SERAPH: Semi-supervised Metric Learning Paradigm with Hyper Sparsity

Gang Niu
Computer Science Dept.
Tokyo Tech
gang@sg.cs.titech.ac.jp

Bo Dai
Automation Inst.
Chinese Acad. Sci.
bda@nlpr.ia.ac.cn

Makoto Yamada
Computer Science Dept.
Tokyo Tech
yamada@sg.cs.titech.ac.jp

Masashi Sugiyama
Computer Science Dept.
Tokyo Tech
sugi@cs.titech.ac.jp

Abstract

We consider the problem of learning a distance metric from a limited amount of pairwise information as effectively as possible. The proposed SERAPH (SEmi-supervised metRic leArning Paradigm with Hyper spar-sity) is a direct and substantially more natural approach for semi-supervised metric learning, since the supervised and unsupervised parts are based on a unified information theoretic framework. Unlike other extensions, the unsupervised part of SERAPH can extract further pairwise information from the unlabeled data according to temporary results of the supervised part, and therefore interacts with the supervised part positively. SERAPH involves both the sparsity of posterior distributions over the unobserved weak labels and the sparsity of the induced projection matrices, which we call the hyper sparsity. The resulting optimization is solved by an EM-like scheme, where the M-Step is convex, and the E-Step has analytical solution. Experimental results show that SERAPH compares favorably with existing metric learning algorithms based on weak labels.

1 Introduction

How to learn a good distance metric for the input data is a crucial issue for many learning algorithms. Over past decades, a large number of metric learning algorithms have been developed. They mainly fall into three categories according to what extra information they use: (a) Supervised metric learning using class labels (e.g., Chiaromonte and Cook 2002 Sugiyama 2007); (b) Supervised metric learning using weak labels, that is, labels that indicate whether two samples are similar or dissimilar (e.g., Xing et al. 2002 Gold-berger et al. 2004 Weinberger et al. 2005 Davis et al. 2007); (c) Unsupervised metric learning using nothing more than the unlabeled data themselves (e.g., Roweis and Saul 2000 Belkin and Niyogi 2001). These categories share slightly different goals: supervised metric learning tries to learn a metric such that the data in the same class are close and the data from different classes are far apart, while unsupervised metric learning tries to find the intrinsic geometric structures of the data.

Notice that those supervised paradigms have strict limitations. Algorithms in (a) require all class labels, and algorithms in (b) still need each sample involved in at least one weak label, which seem severe for real-world applications. Based on the belief that preserving the geometric structure of all data in an unsupervised manner can be better than strongly relying on a few labeled data, semi-supervised metric learning algorithms have emerged. To the best of our knowledge, all previous semi-supervised methods that extend (a) and (b) use unsupervised techniques such as principal component analysis (Yang et al. 2006 Sugiyama et al. 2010) and manifold embedding (Hoi et al. 2008 Baghsashah and Shouraki 2009 Liu et al. 2010) for semi-supervised extensions. It can be viewed as propagating labels through the underlying data structure and then using all original and propagated labels implicitly to learn a distance metric.

However, it is rather a circuitous route and sounds like a dilemma: to learn a better metric, we need a better classification result that relies again on a better metric, whereas in those algorithms the former metric does not affect the latter metric. As mentioned before, supervised and unsupervised metric learning have different goals, so simply preserving the geometric structure is indirect, and perhaps it will not always be helpful but sometimes harmful for our purpose. For instance, when the manifold assumption is violated, the unsupervised part will work against the supervised part that learns from the given problem-
specific knowledge, since they are almost independent and could hardly influence each other directly. This means that good semi-supervised metric learning models should have two interactive parts. What is more, the underlying manifold structure is the geodesic distance between data points, not between pairs of data points. Therefore, the paradigms of those algorithms which combine (c) into (b) [Yang et al. 2006; Hoi et al. 2008; Baghshah and Shouraki 2009; Liu et al. 2010] are still not clearly understood.

In this paper, we propose SERAPH, a novel, direct, natural and unified approach for metric learning based on weak labels, which is named after SEMi-supervised metRIC leArning Paradigm with Hyper sparsity. The technique for semi-supervised extension is completely different from manifold learning, and the manifold assumption is replaced by a posterior sparsity assumption [Graça et al. 2009], which claims that the sparser the posterior distributions are, the better they are, since more posterior sparsity over the unobserved weak labels means more powerful discriminative ability. Both the supervised and unsupervised parts are originated from a unified framework [Bellare et al. 2008], likewise SERAPH is more unified than previous models.

More specifically, we optimize a metric by optimizing an associated probabilistic model. Then, the labeled and unlabeled data are strongly connected by sharing the same probabilistic model to predict the weak labels. Instead of label propagation and transductive classification, we consider solely the metric learning itself, which exhibits the directness of the new approach. The supervised part tries to maximize the log-likelihood over the labeled data, and the unsupervised part tries to maximize the posterior sparsity over the unlabeled data and subsequently maximize the discriminative ability of the probabilistic model. Hence, these two parts interact positively from a theoretical viewpoint. The resulting optimization is solved by an EM-like iterative scheme [Graça et al. 2009] that makes the two parts further positively interact algorithmically. In addition, it may be viewed as a regularized information maximization [Gomes et al. 2010] between the data and the weak labels, so SERAPH is substantially more natural and unified indeed.

Last but not least, learning a full matrix directly sometimes causes problems due to the noises in data [Huang et al. 2009]. The (2,1)-norm method [Argyriou et al. 2006] can reduce the representation dimension and effectively remove noises in data. Following Ying et al. (2009), we employ the (2,1)-norm method to enforce the sparsity of the projection matrix. Combining this projection sparsity with the posterior sparsity, we arrive at a property that we call the hyper sparsity.

Our contributions can be summarized in the following two points:

- The regularized maximum likelihood estimation for metric learning is derived from the generalized maximum entropy estimation [Dudik and Schapire 2006].
- A semi-supervised extension of above estimation is proposed under the framework in Bellare et al. (2008). It can extract further pairwise information from the unlabeled data according to temporary results of the supervised part and feed the extracted information back, and therefore interacts positively with the supervised part.

2 SERAPH, the Model

In this section, we first formulate the generalized maximum entropy estimation for supervised metric learning, and then deal with its semi-supervised extension and the issues of two types of sparsity.

2.1 Notations

Suppose we have a dataset $X = \{x_i \mid x_i \in \mathbb{R}^m\}_{i=1}^n$ that contains $n$ samples with $m$ features. Let the sets of similar and dissimilar pairs of points be

$$S = \{(x_i, x_j) \mid x_i \text{ and } x_j \text{ are similar}\},$$
$$D = \{(x_i, x_j) \mid x_i \text{ and } x_j \text{ are dissimilar}\}.$$

With some abuse of terminology, we refer to $S \cup D$ as the labeled data and

$$U = \{(x_i, x_j) \mid (x_i, x_j) \notin S \cup D\}$$

as the unlabeled data. A weak label $y_{i,j} = 1$ is assigned for $(x_i, x_j) \in S$, and $y_{i,j} = -1$ for $(x_i, x_j) \in D$. We abbreviate $\sum_{(x_i, x_j) \in S \cup D}$ and $\sum_{(x_i, x_j) \in U}$ as $\sum_{S \cup D}$ and $\sum_{U}$, and $\sum_{y \in \{1, -1\}}$ as $\sum y$.

Consider learning a Mahalanobis distance for $z_1, z_2 \in \mathbb{R}^m$ of the form

$$d^A(z_1, z_2) = \|z_1 - z_2\|_A = \sqrt{(z_1 - z_2)^\top A (z_1 - z_2)},$$

where $\top$ is the transpose and $A \in \mathbb{R}^{m \times m}$ is a positive semi-definite matrix to be learned. The probability of labeling $(z_1, z_2) \in \mathbb{R}^m \times \mathbb{R}^m$ with $y \in \{1, -1\}$ is denoted by $p^A(y \mid z_1, z_2)$, and when applied to $(x_i, x_j)$ it is abbreviated as $p^A_{i,j}(y)$.

1 In this paper, $A$ is always assumed to be positive semi-definite and we will omit this constraint of the feasible region from optimizations.
2.2 Generalized Maximum Entropy Estimation

To begin with, we will derive a probabilistic model to investigate the conditional probability of weak labels \( y \) given any \( z_1, z_2 \in \mathbb{R}^m \). That is, in order to make \( p^A(y \mid z_1, z_2) \) well-defined for any \((z_1, z_2) \in \mathbb{R}^m \times \mathbb{R}^m\), we will use a parametric form of \( p^A \) when estimating \( p^A_{i,j}(y) \) at any \((x_i, x_j)\) in the training set to see whether \((x_i, x_j)\) should be a similar pair of points or not, and then we will just focus on this parametric form.

The maximum entropy principle (Berger et al., 1996) suggests that we should choose the probability distribution with the maximum entropy out of all distributions \( q(y \mid z_1, z_2) \) that match the moments of the labeled data. Let:

\[
H(p^A_{i,j}) = - \sum_y p^A_{i,j}(y) \ln p^A_{i,j}(y)
\]

be the entropy of \( p^A(y \mid z_1, z_2) \) at \((x_i, x_j)\), and

\[
f(z_1, z_2, y) \in \mathbb{R}^m \times \mathbb{R}^m \times \{+1, -1\} \rightarrow \mathbb{R}
\]

be a feature function, then the resulting constrained optimization problem is

\[
\max_{A, p^A_{i,j}, \xi} \sum_{S \cup D} H(p^A_{i,j}) - \frac{1}{2\gamma} \xi^2
\]

s.t. \[
\begin{align*}
\sum_{S \cup D} E_{p^A_{i,j}}[f(x_i, x_j, y)] \\
- \sum_{S \cup D} f(x_i, x_j, y_{i,j}) \leq \xi,
\end{align*}
\]

where \( \xi \) is a slack variable and \( \gamma > 0 \) is a regularization parameter to control the degree of the relaxation. The relaxed primal problem (1) is essentially consistent with the generalized maximum entropy estimation (Dudík and Schapire, 2006).

The optimal solution of (1) is given as follows (see Fenchel’s Duality Theorem in Dudík and Schapire, 2006 and the discussions therein)

\[
p^A_{i,j}(y) = \exp(\kappa f(x_i, x_j, y)) / Z^A_{i,j},
\]

where \( \kappa \in \mathbb{R} \) is the dual variable and can be obtained by solving the dual problem of (1),

\[
\min_{A, \kappa} \sum_{S \cup D} Z^A_{i,j} - \sum_{S \cup D} \kappa f(x_i, x_j, y_{i,j}) + \frac{\gamma}{2} \eta^2,
\]

where

\[
Z^A_{i,j} = \sum_{y'} \exp(\kappa f(x_i, x_j, y'))
\]

is the partition function.

Define the regularized log-likelihood function over the labeled data (i.e., over the observed weak labels) as

\[
\mathcal{L}_1(A, \kappa) = \sum_{S \cup D} \ln p^A_{i,j}(y_{i,j}) - \frac{\gamma}{2} \kappa^2.
\]

Then we have the following relationship similarly to Berger et al. (1996):

Theorem 1. The regularized maximum likelihood estimation over the labeled data using probabilistic model \( p^A \) and the generalized maximum entropy estimation for metric learning are equivalent.

When considering the feature function \( f(z_1, z_2, y) \) that should take the data moment about the metric information into consideration, we use

\[
f(z_1, z_2, y) = \frac{y}{2}(\|z_1 - z_2\|_A^2 - \eta)
\]

throughout this paper, where \( \eta > 0 \) is a hyperparameter that serves as the threshold to separate \( S \) and \( D \) under the metric \( d^A \). To the best of our knowledge, this is the first usage of this feature function in the maximum entropy estimation framework. For an ideal \( A \), \( f(x_i, x_j, y_{i,j}) \) must be negative to encourage \( d^A(x_i, x_j) > \eta \) when \( y_{i,j} = -1 \) and vice versa. Moreover, for the unlabeled \((x_i, x_j) \in U\), the sign of \( f(x_i, x_j, y) \) indicates the correctness of \( y \), and the absolute value of \( f(x_i, x_j, y) \) indicates the corresponding confidence. Now the probabilistic model (2) becomes

\[
p^A(y \mid z_1, z_2) = \frac{1}{1 + \exp(-\kappa y(\|z_1 - z_2\|_A^2 - \eta))}.
\]

For the optimal solution \( A^* \) and \( \kappa^* \), we hope for \( p^A_{i,j}(y_{i,j}) > p^A_{i,j}(-y_{i,j}) \) and \( y_{i,j}(\|x_i - x_j\|_A^2 - \eta) < 0 \), so there must be \( \kappa^* < 0 \). Although we use \( f \) as our feature function, other choices are also possible and we can even use multiple feature functions provided they make sense in the metric learning setting.

2.3 Posterior Sparsity

In this subsection, we will extend the paradigm derived above to semi-supervised metric learning so that we can extract further pairwise information from the unlabeled data explicitly and thus utilize the training data much more effectively.

The traditionally used manifold assumption is replaced by a posterior sparsity assumption (Graça et al., 2009). It advocates that the sparser the posterior distribution are, the better they are, since more posterior sparsity means more discriminative ability. Here the posterior sparsity in the sense of semi-supervised metric learning based on weak labels is that for an unlabeled data pairs is different from the sparsity of probabilities of class labels on data points, otherwise it is just the low-density separation principle.
beled pair \((x_i, x_j) \in \mathcal{U}\), \(p_{ij}^A(y)\) is close to either 0 or 1 and away from 1/2. We say that a probability \(\hat{p}_{ij}^A\) is sparser than \(p_{ij}^A\), if \(\hat{p}_{ij}^A(y)\hat{p}_{ij}^A(\bar{y}) < p_{ij}^A(y)p_{ij}^A(\bar{y})\). See Figure 1 as an example. We refer to the horizontal coordinate as the first feature and the vertical coordinate as the second feature. As mentioned before, supervised metric learning tries to learn a metric such that 1), the points in the same class (with the same color) are close and 2), the points in different classes (with different colors) are far apart. This will lead to a metric that ignores the first feature and only uses the second feature as shown in (c) and (f) in Figure 1. However, the abandoned feature is very important in distinguishing the two classes. (b) and (e) in Figure 1 illustrate the posterior sparsity could lead to a better metric avoiding such loss via exploiting unlabeled data effectively. Thus, in addition to achieving the two goals of supervised metric learning, we prefer to take into account the posterior sparsity or equivalently the discriminative ability of \(p^A\) over \(\mathcal{U}\).

The pivotal toolbox is a framework called alternating projections for learning with expectation constraints \cite{Bellare2008}. It enables us to utilize the unlabeled data in forms of auxiliary constraints on posterior expectations. Following the idea of \cite{Bellare2008}, we introduce an auxiliary feature function \(g(z_1, z_2, y)\), and subtract a regularization term \(U_g(\cdot)\) about \(g(x_i, x_j, y)\) over \(\mathcal{U}\) from the objective function when maximizing the log-likelihood \(L_1\)

\[
\max_{A, \kappa} L_1(A, \kappa) - \mu U_g \left( \sum_{\mathcal{U}} \mathbb{E}_{p_{ij}^A} (g(\mathbf{x}_i, \mathbf{x}_j, y)) \right), \quad (5)
\]

where \(\mu \geq 0\) is a regularization parameter to control the influence of the unlabeled data, and \(U_g(\cdot) \in \mathbb{R} \mapsto \mathbb{R}\) is a convex function and required to be a divergence measure from a user-specified target value.

The optimization (5) could encourage the posterior sparsity over the unlabeled data \(\mathcal{U}\) (i.e., over the unobserved weak labels) by a realization of \(U_g(\cdot)\) and \(g(z_1, z_2, y)\) that \(U_g(\cdot)\) over \(\mathcal{U}\) from the objective function when maximizing the log-likelihood \(L_1\)

First of all, \(U_g(\cdot) = |x|\) is certainly a divergence by taking the origin as the user-specified target value. Then, it is sufficient to formulate (5) with the realization of \(U_g(\cdot)\) and \(g(z_1, z_2, y)\) into a soft posterior regularization problem \cite{Graaca2009} and show the effect of auxiliary constraints on posterior expectations. Since \(g(z_1, z_2, y)\) is nonnegative no matter what...
Figure 2: Sparse projection matrices vs. non-sparse projection matrices. Twenty-eight weak labels are constructed according to eight class labels. The top three figures are the original dataset and the data projected by induced projection matrices of metrics learned with/without the projection sparsity respectively. The bottom three figures are the one-nearest-neighbor classification results based on Euclidean distance and two learned distances.

its variables are, $\mathbf{6}$ becomes
\[
\max_{A,\kappa,b_{i,j}} L_1(A,\kappa) = \mu \sum_{i,j} b_{i,j} \\
\text{s.t. } \mathbb{E}_{p_{i,j}}[\ln p_{i,j}^A(y)] \leq b_{i,j}, b_{i,j} \geq 0, \forall (x_i, x_j) \in \mathcal{L},
\]
where $\sum_{i,j} b_{i,j}$ implements an $\ell_1$-regularization on the auxiliary constraints $\mathbb{E}_{p_{i,j}}[\ln p_{i,j}^A(y)] \leq b_{i,j}$. Let $G(q) = -q \ln q - (1 - q) \ln(1 - q)$. Easy to see that keeping $G(p_{i,j}^A(y)) = \mathbb{E}_{p_{i,j}}[\ln p_{i,j}^A(y)]$ as small as possible means that the closer $p_{i,j}^A(y)$ locates to 0 or 1, the better it is. Therefore, the realization of $U_g(\cdot)$ and $g(z_1, z_2, y)$ successfully leads to the desired posterior sparsity by pushing $p_{i,j}^A(y)$ towards 0 or 1.

Substituting the realization of $U_g(x)$ and $g(z_1, z_2, y)$ into $\mathbf{5}$, we get the new optimization problem
\[
\max_{A,\kappa} L_2(A,\kappa) = \sum_{S \in \mathcal{D}} \ln p_{i,j}^A(y_{i,j}) - \frac{\gamma}{2} \kappa^2 \\
+ \mu \sum_{i,j} \sum_y p_{i,j}^A(y) \ln p_{i,j}^A(y).
\]

2.4 Projection Sparsity

Besides the posterior sparsity over $\mathcal{L}$, we also hope for the projection sparsity in our paradigm, that is, the capability of projecting the data into a low-dimensional subspace via the projection induced by $A$. The projection sparsity is helpful in seeking 'intrinsic information' and denoising in high-dimensional datasets. Moreover, it also plays an important role in achieving better generalization.

Denote the (2,1)-norm of a matrix $M$ as
\[
||M||_{(2,1)} = \left( \sum_{i=1}^{m} \left( \sum_{j=1}^{m} M_{i,j}^2 \right)^{1/2} \right)^{1/2},
\]
which is the 1-norm of the vector that contains the 2-norm of the row vectors of $M$. It is shown to be very successful in multi-task feature learning and metric learning problems \cite{Arghyropoulos2006,Ying2009}. See Figure 2 as an illustration of its effect.

Following \cite{Ying2009}, let $P \in \mathbb{R}^{m \times m}$ be a transformation and $W = P^T P$. By introducing an auxiliary transformation $V \in \mathcal{O}$ where $\mathcal{O}$ is the set of $m \times m$ orthonormal matrices, our objective is transformed to
\[
\max_{A,P,V,\kappa} L_2(A,\kappa) - \lambda ||W||_{(2,1)} \\
\text{s.t. } V \in \mathcal{O}, A = (PV)^T (PV), W = P^T P,
\]
where $\lambda \geq 0$ is a regularization parameter. Define
\[
\max_{A, \kappa} \mathcal{L}(A, \kappa) = \sum_{S \cup D} \ln p^A_{i,j}(y_{i,j}) - \frac{\gamma}{2} \kappa^2
+ \mu \sum_{t \in U} \sum_y p^A_{i,j}(y) \ln p^A_{i,j}(y) - \lambda \text{tr}(A),
\]
where $\text{tr}(A)$ is the trace of $A$. Then by Lemma 1 of Ying et al. (2009), the optimal $A$ and $\kappa$ of (6) can be obtained by solving (7). The optimization problem (7) is the final model of SERAPH, and we say that it is equipped with the hyper sparsity when both regularization parameters $\mu$ and $\lambda$ are nonzero.

3 SERAPH, the Algorithm

From now on we will simplify (7) and propose a practical algorithm. Our first goal is to eliminate $\kappa$ from (7). That is why we use a single feature function $f(z_1, z_2, y)$ in $p^A(y \mid z_1, z_2)$.

**Theorem 2.** The optimization problem (7) is equivalent to
\[
\max_A \mathcal{L}(A) = \sum_{S \cup D} \ln \hat{p}^A_{i,j}(y_{i,j})
+ \mu \sum_{t \in U} \sum_y \hat{p}^A_{i,j}(y) \ln \hat{p}^A_{i,j}(y) - \hat{\lambda} \text{tr}(A),
\]
where the simplified probabilistic model is\footnote{The new functions and parameters are denoted by $\hat{\cdot}$ within this theorem for the sake of clarity.}
\[
\hat{p}^A(y \mid z_1, z_2) = \frac{1}{1 + \exp(y(||z_1 - z_2||^2 - \bar{q})}.
\]

After applying Theorem 3, $\gamma$ has been dropped, $\eta$ and $\lambda$ have been modified, but most importantly $\mu$ remains the same, which means that the tradeoff between the supervised and unsupervised parts has not been changed.

The optimization problem (8) could be solved by subgradient projection methods (Polyak 1967). Nonetheless, we would like to pose it as an EM-like iterative scheme to achieve the goal that let the supervised and unsupervised parts positively interact algorithmically.

In the beginning, we initialize a nonparametric probability $q(y \mid x_i, x_j)$. Subsequently, at the $t$-th M-Step, we have for each pair $(x_i, x_j) \in U$ that
\[
\min_{q(y \mid x_i, x_j)} \text{KL}(q \parallel p^A_{i,j}) + \mu \mathbb{E}_{q}[\ln p^A_{i,j}(y)],
\]
where KL is the Kullback-Leibler divergence. Its dual problem is (see the supplementary material of Graça et al. 2009)
\[
\min_{\xi_{i,j}} \ln \left(\sum_y p^A_{i,j}(y) \exp(\xi_{i,j} \ln p^A_{i,j}(y))\right) \\
\text{s.t. } 0 \leq \xi_{i,j} \leq \mu,
\]
and the analytical solution (that is why we use a single auxiliary feature function $g(z_1, z_2, y)$ in $U(y)$) is
\[
q(y \mid x_i, x_j) = \frac{p^A_{i,j}(y) \exp(\mu \ln p^A_{i,j}(y))}{\sum_y p^A_{i,j}(y') \exp(\mu \ln p^A_{i,j}(y'))},
\]
where all $p^A_{i,j}$ here are parameterized by $A^{(t)}$.

At the $t$-th M-Step, we find new $A^{(t)}$ by
\[
\max_A \mathcal{F}(A) = -\lambda \text{tr}(A) + \sum_{S \cup D} \ln p^A_{i,j}(y_{i,j})
+ \mu \sum_{t \in U} \sum_y q(y \mid x_i, x_j) \ln p^A_{i,j}(y),
\]
where $q(y \mid x_i, x_j)$ is the probability generated in the last E-Step. It can be solved by subgradient projection methods without worries about local maxima using the following calculation of $\nabla \mathcal{F}$:
\[
\nabla \mathcal{F}(A) =
- \lambda I_m - \sum_{S \cup D} y_{i,j} (1 - p^A_{i,j}(y_{i,j}; A)) X_{i,j}
- \mu \sum_{t \in U} \sum_y q(y \mid x_i, x_j) (1 - p^A_{i,j}(y; A)) X_{i,j},
\]
where $X_{i,j} = (x_i - x_j)(x_i - x_j)^\top$, because

**Theorem 3.** The optimization problem (9) is convex.

A remarkable property is that $\mathcal{F}(A)$ is smooth enough for applying gradient ascent updates, since its gradient matrix $\nabla \mathcal{F}$ is bounded uniformly, regardless of the scale of $A$, i.e., the magnitude of tr($A$).

**Theorem 4.** $\mathcal{F}(A)$ is Lipschitz continuous with the best Lipschitz constant $\text{Lip}_{\parallel \cdot \parallel_F}(\mathcal{F})$ with respect to the Frobenius norm $\parallel \cdot \parallel_F$ satisfying
\[
\text{Lip}_{\parallel \cdot \parallel_F}(\mathcal{F}) \leq (\#(S \cup D) + \mu \#U)(\text{diam}(\mathcal{X}))^2 + \lambda m,
\]
where $\text{diam}(\mathcal{X}) = \max_{x_i, x_j \in \mathcal{X}} \|x_i - x_j\|_2$ is the diameter of $\mathcal{X}$, and $\#$ measures the cardinality of a set.

4 Related Work

Xing et al. (2002) proposed the Global Distance Metric Learning (GDM) that initiated the metric learning based on weak labels. From different perspectives, many well-known metric learning methods, such as Neighborhood Component Analysis (NCA) (Goldberger et al. 2004), Large Margin Nearest Neighbor (LMNN) (Weinberger et al. 2005), Information-Theoretic Metric Learning (ITML) (Davis et al. 2007) and so on, have emerged. A probabilistic version of GDM was designed intuitively as a baseline in Yang et al. (2009) without strong justification. Compared to our supervised part, it can be viewed as a special case that uses special feature function, abandons the low-rank prior, and sets the variable $\kappa = 1$ which would
cause fatal mistake in their feature function setting that $p_{\eta,i,j}^{\mu}(y_{i,j}) < p_{\eta,i,j}^{\mu}(-y_{i,j})$ for the optimal $A$. Thus, the proposed SERAPH is substantially more general.

Due to limitations of supervised learning paradigms when few labeled data are available, semi-supervised metric learning algorithms that incorporate off-the-shelf unsupervised techniques to existing supervised algorithms were proposed subsequently. Hoi et al. (2008) combined manifold regularization to the minimax principle. Baghshah and Shouraki (2009) incorporated locally linear embedding to manifold regularized linear discriminant analysis (MLDA). Liu et al. (2010) brought the sparse preference to manifold semi-supervised metric learning. However, the method cannot generate an appropriate low-rank matrix based on the element-wise sparsity rather than the row-wise sparsity. Local Distance Metric Learning (LDM) (Yang et al. 2006) completed the NCA to involve dissimilarity constraints. Unlike aforementioned methods considering the unlabeled data within their models, LDM embedded the supervised information into the optimizing process in principal components space. Nevertheless, none of these semi-supervised extensions have considered the discriminative ability of model directly, and thus none of them have aimed at increasing the discriminative ability of the model similarly to SERAPH.

5 Experiments

We first justified the hyper sparsity empirically on two toy datasets. The experimental results on these two datasets were already shown in Figure 1 and 2. The results of (c) and (f) in those two figures were obtained by SERAPH with hyperparameters $\eta = 1, \mu = 10 \cdot \frac{\#(S \cup D)}{\#U}, \lambda = 0$, that is, the maximum entropy estimation with the posterior sparsity only. The result of (b) and (e) in Figure 1 was obtained by SERAPH with hyperparameters $\eta = 1, \mu = 0, \lambda = 250$, that is, the maximum entropy estimation with the projection sparsity only. We obtained a visually comprehensive result about the validity of each sparse style.

By evaluating the misclassification rate of the one-nearest-neighbor classifier based on the learned metric, we compared the proposed SERAPH with the Euclidean distance (EUC), as well as three supervised metric learning methods, including GDM or Probabilistic GDM (PGDM), NCA and LMNN, and two representative semi-supervised metric learning algorithms, LDM and MLDA. Four UCI datasets (Iris, Wine, Ionosphere and Balance) and the USPS benchmark were used in the experiments. Table 1 summaries the details of the statistics of these datasets and the experimental setup. The gray scale images of USPS were down sampled to $8 \times 8$ pixel resolution resulting in 64 dimensions. We selected instances randomly and used the category information to generate the weak labels. To illustrate the advantage of the semi-supervised learning paradigm when lacking supervision, we provided little side information and abundant unlabeled data, meanwhile the remaining data were used for evaluation.

All the parameters were set by preliminary experiments. We used $\eta = 1, \mu = \frac{\#(S \cup D)}{\#U}, \lambda = 0$ on all UCI and $\eta = 1, \mu = \frac{\#(S \cup D)}{\#U}, \lambda = 1$ on USPS. Because of the stochastic property of NCA, we run the algorithm three times each turn and chose the best one. LDM used all the principal components on UCI while 80% principal components on USPS to construct the solution space. The number of nearest neighbors in LLE for MLDA is tuned from the range 5 to 12 and the trade-off parameter $\alpha$ is self-adapted.

We repeated the experiments 15 or 30 times. The results are shown in Table 2 and 3. For each data set, the best method and comparable ones based on the t-test at the significance level 5% are indicated by ‘*’. The proposed SERAPH demonstrated excellent performance in most cases. From the results, we can observe that when just providing the traditional metric learning algorithms little weak labels, some methods might become pale and the learned metrics were even worse than EUC due to over-fitting, e.g., the NCA on Iris and LMNN on Wine, while our semi-supervised method, SERAPH, exhibited advanced generalization. However, some extant semi-supervised extensions may also hurt the performance when the assumptions are not suitable. The phenomenon that LDM and MLDA degraded on USPS illustrated the assertion. In contrast, the proposed SERAPH always outperformed or was comparable to the state-of-the-art methods, indicating that our assumption is reasonable and general, and the SERAPH is more direct and effective than the existing off-the-shelf unsupervised combinations with supervised metric learning methods.

6 Conclusions

By introducing the posterior sparsity assumption to metric learning, we proposed a semi-supervised metric learning method, SERAPH, which embraces the supervised and unsupervised information from a more natural and unified information-theoretic perspective. Compared to the extant semi-supervised metric learning algorithms, the SERAPH focuses on promoting the discriminative ability of model directly rather than

---

2. [http://www.cs.nyu.edu/~roweis/data.html](http://www.cs.nyu.edu/~roweis/data.html)
other indirect ways. Moreover, the projection sparsity preference guarantees the SERAPH could find a metric in low-rank space. Empirical results on various data sets proved that SERAPH could utilize the data effectively and avoid over-fitting when lacking supervision.

References
J. Goldberger, S. Roweis, G. Hinton, and R. Salakhut-

Table 1: Descriptions of the experimental setup

<table>
<thead>
<tr>
<th></th>
<th>#class</th>
<th>#feature</th>
<th>#labeled data</th>
<th>#training data</th>
<th>#test data</th>
<th>#repeat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>3</td>
<td>4</td>
<td>10</td>
<td>100</td>
<td>38</td>
<td>15</td>
</tr>
<tr>
<td>Wine</td>
<td>3</td>
<td>13</td>
<td>10</td>
<td>100</td>
<td>78</td>
<td>15</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>2</td>
<td>34</td>
<td>20</td>
<td>100</td>
<td>251</td>
<td>15</td>
</tr>
<tr>
<td>Balance</td>
<td>3</td>
<td>4</td>
<td>10</td>
<td>100</td>
<td>160</td>
<td>15</td>
</tr>
<tr>
<td>USPS Group 1</td>
<td>5</td>
<td>64</td>
<td>10</td>
<td>100 or 200</td>
<td>5000</td>
<td>15</td>
</tr>
<tr>
<td>Group 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30</td>
</tr>
<tr>
<td>Group 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30</td>
</tr>
</tbody>
</table>

Table 2: Means and standard deviations of the misclassification rate (in %) on four UCI benchmarks. For each dataset, the best method and comparable ones based on the t-test at the significance level 5% are marked by ‘*’.

<table>
<thead>
<tr>
<th></th>
<th>EUC</th>
<th>GDM</th>
<th>NCA</th>
<th>LMNN</th>
<th>LDM</th>
<th>MLDA</th>
<th>SERAPH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>10.2±5.7</td>
<td>8.1±5.4</td>
<td>13.9±6.8</td>
<td>8.5±5.2</td>
<td>*7.0±6.1</td>
<td>7.6±4.6</td>
<td>*4.0±3.3</td>
</tr>
<tr>
<td>Wine</td>
<td>12.3±4.7</td>
<td>11.5±6.6</td>
<td>11.7±4.6</td>
<td>13.0±8.0</td>
<td>19.9±9.8</td>
<td>*9.2±6.6</td>
<td>*7.0±4.5</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>27.2±6.2</td>
<td>*21.2±3.2</td>
<td>26.5±5.8</td>
<td>*23.0±3.7</td>
<td>25.8±4.6</td>
<td>*20.2±4.0</td>
<td>*20.5±3.5</td>
</tr>
<tr>
<td>Balance</td>
<td>28.6±5.9</td>
<td>*22.1±6.9</td>
<td>28.4±6.2</td>
<td>*23.8±12.4</td>
<td>*19.9±8.9</td>
<td>29.2±6.1</td>
<td>*20.3±6.6</td>
</tr>
</tbody>
</table>

Table 3: Means and standard deviations of the misclassification rate (in %) on the benchmark USPS. For each dataset, the best method and comparable ones based on the t-test at the significance level 5% are marked by ‘*’.

<table>
<thead>
<tr>
<th></th>
<th>EUC</th>
<th>PGDM</th>
<th>NCA</th>
<th>LMNN</th>
<th>SERAPH100</th>
<th>SERAPH200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group 1</td>
<td>33.8±4.5</td>
<td>32.1±4.2</td>
<td>34.0±4.9</td>
<td>32.2±4.1</td>
<td>*29.1±3.9</td>
<td>*28.5±3.9</td>
</tr>
<tr>
<td>Group 2</td>
<td>29.1±4.8</td>
<td>28.0±4.2</td>
<td>29.1±4.8</td>
<td>28.3±4.2</td>
<td>*26.1±4.2</td>
<td>*25.6±4.3</td>
</tr>
<tr>
<td>Group 3</td>
<td>26.3±4.0</td>
<td>*25.3±3.0</td>
<td>26.3±4.0</td>
<td>26.5±3.9</td>
<td>*24.4±2.9</td>
<td>*24.2±2.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>LDM100</th>
<th>LDM200</th>
<th>MLDA100</th>
<th>MLDA200</th>
<th>SERAPH100</th>
<th>SERAPH200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group 1</td>
<td>34.9±8.8</td>
<td>35.9±10.6</td>
<td>42.2±6.3</td>
<td>34.7±5.0</td>
<td>*29.1±3.9</td>
<td>*28.5±3.9</td>
</tr>
<tr>
<td>Group 2</td>
<td>29.7±6.7</td>
<td>30.1±8.8</td>
<td>37.5±6.0</td>
<td>30.5±5.4</td>
<td>*26.1±4.2</td>
<td>*25.6±4.3</td>
</tr>
<tr>
<td>Group 3</td>
<td>26.6±5.7</td>
<td>27.5±6.7</td>
<td>33.9±5.8</td>
<td>27.3±4.2</td>
<td>*24.4±2.9</td>
<td>*24.2±2.8</td>
</tr>
</tbody>
</table>


7 APPENDIX—SUPPLEMENTARY MATERIAL

7.1 Proof of Theorem 1

The optimization of regularized maximum likelihood estimation is

$$\max_{A,\kappa} \mathcal{L}_1(A, \kappa).$$

Putting the probabilistic model (2) into above problem we will get (3) exactly, which is the dual problem of generalized maximum entropy estimation for metric learning (1).

7.2 Proof of Theorem 2

Suppose that $(A^*, \kappa^*)$ is an optimal solution of (4). As mentioned before there must be $\kappa^* < 0$. Moreover, $\kappa^* \neq -\infty$ and $\text{tr}(A^*) \neq +\infty$ since they are penalized in (7).

Let $\hat{A} = -\kappa^* A^*$, $\hat{\eta} = -\kappa^* \eta$ and $\hat{\lambda} = -\lambda / \kappa^*$. Then $\hat{\eta}$ and $\hat{\lambda}$ are well-defined hyperparameters as finite positive real numbers, and $\hat{A}$ is a feasible solution of (8) as a finite trace positive semi-definite matrix.

Differentiate $p^A$ and $\hat{p}^A$ with respect to $A$,

$$\frac{\partial p^A}{\partial A} = \kappa y p^A (1 - p^A)(z_1 - z_2)(z_1 - z_2)^\top,$$

$$\frac{\partial \hat{p}^A}{\partial A} = -y \hat{p}^A (1 - \hat{p}^A)(z_1 - z_2)(z_1 - z_2)^\top. \quad (11)$$

Note that from

$$\hat{p}^A(y \mid z_1, z_2; \hat{A}, \hat{\eta}) = p^A(y \mid z_1, z_2; A^*, \kappa^*, \eta) \quad (12)$$

we have

$$\frac{\partial \hat{L}}{\partial \hat{p}_{i,j}^A} \bigg|_{A = \hat{A}} = \frac{\partial \mathcal{L}}{\partial p_{i,j}^A} \bigg|_{A = A^*, \kappa = \kappa^*}.$$

Thus from

$$\frac{\partial \hat{F}}{\partial A} \bigg|_{A = \hat{A}} = -\frac{1}{\kappa^*} \frac{\partial p^A}{\partial A} \bigg|_{A = A^*, \kappa = \kappa^*},$$

$$d \text{tr}(A) / dA = I_m$$ where $I_m$ is the identity matrix, and a Karush-Kuhn-Tucker stationarity condition of optimization (7)

$$\frac{\partial \mathcal{L}}{\partial A}(A^*, \kappa^*) = 0_{m \times m}$$

where $0_{m \times m}$ is the zero matrix in $\mathbb{R}^{m \times m}$, we get

$$\frac{\partial \hat{L}}{\partial A}(\hat{A}) = -\frac{1}{\kappa^*} \frac{\partial \mathcal{L}}{\partial A}(A^*, \kappa^*) = 0_{m \times m}.$$

This implies that $\hat{A}$ is actually a local maximum of $\hat{L}(\hat{A})$.

Remember (12) that $\hat{p}^A(y \mid z_1, z_2; \hat{A}, \hat{\eta})$ is identical to $p^A(y \mid z_1, z_2; A^*, \kappa^*, \eta)$. The theorem follows.

7.3 Proof of Theorem 3

To see the convexity of (9), note that the feasible region is a convex set. Rewrite $\ln p^A(y \mid z_1, z_2)$ as

$$\ln p^A(y \mid z_1, z_2) = f(z_1, z_2, y) - \sum_{y'} \exp \left( f(z_1, z_2, y') \right),$$

where the first term

$$f(z_1, z_2, y) = \frac{y}{2} ((z_1 - z_2)^\top A(z_1 - z_2) - \eta)$$

is linear w.r.t. $A$. The negative of the second term

$$\sum_{y'} \exp(f(z_1, z_2, y'))$$

is a convex log-sum-exp function (see Boyd and Vandenberghe 2004, page 74) of $f(z_1, z_2, y')$. Hence we know $\ln p^A(y \mid z_1, z_2)$ is concave w.r.t. $A$. At last, $\mathcal{F}(A)$ is linear w.r.t. $\ln p^A(y \mid z_1, z_2)$ and tr($A$) is also linear w.r.t. $A$. Consequently, $\mathcal{F}(A)$ is a concave function. Combining the convexity of the feasible region and the concavity of the objective function completes the proof.

7.4 Proof of Theorem 4

Obviously $\mathcal{F}(A)$ is differentiable if we allow that the derivative goes to infinity. The conjugate norm of $\| \cdot \|_F$ is still the Frobenius norm, that is, $\|B\|_F^* = \max_{\|A\|_F \leq 1} \langle A, B \rangle = \|B\|_F$. Then the best Lipschitz constant of $\mathcal{F}$ w.r.t. $\| \cdot \|_F$ can be expressed as

$$\text{Lip}_{\| \cdot \|_F} (\mathcal{F}) = \sup_{A \geq 0} \| \nabla \mathcal{F} \|_F,$$

so it is sufficient to bound $\| (\partial \mathcal{F} / \partial p^A_{i,j}) \cdot (\partial p^A_{i,j} / \partial A) \|_F$ from above.

Recall that the partial derivative of $p^A$ w.r.t. $A$ was given by (11). On the other hand,

$$\frac{\partial \mathcal{F}}{\partial p^A_{i,j}} = \begin{cases} \frac{1}{p^A_{i,j}(y_i)}, & \text{if } (x_i, x_j) \in \mathcal{S} \cup \mathcal{D} \\ \mu q(y \mid x_i, x_j), & \text{if } (x_i, x_j) \in \mathcal{U}, y \in \{1, -1\}. \end{cases}$$

Hence when $(x_i, x_j) \in \mathcal{S} \cup \mathcal{D}$,

$$\left\| \frac{\partial \mathcal{F}}{\partial p^A_{i,j}} \cdot \frac{\partial p^A_{i,j}}{\partial A} \right\|_F = \left\| -y_{i,j} (1 - p^A_{i,j}(y_{i,j})) (x_i - x_j)(x_i - x_j)^\top \right\|_F \leq \left\| (x_i - x_j)(x_i - x_j)^\top \right\|_F = \|x_i - x_j\|^2 \leq (\text{diam}(\mathcal{X}))^2,$$
where we use a fact \( \|zz^\top\|_F = \sqrt{\sum_{i,j=1}^m (z_i z_j)^2} = \sqrt{(\sum_{i=1}^m z_i^2)(\sum_{j=1}^m z_j^2)} = \|z\|_2^2 \). When \( (x_i, x_j) \in \mathcal{U} \), for fixed \( y \) we have
\[
\left\| \frac{\partial F}{\partial p_{i,j}} \cdot \frac{\partial p_{i,j}}{\partial A} \right\|_F \leq \mu q(y | x_i, x_j) (\text{diam}(\mathcal{X}))^2.
\]

As a result, there exists a finite Lip\( \|\cdot\|_F (\mathcal{F}) \). The inequality (10) is obtained by applying the triangle inequality of the Frobenius norm.