2006). On the other side, discriminative models typically have to rely on heuristics, such as imputation techniques, if the given data contains missing features and the extension to semi-supervised learning is often not straightforward.

In this paper, we consider a hybrid generative-discriminative treatment of Gaussian mixture models (GMMs) to benefit from the advantages of both approaches. While classical ML optimized GMMs are usually not competitive in terms of classification performance, some previous work investigated discriminative strategies for GMMs to improve the classification performance (Sha and Saul, 2006; Pernkopf and Wohlmayr, 2010). However, none of these approaches aims to maintain the generative character of the GMM, and any inference task over the input features, e.g. marginalizing missing features during test time, is actually not well justified. The fact that these models still deliver reasonable results is often merely an optimization artifact – discriminative training is frequently initialized with the ML solution – and not a design goal in itself. To close this gap, we introduce hybrid generative-discriminative learning by formulating an objective that trades off between a generative likelihood term and a discriminative term. In fact, discriminative and generative learning are not diametrically opposed: It is well known that the data generating distribution – the ultimate object of interest in generative learning – also delivers the Bayes optimal classifier. On the other hand, the class labels can be seen as an abstract representation of the input data, and incorporating a discriminative term in generative learning can be interpreted as an informative regularizer, helping to overcome data scarcity in generative modeling.

For the discriminative term, we investigate two common objectives: (i) the conditional log-likelihood (CLL) criterion estimates parameters so as to maximize the class posterior probabilities

$$\boldsymbol{\theta}_{CLL} = \arg\max_{\boldsymbol{\theta}} \prod_{n=1}^{N} p(c_n|\boldsymbol{x}_n, \boldsymbol{\theta}), \tag{4}$$

and (ii) a probabilistic large margin (LM) criterion where samples, whose class posterior probability $p(c_n|\boldsymbol{x}_n, \boldsymbol{\theta})$ of the true class c_n is not sufficiently larger than the class posterior probabilities of all other classes, are penalized (see Section 3 for more details). These hybrid objectives have already been considered in (Bouchard and Triggs, 2004; Peharz et al., 2013), but they did not consider GMMs. Lasserre et al. (2006) proposed a different approach where the trade-off between generative and discriminative semantics is governed at model level rather than at objective level.

Since the covariance matrices of GMMs can be very large in case of high-dimensional input spaces, we propose to use a diagonal plus low-rank structure for the covariance matrices to reduce the parameter space considerably while still allowing important dependencies among the variables to be captured. This special matrix structure can be utilized to reduce the costly operations of matrix inversion and determinant computation of large matrices to matrices of much smaller size. We show that this is beneficial at both training time and testing time, especially in the presence of missing features where the matrix inversions and determinants cannot be precomputed if the missing data patterns are not known beforehand. Furthermore, we show that non-diagonal covariance matrices outperform diagonal covariance matrices by a large margin.

We performed several experiments on real-world data sets using different kinds of GMMs. The classification performance of purely generatively trained GMMs can be substantially improved while at the same time retaining a good performance in the presence of missing features. The hybrid model also outperforms pure discriminative models indicating that the generative likelihood term is a proper regularizer. Furthermore, we show that unlabeled data can be used in a semi-supervised scenario to improve performance considerably compared to a purely supervised scenario having only access to the labeled data.

The paper is structured as follows. In Section 2, we introduce the notation and review related work. In Section 3, we present the hybrid generative-discriminative objective and show how to handle missing data scenarios. Section 4 shows extensive experiments on real-world data and Section 5 concludes the paper.

2. Background and Related Work

The joint distribution of a parameterized model $p(\boldsymbol{x}, c|\boldsymbol{\theta})$ can be factorized as $p(\boldsymbol{x}|c, \boldsymbol{\theta})\ p(c|\boldsymbol{\theta})$, i.e., a class conditional distribution and a class prior probability for each class. The class conditional distribution of class c is modeled by a GMM with K_c components, i.e.,

$$p(\boldsymbol{x}|c, \boldsymbol{\theta}) = \sum_{k=1}^{K_c} \alpha_{c,k}\ \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_{c,k}, \boldsymbol{\Sigma}_{c,k}), \tag{5}$$

where $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_{c,k}, \boldsymbol{\Sigma}_{c,k})$ denotes the Gaussian probability density with mean $\boldsymbol{\mu}_{c,k}$ and covariance matrix $\boldsymbol{\Sigma}_{c,k}$. The parameters of the joint distribution $p(\boldsymbol{x}, c|\boldsymbol{\theta})$ are given by $\boldsymbol{\theta} = (\pi_1, \ldots, \pi_C, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_C)$ where $p(c|\boldsymbol{\theta}) = \pi_c$, $\pi_c \geq 0$, $\sum_{c=1}^{C} \pi_c = 1$, and $\boldsymbol{\theta}_c = (\boldsymbol{\alpha}_c, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$. $\boldsymbol{\alpha}_c = (\alpha_{c,1}, \ldots, \alpha_{c,K_c})$ contains the component priors of class c, $\alpha_{c,k} \geq 0$, $\sum_{k=1}^{K_c} \alpha_{c,k} = 1$. $\boldsymbol{\mu}_c = (\boldsymbol{\mu}_{c,1}, \ldots, \boldsymbol{\mu}_{c,K_c})$ and $\boldsymbol{\Sigma}_c = (\boldsymbol{\Sigma}_{c,1}, \ldots, \boldsymbol{\Sigma}_{c,K_c})$ define the Gaussian means and covariance matrices for class c.

GMMs are usually trained generatively by maximizing the likelihood using the expectation maximization (EM) algorithm (Dempster et al., 1977). In recent years, several different approaches to train GMMs discriminatively have been proposed. Pernkopf and Wohlmayr (2010) trained GMMs according to the LM criterion using the extended Baum-Welch algorithm. Sha and Saul (2006) introduced LM training of GMMs using the Mahalanobis distance with respect to the covariance matrices. However, none of these approaches additionally incorporates the likelihood in the objective which effectively abandons the generative aspect of the trained models.

Bouchard and Triggs (2004) proposed to optimize an objective that trades off between a generative log-likelihood term and a discriminative CLL term. Their objective is equivalent to the hybrid CLL objective used in this paper, but they did not consider GMMs. Peharz et al. (2013) trained Bayesian networks with the hybrid LM objective used in this paper. They utilize the structure of discrete-valued Bayesian networks to formulate an equivalent convex support vector machine (SVM)-like objective that trades off between a weighted ℓ^1-norm over the parameters and a probabilistic margin term. However, the convexity properties of their objective do not translate to GMMs, rendering GMM training using the hybrid LM objective intrinsically more difficult. Lasserre et al. (2006) proposed to parameterize the joint distribution with two semantically equivalent sets of parameters $(\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}})$ as $p(\boldsymbol{x}, c|\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}) = p(c|\boldsymbol{x}, \boldsymbol{\theta})p(\boldsymbol{x}|\tilde{\boldsymbol{\theta}})$. Rather than formulating a hybrid objective, their approach trades off between the generative and discriminative characteristics at model level by selecting certain prior distributions $p(\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}})$ over the parameters. In particular, an independent prior $p(\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}) = p(\boldsymbol{\theta})p(\tilde{\boldsymbol{\theta}})$ results in a purely discriminative model whereas an equivalence enforcing prior $p(\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}) = p(\boldsymbol{\theta})\delta(\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}})$ results in a purely generative model. Although their approach allows for generative, discriminative, and hybrid training in the same statistically sound framework by maximizing a posterior distribution, respectively, their approach requires the model size to be doubled which limits the practical applicability. Furthermore, the discriminative part of their model is based on the CLL which is inferior to LM optimization on most data sets in our experiments (cf. Section 4). They also showed experiments with GMMs on a rather small data set, but details about the GMMs are not provided.

Besides generative models, that offer a principled framework for missing data scenarios, there exist a wide range of non-generative techniques. A common approach to handle missing features are imputation techniques, such as *mean imputation* (Little and Rubin, 1986), *k-nearest neighbor (k-NN)* imputation (Jönsson and Wohlin, 2004), and *distribution based imputation* (Saar-Tsechansky and Provost, 2007). *Reduced-feature models* classify samples with missing features using a model that was learned by ignoring the corresponding features of the training set (Saar-Tsechansky and Provost, 2007). Common approaches for semi-supervised learning include *self-learning*, *graph based approaches* and *transductive SVMs* (Chapelle et al., 2010).

3. Hybrid Gaussian Mixture Models

The hybrid generative-discriminative objective for probabilistic models consists of a generative log-likelihood (LL) term and a discriminative term. Both terms are weighted according to a hyperparameter $\lambda \in [0, 1]$ such that for $\lambda \to 1$ the purely generative objective is recovered, and for $\lambda \to 0$ the purely discriminative objective is recovered. We investigate two discriminative criterions: (i) the CLL criterion and (ii) the LM criterion. For the CLL criterion, the hybrid objective[1] is given by

$$l^{cll}_{hybrid}(\boldsymbol{\theta}) = \underbrace{-\lambda \sum_{n=1}^{N} \log p\left(\boldsymbol{x}_n, c_n|\boldsymbol{\theta}\right)}_{\text{generative LL term}} \underbrace{- (1-\lambda) \sum_{n=1}^{N} \log p\left(c_n|\boldsymbol{x}_n, \boldsymbol{\theta}\right)}_{\text{discriminative CLL term}}. \tag{6}$$

This objective is equivalent to the objective used in Bouchard and Triggs (2004). For the LM criterion, we begin with some definitions before stating the hybrid objective. The probabilistic margin (Pernkopf et al., 2012; Peharz et al., 2013; Guo et al., 2005) of the n^{th} data sample $(\boldsymbol{x}_n, c_n)$ with parameters $\boldsymbol{\theta}$ is defined as

$$\delta_n\left(\boldsymbol{\theta}\right) = \frac{p(c_n|\boldsymbol{x}_n, \boldsymbol{\theta})}{\max_{c \neq c_n} p(c|\boldsymbol{x}_n, \boldsymbol{\theta})} = \frac{p(\boldsymbol{x}_n, c_n|\boldsymbol{\theta})}{\max_{c \neq c_n} p(\boldsymbol{x}_n, c|\boldsymbol{\theta})}. \tag{7}$$

Since all the classes are considered in (7), the multiclass case is naturally handled. This is similar to the way SVMs have been generalized to the multiclass case by Crammer and Singer (2001). Applying the logarithm to (7) yields

$$\beta_n\left(\boldsymbol{\theta}\right) = \log \delta_n\left(\boldsymbol{\theta}\right) = \log p(\boldsymbol{x}_n, c_n|\boldsymbol{\theta}) - \max_{c \neq c_n} \log p(\boldsymbol{x}_n, c|\boldsymbol{\theta}). \tag{8}$$

A data sample $(\boldsymbol{x}_n, c_n)$ is correctly classified if $\delta_n > 1$, or, similarly, if $\beta_n > 0$. In this case, $p(\boldsymbol{x}_n, c_n|\boldsymbol{\theta})$ is larger than $\max_{c \neq c_n} p(\boldsymbol{x}_n, c|\boldsymbol{\theta})$ where the class $c \neq c_n$ is referred to as the *best competitor class*. Following the intuition of SVMs (Cortes and Vapnik, 1995), it is now desired to obtain a classifier such that the log-margin $\beta_n(\boldsymbol{\theta})$ of all samples is large and no samples are located close to the decision boundary at $\beta = 0$. Therefore, we introduce a *desired log-margin* hyperparameter $\gamma > 0$ and design an objective that penalizes samples whose log-margin $\beta_n(\boldsymbol{\theta})$ is less than γ. The hybrid objective with discriminative LM term is then given by

$$l^{lm}_{hybrid}(\boldsymbol{\theta}) = \underbrace{-\lambda \sum_{n=1}^{N} \log p\left(\boldsymbol{x}_n, c_n|\boldsymbol{\theta}\right)}_{\text{generative LL term}} + \underbrace{(1-\lambda) \sum_{n=1}^{N} \operatorname{hinge}\left(\gamma - \beta_n(\boldsymbol{\theta})\right)}_{\text{discriminative LM term}}, \tag{9}$$

where $\operatorname{hinge}(x) = \max(0, x)$. This objective has a similar structure as the soft-margin objective for SVMs. Note, however, that the soft-margin objective for SVMs has a constant 1 in place of the additional desired log-margin parameter γ. For SVMs, the constant arises due to a particular normalization such that points that satisfy the margin with equality are one unit away from the decision boundary in terms of the linear decision function $\boldsymbol{w}^T\boldsymbol{x} + b$. Such a normalization is not possible for our scenario, and, indeed, in many cases an arbitrary large margin is achievable by shrinking the component covariances or moving the component means far away from the data. The following proposition gives further insights into the discriminative LM term.

Proposition 1. $\left(\sum_{n=1}^{N} \operatorname{hinge}\left(\gamma - \beta_n\left(\boldsymbol{\theta}\right)\right)\right)/\gamma$ *is an upper bound on the number of wrongly classified samples.*

[1] Throughout the paper, all objectives are minimization problems.

Proof. Let $\mathcal{I}_w(\boldsymbol{\theta})$ be the indices of the wrongly classified samples. If a sample is wrongly classified, we have $\beta_n(\boldsymbol{\theta}) \leq 0$ and therefore $\text{hinge}(\gamma - \beta_n(\boldsymbol{\theta})) \geq \gamma$. We have $\sum_{n=1}^{N} \text{hinge}\left(\gamma - \beta_n\left(\boldsymbol{\theta}\right)\right) \geq \sum_{n \in \mathcal{I}_w(\boldsymbol{\theta})} \text{hinge}\left(\gamma - \beta_n\left(\boldsymbol{\theta}\right)\right) \geq |\mathcal{I}_w(\boldsymbol{\theta})|\gamma$. Dividing by γ concludes the proof. □

3.1. Reparameterization of GMM Parameters

The parameters of GMMs are subject to several constraints. The class prior probabilities π_c and the component priors $\alpha_{c,k}$ are constrained to be non-negative and sum up to one. Each component prior $\alpha_{c,k}$ can be reparameterized as a function of $\boldsymbol{z}_c = (z_{c,1}, \ldots, z_{c,K_c})$ by

$$\alpha_{c,k}\left(\boldsymbol{z}_c\right) = \frac{\exp\left(z_{c,k}\right)}{\sum_{k'=1}^{K_c} \exp\left(z_{c,k'}\right)}, \tag{10}$$

where $z_{c,k}$ is unconstrained (Pernkopf et al., 2012). The new values $\alpha_{c,k}$ satisfy the non-negativity and sum-to-one constraints due to the non-negativity of the exponential.[2] The constraints on the class priors π_c can be handled analogously.

Moreover, the covariance matrices $\boldsymbol{\Sigma}_{c,k}$ are constrained to be symmetric and positive definite. We reparameterize the covariance matrices using a diagonal matrix plus low-rank matrix (DPLR) approximation as

$$\boldsymbol{\Sigma}_{c,k}(\boldsymbol{d}_{c,k}, \boldsymbol{S}_{c,k}) = \boldsymbol{I}\varepsilon + \text{diag}(\boldsymbol{d}_{c,k}) + \boldsymbol{S}_{c,k}\boldsymbol{S}_{c,k}^T \tag{11}$$

where $\boldsymbol{S}_{c,k}$ is an unconstrained $D \times R$ matrix, $\boldsymbol{d}_{c,k}$ are positive entries of a diagonal matrix, $\boldsymbol{I}$ is the identity matrix, and $\varepsilon > 0$ is a small regularizer for the diagonal. It is straightforward to show that this reparameterization leads to positive definite matrices. Note that $\boldsymbol{d}_{c,k}$ are itself constrained to be positive. For these parameters, we apply the softplus reparameterization

$$d(z) = \log(1 + \exp z) \tag{12}$$

with unconstrained z. The parameter R determines the rank of the low-rank approximation $\boldsymbol{S}_{c,k}\boldsymbol{S}_{c,k}^T$. This reparameterization does not only come with the advantage of making the optimization problem unconstrained but it also reduces the number of parameters and increases the computational efficiency while still allowing us to model important correlations. The Gaussian pdf requires a costly matrix inversion and determinant computation, both of which have a time complexity of $O(D^3)$. This can be prohibitive if D is large, especially since these operations must be computed in each iteration of gradient based optimization schemes. Let $\boldsymbol{A} = \boldsymbol{I}\varepsilon + \text{diag}(\boldsymbol{d})$ be a diagonal matrix. The special form of (11) can be utilized by the matrix determinant lemma to compute the determinant as

$$\det(\boldsymbol{A} + \boldsymbol{S}\boldsymbol{S}^T) = \det(\boldsymbol{I} + \boldsymbol{S}^T\boldsymbol{A}^{-1}\boldsymbol{S})\det(\boldsymbol{A}), \tag{13}$$

and the Woodburry matrix identity to compute the matrix inversion as

$$(\boldsymbol{A} + \boldsymbol{S}\boldsymbol{S}^T)^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{S}(\boldsymbol{I} + \boldsymbol{S}^T\boldsymbol{A}^{-1}\boldsymbol{S})^{-1}\boldsymbol{S}^T\boldsymbol{A}^{-1}. \tag{14}$$

Equations (13) and (14) require only determinants and inversions of diagonal matrices and $R \times R$ matrices, respectively, which reduces computation time substantially in case $R \ll D$. In practice, it turns out that modeling correlations is crucial to achieve a good performance but already a relatively small value for R is sufficient to improve the performance substantially compared to diagonal covariance matrices (cf. Section 4).

3.2. Smoothed maximum function

Since the gradient of the maximum function is zero for entries that are not maximal, we introduce the smooth *soft-max* approximation (Pernkopf et al., 2012; Peharz et al., 2013)

$$\text{smax}\left(t_1, \ldots, t_L\right) = \frac{1}{\nu} \log \sum_{i=1}^{L} \exp\left(\nu\, t_i\right) \tag{15}$$

for the maximum function in the log-probabilistic margin (8). The parameter $\nu > 0$ governs the smoothness of the soft-max function. As $\nu \to \infty$, the maximum function is recovered.

3.3. Hybrid GMMs for Missing Features and Semi-supervised Learning

Given that the MAR property holds, marginalization over the missing features is valid to perform both ML estimation and classification according to the class with the highest posterior probability (Marlin, 2008). To estimate ML parameters of a GMM, the EM algorithm can be extended to handle missing features (Ghahramani and Jordan, 1993) or general optimization algorithms can be used to maximize the marginal likelihood directly. To marginalize out the missing features in GMMs, it suffices to remove the entries of the component means and the rows and columns of the component covariance matrices that correspond to the missing features (Bishop, 2006). Note that the DPLR structure from Section 3.1 can also be used to obtain the resulting covariance matrix by removing the corresponding entries and rows from $\boldsymbol{d}_{c,k}$ and $\boldsymbol{S}_{c,k}$, respectively. This allows us to compute the necessary matrix inverses and determinants of the reduced (but still larger than $R \times R$) matrices efficiently using (13) and (14). This is especially important at test time when the missing feature patterns are not known beforehand and the matrix inverses and determinants cannot be precomputed.

To use additional unlabeled data, we include the marginal distribution $p(\boldsymbol{x}|\boldsymbol{\theta})$ of the unlabeled data in the hybrid objective. To mitigate the possible negative effect of unlabeled data samples, we introduce another trade-off parameter $\kappa \in [0, 1]$ that governs the influence of the unlabeled data (Nigam et al., 2000). The hybrid objective for semi-supervised learning with large margin criterion is defined as

$$\begin{aligned} l_{ssl}^{lm}\left(\boldsymbol{\theta}\right) = \lambda & \left(-\kappa \sum_{n=1}^{N_l} \log p(\boldsymbol{x}_n^l, c_n|\boldsymbol{\theta}) - (1-\kappa) \sum_{n=1}^{N_u} \log p\left(\boldsymbol{x}_n^u|\boldsymbol{\theta}\right) \right) \\ & + (1-\lambda) \sum_{n=1}^{N_l} \text{hinge}\left(\gamma - \beta_n\left(\boldsymbol{\theta}\right)\right). \end{aligned} \tag{16}$$

For $\kappa \to 1$, the pure supervised objective is recovered. The additional trade-off parameter κ is useful especially in case the

[2] Note that the newly introduced variables $z_{c,k}$ are not unique.

number of unlabeled data samples N_u exceeds the number of labeled data samples N_l by orders of magnitudes. In this case, the influence of the labeled data would become weak and the parameters are solely determined by the unlabeled data.

4. Experiments

We performed our experiments on MNIST (Lecun et al., 1998), variants of MNIST (Larochelle et al., 2007), CIFAR-10 (Krizhevsky, 2009), and TIMIT (Zue et al., 1990). Additional experiments on synthetic toy data sets are provided in Section 2 of the supplementary material.

4.1. Data Sets

MNIST: The MNIST data set for handwritten digit recognition (Lecun et al., 1998) contains 60000 training images and 10000 test images of size 28×28 with grayscale color values in $[0, 1]$. We split the training set into 50000 training samples and 10000 validation samples and treat each pixel as feature, i.e., $\boldsymbol{x} \in \mathbb{R}^{784}$. We also used five variants of the MNIST data set (Larochelle et al., 2007) where the images of the standard MNIST data set have been transformed by various operations, namely rotations and/or the insertion of either images or random pixel values as background. Each of these data sets contain 10000 training images, 2000 validation images, and 50000 test images. More details about the transformations of the MNIST variants are provided in Section 1 of the supplementary material.

CIFAR-10: The CIFAR-10 data set (Krizhevsky, 2009) contains 50000 training images and 10000 test images of size 32×32 pixels with RGB color values in $[0, 1]$. The images depict one out of ten object classes, i.e., airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. We split the training set into 40000 training images and 10000 validation images. Again, we treat each color value of each pixel as feature without further preprocessing, i.e., $\boldsymbol{x} \in \mathbb{R}^{3072}$.

TIMIT: The TIMIT data set is used for speech classification (Zue et al., 1990). Each sample consists of 92 features which represent a phonetic segment that is classified to one of 39 phonemes. The data is split into 140173 training samples, 50735 validation samples (test) and 7211 test samples (core test). Details on data preprocessing can be found in (Halberstadt and Glass, 1997).

Except for TIMIT, we also conducted all experiments by first whitening the data and reducing the number of dimensions to 50 with PCA. Some exemplary samples of the image data sets are shown in Section 1 of the supplementary material.

4.2. Classification Experiments

We compare the classification performance of GMMs trained with different objectives and covariance structures, i.e., LL, CLL, LM optimized GMMs, and their hybrid counterparts, respectively, for both diagonal and full covariance matrices.

4.2.1. Experimental Setup

We trained generative GMMs with the EM algorithm (Dempster et al., 1977) using 20 random restarts and $K_c \in \{1, \ldots, 20\}$ for full covariance matrices, and $K_c \in \{1, \ldots, 100\}$ for diagonal covariance matrices. We used a uniform class prior for MNIST and CIFAR-10, since the number of samples of each class is almost identical for these data sets. For TIMIT, we used the empirical prior, obtained by the fraction of samples of each class in the training set, since the class distribution of TIMIT is non-uniform. We selected the number of components $(K_1, \ldots, K_C)$ by jointly optimizing them to maximize the classification performance on a held-out validation set using 500 iterations of Bayesian optimization (Snoek et al., 2012).[3] The corresponding test errors are reported as LL in Table 1. We fix $(K_1, \ldots, K_C)$ and use the resulting model to initialize the discriminative and hybrid models. For the non-diagonal case, we used the following approach to compute $\boldsymbol{S}_{c,k}$ and $\boldsymbol{d}_{c,k}$ of the DPLR matrices from the full covariance matrices $\boldsymbol{\Sigma}_{c,k}$ obtained with the EM algorithm: Let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_D$ and $w_1 \geq \ldots \geq w_D$ be the normalized eigenvectors and corresponding eigenvalues of $\boldsymbol{\Sigma}_{c,k}$. We initialize $\boldsymbol{S}_{c,k} = [\,\sqrt{w_1}\boldsymbol{v}_1, \cdots, \sqrt{w_R}\boldsymbol{v}_R]$ and $\boldsymbol{d}_{c,k}$ to be the diagonal entries of $\boldsymbol{\Sigma}_{c,k} - \boldsymbol{S}_{c,k}\boldsymbol{S}_{c,k}^T$. The discriminative and hybrid models are then optimized using the stochastic optimization algorithm ADAM (Kingma and Ba, 2015). We approximate the gradient of the objective using minibatches of 100 samples for MNIST and CIFAR-10, and 300 samples for TIMIT, respectively. We optimized for 500 epochs on the variants of MNIST and for 100 epochs on the remaining data sets. We report the test error for the best model on the validation set during optimization rather than the test error at the end of optimization. We optimized the hyperparameters $\lambda, \gamma \in [10^{-2}, 10^{2}]$, $R \in \{1, \ldots, 25\}$ and the step size of ADAM $\eta \in [10^{-5}, 10^{-2}]$ jointly on a separate held-out validation set using 100 iterations of Bayesian optimization.[4] We fixed the smoothness parameter of the softmax approximation $\nu = 10$ as in (Peharz et al., 2013). The whole setup was executed four times for the fixed regularizers $\varepsilon \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$ of the diagonals of *all* covariance matrices and we report the result leading to the best validation error. All experiments were performed in Python using the automatic differentiation framework Theano (Theano Development Team, 2016).[5]

4.2.2. Results

The results are shown in Table 1. For full covariance matrices, the generative objective (LL) gets outperformed by the hybrid and discriminative objectives on almost all data sets. More importantly, the hybrid objectives also tend to outperform their purely discriminative counterparts on almost all data sets. This behavior occurs less frequently for diagonal covariance matrices where discriminative LM GMMs perform best on most data sets. This is explained by the fact that in the diagonal case the ML parameters obtained with the EM algorithm are used di-

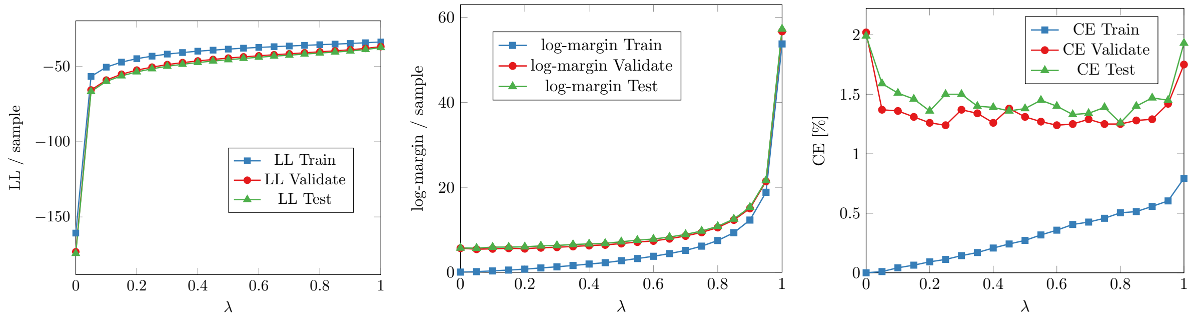
[3] Note that each class c has its individual number of components K_c.

[4] Note that not all parameters are needed for all objectives. For instance, the λ parameter is not used for purely discriminative models.

[5] Code available online at https://github.com/wroth8/hybrid-gmm

Table 1. Test classification errors [%] of various GMMs for diagonal and full covariance matrices. The following objectives are compared: LL (log-likelihood), LM (large margin), CLL (conditional log-likelihood). For full covariance matrices, the pure generative model (LL) uses general full covariance matrices, whereas the remaining models use DPLR covariance matrices. The best results for each data set are shown in bold face.

	full covariance					diagonal covariance				
dataset	LL	LL+LM	LL+CLL	LM	CLL	LL	LL+LM	LL+CLL	LM	CLL
MNIST	1.86	1.76	1.73	**1.57**	1.68	4.08	2.40	3.47	**2.32**	3.22
MNIST (pca50)	1.66	**1.42**	1.81	1.51	1.66	4.09	**2.53**	3.05	**2.53**	2.79
MNIST Basic	3.198	**2.862**	2.968	2.984	3.030	5.758	4.516	5.470	**4.284**	5.088
MNIST Basic (pca50)	3.162	**3.122**	3.346	3.146	3.408	6.410	4.754	6.478	**4.738**	5.552
MNIST Background	24.842	**19.228**	19.392	19.894	19.992	43.522	22.512	26.416	**22.280**	27.376
MNIST Background (pca50)	21.470	**18.840**	20.684	19.392	22.312	30.540	**22.446**	28.402	22.944	28.150
MNIST Background Random	16.132	11.356	**10.786**	14.762	13.624	12.972	12.644	12.790	13.000	**12.532**
MNIST Background Random (pca50)	8.238	7.878	**7.814**	8.432	9.816	11.890	9.912	10.994	**9.660**	11.712
MNIST Rotated	11.284	**9.854**	10.610	11.116	11.662	19.104	**14.010**	16.426	14.114	17.058
MNIST Rotated (pca50)	11.710	**9.962**	11.710	11.656	12.224	19.686	16.466	19.998	**16.356**	18.124
MNIST Rotated Background	57.750	**50.578**	52.192	52.002	53.482	75.718	57.014	64.032	**56.416**	65.194
MNIST Rotated Background (pca50)	52.134	**49.414**	50.184	49.466	51.858	60.512	**52.430**	57.416	52.562	59.094
CIFAR-10	49.96	**46.27**	49.48	46.42	49.70	61.32	**53.72**	57.70	53.96	57.90
CIFAR-10 (pca50)	50.67	**48.58**	51.14	50.34	53.00	56.79	52.59	56.23	**52.52**	55.47
TIMIT	25.433	**20.386**	23.423	20.649	23.381	28.373	22.535	26.834	**22.355**	26.751

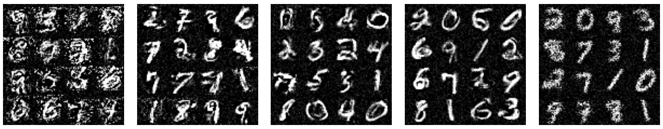


(a) (b) (c)

Fig. 1. Influence of the trade-off parameter λ on (a) the log-likelihood (LL), (b) the log-margin, and (c) the classification error (CE). The plots were computed on MNIST (pca50) for fixed $\gamma = 100$.

rectly as initial parameters and performing only a few iterations of ADAM optimization with early stopping preserves much of the model's generative semantics. However, when we apply the proposed procedure to obtain DPLR matrices from general covariance matrices, the local optimality of the parameters obtained with the EM algorithm with respect to the log-likelihood is in general lost. This allows the generative term of the hybrid objective to recover the generative semantics which is not possible for the discriminative models. Nevertheless, modeling correlations in the covariance matrices is crucial to achieve a good performance as the models using DPLR covariance matrices outperform the models using diagonal covariance matrices consistently on all data sets by a large margin. Interestingly, the PCA transformation improves the performance especially on data sets with *Background* artifacts as the noise is not modeled in the first principal components, whereas on other data sets too much valuable information gets lost. The best hyperparameters that correspond to the results in Table 1 are summarized in Section 3 of the supplementary material.

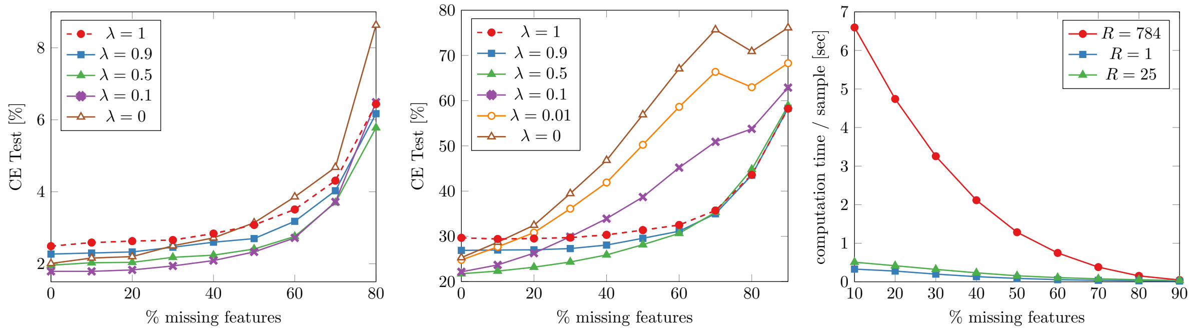
Figure 1 illustrates how the generative-discriminative trade-off parameter λ influences several aspects of the model. The figure illustrates the results after 100 epochs of optimizing the hybrid LM objective and fixed $\gamma = 100$ on MNIST (pca50). Figure 1(a) shows how the generative semantics of the model, measured by the log-likelihood of the data, degrades by moving from the generative model ($\lambda = 1$) to the discriminative model ($\lambda = 0$). The generative semantics is completely abandoned for the discriminative model. The behavior of the margin is shown in Figure 1(b). In the purely generative case, the model is trained completely unaware of the margin term. As we move from the generative to the discriminative objective, the margin consistently decreases until almost all training samples satisfy the desired margin at $\lambda = 0$. The classification error depending on λ is shown in Figure 1(c). For $\lambda = 1$, the model clearly underfits the data as there is a large error on both the training set and the test set. The opposite happens for $\lambda = 0$: The model appears to overfit the data as all the training samples are classified correctly but the test error increases again. An intermediate value for λ, where both the generative and discriminative aspects are considered, appears to be just right.



(a) $\lambda = 0$ (b) $\lambda = 0.01$ (c) $\lambda = 0.1$ (d) $\lambda = 1$ (e) $\lambda = 1$, diag

Fig. 2. Samples generated by hybrid LM GMMs that were trained with different values of λ and fixed $\gamma = 100$. (a)-(d) GMMs with DPLR covariance matrix structure using $R = 25$. (e) GMMs with diagonal covariance matrices.

In the next experiment, we investigated the generative semantics of hybrid LM GMMs by sampling from the generative model. We trained several models on MNIST for different



(a) MNIST

(b) TIMIT

(c) MNIST

Fig. 3. Influence of λ on the test classification error (CE) of hybrid LM GMMs (fixed $\gamma = 100$) in the presence of missing features on (a) MNIST and (b) TIMIT. (c) Running time of classifying a single MNIST sample for different amounts of missing features. $R = 784$ corresponds to full covariance matrices. For $R < 784$, the DPLR matrix structure is exploited.

trade-off parameters λ and fixed $\gamma = 100$. We performed 100 epochs of ADAM training using minibatches of 100 samples without early stopping. The sampled digits are shown in Figure 2. Figures 2(a)-(d) show samples from GMMs with DPLR covariance matrices using $R = 25$. The purely discriminative model in Figure 2(a) has lost most of its generative semantics, and it is difficult to identify the digits shown in the sampled images. In Figure 2(b), the model is trained with a generative contribution and, consequently, the image quality has improved substantially. In Figure 2(c), λ is further increased causing the number of artifacts in the images to decrease. Images sampled from the purely generative model are shown in Figure 2(d). Figure 2(e) shows samples from a generative GMM with *diagonal* covariance matrices. The images appear noisy as correlations, causing neighboring pixels to have similar intensities, are not modeled. This gives an intuition why diagonal covariances perform worse than full covariance matrices. Nevertheless, the images generated for full covariances indicate that a relatively small $R = 25$ is sufficient to model the most important correlations.

4.3. *Missing Feature Experiments*

We conducted classification experiments with different numbers of missing features. Let p be the fraction of missing features. We randomly selected for each sample a set of $round(Dp)$ feature indices with uniform probability and marked these features as unobserved. As a result, the MAR assumption holds and we treat missing features at test time by marginalization as described in Section 3.3. We trained several models with varying λ and fixed $\gamma = 100$ for 100 epochs without early stopping on MNIST (Figure 3(a)) and TIMIT (Figure 3(b)).

We observe a similar behavior as Peharz et al. (2013); generative GMMs have a relatively large error when no features are missing but its performance does not degrade severely for up to 40-50% missing features. The performance of very discriminative models with $\lambda \leq 0.1$ degrade faster such that they only outperform the generative model for a few missing features. On both data sets, the curve for $\lambda = 0.5$ shows substantial improvements over the generative model for a few missing features while at the same time performing equally well in case of many missing features.

Furthermore, we want to stress that the DPLR structure can be utilized to substantially improve the running time in case of missing features. If there are no missing features or the number of different missing feature patterns is small, the matrix inversions and determinant computations of the covariance matrices can be precomputed. However, if the patterns are not known beforehand, these operations must be computed anew for each individual sample. As shown in Section 3.3, the DPLR structure can be utilized to perform these operations on smaller $R \times R$ matrices and diagonal matrices in case $D > R$. The average computation times for classifying a *single* MNIST sample with a total of 170 covariance matrices to process are shown in Figure 3(c). The average computation times were obtained by classifying 1000 samples on an Intel Core i5-4690 CPU with multithreading disabled. Note that, for instance, in case of only 10% missing features, this requires inverting and computing the determinants of 170 706×706 matrices. In this case, classification of a single sample takes up to 6.6 seconds for 10% missing features. The same procedure scales much better for the DPLR matrix structure and only requires at most 0.5 seconds for $R = 25$. For fully observed data, classifying a *single* sample with precomputed determinants and matrix inversions took 70 ms if each sample is processed individually and 5 ms if they are processed in batch mode, i.e., all samples are processed at once by utilizing efficient matrix operations.

4.4. *Semi-Supervised Experiments*

In our last experiment, we compare the supervised and semi-supervised setting of hybrid GMMs for both discriminative LM and discriminative CLL term on the MNIST (pca50) data set. We trained GMMs with $N_l \in \{100, 250, 500, 1000, 2500, 5000, 10000, 25000\}$ labeled samples in the supervised setting and used the remaining $N_u = 50000 - N_l$ samples as additional unlabeled data. After training the initial parameters with the EM algorithm on the labeled data only, we optimized the hybrid objective with ADAM for 100 epochs. For semi-supervised learning, we used a minibatch size of 100. Since supervised learning uses less samples, we selected smaller minibatch sizes of $(1, 2, 5, 10, 25, 50, 100, 100)$ for the different values of N_l to have a similar number of parameter updates as in the semi-supervised setting. To reduce

Table 2. Test classification errors [%] for semi-supervised learning (SSL) and supervised learning (SV) using both LM and CLL criterion on MNIST (pca50). For SSL, the total number of samples is $N_l + N_u = 50000$. For SV, we have $N_u = 0$.

N_l	100	250	500	1000	2500	5000	10000	25000
SV (LM)	28.32	15.65	9.93	6.42	3.91	3.29	2.42	1.72
SSL (LM)	5.08	4.36	4.65	3.54	2.84	3.30	2.20	1.80
SV (CLL)	27.62	15.78	10.44	6.96	4.52	3.64	3.01	2.27
SSL (CLL)	5.00	4.74	4.21	4.45	2.90	2.95	2.79	2.19

the number of hyperparameters, we used a *shared* number of components per class $K = K_c$. We jointly optimized K, λ, κ, γ, η and R using 50 iterations of Bayesian optimization. We evaluated the numbers of components $K \in \{1, 2, 3\}$ for $N_l \leq 250$ and $K \in \{1, \dots, 5\}$ for $N_l > 250$ and used the same ranges for the remaining hyperparameters as in Section 4.2. We report the test error leading to the best validation error. The best hyperparameters are shown in Section 3 of the supplementary material.

The results are shown in Table 2. The semi-supervised objective benefits consistently from the unlabeled data and outperforms the supervised objective. Especially in the regime of only a few labeled samples and many unlabeled samples, there is a considerable performance gap between the two approaches. The fluctuations in test error, where the performance gets worse for more labeled samples, are not present in the validation error.

5. Conclusion

Recent work has proposed several ways to train GMMs discriminatively to achieve a higher classification performance than generative ML optimized GMMs (Sha and Saul, 2006; Pernkopf and Wohlmayr, 2010). Since none of these approaches takes the likelihood into account, we proposed a principled method to train GMMs in a hybrid generative-discriminative way by optimizing objectives that trade off between a generative likelihood term and either a conditional log-likelihood (CLL) term or a large margin (LM) term to utilize the advantages of both worlds.

We compared our model with standard ML GMMs and pure discriminative CLL and LM optimized GMMs on several real-world data sets. Our hybrid model considerably outperforms ML GMMs. Especially hybrid GMMs with LM term achieve the best classification error on 12 out of 15 data sets. We also showed that non-diagonal covariance matrices are crucial to achieve a good performance. Therefore, we used a diagonal plus low-rank structure (DPLR) for the covariance matrices to keep the number of parameters small. Furthermore, we evaluated several missing data scenarios. We performed experiments by removing various numbers of features at test time and showed that hybrid GMMs outperform generative GMMs substantially for a small number of missing features while at the same time performing equally well in case of many missing features. Moreover, we showed that the DPLR structure of the covariance matrices can be utilized to reduce the classification time for data containing missing features considerably. Finally, hybrid GMMs showed convincing performance on a semi-supervised task. Especially in case of only a few labeled samples, the additional unlabeled data leads to a substantial increase in classification performance.

Acknowledgments

This work was supported by the Austrian Science Fund (FWF) under the project numbers P27803-N15 and I2706-N31. This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Grant Agreement No. 797223 — HYBSPN. Furthermore, we acknowledge NVIDIA for providing GPU computing resources.

References

Bishop, C.M., 2006. Pattern Recognition and Machine Learning. volume 1. Springer New York.

Bouchard, G., Triggs, B., 2004. The tradeoff between generative and discriminative classifiers, in: IASC International Symposium on Computational Statistics (COMPSTAT), pp. 721–728.

Chapelle, O., Schölkopf, B., Zien, A. (Eds.), 2010. Semi-Supervised Learning. The MIT Press.

Cortes, C., Vapnik, V., 1995. Support-vector networks. Machine Learning 20, 273–297.

Crammer, K., Singer, Y., 2001. On the algorithmic implementation of multiclass kernel-based vector machines. JMLR 2, 265–292.

Dempster, A.P., Laird, N.M., Rubin, D.B., 1977. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. Series B 39, 1–38.

Ghahramani, Z., Jordan, M.I., 1993. Supervised learning from incomplete data via an EM approach, in: NIPS, pp. 120–127.

Guo, Y., Wilkinson, D.F., Schuurmans, D., 2005. Maximum margin Bayesian networks, in: UAI, pp. 233–242.

Halberstadt, A.K., Glass, J.R., 1997. Heterogeneous acoustic measurements for phonetic classification, in: EUROSPEECH.

Jönsson, P., Wohlin, C., 2004. An evaluation of k-nearest neighbour imputation using Likert data, in: International Software Metrics Symposium (METRICS), pp. 108–118.

Kingma, D.P., Ba, J., 2015. Adam: A method for stochastic optimization, in: ICLR. ArXiv: 1412.6980.

Krizhevsky, A., 2009. Learning multiple layers of features from tiny images. Technical Report. University of Toronto.

Larochelle, H., Erhan, D., Courville, A.C., Bergstra, J., Bengio, Y., 2007. An empirical evaluation of deep architectures on problems with many factors of variation, in: ICML, pp. 473–480.

Lasserre, J.A., Bishop, C.M., Minka, T.P., 2006. Principled hybrids of generative and discriminative models, in: CVPR, pp. 87–94.

Lecun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE 86, 2278–2324.

Little, R.J.A., Rubin, D.B., 1986. Statistical Analysis with Missing Data. John Wiley & Sons, Inc.

Marlin, B.M., 2008. Missing Data Problems in Machine Learning. Ph.D. thesis. University of Toronto.

Nigam, K., McCallum, A., Thrun, S., Mitchell, T.M., 2000. Text classification from labeled and unlabeled documents using EM. Machine Learning 39, 103–134.

Peharz, R., Tschiatschek, S., Pernkopf, F., 2013. The most generative maximum margin Bayesian networks, in: ICML, pp. 235–243.

Pernkopf, F., Wohlmayr, M., 2010. Large margin learning of Bayesian classifiers based on Gaussian mixture models, in: ECML PKDD, pp. 50–66.

Pernkopf, F., Wohlmayr, M., Tschiatschek, S., 2012. Maximum margin Bayesian network classifiers. Transactions on Pattern Analysis and Machine Intelligence (TPAMI) 34, 521–532.

Saar-Tsechansky, M., Provost, F.J., 2007. Handling missing values when applying classification models. JMLR 8, 1623–1657.

Sha, F., Saul, L.K., 2006. Large margin Gaussian mixture modeling for phonetic classification and recognition, in: ICASSP, pp. 265–268.

Snoek, J., Larochelle, H., Adams, R.P., 2012. Practical Bayesian optimization of machine learning algorithms, in: NIPS, pp. 2960–2968.

Theano Development Team, 2016. Theano: A Python framework for fast computation of mathematical expressions. arXiv e-prints abs/1605.02688.

Zue, V., Seneff, S., Glass, J., 1990. Speech database development at MIT: Timit and beyond. Speech Communication 9, 351–356.