Scalable Metric Learning for Co-embedding

Farzaneh Mirzazadeh[1], Martha White[2], András György[1], and Dale Schuurmans[1]

[1] Department of Computing Science, University of Alberta
[2] Department of Computer Science, Indiana University

Abstract. We present a general formulation of metric learning for co-embedding, where the goal is to relate objects from different sets. The framework allows metric learning to be applied to a wide range of problems—including link prediction, relation learning, multi-label tagging and ranking—while allowing training to be reformulated as convex optimization. For training we provide a fast iterative algorithm that improves the scalability of existing metric learning approaches. Empirically, we demonstrate that the proposed method converges to a global optimum efficiently, and achieves competitive results in a variety of co-embedding problems such as multi-label classification and multi-relational prediction.

1 Introduction

The goal of *metric learning* is to learn a distance function that is tuned to a target task. For example, a useful distance between person images would be significantly different when the task is pose estimation versus identity verification. Since many machine learning algorithms rely on distances, metric learning provides an important alternative to hand-crafting a distance function for specific problems. For a single modality, metric learning has been well explored (Xing et al., 2002; Globerson & Roweis, 2005; Davis et al., 2007; Weinberger & Saul, 2008, 2009; Jain et al., 2012). However, for multi-modal data, such as comparing text and images, metric learning has been less explored, consisting primarily of a slow semi-definite programming approach (Zhang et al., 2011) and local alternating descent approaches (Xie & Xing, 2013).

Concurrently, there is a growing literature that tackles *co-embedding problems*, where *multiple* sets or modalities are embedded into a common space to improve prediction performance, reveal relationships and enable zero-shot learning. Current approaches to these problems are mainly based on deep neural networks (Ngiam et al., 2011; Srivastava & Salakhutdinov, 2012; Socher et al., 2013a,b; Frome et al., 2013) and simpler non-convex objectives (Chopra et al., 2005; Larochelle et al., 2008; Weston et al., 2010; Cheng, 2013; Akata et al., 2013). Unlike metric learning, the focus of this previous work has been on exploring heterogeneous data, but without global optimization techniques. This disconnect appears to be unnecessary however, since the standard association scores used for co-embedding are related to a Euclidean metric.

In this paper, we demonstrate that co-embedding can be cast as metric learning. Once formalized, this connection allows metric learning methods to be applied to a wider class of problems, including link prediction, multi-label and multi-class tagging, and ranking. Previous formulations of co-embedding as metric learning were either non-convex (Zhai et al., 2013; Duan et al., 2012), introduced approximation (Akata et al.,

2013; Huang et al., 2014), dropped positive semi-definiteness (Chechik et al., 2009; Kulis et al., 2011), or required all data to share the same dimensionality (Garreau et al., 2014). Instead, we provide a convex formulation applicable to heterogeneous data.

Once the general framework has been established, the paper then investigates optimization strategies for metric learning that guarantee convergence to a global optimum. Although many metric learning approaches have been based on convex formulations, these typically introduce a semi-definite constraint over a matrix variable, $C \succeq 0$, which hampers scalability. An alternative approach that has been gaining popularity has been to work with a low-rank factorization Q that implicitly maintains positive semi-definiteness through $C = QQ'$ (Burer & Monteiro, 2003). This approach allows one to optimize over smaller matrices while avoiding the semi-definite constraint. Recently, Journée et al. (2010) proved that if Q has more columns than the globally optimal rank, a locally optimal Q^* provides a *global* solution $C^* = Q^*Q^{*\prime}$, provided that the objective is smooth and convex *in* C. This result is often neglected in the metric learning literature. However, by using this result, we are able to develop a fast approach to metric learning that improves previous approaches (Journée et al., 2010; Zhang et al., 2012).

The paper then concludes with an empirical investigation of a metric learning task and two co-embedding tasks: multi-label classification and tagging. We demonstrate that the diversity of local minima contracts rapidly in these problems and that local solutions approach global optimality well before the true rank is attained.

2 Metric Learning

The goal of metric learning is to learn a distance function between data instances that helps solve prediction problems. To obtain task-specific distances without extensive manual design, supervised metric learning formulations attempt to exploit task-specific information to guide the learning process. For example, to recognize individual people in images a distance function needs to emphasize certain distinguishing features (such as hair color, etc.), whereas to recognize person-independent facial expressions in the same data, different features should be emphasized (such as mouth shape, etc.).

Suppose one has a sample of t observations, $\mathbf{x}_i \in \mathcal{X}$, and a feature map $\phi : \mathcal{X} \to \mathbb{R}^n$. Then a training matrix $\phi(X) = [\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_t)] \in \mathbb{R}^{n \times t}$ can be obtained by applying ϕ to each of the original data points.[3] A natural distance function between points $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$ can then be given by a Mahalanobis distance over the feature space

$$d_C(\mathbf{x}_1, \mathbf{x}_2) = (\phi(\mathbf{x}_1) - \phi(\mathbf{x}_2))' C (\phi(\mathbf{x}_1) - \phi(\mathbf{x}_2)) \tag{1}$$

specified by some positive semi-definite inverse covariance matrix $C \in \mathcal{C} \subset \mathbb{R}^{n \times n}$.

Although an inverse covariance in this form can be learned in an unsupervised manner, there is often side information that should influence the learning. As a general framework, Kulis (2013) unifies metric learning problems as learning a positive semi-definite matrix C that minimizes a sum of loss functions plus a regularizer:[4]

$$\min_{C \succeq 0, C \in \mathcal{C}} \sum_i L_i(\phi(X)' C \phi(X)) + \beta \, \text{reg}(\mathbf{C}). \tag{2}$$

[3] Throughout the paper we extend functions $\mathbb{R} \to \mathbb{R}$ to vectors or matrices elementwise.

[4] Kulis (2013) equivalently places the trade-off parameter on the loss rather than the regularizer.

For example, in large margin nearest neighbor learning, one might want to minimize

$$L(\phi(X)'C\phi(X)) = \sum_{(i,j)\in\mathcal{S}} d_C(\mathbf{x}_i, \mathbf{x}_j) + \sum_{(i,j,k)\in\mathcal{R}} [1 + d_C(\mathbf{x}_i, \mathbf{x}_j) - d_C(\mathbf{x}_i, \mathbf{x}_k)]_+$$

where $\mathcal{S}$ is a set of "should link" pairs, and $\mathcal{R}$ provides a set of triples (i, j, k) specifying that if $(i, j) \in \mathcal{S}$ then $\mathbf{x}_k$ should have a different label than $\mathbf{x}_i$.

Although supervised metric learning has typically been used for classification, it can also be applied to other settings where distances between data points are useful, such as for kernel regression or ranking. Interestingly, the applicability of metric learning can be extended well beyond the framework (2) by additionally observing that *co-embedding* elements from different sets can also be expressed as a *joint* metric learning problem.

3 Co-embedding as Metric Learning

Co-embedding considers the problem of mapping elements from distinct sets into a common (low dimensional) Euclidean space. Once so embedded, simple Euclidean proximity can be used to determine associations between elements from different sets. This idea underlies many useful formulations in machine learning. For example, in retrieval and recommendation, Bordes et al. (2014) use co-embedding of questions and answers to rank appropriate answers to a query, and Yamanishi (2008) embeds nodes of a heterogeneous graph for link prediction. In natural language processing, Globerson et al. (2007) embed documents, words and authors for semantic document analysis, while Bordes et al. (2012) embed words and senses for word sense disambiguation.

Despite the diversity of these formulations, we show that co-embedding can be unified in a simple metric learning framework. Such a unification is inspired by (Mirzazadeh et al., 2014), who proposed a general framework for bi-linear co-embedding models but did not investigate the extension to metric learning. Here we develop a full formulation of co-embedding as metric learning and develop algorithmic advances.

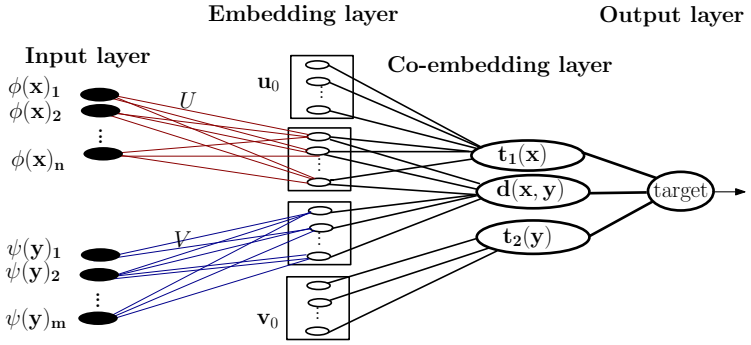

Fig. 1. A neural network view of co-embedding

For co-embedding, assume we are given two sets of data objects $\mathcal{X}$ and $\mathcal{Y}$ with feature maps $\phi(\mathbf{x}) \in \mathbb{R}^n$ and $\psi(\mathbf{y}) \in \mathbb{R}^m$ respectively. Without loss of generality, we assume that the number of samples from $\mathcal{Y}$, t_y, is no more than t, the number of samples

from $\mathcal{X}$; that is, $t_y \leq t$. The goal is to map the elements $\mathbf{x} \in \mathcal{X}$ and $\mathbf{y} \in \mathcal{Y}$ from each set into a common Euclidean space.[5]

A standard approach is to consider linear maps into a common d dimensional space where $U \in \mathbb{R}^{d \times n}$ and $V \in \mathbb{R}^{d \times m}$ are parameters. To provide decision thresholds two dummy items can also be embedded from each space, parameterized by $\mathbf{u}_0$ and $\mathbf{v}_0$ respectively. Figure 1 depicts this standard co-embedding set-up as a neural network, where the trainable parameters, U, V, $\mathbf{u}_0$ and $\mathbf{v}_0$, are in the first layer. The inputs to the network are the feature representations $\phi(\mathbf{x}) \in \mathbb{R}^n$ and $\psi(\mathbf{y}) \in \mathbb{R}^m$. The first hidden layer, the *embedding layer*, linearly maps input to embeddings in a common d dimensional space via:

$$\mathbf{u}(\mathbf{x}) = U\phi(\mathbf{x}), \qquad \mathbf{v}(\mathbf{y}) = V\psi(\mathbf{y}).$$

The second hidden layer, the *co-embedding layer*, computes the distance function between embeddings, $d(\mathbf{x}, \mathbf{y})$, and decision thresholds, $t_1(\mathbf{x})$ and $t_2(\mathbf{y})$:

$$d(\mathbf{x}, \mathbf{y}) = \|\mathbf{u}(\mathbf{x}) - \mathbf{v}(\mathbf{y})\|^2, \quad t_1(\mathbf{x}) = \|\mathbf{u}(\mathbf{x}) - \mathbf{u}_0\|^2, \quad t_2(\mathbf{y}) = \|\mathbf{v}(\mathbf{y}) - \mathbf{v}_0\|^2. \tag{3}$$

The output layer nonlinearly combines the association scores and thresholds to predict targets. For example, in a multi-label classification problem, given an element $\mathbf{x} \in \mathcal{X}$, its association to each $\mathbf{y} \in \mathcal{Y}$ can be determined via: $\text{label}(\mathbf{y}|\mathbf{x}) = \text{sign}(t_1(\mathbf{x}) - d(\mathbf{x}, \mathbf{y}))$. Alternatively, in a symmetric (i.e. undirected) link prediction problem, the association between a pair of elements $\mathbf{x} \in \mathcal{X}$, $\mathbf{y} \in \mathcal{Y}$ can be determined by $\text{label}(\mathbf{x}, \mathbf{y}) = \text{sign}(\min(t_1(\mathbf{x}), t_2(\mathbf{y})) - d(\mathbf{x}, \mathbf{y}))$, and so on.

Although the relationship to metric learning might not be obvious, it is useful to observe that the quantities in (3) can be expressed in terms of underlying covariances:

$$\begin{aligned}
d(\mathbf{x}, \mathbf{y}) &= \begin{bmatrix} \boldsymbol{\phi}(\mathbf{x}) \\ -\boldsymbol{\psi}(\mathbf{y}) \end{bmatrix}' \begin{bmatrix} U'U & U'V \\ V'U & V'V \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}(\mathbf{x}) \\ -\boldsymbol{\psi}(\mathbf{y}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\phi}(\mathbf{x}) \\ -\boldsymbol{\psi}(\mathbf{y}) \end{bmatrix}' C_1 \begin{bmatrix} \boldsymbol{\phi}(\mathbf{x}) \\ -\boldsymbol{\psi}(\mathbf{y}) \end{bmatrix} \\
t_1(\mathbf{x}) &= \begin{bmatrix} \boldsymbol{\phi}(\mathbf{x}) \\ -1 \end{bmatrix}' \begin{bmatrix} U'U & U'\mathbf{u}_0 \\ \mathbf{u}_0'U & \mathbf{u}_0'\mathbf{u}_0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}(\mathbf{x}) \\ -1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\phi}(\mathbf{x}) \\ -1 \end{bmatrix}' C_2 \begin{bmatrix} \boldsymbol{\phi}(\mathbf{x}) \\ -1 \end{bmatrix} \\
t_2(\mathbf{y}) &= \begin{bmatrix} \boldsymbol{\psi}(\mathbf{y}) \\ -1 \end{bmatrix}' \begin{bmatrix} V'V & V'\mathbf{v}_0 \\ \mathbf{v}_0'V & \mathbf{v}_0'\mathbf{v}_0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\psi}(\mathbf{y}) \\ -1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\psi}(\mathbf{y}) \\ -1 \end{bmatrix}' C_3 \begin{bmatrix} \boldsymbol{\psi}(\mathbf{y}) \\ -1 \end{bmatrix}
\end{aligned}$$

where C_1, C_2 and C_3 are symmetric positive semi-definite matrices.

Although our previous work on bi-linear coembedding (Mirzazadeh et al., 2014) did not suggest embedding the thresholds, these turn out to be essential. In fact, to ensure the construction of a common metric space where the inverse covariances are mutually consistent (but without introducing auxiliary equality constraints), one must merge C_1, C_2 and C_3 into a common inverse covariance matrix, $C \in \mathbb{R}^{p \times p}$, $p = n + m + 2$, via:

$$C = \begin{bmatrix} U & V & \mathbf{u}_0 & \mathbf{v}_0 \end{bmatrix}' \begin{bmatrix} U & V & \mathbf{u}_0 & \mathbf{v}_0 \end{bmatrix} \tag{4}$$

From (4), the distance functions d, t_1 and t_2, can then be expressed by

$$\begin{aligned}
d(\mathbf{x}, \mathbf{y}) &= [\phi(\mathbf{x}),\ -\psi(\mathbf{y}),\ 0,\ 0]\ C\ [\phi(\mathbf{x}),\ -\psi(\mathbf{y}),\ 0,\ 0]' \\
t_1(\mathbf{x}) &= [\phi(\mathbf{x}),\ 0,\ -1,\ 0]\ C\ [\phi(\mathbf{x}),\ 0,\ -1,\ 0]' \\
t_2(\mathbf{y}) &= [0,\ -\psi(\mathbf{y}),\ 0,\ -1]\ C\ [0,\ -\psi(\mathbf{y}),\ 0,\ -1]'.
\end{aligned} \tag{5}$$

[5] The extension to more than two sets can be achieved by considering tensor representations.

This yields a novel distance function representation with mutually consistent thresholds.

Finally, based on this new representation, we can extend the general framework (2) to encompass co-embedding in a novel formulation. Let $Y \in \mathbb{R}^{t_y \times m}$ denote the data matrix from the $\mathcal{Y}$ space and let $\widehat{\psi}(Y) \in \mathbb{R}^{t \times m}$ denote a zero-padded version of $\psi(Y)$; that is, a matrix whose top $t_y \times m$ block is $\psi(Y)$ with the remaining $t - t_y$ rows being all zero. Then, defining $\mathbf{f}(X, Y) = [\phi(X)', -\widehat{\psi}(Y)', -\mathbf{1}, -\mathbf{1}]' \in \mathbb{R}^{t \times (n+m+2)}$, where $\mathbf{1}$ denotes an all-one vector (of dimension t in this case), we propose to find C by solving

$$\min_{C \in \mathbb{R}^{p \times p}, C \succeq 0} \sum_i L_i(\mathbf{f}(X, Y)' \, C \, \mathbf{f}(X, Y)) + \beta \, \mathrm{reg}(C) \; . \tag{6}$$

Duan et al. (2012) developed a similar algorithm for domain adaptation, which learned a matrix $C \succeq 0$ instead of U and V; however, they approached a less general setting, which, for example, did not include thresholds nor general losses. Furthermore, their formulation leads to a non-convex optimization problem.

Regularization Regularization is also an important consideration since the risk of over-fitting is ever present. We focus on the most widely used regularizer, the Frobenius norm, which if applied to the factors yields the trace norm regularizer on C:

$$\|U\|_F^2 + \|V\|_F^2 + \|\mathbf{u}_0\|_F^2 + \|\mathbf{v}_0\|_F^2 = \mathrm{tr}(C) = \|C\|_{\mathrm{tr}}.$$

The trace norm (aka nuclear norm) is the sum of the singular values of C. This is a common choice for metric learning since it is the tightest convex lower bound to the rank of a matrix, a widely desired objective for compact learned models and generalization. Moreover, for metric learning, since we have the constraint $C \succeq 0$, the trace norm simplifies to $\|C\|_{\mathrm{tr}} = \mathrm{tr}(C)$, which allows efficient optimization.

4 Algorithm

Given the formulation (6), we consider how to efficiently solve it. First note that the objective can be written, using $L(C) = \sum_i L_i(\mathbf{f}(X, Y)' C \mathbf{f}(X, Y))$, as

$$\min_{C \in \mathbb{R}^{p \times p}, C \succeq 0} f(C) \qquad \text{where } f(C) = L(C) + \beta \, \mathrm{tr}(C). \tag{7}$$

One way to encode the semi-definite constraint is via a change of variable $C = QQ'$:

$$\min_{Q \in \mathbb{R}^{p \times d}} f(QQ') = \min_{Q \in \mathbb{R}^{p \times d}} L(QQ') + \beta \, \mathrm{tr}(QQ'). \tag{8}$$

This optimization, however, becomes non-convex in Q. Recently, however, Journée et al. (2010) showed that local optimization of a related trace constrained problem attains global solutions for rank-deficient local minima $Q \in \mathbb{R}^{p \times d}$; that is, if Q is a local minimum of (8) with $\mathrm{rank}(Q) < d$, then QQ' is a global optimum of (7). In what follows, C^* will denote an optimum of (7) and d^* its rank. Although we have inequality rather than equality constraints, the proof follows easily for our case using the techniques developed in (Bach et al., 2008; Journée et al., 2010; Haeffele et al., 2014), and is an easy consequence of the following, more general result.

Proposition 1. *Consider a local solution of* (8)*, yielding a* Q *such that* $\nabla L(QQ')Q + \beta Q = 0$*. Let* $\mathbf{u}_1, ..., \mathbf{u}_k$ *be the eigenvectors corresponding to the top* k **positive** *eigenvalues* $\lambda_1, ..., \lambda_k$ *of* $-\nabla L(C) - \beta I$*. Then, if* C *is not a solution to* (7)*, it follows that*

1. $k > 0$
2. $\mathbf{u}_1, ..., \mathbf{u}_k$ *are orthogonal to* Q*, yielding* $Q_k = [Q\ \mathbf{u}_1\ ...\ \mathbf{u}_k]$ *such that* $C_k = Q_k Q_k' = C + \sum_{i=1}^k \mathbf{u}_i\mathbf{u}_i'$ *satisfies* $\operatorname{rank}(C_k) = \operatorname{rank}(C) + k$*; and*
3. *the descent direction* $\sum_{i=1} \mathbf{u}_i\mathbf{u}_i'$ *is the solution to*

$$\underset{\substack{\|\mathbf{u}_i\|\leq 1, i=1,...,k \\ \mathbf{u}_i'\mathbf{u}_j = 0, i\neq j,\ \mathbf{u}_i \neq 0}}{\operatorname{argmin}} \Big\langle -\nabla L(C) - \beta I, \textstyle\sum_{i=1}^k \mathbf{u}_i\mathbf{u}_i' \Big\rangle. \tag{9}$$

Proof. **Part 1:** First, form the Lagrangian of (7), given by $L(C) + \beta \operatorname{tr}(C) - \operatorname{tr}(SC)$ with $S \succeq 0$, and consider the KKT conditions:

$$S = \nabla L(C) + \beta I, \quad S \succeq 0, \quad C \succeq 0, \quad SC = 0. \tag{10}$$

The problem is strictly feasible, since $C = I$ is a strictly feasible point; therefore, Slater's condition holds and (10) is sufficient for optimality. Consequently, an optimal solution is reached when $-S \preceq 0$; that is, the largest eigenvalue of $-\nabla L(C) - \beta I$ is negative or zero. We assumed that C is not optimal, therefore $k > 0$.

Part 2: We know that $0 = \nabla L(QQ')Q + \beta Q = SQ$. Therefore, either $S = 0$, in which case we are at a global minimum (which we assumed was not the case) or S is orthogonal to Q. It follows that $-\lambda_i \mathbf{u}_i'Q = (\mathbf{u}_i'S')Q = \mathbf{u}_i'(S'Q) = \mathbf{u}_i'\mathbf{0} = \mathbf{0}$ since $\mathbf{u}_i$ is an eigenvector of S and S is symmetric.

Part 3: To optimize the inner product (9), introduce Lagrange multipliers $\xi_i > 0$ for the norm constraints. Since $-S$ is symmetric, we can re-express the inner objective as

$$\underset{\substack{\mathbf{u}_1,...,\mathbf{u}_k \\ \mathbf{u}_i'\mathbf{u}_j = 0, i\neq j,\ \mathbf{u}_i \neq 0}}{\operatorname{argmin}} \sum_i \mathbf{u}_i'(-S)'\mathbf{u}_i - \sum_i \xi_i \mathbf{u}_i'\mathbf{u}_i.$$

Considering the gradients yields $\frac{\partial}{\partial \mathbf{u}_i} = -S\mathbf{u}_i - 2\xi_i\mathbf{u}_i = 0$, which implies $(-S)\mathbf{u}_i = 2\xi_i\mathbf{u}_i$; that is $\mathbf{u}_i$ is an eigenvector of $-S$ corresponding to eigenvalue $\lambda_i = 2\xi_i > 0$. □

Corollary 1. *Let* $Q \in \mathbb{R}^{p\times d}$*. If (i)* Q *is a local minimum of* $f(QQ')$ *with* $\operatorname{rank}(Q) < d$ *or (ii)* Q *is a critical point of* $f(QQ')$ *with* $\operatorname{rank}(Q) = p$*, then* QQ' *is a solution of* (7)*.*

Proof. First assume condition (i) holds and argue by contradiction. Assume QQ' is not a global optimum of (7), and let $\mathbf{u}_1 \in \mathbb{R}^p$ be as defined as in Proposition 1. Then, $f(QQ' + \beta\mathbf{u}_1\mathbf{u}_1') < f(QQ')$ for a sufficiently small $\beta > 0$. Furthermore, since $\operatorname{rank}(Q) < d$, there exists an orthogonal matrix $V \in \mathbb{R}^{d\times d}$ such that QV has a zero column. Let $\widehat{Q}_\alpha$ be the matrix obtained from QV by replacing this zero column by $\alpha\mathbf{u}_1$, $\alpha = \sqrt{\beta}$. Then $\lim_{\alpha\to 0} \widehat{Q}_\alpha V' = QVV' = Q$. Moreover, since $\mathbf{u}_1$ is orthogonal to the columns of Q, it is also orthogonal to the columns of QV, so $\widehat{Q}_\alpha V(\widehat{Q}_\alpha V)' = QV(QV)' + \alpha^2\mathbf{u}_1\mathbf{u}_1' = QQ' + \beta\mathbf{u}_1\mathbf{u}_1'$. Therefore, $f(\widehat{Q}_\alpha\widehat{Q}_\alpha') = f(QQ' + \beta\mathbf{u}_1\mathbf{u}_1') < f(QQ')$ for $Q_\alpha \in \mathbb{R}^{p\times d}$, hence Q is not a local optimum of f.

Next assume (ii). Since Q is a critical point of $f(QQ')$, $\nabla f(QQ')Q = 0$. Since Q has rank p, the null-space of $\nabla f(QQ')$ is of dimension p, yielding that $\nabla f(QQ') = 0$. Since $QQ' \succeq 0$ and f is convex, $C = QQ'$ is an optimum of (7). □

Algorithm 1 Iterative local algorithm (ILA)

1: **Input:** $L : \mathcal{C} \to \mathbb{R}$, $\beta > 0$
2: **Output:** Q, such that $QQ' = \min_{C:C\succeq 0} L(C) + \beta \operatorname{tr}(C)$
3: $Q \leftarrow 0, k \leftarrow 1, \epsilon \leftarrow 10^{-6}$ ▷ Note $L(QQ') + \operatorname{tr}(QQ')$ is evaluable without forming QQ'
4: **while** not converged **do**
5: $\{\mathbf{u}_1, ..., \mathbf{u}_j\} \leftarrow$ up-to-k-top-positive-eigenvectors$(-\nabla L(QQ') - \beta I)$
6: $\{\lambda_1, ..., \lambda_j\} \leftarrow$ up-to-k-top-positive-eigenvalues$(-\nabla L(QQ') - \beta I)$
7: **if** $k = 0$ or $\lambda_1 \leq \epsilon$ **then** break ▷ converged
8: $k \leftarrow j$
9: $U \leftarrow \sum_i \mathbf{u}_i \mathbf{u}_i'$
10: $(a, b) \leftarrow \operatorname*{argmin}_{a\geq 0, b\geq 0} L(aQQ' + bU) + \beta a \operatorname{tr}(QQ') + \beta bk$ ▷ Line search
11: $Q_{init} \leftarrow [\sqrt{a}Q, \sqrt{b}\mathbf{u}_1, ..., \sqrt{b}\mathbf{u}_k]$ ▷ Start local optimization from Q_{init}
12: $Q \leftarrow$ locally_optimize$(Q_{init}, L(QQ') + \beta \operatorname{tr}(QQ'))$
13: $k \leftarrow 2k$
14: **return** $C = QQ'$

To efficiently solve (7), we therefore propose the Iterative Local Algorithm (ILA) shown in Algorithm 1. ILA iteratively adds multiple columns to an initially empty Q and performs a local optimization over $Q \in \mathbb{R}^{p\times d}$ until convergence. The main advantage of this approach over simply setting $d = p$ is that good initial points are generated, and if $d^* \ll p$, then incrementally growing d optimizes over much smaller Q variables. Furthermore, one hopes that when the number of columns d of Q_{init} is at least d^*, ILA finds the global optimum. In particular, if the local optimizer in line 12 of ILA always returns a local optimum whose rank is smaller than d if $d > d^*$ (we call this a *nice local optimizer*), then the optimality of a rank-deficient local minimum implies that ILA finds the global optimum when $d > d^*$. While in theory we cannot guarantee such a behavior of the local algorithm, it always happened in our experiments, similarly to what was reported in earlier work (Journée et al., 2010; Haeffele et al., 2014).

The main novelty of ILA over previous approaches is in the initialization and expansion of columns in Q, which reduces the number of iterations from d^* to $O(\log d^*)$ for nice local optimizers. In particular, motivated by Proposition 1, to generate the candidate columns, ILA uses eigenvectors corresponding to the top k positive eigenvalues of $-\nabla L(C) - \beta I$ capped at 2^{i-1} columns on the ith iteration. Such an exponential search quickly covers the space of possible d, even when d^* is large, while still initially optimizing over smaller Q matrices. This approach can be significantly faster than the typical single column increment (Journée et al., 2010; Zhang et al., 2012), whose complexity typically grows linearly with d^*.[6]

Compared to earlier work, there are also small differences in the optimization: Zhang et al. (2012) do not constrain C to be positive semi-definite. Journée et al. (2010) assume an equality constraint on the trace of C; their Lagrange variable (i.e., regularization parameter) can therefore be negative. Finally, ILA more efficiently exploits the local algorithm. The convergence analysis of Zhang et al. (2012) does not include local

[6] One can create problems where adding single columns improves performance, but we observe in our experiments that the proposed approach is more effective in practice.

training. In practice, we find that solely using boosting (with the top eigenvector as the weak learner) without local optimization, results in much slower convergence.

Corollary 1 implies ILA solves (7) when the local optimizer avoids saddle points.

Corollary 2. *Suppose the local optimizer always finds a local optimum, where d is the number of columns in Q. Then ILA stops with a solution to* (7) *in line 12 with* $\mathrm{rank}(Q)<d$ *or* $d=p$. *If, in addition, the local optimizer is* nice*, this happens for* $d>d^*$.

Due to the exponential search in ILA, the algorithm stops in essentially at most $\log(p)$ iterations when the local optimizer avoids saddle points, and in about $\log(d^*)$ iterations for *nice* local optimizers. However, ILA can potentially be slower if there are not enough eigenvectors to add in a given iteration; i.e., $j < k$ in line 5.

Similarly to (Journée et al., 2010; Zhang et al., 2012; Haeffele et al., 2014) we have found that the local optimizer always returns local minima in practice. However, all of these search-based algorithms risk strange behavior if the local optimizer returns saddle points. Note that even in this case, if d reaches p in any iteration, ILA finds an optimum by Corollary 1. However, there is no guarantee that the rank of Q is not reduced in the local optimization step. If this happens and Q is a local optimum, QQ' is optimal by Corollary 1 and the algorithm halts. Unfortunately, this is not the only possibility: in every iteration of ILA we obtain Q_{init} by increasing the rank of the previous Q, but the ranks might be subsequently reduced during the local optimization step. This creates the potential for a loop where $\mathrm{rank}(Q)$ never reaches p.

Such potential effects of saddle points have not been considered in previous papers. However, we close this section by showing that ILA is still consistent under mild technical conditions on L, even if the local optimizer can get trapped in saddle points.

Proposition 2. *Suppose that f is ν-smooth; that is, $\|\nabla f(C+S)-\nabla f(C)\|_{\mathrm{tr}} \leq \nu\rho(S)$ for all $C, S \in \mathbb{R}^{p\times p}$, $C, S \succeq 0$ and some $\nu \geq 0$, where $\rho(S)$ denotes the spectral norm of S. Assume furthermore, for simplicity, that $L(C) \geq 0$ for all $C \succeq 0$. If the local optimizer in line 12 always returns a Q such that $\nabla f(QQ')Q = 0$, then QQ' in ILA converges to the globally optimal solution of* (7).

Proof. Let Q_m and U_m denote the matrix Q and U in ILA when line 10 is executed the mth time, and let $Q_{init,m}$ denote Q_{init} obtained from Q_m. Note that $Q_{init,m} = \sqrt{a}Q_m + \sqrt{b}U_m$ and Q_{m+1} is obtained from $Q_{init,m}$ via local optimization in line 12. Furthermore, let $C_m = Q_m Q_m'$ and $C_{init,m} = Q_{init,m}Q_{init,m}' = a_m C_m + b_m U_m U_m'$.

If C_m is not a global optimum of (7), then $f(C_{init,m}) < f(C_m)$ by Proposition 1. Furthermore, we assume that the local optimizer in line 12 cannot increase the function value f of $C_{init,m}$, hence $f(C_{m+1}) \leq f(C_{init,m})$, and consequently $f(C_{m+1}) < f(C_m)$. Note that since $L(C_m) \geq 0$, we have $\|Q_m\|_F = \mathrm{tr}(C_m) \leq f(C_0)$, thus the entries of C_m are uniformly bounded for all m. Therefore, $(C_m)_m$ has a convergent subsequence, and denote its limit point by $\widehat{C}$. We will show that $\widehat{C}$ is an optimal solution of (7) by verifying the KKT conditions (10) with $S = \nabla f(\widehat{C})$. First notice that $\widehat{C}$ is positive semi-definite, $\nabla f(\widehat{C})\widehat{C} = 0$ by continuity since $\nabla f(C_m)C_m = \nabla f(QQ')QQ' = 0$. Thus, we only need to verify that $\nabla f(\widehat{C})$ is positive semi-definite.

To show the latter, we first apply Lemma 1 (provided in the appendix) to obtain a lower bound ILA's progress:

$$\begin{aligned} f(C_{m+1}) \leq f(C_{init,m+1}) = f(aC_m + bU_m U_m') &\leq f(C_m + \hat{b} U_m U_m') \\ &\leq f(C_m) + \operatorname{tr}((\hat{b} U_m U_m')' \nabla f(C_m)) + \frac{\nu}{2} \rho (\hat{b} U_m U_m')^2 \\ &= f(C_m) + \operatorname{tr}(\hat{b} U_m' \nabla f(C_m) U_m) + \frac{\nu \hat{b}^2}{2} \end{aligned} \tag{11}$$

for any $\hat{b} \geq 0$, where the last equality holds since $U_m U_m'$ has k_m eigenvalues equal 1, and $p - k_m$ equal 0, where k_m denotes the number of columns of U_m. Now consider

$$\hat{b} = -\frac{\operatorname{tr}(U_m' \nabla f(C_m) U_m)}{\nu} = \frac{\operatorname{tr}(U_m' \Lambda_m U_m)}{\nu} = \frac{1}{\nu} \sum_{i=1}^{k_m} \lambda_{m,i},$$

where $\lambda_1 \geq \cdots \geq \lambda_{k_m} > 0$ are the eigenvalues of $-\nabla f(C_m)$, and Λ_m is the diagonal matrix of the eigenvalues padded with $p - m_k$ zeros. Then $\operatorname{tr}(\hat{b} U_m' \nabla f(C_m) U_m) = -\nu \hat{b}^2$, hence (11) yields

$$f(C_m) - f(C_{m+1}) \geq \frac{\nu}{2} \hat{b}^2 = \frac{1}{\nu} \left(\sum_{i=1}^{k_m} \lambda_{m,i} \right)^2 \geq \frac{\lambda_{m,1}^2}{2\nu} \,.$$

By our assumptions, $f(C_0) \geq 0$, and so using the monotonicity of $f(C_m)$, we have

$$f(C_0) \geq \lim_{m \to \infty} f(C_0) - f(C_{m+1}) = \lim_{m \to \infty} \sum_{i=0}^{m} f(C_i) - f(C_{i+1}) \geq \frac{1}{2\nu} \sum_{m=0}^{\infty} \lambda_{m,1}^2.$$

Therefore, $\lim_{m \to \infty} \lambda_{m,1} = 0$. Thus, by continuity, $-\nabla f(\widehat{C})$ has no positive eigenvalues, implying that $\nabla f(\widehat{C})$ is positive semi-definite, concluding the proof. □

5 Empirical Computational Complexity

To compare the exponential versus linear rank expansion strategies for ILA we first consider a standard metric learning problem. In this experiment to control the rank of the solution, we generated synthetic data $X \in \mathbb{R}^{n \times t}$ from a standard normal distribution, systematically increasing the data dimension from $n = 1$ to $n = 1000$ and increasing the sample sizes from $t = 250$ to $t = 2000$. The training objective was set to

$$\min_{C \succeq 0} \|X'X - X'CX\|_F^2 + \beta \operatorname{tr}(C) \tag{12}$$

with a regularization parameter $\beta = 0.5$.

Figure 2 compares the run times of the linear versus exponential expansion strategies, both of which optimize over Q of increasing width rather than $C = QQ'$. Both methods used the same local optimizer but differed in how many new columns were

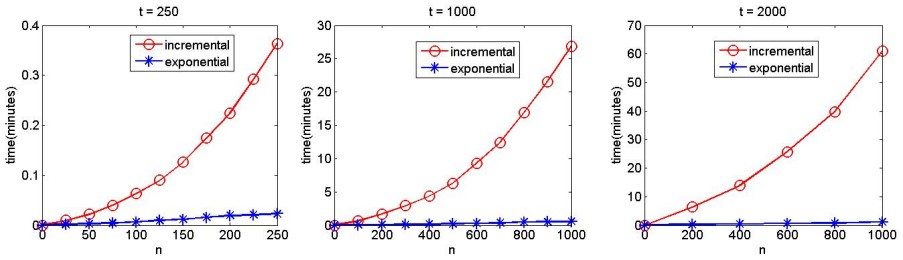

Fig. 2. Comparing the run time in minutes (y-axis) of linear versus exponential strategies in ILA as data dimension (x-axis) is increased. Left shows $t = 250$, middle shows $t = 1000$, and right shows $t = 2000$.

generated for Q in ILA Line 8. For the smaller sample size $t = 250$, the exponential search already demonstrates an advantage as data dimension is increased. However, for larger sample sizes, the advantage of the exponential approach becomes even more pronounced. In this case, when n is increased from 0 to 1000 the run time of the linear expansion strategy goes from being about the same as of the exponential strategy to much slower. The trend indicates that the exponential search becomes more useful as the data dimension and number of samples increases.

6 Case Study: Multi-label Classification

Next, we evaluated ILA on a challenging problem setting—multi-label classification—with real data. In this setting one can view the labels themselves as objects to be co-embedded with data instances; given such an embedding, the multi-label classification of an input instance $\mathbf{x}$ can be determined by comparing the distance of its embedding to the embedded locations of each label. In particular, given a feature representation $\phi(\mathbf{x}) \in \mathbb{R}^n$ for data instances $\mathbf{x} \in \mathcal{X}$, we introduce a simple indicator feature map $\psi(\mathbf{y}) \in \mathbb{R}^m$ over $\mathbf{y} \in \mathcal{Y}$, which specifies a vector of all zeros with a single 1 in the entry corresponding to label $\mathbf{y}$. From a co-embedding perspective, the training problem then becomes to map the feature representations of both the input instances $\mathbf{x} \in \mathcal{X}$ and target labels $\mathbf{y} \in \mathcal{Y}$ into a common Euclidean space.

Based on this observation, we can then cast multi-label learning as an equivalent *metric learning* problem where one learns the inverse covariance C. Following the development in Section 3 (but here not using the threshold for $\mathbf{y}$ since it is not needed), the co-embedding parameters U, V and $\mathbf{u}_0$ can first be combined into a joint matrix $Q = \begin{bmatrix} U, & V, & \mathbf{u}_0 \end{bmatrix} \in \mathbb{R}^{p \times d}$, where $p = n + m + 1$. Then, as in (4), the co-embedding problem of optimizing U, V and $\mathbf{u}_0$ can be equivalently expressed as a metric learning problem of optimizing the inverse covariance $C = QQ' \in \mathbb{R}^{p \times p}$.

Training objective To develop a novel metric learning based approach to multi-label classification, we adopt a standard training loss that encourages small distances between an instance's embedding and the embeddings of its associated labels while encouraging large distances to embeddings of disassociated labels. In particular, we investigate the convex large margin loss suggested by Mirzazadeh et al. (2014) which was reported to

yield good performance for multi-label classification (in a bilinear co-embedding model but not a metric learning model):

$$\min_{C\succeq 0} \beta \operatorname{tr}(C) + \sum_{x\in\mathcal{X}} \Big[\operatorname*{sftmx}_{y\in\mathcal{Y}(\mathbf{x})} \tilde{h}(d_C(\mathbf{x},\mathbf{y}) - t_C(\mathbf{x})) + \operatorname*{sftmx}_{\bar{\mathbf{y}}\in\bar{\mathcal{Y}}(\mathbf{x})} \tilde{h}(t_C(\mathbf{x}) - d_C(\mathbf{x},\bar{\mathbf{y}})) \Big] \quad (13)$$

where $\operatorname{sftmx}_{y\in\mathcal{Y}}(z_y) = \ln \sum_{y\in\mathcal{Y}} \exp(z_y)$, $t_C(x) = [\phi(\mathbf{x}),\, \mathbf{0},\, -1]\; C\; [\phi(\mathbf{x}),\, \mathbf{0},\, -1]'$, $d_C(\mathbf{x},\mathbf{y}) = [\phi(\mathbf{x}),\, -\psi(\mathbf{y}),\, 0]\; C\; [\phi(\mathbf{x}),\, -\psi(\mathbf{y}),\, 0]'$ and $\tilde{h}(z) = (2+z)_+^2/4$ if $0 \le z \le 2$; $(1+z)_+$ otherwise. Here we are using $\mathcal{Y}(\mathbf{x}) \subset \mathcal{Y}$ to denote the subset of labels associated with $\mathbf{x}$, and $\bar{\mathcal{Y}}(\mathbf{x}) \subset \mathcal{Y}$ to denote the subset of labels disassociated with $\mathbf{x}$.

Note that in (13) we also use Frobenius norm regularization on the co-embedding parameters U, V and $\mathbf{u}_0$, which was shown in Section 3 to yield trace regularization of C: $\|U\|_F^2 + \|V\|_F^2 + \|\mathbf{u}_0\|_2^2 = \operatorname{tr}(U'U) + \operatorname{tr}(V'V) + \mathbf{u}_0'\mathbf{u}_0 = \operatorname{tr}(C)$.

Results We investigate the behavior of ILA on five widely used multi-label classification data sets, summarized in Table 1. To establish the suitability of metric learning for multi-label classification, we evaluated test performance using three commonly used criteria for multi-label classification: Hamming score (Table 2), micro averaged F1 measure (Table 3) and macro averaged F1 measure (Table 4). Here β was chosen by cross-validation over $\{1, 0.5, 0.1, 0.05, 0.01, 0.005\}$. We compared the performance of the proposed approach against six standard competitors: BR(SMO), an independent SVM classifiers for each label (Platt, 1998); BR(LOG), an independent logistic regression (LOG) classifiers for each label (Hastie et al., 2009); CLR(SMO) and CLR(LOG), the calibrated pairwise label ranking method of Fürnkranz et al. (2008) with SVM and LOG, respectively; and CC(SMO) and CC(LOG), a chain of SVM classifiers and a chain of logistic regression classifiers for multi-label classification by Read et al. (2011). The results in Tables 2–4 are averaged over 10 splits and demonstrate comparable performance to the best competitors consistently in all three criteria for all data sets.

Data set	examples	features	labels
Emotion	593	72	6
Scene	2407	294	6
Yeast	2417	103	14
Mediamill	3000	120	30
Corel5K	4609	499	30

Table 1. Data properties for multi-label experiments. 1000 used for training and the rest for testing ($2/3$-$1/3$ split for Emotion).

Next, to also investigate the properties of the local optima achieved we ran local optimization from 1000 random initializations of Q at successive values for d, using $\beta = 1$. The values of the local optima we observed are plotted in Figure 3 as a function of d.[7] As expected, the local optimizer always achieves the globally optimal value when $d \ge d^*$. Interestingly, for $d < d^*$ we see that the initially wide diversity of local optimum values contracts quickly to a singleton, with values approaching the global minimum before reaching $d = d^*$. Although not displayed in the graphs, other useful properties can be observed. First, for $d \ge d^*$, the global optimum is achieved by local optimization under random initialization, but not with initialization to any of the critical points of smaller d observed in Figure 3, which traps the optimization in a saddle point.

[7] Note that Q is not unique since $C = QQ'$ is invariant to transform QR for orthonormal R.

	BR(SMO)	BR(LOG)	CLR(SMO)	CLR(LOG)	CC(SMO)	CC(LOG)	ILA
Emotion	80.9 ±1.0	77.1 ±1.2	79.9 ±0.7	76.0 ±1.4	79.0 ±0.9	75.2 ±1.1	80.2 ±0.8
Scene	88.7 ±0.4	81.9 ±0.6	89.7 ±0.3	85.7 ±0.4	88.9 ±0.4	80.9 ±0.4	88.0 ±0.5
Yeast	79.8 ±0.2	77.0 ±0.2	77.2 ±0.2	75.3 ±0.3	78.9 ±0.5	76.0 ±0.2	78.9 ±0.3
Mediamill	90.3 ±0.1	87.4 ±0.2	87.8 ±0.1	87.7 ±0.1	89.9 ±0.1	86.3 ±0.3	90.4 ±0.5
Corel5K	89.8 ±0.1	88.5 ±0.2	88.8 ±0.1	88.0 ±0.1	89.6 ±0.1	83.1 ±0.4	87.8 ±0.4

Table 2. Comparison of ILA with competitors in terms of Hamming score.

	BR(SMO)	BR(LOG)	CLR(SMO)	CLR(LOG)	CC(SMO)	CC(LOG)	ILA
Emotion	66.3 ±2.3	63.2 ±1.8	70.1 ± 1.2	64.5 ± 2.1	65.9 ± 1.8	60.3 ± 1.9	65.9 ± 1.3
Scene	66.8 ±1.0	49.5 ±1.5	72.2 ± 0.7	61.8 ± 1.3	68.8 ± 1.1	50.1 ± 1.1	65.9 ± 0.8
Yeast	63.2 ±0.3	62.0 ±0.4	65.0 ± 0.3	61.9 ± 0.4	63.7 ± 0.8	60.0 ± 0.4	62.4 ± 0.5
Mediamill	55.4 ±0.5	55.1 ±0.6	59.7 ± 0.4	58.7 ± 0.4	50.7 ± 0.9	53.1 ± 0.7	58.0 ± 0.7
Corel5K	21.9 ±0.7	17.4 ±0.5	27.6 ± 0.4	26.3 ± 0.5	21.9 ± 0.5	16.7 ± 0.6	21.9 ± 0.6

Table 3. Comparison of ILA with competitors in terms of Micro F1.

	BR(SMO)	BR(LOG)	CLR(SMO)	CLR(LOG)	CC(SMO)	CC(LOG)	ILA
Emotion	62.3 ±3.1	62.0 ±1.9	69.0 ±1.0	63.8 ±2.0	64.3 ±1.8	59.3 ±2.0	64.4 ±1.4
Scene	67.6 ±0.9	50.6 ±1.6	73.3 ±0.6	63.3 ±1.3	69.8 ±1.0	50.9 ±1.0	66.8 ±0.9
Yeast	32.9 ±0.7	41.9 ±0.8	40.3 ±0.6	42.6 ±0.7	35.1 ±0.4	40.4 ±0.4	37.8 ±0.8
Mediamill	10.0 ±0.4	29.9 ±0.7	21.4 ±0.7	31.7 ±0.8	8.9 ±1.0	29.5 ±0.8	16.2 ±0.9
Corel5K	17.8 ±0.4	11.6 ±0.4	21.4 ±0.5	22.0 ±0.5	17.6 ±0.5	14.4 ±0.6	17.8 ±0.6

Table 4. Comparison of ILA with competitors in terms of Macro F1.

Overall, empirically and theoretically, we find that ILA quickly finds global solutions for the multi-label objective, while typically producing good solutions before $d = d^*$.

7 Case Study: Tagging via Tensor Completion

Finally, we investigated Task 2 of the 2009 ECML/PKDD Discovery Challenge: a multi-relational problem involving users, items and tags, where users have tagged subsets of the items and the goal is to predict which tags the users will assign to other items. Here the training data is given in a tensor T, where $T(x, y, z) = 1$ indicates that x has tagged z with y, $T(x, y, z) = -1$ indicates that y is not a tag of z according to x, and $T(x, y, z) = 0$ denotes an unknown entry. The goal is to predict the unknown values, subject to a constraint that at most five tags can be active for any user-item pair. The "core at level 10" subsample reduces the data to 109, 192, 229 unique users, items, and tags respectively (Jäschke et al., 2008). The winner of this challenge (Rendle & Schmidt-Thieme, 2009) used a multi-linear co-embedding model that assumed the completed tensor has a low rank structure.

Training Objective To show that this multi-relational prediction problem can be tackled from the novel perspective of metric learning, we first express the problem in terms of a multi-way co-embedding where users, tags and items are mapped to a joint embedding space: $x \mapsto \sigma$, $y \mapsto \tau$ and $z \mapsto \rho$ where $\sigma, \tau, \rho \in \mathbb{R}^d$. The training problem can then be expressed in terms of proximities between embeddings. In particular,

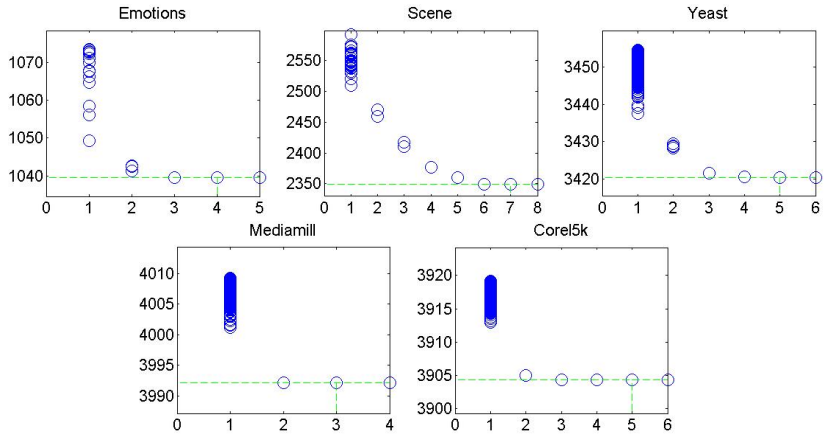

Fig. 3. Objective values achieved by local optimization given 1000 initializations of $Q \in \mathbb{R}^{p \times d}$. For small d a diversity of local minima are observed, but the set of local optima contracts rapidly as d increases, reaching a singleton at the global optimum by $d = d^*$.

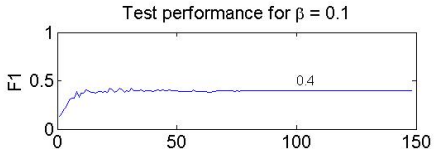

Fig. 4. F1 measure achieved by ILA on test data with an increasing number of columns (optimal rank is 84 in this case).

following Rendle & Schmidt-Thieme (2009), we summarize the three-way interaction between a user, item and tag by the squared distance between the user and tag embeddings, and between the item and tag embeddings: $d(x, y, z) := d(x, y) + d(z, y) = \|\sigma - \tau\|^2 + \|\rho - \tau\|^2$. Given this definition, tags can be predicted from a given user-item pair (x, z) via

$$\hat{T}(x, y, z) = \begin{cases} 1 & \text{if } d(x, y, z) \text{ among smallest five } d(x, \cdot, z) \\ -1 & \text{otherwise} \end{cases}.$$

The training problem can be expressed as metric learning by exploiting a construction reminiscent of Section 3: the embedding vectors can conceptually be stacked in matrix factor $Q = \begin{bmatrix} \sigma, & \tau, & \rho \end{bmatrix}'$, which defines the inverse covariance $C = QQ'$. To learn C, we use the same loss proposed by Rendle & Schmidt-Thieme (2009), regularized by the Frobenius norm over σ, τ and ρ (which again corresponds to trace regularization of C), yielding the convex training problem

$$\min_{C \succeq 0} \quad \beta \operatorname{tr}(C) + \sum_{x,z} \sum_{y \in tag(x,z)} \sum_{\bar{y} \notin tag(x,z)} L(d_C(x, z, \bar{y}) - d_C(x, z, y)). \tag{14}$$

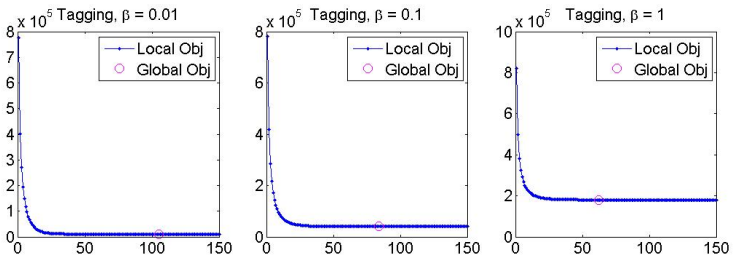

Fig. 5. Training objectives for $\beta \in \{0.01, 0.1, 1\}$ as a function of the rank of C, where the optimal ranks are 105, 84 and 62 respectively.

Results To establish the suitability of metric learning for multi-relational prediction, we first evaluated the test performance achieved on the down-sampled Discovery Challenge data. Figure 4 shows that ILA efficiently approaches the state of the art F1 performance of 0.42 reported by Mirzazadeh et al. (2014). Furthermore, we also investigated the behavior of local minima at different d by comparing the training objective values achieved by local optimization compared to the global minimum, here using $\beta \in \{0.01, 0.1, 1\}$. Figure 5 shows that although the optimal rank can be larger in this scenario, the properties of the local solutions become even more apparent: interestingly, the local minima approach the training global minimum at ranks much smaller than the optimum. These results further support the effectiveness of metric learning and the potential for ILA to solve these problems much more efficiently than standard semi-definite programming approaches.

8 Conclusion

We have demonstrated a unification of co-embedding and metric learning that enables a new perspective on several machine learning problems while expanding the range of applicability for metric learning methods. Additionally, by using recent insights from semi-definite programming theory, we developed a fast local optimization algorithm that is able to preserve global optimality while significantly improving the speed of existing methods. Both the framework and the efficient algorithm were investigated in different contexts, including metric learning, multi-label classification and multi-relational prediction—demonstrating their generality. The unified perspective and general algorithm show that a surprisingly large class of problems can be tackled from a simple perspective, while exhibiting a local-global property that can be usefully exploited to achieve faster training methods.

Bibliography

Akata, Z., Perronnin, F., Harchaoui, Z., and Schmid, C. Label-embedding for attribute-based classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

Bach, F., Mairal, J., and Ponce, J. Convex sparse matrix factorizations. *CoRR*, 2008.

Bordes, A., Glorot, X., Weston, J., and Bengio, Y. Joint learning of words and meaning representations for open-text semantic parsing. In *Proceedings AISTATS*, 2012.

Bordes, A., Weston, J., and Usunier, N. Open question answering with weakly supervised embedding models. In *European Conference on Machine Learning*, 2014.

Burer, S. and Monteiro, R. D. C. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Math. Program.*, 95(2):329–357, 2003.

Chechik, G., Shalit, U., Sharma, V., and Bengio, S. An online algorithm for large scale image similarity learning. In *Neural Information Processing Systems*, 2009.

Cheng, L. Riemannian similarity learning. In *Internat. Conference on Machine Learning*, 2013.

Chopra, S., Hadsell, R., and LeCun, Y. Learning a similarity metric discriminatively, with application to face verification. In *IEEE Conf. on Computer Vision and Pattern Recogn.*, 2005.

Davis, J., Kulis, B., Jain, P., Sra, S., and Dhillon, I. S. Information-theoretic metric learning. In *International Conference on Machine Learning*, 2007.

Duan, L., Xu, D., and Tsang, I. Learning with augmented features for heterogeneous domain adaptation. In *International Conference on Machine Learning*, 2012.

Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., Mikolov, T., et al. Devise: A deep visual-semantic embedding model. In *Neural Information Processing Systems*, 2013.

Fürnkranz, J., Hüllermeier, E., Loza Mencía, E., and Brinker, K. Multilabel classification via calibrated label ranking. *Machine Learning*, 73(2):133–153, 2008.

Garreau, D., Lajugie, R., Arlot, S., and Bach, F. Metric learning for temporal sequence alignment. In *Neural Information Processing Systems*, 2014.

Globerson, A., Chechik, G., Pereira, F., and Tishby, N. Euclidean embedding of co-occurrence data. *Journal of Machine Learning Research*, 8:2265–2295, 2007.

Globerson, A. and Roweis, S. T. Metric learning by collapsing classes. In *NIPS*, 2005.

Haeffele, B., Vidal, R., and Young, E. Structured low-rank matrix factorization: Optimality, algorithm, and applications to image processing. In *Proceedings ICML*, 2014.

Hastie, T., Tibshirani, R., and Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer, 2 edition, 2009.

Huang, Z., Wang, R., Shan, S., and Chen, X. Learning Euclidean-to-Riemannian metric for point-to-set classification. In *IEEE Conference on Computer Vision and Pattern Recogn.*, 2014.

Jain, P., Kulis, B., Davis, J. V., and Dhillon, I. S. Metric and kernel learning using a linear transformation. *Journal of Machine Learning Research*, 13:519–547, 2012.

Jäschke, R., Marinho, L. B., Hotho, A., Schmidt-Thieme, L., and Stumme, G. Tag recommendations in social bookmarking systems. *AI Communications*, 21(4):231–247, 2008.

Journée, M., Bach, F. R., Absil, P.-A., and Sepulchre, R. Low-rank optimization on the cone of positive semidefinite matrices. *SIAM Journal on Optimization*, 20(5):2327–2351, 2010.

Kulis, B. Metric learning: A survey. *Foundat. and Trends in Mach. Learn.*, 5(4):287–364, 2013.

Kulis, B., Saenko, K., and Darrell, T. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *Proceedings CVPR*, 2011.

Larochelle, H., Erhan, D., and Bengio, Y. Zero-data learning of new tasks. In *AAAI*, 2008.

Mirzazadeh, F., Guo, Y., and Schuurmans, D. Convex co-embedding. In *AAAI*, 2014.

Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., and Ng, A. Y. Multimodal deep learning. In *International Conference on Machine Learning*, 2011.

Platt, J. C. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical report, Advances in Kernel Methods, 1998.

Read, J., Pfahringer, B., Holmes, G., and Frank, E. Classifier chains for multi-label classification. *Machine Learning*, 85(3):333–359, 2011.

Rendle, S. and Schmidt-Thieme, L. Factor models for tag recommendation in bibsonomy. In *ECML/PKDD Discovery Challenge*, 2009.

Socher, R., Chen, D., Manning, C. D., and Ng, A. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems*, 2013a.

Socher, R., Ganjoo, M., Manning, C. D., and Ng, A. Zero-shot learning through cross-modal transfer. In *Advances in Neural Information Processing Systems*, pp. 935–943, 2013b.

Srivastava, N. and Salakhutdinov, R. Multimodal learning with deep boltzmann machines. In *Advances in neural information processing systems*, 2012.

Weinberger, K. and Saul, L. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244, 2009.

Weinberger, K. and Saul, L. K. Fast solvers and efficient implementations for distance metric learning. In *International Conference on Machine Learning*, 2008.

Weston, J., Bengio, S., and Usunier, N. Large scale image annotation: learning to rank with joint word-image embeddings. *Machine learning*, 81(1):21–35, 2010.

Xie, P. and Xing, E. Multi-modal distance metric learning. In *Proceedings IJCAI*, 2013.

Xing, E., Ng, A., Jordan, M., and Russell, S. Distance metric learning with application to clustering with side-information. In *Neural Information Processing Systems*, 2002.

Yamanishi, Y. Supervised bipartite graph inference. In *Proceedings NIPS*, 2008.

Zhai, X., Peng, Y., and Xiao, J. Heterogeneous metric learning with joint graph regularization for cross-media retrieval. In *AAAI Conference on Artificial Intelligence*, 2013.

Zhang, H., Huang, T. S., Nasrabadi, N. M., and Zhang, Y. Heterogeneous multi-metric learning for multi-sensor fusion. In *International Conference on Information Fusion*, 2011.

Zhang, X., Yu, Y., and Schuurmans, D. Accelerated training for matrix-norm regularization: A boosting approach. In *Neural Information Processing Systems*, 2012.

A An Auxiliary Lemma

Lemma 1. *Suppose f is ν-smooth. Then for any positive semi-definite $C, S \in \mathbb{R}^{p\times p}$,*

$$f(C+S) \leq f(C) + \operatorname{tr}(S'\nabla f(C)) + \frac{\nu}{2}\rho(S)^2 \,. \tag{15}$$

Proof. Define $h(\eta) = f(C+\eta S)$ for $\eta \in [0,1]$. Note that $h(0) = f(C)$, $h(1) = f(C+S)$, and $h'(\eta) = \operatorname{tr}(S'\nabla f(C+\eta S))$ for any $\eta \in (0,1)$. Then

$$\begin{aligned}
&f(C+S) - f(C) - \operatorname{tr}(S'\nabla f(C)) \\
&= h(1) - h(0) - \operatorname{tr}(S'\nabla f(C)) = \int_0^1 h'(\eta)d\eta - \operatorname{tr}(S'\nabla f(C)) \\
&= \int_0^1 \operatorname{tr}(S'\nabla f(C+\eta S))d\eta - \operatorname{tr}(S'\nabla f(C)) = \int_0^1 \operatorname{tr}\left(S'(\nabla f(C+\eta S) - \nabla f(C))\right)d\eta \\
&\leq \int_0^1 \rho(S)\|\nabla f(C+\eta S) - \nabla f(C)\|_{\operatorname{tr}}\, d\eta \leq \int_0^1 \nu\rho(S)\rho(\eta S)\eta = \int_0^1 \nu\eta\rho(S)^2 d\eta = \frac{\nu}{2}\rho(S)^2
\end{aligned}$$

where the first inequality holds by the Cauchy-Schwarz inequality, and the second by the Lipschitz condition on ∇f. Reordering the inequality establishes the lemma. □