Sparse Compositional Local Metric Learning

Joseph St.Amand, Jun Huan
Electrical Engineering & Computer Science
University of Kansas
1450 Jayhawk Blvd
Lawrence, Kansas 66045

ABSTRACT

Mahalanobis distance metric learning becomes an especially challenging problem as the dimension of the feature space p is scaled upwards. The number of parameters to optimize grows with space complexity of order $O(p^2)$, making storage infeasible, interpretability poor, and causing the model to have a high tendency to overfit. Additionally, optimization while maintaining feasibility of the solution becomes prohibitively expensive, requiring a projection onto the positive semi-definite cone after every iteration. In addition to the obvious space and computational challenges, vanilla distance metric learning is unable to model complex and multi-modal trends in the data.

Inspired by the recent resurgence of Frank-Wolfe style optimization, we propose a new method for sparse compositional local Mahalanobis distance metric learning. Our proposed technique learns a set of distance metrics which are composed of local and global components. We capture local interactions in the feature space, while ensuring that all metrics share a global component, which may act as a regularizer. We optimize our model using an alternating pairwise Frank-Wolfe style algorithm. This serves a dual purpose, we can control the sparsity of our solution, and altogether avoid any expensive projection operations. Finally, we conduct an empirical evaluation of our method with the current state of the art and present the results on five datasets from varying domains.

CCS CONCEPTS

• **Theory of computation** → *Semidefinite programming*; *Semi-supervised learning*; • **Computing methodologies** → *Supervised learning by classification*; *Feature selection*; *Regularization*;

KEYWORDS

Metric Learning, Mahalanobis, Sparse Learning, High Dimensional Feature Space

ACM Reference format:
Joseph St.Amand, Jun Huan. 2017. Sparse Compositional Local Metric Learning. In *Proceedings of KDD '17, Halifax, NS, Canada, August 13-17, 2017,* 8 pages.
https://doi.org/10.1145/3097983.3098153

KDD '17, August 13-17, 2017, Halifax, NS, Canada

ACM ISBN 978-1-4503-4887-4/17/08…$15.00
https://doi.org/10.1145/3097983.3098153

1 INTRODUCTION

Distance metrics are the core of many machine learning algorithms, including k-Means clustering[34], ranking[27], k-Nearest Neighbors[33] and many others. In this paper, we address the problem of learning a locally-adaptive distance metric for data with high dimensional input (i.e. feature) spaces. Specifically, we propose a method for local distance metric learning which learns the matrix parameterizing the metric directly in the input space. A distinguishing property of our proposed method is that it is both *locally adaptive* and *scalable* with respect to the dimension of the input space.

Learning a distance metric is a well-studied problem, refer to the surveys [1] and [20] for a summary of recent works. In general, learning a distance metric is a computationally expensive procedure, with most existing algorithms having from $O(p^2)$ to $O(p^{6.5})$ computational complexity, where p is the dimension of the input space. This computational expense stems from the requirement that the metric matrix $M \in \mathbb{R}^{p \times p}$ must be symmetric and positive semi-definite (p.s.d). The most common approach to maintain that M is p.s.d is via a projection onto the positive semi-definite cone after each iteration, with a computational cost on the order of $O(p^3)$. In addition, as p is scaled upwards it may simply become infeasible to store M in memory. Finally, learning on the order of $O(p^2)$ parameters greatly increases the chance of over-fitting.

In light of the above challenges, relatively little work has been done on learning M directly in the high dimensional feature space [6, 9, 25, 26]. Until recently, most work on distance metric learning has addressed high dimensional input spaces by first compressing the space via a projection onto a low-dimensional manifold[5, 24, 32, 33, 36]. This is typically done via an eigen-decomposition, the computational cost of which is obviously prohibitive. Additionally, projections of this nature are sensitive to unit changes or re-scaling of the data, and may not preserve information which is well-suited to learning a distance from. This is in contrast to our work, which does not compress or reduce the input space dimension via a projection or other means.

A number of works address the case of metric learning for high-dimensional data by taking a low-rank(structured) approach [10, 18, 33]. These works operate by indirectly learning an L such that $M = L^T L$, this removes the need to enforce that M is positive semi-definite, and the rank of L can be controlled by selecting the trailing dimension of L. This reduces the total number of independent parameters to be learned, but also introduces some drawbacks. The matrix L is typically full rank and may have challenging storage requirements. Additionally, virtually all works of this type result in a non-convex objective, which may be difficult to optimize and is often plagued with many local minima [20]. One work has resulted in a convex objective[4], but limits the solution space to the span

of L. These works also have no means to control the sparsity of M, which may harm the interpretability of the model. Our proposed method is convex, has no restrictions introduced by the range of a low-rank projection operator, and encourages interpretability through the use of sparsity inducing penalties.

Traditional approaches to metric learning take a global approach, and make the assumption that feature interactions are consistent across the input space. In cases where the decision boundary is too complex, or the data is multi-modal, a global metric may be too inflexible. A popular approach is to focus on learning multiple metrics, each of which is local to a different spacial area of the input space. Works of this type have the ability to capture non-heterogeneous feature interactions, and have in many cases been show capable of outperforming their global counterparts [6, 8, 12, 32, 33, 37]. In our work we propose the use of a two part compositional metric consisting of both global and local components. Our approach is to separate portions which model global data trends, and those interactions which are confined to a local area. We structure the metric such that the global portion is limited to the diagonal, and the local portion is sparse and positive semi-definite.

Specifically, we introduce a sparse compositional metric for high-dimensional data which is locally adaptive. We propose a two part compositional metric, allowing it to capture global trends, while remaining sensitive to local interactions. We develop an efficient Frank-Wolfe style alternating optimization algorithm which maintains that M remains within the constraint region between all iterates. This allows us to avoid any expensive projections onto the positive semidefinite cone, enforces our choice of structure on the global portion of the metric, and removes the requirements of ever having to store a full matrix M. Finally, because of the compositional design of our algorithm, it lends well to an efficient implementation.

In summary, in this paper we claim the following contributions:

- We propose a new algorithm for learning a sparse compositional metric for high-dimensional data, which consists of both global and local components.
- To our knowledge, our proposed method is the first locally-adaptive distance metric which is learned directly in the input space.
- We provide an empirical evaluation of our method against the current state of the art via a classification scenario.
- Finally, we make our code freely available for download, to aid in research reproducibility.

2 BACKGROUND & RELATED WORK

2.1 Frank-Wolfe Style Optimization

Frank-Wolfe optimization (also referred to as the conditional gradient method) [7] has recently experienced a resurgence of interest, primarily due to its capability of producing projection-free updates. Much recent work has focused on variants of Frank-Wolfe methods and their convergence properties ([14, 15, 21] and others), with numerous applications including video co-localization [17], particle filtering [22], any many others. Updates in the Frank-Wolfe method are typically much cheaper in comparison to full projections, but often require many more iterations for convergence. Additionally, Frank-Wolfe suffers from sublinear convergence rates when the solution lies on the border of the constraint region [21].

Frank-Wolfe based techniques operate on problems of the form:

$$\min_{\vec{x} \in \mathcal{S}} f(\vec{x})$$

where f is a convex function with a smooth Lipshitz continuous gradient and $\mathcal{S}$ is a compact convex set. Each Frank-Wolfe update is calculated by linearizing f about the current iterate $\vec{x} \in \mathcal{S}$ and solving a linear subproblem for the descent direction. The solution to each subproblem is given by the linear minimization oracle (LMO) (equation 1), where $< \cdot, \cdot >$ denotes the inner Frobenius norm and $\nabla f(\vec{x})$ is the gradient of f at $\vec{x}$.

At any point during the Frank-Wolfe optimization procedure, the solution may be given by the combination $\alpha_1\vec{x}_1 + \alpha_2\vec{x}_2 + ... + \alpha_k\vec{x}_k$, where $\sum_{i=1}^{k} \alpha_i = 1$ and $\alpha_i > 0, \quad i \in \{1, ..., k\}$. Each $\vec{x}_i$ is a solution found by solving the LMO at some point in the optimization. Note that the vectors $\vec{x}_i$ are commonly referred to as "atoms". The power of this method lies in the fact that as long as $\mathcal{S}$ is convex and compact, the solution will never leave the constraint region.

$$\min_{\vec{x} \in \mathcal{S}} < \vec{x}, \nabla f(\vec{x}) > \tag{1}$$

Sparsity may be induced by the Frank-Wolfe technique in two ways:

(1) During each iteration, a single atom is added to the solution. The total number of atoms is upper-bounded by the maximum number of iterations, the solution is then the result of a sparse combination of basis atoms.
(2) In the case where each atom $\vec{x} \in \mathcal{S}$ itself is sparse, sparsity is produced as a combination of sparse basis atoms. In this situation the solution need not be stored directly, only the basis atoms and their corresponding weights are stored. This is the approach we take in this paper.

Finally, the quality of the solution may be monitored via an upper bound on the duality gap (equation 2), this is often referred to as the Frank-Wolfe gap. In equation 2, $\vec{d}_{FW}$ denotes the Frank-Wolfe direction found by the linear minimization oracle.

$$\text{Frank-Wolfe Gap} = \quad < -\nabla f(\vec{x}), \vec{d}_{FW} > \tag{2}$$

2.2 Distance Metric Learning in High Dimensional Input Spaces

The majority of metric learning has been done under the framework of the Mahalanobis distance function (see the surveys [1, 20]). With many local metric learning works shown capable of outperforming their global counterparts [13, 16, 23, 28, 30, 35, 37, 38]. However, these works are not able to scale to very high dimensions. Until recently, very few works have concentrated on metric learning directly in high dimensional input spaces [6, 9, 25, 31]. We restrict our discussion of the related work only to those papers which focus on handling the case of high dimensional data.

The work of [6] learns a Mahalanobis distance metric from only dissimilarities, and shows that this problem is equivalent to learning a Support Vector Machine(SVM) with a quadratic kernel. The other contribution of [6] is that of local invariance to transformations of the data. However, the type of transformation must be known

apriori and only the case of rotational invariance for image data is studied.

In [9], online distance metric learning is considered in the scenario of multimedia content retrieval. They accelerate the optimization process by constraining the metric matrix M such that all off-diagonal matrix entries are equal to zero. This prevents the metric from learning any iterations between features, whether on a local or global level. It is our belief that this assumption is too restrictive, as large and complex data spaces may have many 2nd-order interactions which could be pertinent for performance.

The recent works [25, 36] also focus on the case of metric learning via a Frank-Wolfe style optimization procedure. The work of [36] is a sparse compositional metric learning technique. While similar in name to this work, the approach is quite different. They focus finding a sparse combination of basis functions with which to construct the overall metric matrix M. Each basis function is found as the leading eigenvector of the gradient matrix, the expense of solving for the leading eigenvector may limit its scalability. The solution of [36] consists of a sparse combination of basis atoms (each possibly full), while in our proposed technique the solution is a composition of sparse atoms.

Finally, the work of [25] learns a similarity metric directly in the high dimensional feature space. The method of this paper shares some similarities with this work in that both utilize Frank-Wolfe optimization techniques. Besides the obvious difference in the metric learned(similarity vs. Mahalanobis), there are several contrasts. In [25], a single global similarity is learned, and local information in the dataset is not considered. In our work, we learn a set of local metrics which share a global component. The objective of [25] is to solve a method in a high dimensional input space quickly, in this paper we go beyond [25] through the addition of local adaptability.

3 METHOD

3.1 Overview

Given a subset of the sample space, we wish to learn a Mahalanobis distance metric of the form $d(\vec{x},\vec{y}) = (\vec{x}-\vec{y})^T\ M(\vec{x}-\vec{y})$, where $M = M_G + M_L$ consists of a diagonal metric M_G capturing long range trends and M_L modeling local interactions between variables. We constrain both matrices M_G and M_L to be symmetric positive semi-definite $M_G,\ M_L \in \mathbb{S}^d_+$.

Following many previous works in metric learning [29, 33], we make use of relative distance constraints in the form of triplets (equation 3). In 3, $\vec{x}_i$ and $\vec{y}_i$ share the same label, and $\vec{z}_i$ has a different label.

$$\mathcal{T} = \{\vec{x}_i \text{ should be close to } \vec{y}_i \text{ than to } \vec{z}_i\}_{i=1}^T \tag{3}$$

For each point $\vec{x}_i$, a set of triplet constraints may be formed by finding the nearest neighbors to that point, and labeling points as either similar or dissimilar, depending on the similarity to the label of $\vec{x}_i$. Alternatively, in the case where there are too many samples to form a distance matrix, points can be sampled randomly to form constraints. We choose to take a large-margin approach, and constrain the distance between dissimilar points to be larger than the distance between similar points by some margin m_i. Once the triplet constraints are formed, we guide the learning process of M with constraint violations.

Given a Mahalanobis distance function parameterized by M $d_M(\cdot,\cdot)$, a set of triplet points $(\vec{x}_i,\vec{y}_i,\vec{z}_i)$ and the corresponding margin m_i, the constraint is violated when $d_M(\vec{x}_i,\vec{y}_i)+m_i > d_M(\vec{x}_i,\vec{z}_i)$. This may be represented as a hinge loss function to be minimized (equation 4).

$$[d_M(\vec{x}_i,\vec{y}_i) + m_i - d_M(\vec{x}_i,\vec{z}_i)]_+ \tag{4}$$

The hinge loss (equation 4) has a discontinuous gradient and is ineligible to be solved with first-order techniques. To alleviate this we approximate the hinge loss with a smooth version (equation 5). Note that equation 4 may accommodate different margin values by simply shifting the input to the smooth hinge loss function by some desired amount.

$$h(x) = \begin{cases} x - \frac{1}{2} & x \geq 1 \\ \frac{1}{2}x^2 & 0 < x < 1 \\ 0 & x \leq 0 \end{cases} \tag{5}$$

3.2 Sparse Compositional Local Metrics

Given L local areas, each of which contains a set of samples $\mathcal{X}_i, i \in \{1,...,L\}$, the constraint set and associated margins may be constructed. The set of constraints associated with each sample set $\mathcal{X}_i$ is denoted as C_i. Each element of C_i is a 4-tuple consisting of three samples $\vec{x},\vec{y},\vec{z}$ and the desired margin m. For local sample set $\mathcal{X}_i$, we learn a compositional metric which consists of an exclusive local component parameterized by the local metric matrix $M_i \in \mathcal{S}^d_+$ and a shared component $M_G \in \mathcal{S}^d_+$ which is constrained such that all non-diagonal elements are equal to zero.

Integrating the above information, the objective is formed to optimize over(equation 6). Sparsity is introduced in each metric through the use of L_1-regularization, with the variables $\lambda_1,...,\lambda_L,\lambda_G$ forming the boundaries of the constraint region associated with each metric matrix. Note that in equation 6 the local and global portions of the metric have been separated, as $(\vec{x}-\vec{y})^T(M_G + M_i)(\vec{x}-\vec{y}) \equiv (\vec{x}-\vec{y})^T\ M_G(\vec{x}-\vec{y}) + (\vec{x}-\vec{y})^T\ M_i(\vec{x}-\vec{y})$.

$$\begin{aligned} \text{min.} \quad & \sum_{l=1}^{L}\sum_{i=1}^{C_l}[d_{M_G}(\vec{x},\vec{y}) + d_{M_l}(\vec{x},\vec{y}) + m_i^l \\ & - d_{M_G}(\vec{x},\vec{z}) - d_{M_l}(\vec{x},\vec{z})]_+ \\ \text{s.t.} \quad & M_G \in \mathcal{S}^d_+ \\ & M_G(i,j) = 0, \quad \forall i \neq j \\ & M_l \in \mathcal{S}^d_+, \quad \forall l \in \{1,2,...,L\} \\ & ||\ M_G||_1 < \lambda_G \\ & ||\ M_l||_1 < \lambda_l, \quad \forall l \in \{1,2,...,L\} \end{aligned} \tag{6}$$

3.3 Maintaining Feasibility of Iterates

A pair-wise variant of the Frank-Wolfe procedure was selected to minimize equation 6. Besides removing the need for projection operators, the pair-wise variant only requires the update of two atoms per iterate, in contrast with vanilla Frank-Wolfe, which requires the weights of all atoms to be updated at every iteration. Our objective function (equation 6) is a function of L+1 matrices $(M_1,...,M_l,M_G)$. The function is convex in each matrix with respect to the other matrices. We optimize this using an alternating

pairwise Frank-Wolfe method where the local matrices are updated, followed by the global metric. We leave the variables $\lambda_1, ..., \lambda_L, \lambda_G$ which determine the constraint region bounds as tunable parameters.

Projections are avoided by careful selection of the update directions and step sizes, ensuring that $M_1, ..., M_L$ never leaves the constraint region. We use the following rank-1 update matrices $P^{(i,j)}$, $N^{(i,j)}$ in updating $M_1, ..., M_L$ and restrict the updates of M_G to $P_\lambda^{(i,j)}$ where $i = j$. These update types were first proposed by [14] and subsequently used in [25]. Updating the matrices in this manner allows us to maintain the positive semi-definiteness of $M_1, ..., M_l, M_G$ while still producing a sparse iterate. This is key to the scalability of our algorithm with respect to the size of the feature space.

$$P_\lambda^{(i,j)} = \lambda(\vec{e}_i + \vec{e}_j)(\vec{e}_i + \vec{e}_j)^T = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \lambda & \cdot & \lambda & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \lambda & \cdot & \lambda & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}$$

$$N_\lambda^{(i,j)} = \lambda(\vec{e}_i - \vec{e}_j)(\vec{e}_i - \vec{e}_j)^T = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \lambda & \cdot & -\lambda & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & -\lambda & \cdot & \lambda & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}$$

The design of the update steps handles the constraint, leaving a smooth unconstrained objective function to minimize 7, where $h(\cdot)$ denotes the smooth hinge loss function 5. The gradient for each M_i is calculated on a per-constraint basis, with each constraint violation contributing a direction and a scaling factor determined by the gradient of the smooth hinge loss function. This is shown in equations 8 and 9.

$$\begin{aligned} \text{min.} \quad & \sum_{l=1}^{L} \sum_{i=1}^{C_l} h(d_{M_G}(\vec{x}, \vec{y}) + d_{M_l}(\vec{x}, \vec{y}) + m_i^l \\ & - d_{M_G}(\vec{x}, \vec{z}) - d_{M_l}(\vec{x}, \vec{z})) \end{aligned} \tag{7}$$

$$g(f(x), \frac{\partial f(x)}{\partial x}) = \begin{cases} \frac{\partial f(x)}{\partial x} & x \geq 1 \\ f(x) \cdot \frac{\partial f(x)}{\partial x} & 0 < x < 1 \\ 0 & x \leq 0 \end{cases} \tag{8}$$

$$\frac{\partial d_{M_l}(\vec{x}, \vec{y}, \vec{z})}{\partial\, M_l} = (\vec{x} - \vec{y}) \cdot (\vec{x} - \vec{y})^T - (\vec{x} - \vec{z}) \cdot (\vec{x} - \vec{z})^T \tag{9}$$

3.4 Algorithm

Given the desired number of local metrics c, we begin by clustering the samples into c distinct groups. This may be accomplished via any clustering method, in our experiments we use either a sparse k-means approach or k-means clustering with the cosine distance. Constraints are formed for each point, this is done by taking the nearest neighbors which are "friends" (same label) and "impostors" (different label), then generating relative distance constraints in the form of triplets.

Given the metric matrices $M_i, i \in \{1, .., c\}$ corresponding to each cluster and the global metric matrix M_g, the task is to minimize the objective. The objective is convex with respect to each M_i when the others are held constant, allowing us to conduct the optimization in an alternating fashion. (Note that the local components $M_1, ..., M_c$ are independent when M_g is held constant and may be updated simultaneously if desired.)

The iterates $M_1, ..., M_c, M_g$ must be initialized to lie within the feasible set. This is accomplished by initializing each M_i to any valid atom in $\mathbb{S}_d^+$ and assigning it a weight of one. Note that the matrices $M_1, ..., M_c, M_g$ are never formed explicitly, instead each matrix M_i is represented by a set of atoms $\mathcal{A}_i$ and atom weights $\mathcal{W}_i$ such that $\sum \mathcal{W}_i = 1$. This allows the addition, removal, and weight manipulation of individual atoms in a fast and efficient manner which pairs naturally with Frank-Wolfe style algorithms.

$$\begin{aligned} & \min_S \langle \nabla f(M_G), S \rangle \\ & S(i,j) = 0, \forall i \neq j \quad s.t. \quad S \in \mathbb{S}_d^+ \end{aligned} \tag{10}$$

Once each M_i has been initialized, the algorithm proceeds by alternating through each local metric and the shared diagonal metric, updating each in turn. Each update consists of a calculation of the gradient, then solving the corresponding global or local linear minimization oracle (equation 10 & 11, respectively) for the Frank-Wolfe direction d_{FW}. In the pairwise variant of Frank-Wolfe, the best atom is identified from which to pivot weight away from. The advantage of this approach is that only two atoms must be manipulated for each iteration, this is in contrast to the full Frank-Wolfe method in which the weight of every atom in the active set must be adjusted at every iteration.

$$\begin{aligned} & \min_S \langle \nabla f(M_l), S \rangle \\ & s.t. \quad S \in \mathbb{S}_d^+ \end{aligned} \tag{11}$$

The global linear minimization oracle is simple to optimize, as it consists of p Frobenius inner products between each $P(i,i) \quad \forall i \in \{1, .., d\}$ and the elements of the gradient. The linear minimization oracle for each local metric does not scale as easily, as there are

Table 1: Summary of Data Characteristics

Dataset	# classes	# Features	# Samples
CNAE-9	9	856	1080
BBC-Sports	5	4613	737
BBC-News	5	9635	2225
TDT2-30	30	36771	9394
Madelon	2	500	2600

p^2 atoms to consider. In cases where p is rather large, we use the heuristic proposed in [25]. The general idea of the heuristic is to randomly select a row of the matrix, and consider all columns. Once the column containing the lowest LMO score is found, that column is used while all rows are considered. This technique has order $O(p)$ complexity and we find that it works relatively well in practice.

Once an atom S_{FW} has been selected to increase the weight on, an atom must be selected to remove weight from (the "away" atom). This problem is extremely similar to the linear minimization oracle problem, and consists of calculating the inner Frobenius norm between the gradient and each atom in the active set, taking that which has the maximum value. This is shown as the following, with G representing the gradient at the current iterate.

$$\max_{S} \langle G, S \rangle, \quad \forall\, S \in \mathcal{A}$$

Within a small number of iterations of the algorithm, it is common for the number of active constraints to quickly drop to a fraction of the original number. Inactive constraints have no contribution to the gradient and may be ignored. We leverage this fact in our implementation which results in significant speedups.

We provide a Matlab implementation of our code available for download[1].

4 EXPERIMENT

The proposed metric learning technique, Sparse Compositional Local Metric (SCLM) learning is evaluated under a classification scenario using a K-Nearest Neighbors(K-NN) approach. There are few sparse metric learning techniques capable of handling high dimensional data. The high dimensional similarity learning (HDSL) algorithm developed in [25] is selected as the current state of the art algorithm for comparison. The support vector machine (SVM) with a linear kernel is included in the experimental evaluations to serve as a baseline algorithm. In our experiments the implementation provided by the LIBSVM library [3] is used.

[1]https://github.com/jstamand/

4.1 Datasets for Evaluation

Five datasets were selected to run the experimental evaluations on, the characteristics of these datasets are summarized in Table 1.

4.1.1 CNAE-9. The classification nacional de atividae econamicas(CNAE) dataset[2] consists of 1080 documents of text business descriptions from Brazilian companies. Each of the companies is placed into one of nine categories, based on their national economic activities. The data is preprocessed such that punctuation and the most frequently occurring words are removed. Each document is then represented as a vector with each entry weighted occurring to word frequencies.

4.1.2 BBC-Sports & BBC-News. The BBC-Sports and BBC-News datasets[11][3] originate from BBC News. The BBC-Sports dataset consists of 737 documents, and the BBC-News dataset consists of 2225 documents. We use the provided pre-processed form, where each dataset is given as a term-document frequency matrix. Each of the two datasets have five distinct classes.

4.1.3 TDT2-30. The TDT2-30 dataset[4] is a subset of the original NIST Topic Detection and Tracking corpus. The version we use was prepared by [2], and contains only the 30 most frequently appearing labels. The dataset consists of 9,394 documents, each with 36,771 features.

4.1.4 Madelon. The Madelon[5] dataset is an artificial dataset created for the NIPS 2003 feature selection challenge. It consists of 2500 samples which form 32 separate clusters, each cluster is located on a vertice of a five-dimensional hypercube. The data points are randomly assigned a binary label, making this an especially challenging classification task.

[2]https://archive.ics.uci.edu/ml/datasets/CNAE-9
[3]http://mlg.ucd.ie/datasets/bbc.html
[4]http://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html
[5]https://archive.ics.uci.edu/ml/datasets/Madelon

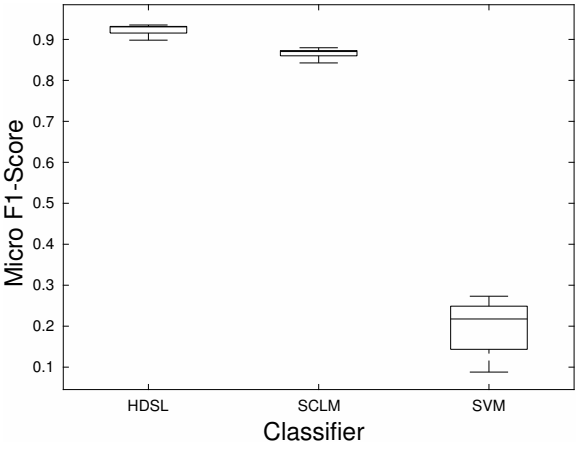

Figure 1: Reported micro-averaged F1 scores of classification experiment on CNAE-9 dataset.

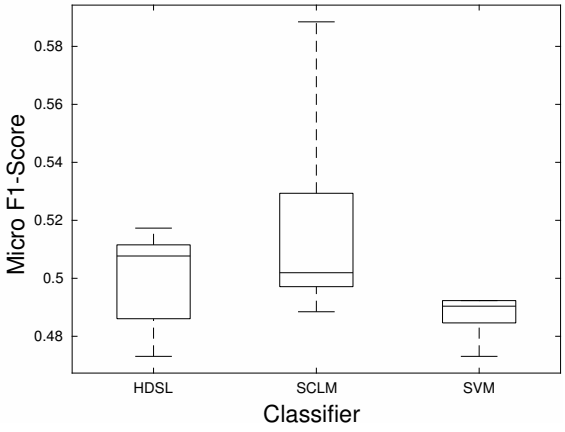

Figure 2: Reported micro-averaged F1 scores of classification experiment on Madelon dataset.

4.2 Experimental Setup

A cross validation procedure is utilized to separate the model selection and model evaluation processes. The dataset is partitioned into training and testing sets using a five fold cross validation procedure. The training set is then further partitioned into additional training and validation sets using an internal round of cross validation. The training and validation sets are used optimize the model using a grid search over the parameter space. After the best performing model is located, it is trained on the combined training and validation sets, and predictions are made and evaluated on the test set. This is repeated for each of the five folds of cross-validated data.

The Support Vector Machine is trained using a linear kernel with the regularization parameter c is tuned to the best value such that $c \in \{1c-6, 1e-5, 1e-5, 1e-4, 1e-3, 1e-2, 1e-2, 1e0, 1e1, 1e2, 1e3, 1e4, 1e5\}$. The high dimensional similarity learning algorithm does not have many tunable parameters, we vary the scaling parameter γ such that $\gamma \in \{1e0, 1e1, 1e2, 1e3, 1e4\}$. The proposed algorithm has two tunable parameters $(\lambda_{global}, \lambda_{local})$ controlling the balance between local adaptivity and global consistency, we allow both parameters to vary on the set of $\{1e0, 1e1, 1e2, 1e3, 1e4\}$.

The proposed and state of the art techniques (SCLM and HDSL, respectively) both utilize a Frank-Wolfe style optimization technique. It is well known that Frank-Wolfe optimization converges at a sublinear rate when the solution lies on the boundary region [21]. We observed that for SCLM and HDSL, the objective value is quickly reduced in the first hundred or so iterations and then makes minimal progress towards the solutions. The maximum allowable number of iterations was set to 500 to ensure both algorithms terminated in a reasonable amount of time.

The SCLM algorithm relies on an external technique to provide the subset of samples which correspond to each local metric. In the experiments we cluster the data using unsupervised clustering methods. The CNAE-9, BBC-Sports, and Madelon datasets were clustered using the Sparse and Robust K-means Clustering (RSKC) algorithm of [19]. We found that RSKC had difficulty operating on the larger datasets, so the k-NN clustering algorithm with the cosine similarity function was utilized to cluster the BBC-News and TDT2-30 datasets.

The HDSL and SCLM algorithm share several similarities, our aim was to train them in the same fashion. Both algorithms are trained using relative distance constraints in the form of triplets, which are formed for each point using 3 friends (same-label points) and 8 impostors (different label points). After training, test points are classified using a k-NN approach based on the learned metric with $k = 3$.

We evaluate each classfier using the micro-averaged F1-score. The vanilla F1-score is the harmonic mean of precision and recall. Precision and recall may be calculated on a per-class basis by taking the predicted and true label values and calculating the number of positive (P) and negative (N) samples, and the number of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions. One found, precision and recall are calculated as follows:

$$\text{precision} = \frac{TP}{TP + FP} \qquad \text{recall} = \frac{TP}{TP + FN}$$

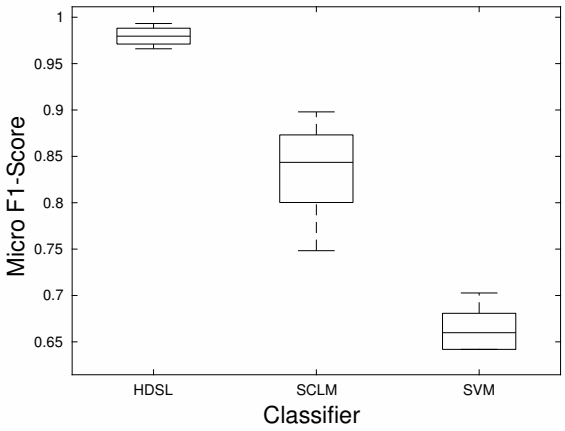

Figure 3: Reported micro-averaged F1 scores of classification experiment on BBC-Sports dataset.

The F1-score is the harmonic mean of precision and recall, and is calculated as shown below:

$$\text{F1-score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{12}$$

Typically, the F1-score is calculated on a per-class basis, simple averaging may bias the score towards the classes which contain fewer samples. This effect is alleviated by calculating the micro-averaged F1-score, which is the F1-score found by calculating precision and recall in a global manner (by summing the TN, TP, FN, FP from each class). Finally, the micro-averaged F1-scores for each cross validation fold are averaged, which is how the reported metric is calculated for each experiment.

4.3 Results

In the results, we refer to our proposed method as Sparse Compositional Local Metrics (SCLM) and the method of [25] as High Dimensional Sparse Learning (HDSL). When compared with the support vector machine baseline algorithm, our proposed technique was able to outperform it on all datasets except for BBC-News. Though support vector machines typically perform well on sparse datasets, we found that it had a particularly hard time on the CNAE-9, Madelon and TDT2-30 datasets. It was observed that in these cases, it had a tendency to make predictions nearly all of one type, which brought the performance measure averages down significantly.

An inspection of the results show that for the CNAE-9, BBC-Sports, and TDT2-30 datasets, the HDSL method has a clear advantage. The Madelon dataset was particularly challenging, with HDSL and SCLM producing around the same level of performance on average. We note that the performance of SCLM on the Madelon dataset was extremely good for some folds, and comparative to the HDSL method on others. Finally, in the BBC-News dataset, the SCLM method appears to have the advantage. The complete performance results of all methods and datasets are shown as Figures 1, 2, 3, 4, and 5.

5 DISCUSSION & FUTURE WORK

5.1 Impact of Top-level Clusterings

The HDSL method outperformed the SCLM method by a good margin on the TST2-30 and BBC-Sports datasets, in these cases the SCLM method demonstrated a larger variance. One source of this phenomenon could be the overlying clustering algorithm which determines the "local groupings" or clustering of the data. Clustering algorithms are typically not convex and are often sensitive to the initialization. Widely varying clusterings of the data between cross validation folds could be one source of the variance in classification performance.

In cases where SCML underperformed HDSL, one explanation could be that the data may not be multi-model. Given a dataset which exhibits a single mode, one can expect that splitting the data and training two (or more) classifiers will not produce the same results as training a single classifier on the complete data, especially in the case of a high-dimensional feature space. A less extreme version of this scenario could be a mismatch between the number of modes and the target number of local groups. In summary, the top-level clustering method has no knowledge of the labels and the metric to be learned, and could be creating local groups which are not useful.

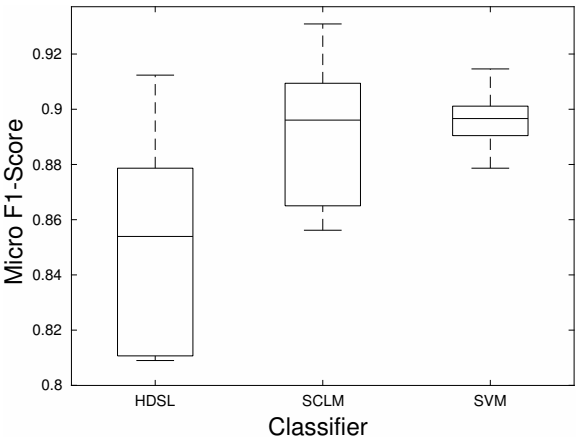

Figure 4: Reported micro-averaged F1 scores of classification experiment on BBC-News dataset.

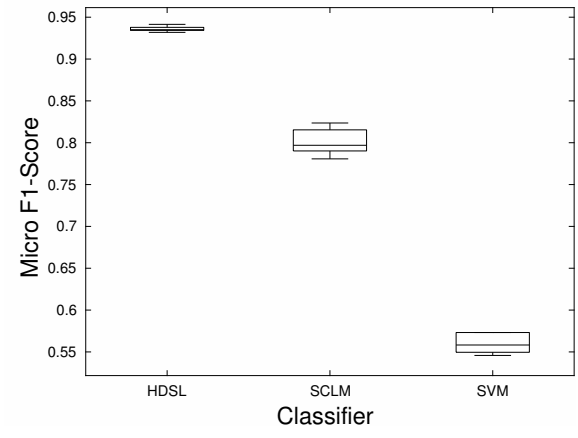

Figure 5: Reported micro-averaged F1 scores of classification experiment on TDT2-30 dataset.

A promising direction for future work is the integration of the top-level clustering procedure with the local metric learning algorithm. The ability to exchange information between the clustering and the local metrics could result in local groups which are meaningful with respect to the metric learning task.

5.2 Algorithm Acceleration

To accelerate the runtime of the HDSL algorithm and our own, we both make use of a stochastic estimate of the gradient. As noted in [25], the accuracy of this approximation is bounded and works relatively well in practice. However, we point out that this approximation also serves to prevent the gradient matrix from "filling-in". This is particularly noteworthy, as it may be infeasible to store a non-sparse gradient matrix. For the diagonal matrix in our technique, we only calculate the diagonal elements of the gradient, upper bounding the space complexity to $O(p)$. However, for the local metric matrices(and that in [25]), as the number of constraints grows, it could be possible to get a full gradient matrix. Placing a limit on the number of constraints used in estimating the gradient could prevent this, depending on the sparsity level of each sample.

The proposed work and others [14, 25] make use of updates using "Jaggi atoms". Another promising direction of research could be the development of a specialized Frank-Wolfe technique which takes advantage of updates with this structure. This type of work would be widely applicable, as there are many positive semi-definite programming problems in machine learning.

6 CONCLUSION

In this paper, we presented an algorithm for sparse compositional local metric learning, where each local metric consists of a local and shared global component. A pairwise alternating Frank-Wolfe style optimization algorithm was used to optimize the objective in an efficient projection-free manner. The proposed method was able to maintain sparsity of the solution through the optimization process, which allowed scaling to datasets with over 30,000 features. An empirical evaluation of the proposed technique was executed against a solid baseline algorithm and the current state of the art in sparse similarity metric learning. The results of the empirical evaluation demonstrate that our method is more effective than the baseline measure, and is comparable to the competing method for some datasets. Finally, a discussion was presented and directions for future work in this area was outlined.

REFERENCES

[1] Aurélien Bellet, Amaury Habrard, and Marc Sebban. 2013. A Survey on Metric Learning for Feature Vectors and Structured Data. *CoRR* abs/1306.6709 (2013). http://arxiv.org/abs/1306.6709

[2] Deng Cai, Xuanhui Wang, and Xiaofei He. 2009. Probabilistic Dyadic Data Analysis with Local and Global Consistency. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML '09)*. ACM, New York, NY, USA, 105–112. https://doi.org/10.1145/1553374.1553388

[3] Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2 (2011), 27:1–27:27. Issue 3. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[4] Jason V. Davis and Inderjit S. Dhillon. 2008. Structured Metric Learning for High Dimensional Problems. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08)*. ACM, New York, NY, USA, 195–203. https://doi.org/10.1145/1401890.1401918

[5] Jason V. Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S. Dhillon. 2007. Information-theoretic Metric Learning. In *Proceedings of the 24th International Conference on Machine Learning (ICML '07)*. ACM, New York, NY, USA, 209–216. https://doi.org/10.1145/1273496.1273523

[6] Ethan Fetaya and Shimon Ullman. 2015. Learning Local Invariant Mahalanobis Distances. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, David Blei and Francis Bach (Eds.). JMLR Workshop and Conference Proceedings, 162–168. http://jmlr.org/proceedings/papers/v37/fetaya15.pdf

[7] Marguerite Frank and Philip Wolfe. 1956. An algorithm for quadratic programming. *Naval Research Logistics Quarterly* 3, 1-2 (1956), 95–110. https://doi.org/10.1002/nav.3800030109

[8] Andrea Frome, Yoram Singer, Fei Sha, and Jitendra Malik. 2007. Learning Globally-Consistent Local Distance Functions for Shape-Based Image Retrieval and Classification. In *IEEE 11th International Conference on Computer Vision, ICCV 2007, Rio de Janeiro, Brazil, October 14-20, 2007*. 1–8. https://doi.org/10.1109/ICCV.2007.4408839

[9] Xingyu Gao, Steven C. H. Hoi, Yongdong Zhang, Ji Wan, and Jintao Li. 2014. SOML: Sparse Online Metric Learning with Application to Image Retrieval. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada*. 1206–1212. http://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/view/8369

[10] Jacob Goldberger, Geoffrey E Hinton, Sam T. Roweis, and Ruslan R Salakhutdinov. 2005. Neighbourhood Components Analysis. In *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou (Eds.). MIT Press, 513–520. http://papers.nips.cc/paper/2566-neighbourhood-components-analysis.pdf

[11] Derek Greene and Pádraig Cunningham. 2006. Practical Solutions to the Problem of Diagonal Dominance in Kernel Document Clustering. In *Proc. 23rd International Conference on Machine learning (ICML'06)*. ACM Press, 377–384.

[12] Yi Hong, Quannan Li, Jiayan Jiang, and Zhuowen Tu. 2011. Learning a mixture of sparse distance metrics for classification and dimensionality reduction. In *2011 International Conference on Computer Vision*. 906–913. https://doi.org/10.1109/ICCV.2011.6126332

[13] Sung Ju Hwang, Kristen Grauman, and Fei Sha. 2011. *Learning a tree of metrics with disjoint visual features*.

[14] Martin Jaggi. 2011. *Sparse Convex Optimization Methods for Machine Learning*. Ph.D. Dissertation. ETH Zurich. https://doi.org/10.3929/ethz-a-007050453

[15] Martin Jaggi. 2013. Revisiting Frank-Wolfe: Projection-Free Sparse Convex Optimization. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, Sanjoy Dasgupta and David Mcallester (Eds.), Vol. 28. JMLR Workshop and Conference Proceedings, 427–435. http://jmlr.csail.mit.edu/proceedings/papers/v28/jaggi13.pdf

[16] D. M. Johnson, C. Xiong, and J. J. Corso. 2014. Semi-Supervised Nonlinear Distance Metric Learning via Forests of Max-Margin Cluster Hierarchies. *ArXiv e-prints* (Feb. 2014). arXiv:stat.ML/1402.5565

[17] Armand Joulin, Kevin Tang, and Li Fei-Fei. 2014. Efficient image and video co-localization with frank-wolfe algorithm. In *European Conference on Computer Vision*. Springer, 253–268.

[18] Dor Kedem, Stephen Tyree, Fei Sha, Gert R. Lanckriet, and Kilian Q Weinberger. 2012. Non-linear Metric Learning. In *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 2573–2581. http://papers.nips.cc/paper/4840-non-linear-metric-learning.pdf

[19] Yumi Kondo, Matias Salibian-Barrera, and Ruben Zamar. 2016. RSKC: An R Package for a Robust and Sparse K-Means Clustering Algorithm. *Journal of Statistical Software, Articles* 72, 5 (2016), 1–26. https://doi.org/10.18637/jss.v072.i05

[20] Brian Kulis. 2013. Metric Learning: A Survey. *Foundations and TrendsÂő in Machine Learning* 5, 4 (2013), 287–364. https://doi.org/10.1561/2200000019

[21] Simon Lacoste-Julien and Martin Jaggi. 2015. On the Global Linear Convergence of Frank-Wolfe Optimization Variants. In *Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS'15)*. MIT Press, Cambridge, MA, USA, 496–504. http://dl.acm.org/citation.cfm?id=2969239.2969295

[22] Simon Lacoste-Julien, Fredrik Lindsten, and Francis Bach. 2015. Sequential kernel herding: Frank-Wolfe optimization for particle filtering. *arXiv preprint arXiv:1501.02056* (2015).

[23] Hao Lei, Kuizhi Mei, Jingmin Xin, Peixiang Dong, and Jianping Fan. 2016. Hierarchical learning of large-margin metrics for large-scale image classification. *Neurocomputing* 208 (2016), 46 – 58. https://doi.org/10.1016/j.neucom.2016.01.100 SI: BridgingSemantic.

[24] Daryl Lim, Gert R. G. Lanckriet, and Brian McFee. 2013. Robust Structural Metric Learning.. In *ICML (1) (JMLR Workshop and Conference Proceedings)*, Vol. 28. JMLR.org, 615–623.

[25] Kuan Liu, Aurélien Bellet, and Fei Sha. 2015. Similarity Learning for High-Dimensional Sparse Data. In *AISTATS*.

[26] Wei Liu, Cun Mu, Rongrong Ji, Shiqian Ma, John R. Smith, and Shih-Fu Chang. 2015. Low-rank Similarity Metric Learning in High Dimensions. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI'15)*. AAAI Press, 2792–2799. http://dl.acm.org/citation.cfm?id=2886521.2886710

[27] Brian Mcfee and Gert Lanckriet. 2010. Metric learning to rank. In *In Proceedings of the 27th annual International Conference on Machine Learning (ICML*.

[28] Shreyas Saxena and Jakob Verbeek. 2015. Coordinated Local Metric Learning. In *ICCV ChaLearn Looking at People workshop (Proceedings IEEE International Conference on Computer Vision Workshops)*. IEEE, Santiago, Chile, 369–377. https://doi.org/10.1109/ICCVW.2015.56

[29] Matthew Schultz and Thorsten Joachims. 2004. Learning a Distance Metric from Relative Comparisons. In *Advances in Neural Information Processing Systems 16*, S. Thrun, L. K. Saul, and P. B. Schölkopf (Eds.). MIT Press, 41–48. http://papers.nips.cc/paper/2366-learning-a-distance-metric-from-relative-comparisons.pdf

[30] Chunhua Shen, Junae Kim, Lei Wang, and Anton Van Den Hengel. 2012. Positive Semidefinite Metric Learning Using Boosting-like Algorithms. *J. Mach. Learn. Res.* 13, 1 (April 2012), 1007–1036. http://dl.acm.org/citation.cfm?id=2503308.2343679

[31] Yuan Shi, Aurélien Bellet, and Fei Sha. 2014. Sparse Compositional Metric Learning. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI'14)*. AAAI Press, 2078–2084. http://dl.acm.org/citation.cfm?id=2892753.2892841

[32] Jun Wang, Adam Woznica, and Alexandros Kalousis. 2012. Parametric Local Metric Learning for Nearest Neighbor Classification. In *Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS'12)*. Curran Associates Inc., USA, 1601–1609. http://dl.acm.org/citation.cfm?id=2999134.2999313

[33] K.Q. Weinberger and L.K. Saul. 2009. Distance metric learning for large margin nearest neighbor classification. *The Journal of Machine Learning Research* 10 (2009), 207–244.

[34] Eric P. Xing, Michael I. Jordan, Stuart J Russell, and Andrew Y. Ng. 2003. Distance Metric Learning with Application to Clustering with Side-Information. In *Advances in Neural Information Processing Systems 15*, S. Becker, S. Thrun, and K. Obermayer (Eds.). MIT Press, 521–528. http://papers.nips.cc/paper/2164-distance-metric-learning-with-application-to-clustering-with-side-information.pdf

[35] Caiming Xiong, David Johnson, Ran Xu, and Jason J. Corso. 2012. Random Forests for Metric Learning with Implicit Pairwise Position Dependence. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '12)*. ACM, New York, NY, USA, 958–966. https://doi.org/10.1145/2339530.2339680

[36] Yiming Ying and Peng Li. 2012. Distance Metric Learning with Eigenvalue Optimization. *J. Mach. Learn. Res.* 13 (Jan. 2012), 1–26. http://dl.acm.org/citation.cfm?id=2188385.2188386

[37] De-Chuan Zhan, Ming Li, Yu-Feng Li, and Zhi-Hua Zhou. 2009. Learning Instance Specific Distances Using Metric Propagation. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML '09)*. ACM, New York, NY, USA, 1225–1232. https://doi.org/10.1145/1553374.1553530

[38] Yu Zheng, Jianping Fan, Ji Zhang, and Xinbo Gao. 2017. Hierarchical learning of multi-task sparse metrics for large-scale image classification. *Pattern Recognition* 67 (2017), 97 – 109. https://doi.org/10.1016/j.patcog.2017.01.029