Convex Co-embedding

Farzaneh Mirzazadeh
Department of Computing Science
University of Alberta
Edmonton, AB T6G 2E8, Canada
mirzazad@cs.ualberta.ca

Yuhong Guo
Department of Computer and
Information Sciences, Temple University
Philadelphia, PA 19122, USA
yuhong@temple.edu

Dale Schuurmans
Department of Computing Science
University of Alberta
Edmonton, AB T6G 2E8, Canada
dale@cs.ualberta.ca

Abstract

We present a general framework for association learning, where entities are embedded in a common latent space to express relatedness via geometry—an approach that underlies the state of the art for link prediction, relation learning, multi-label tagging, relevance retrieval and ranking. Although current approaches rely on local training methods applied to non-convex formulations, we demonstrate how general convex formulations can be achieved for entity embedding, both for standard multi-linear and prototype-distance models. We investigate an efficient optimization strategy that allows scaling. An experimental evaluation reveals the advantages of global training in different case studies.

1 Introduction

Associating items between sets is a fundamental problem in applications as diverse as ranking, retrieval, recommendation, link prediction, association mining, relation learning, tagging, and multi-label classification. Despite the diversity of these tasks, a unified approach can be achieved through the concept of an *association score function* that evaluates associative strength between items. For example, *retrieval* and *recommendation* can be expressed as identifying items from a collection that exhibit the strongest association to a given query object; *ranking* can be expressed as sorting items based on their associative strength to a given object; multi-label *tagging* can be expressed as predicting which of a set of label items are associated with a given query object; *link prediction* involves determining which items from a set are related to items from another (possibly identical) set; and so on. These problems can be extended to a *multi-relational* setting by introducing context or side-information to the association scores. Despite their varied histories, different subcommunities have converged on a common approach of using score functions to determine item associations.

Another recent convergence has been the approach of *co-embedding*: one natural way to evaluate associations between objects is to first embed them in a common space and use Euclidean geometry to determine relatedness. For example, *alignment* (i.e., inner product) between embedding vectors can be used to determine the association strength. Another approach is to use the *Euclidean distance* (or related proximity) between embedding vectors to determine relatedness. Such methods provide the current state of the art in association learning, leading to improved prediction performance in applications ranging from image tagging (Akata et al. 2013; Weston, Bengio, and Usunier 2010) to recommendation (Rendle et al. 2009). Beyond improved association quality, these approaches can also provide additional insight by revealing relationships between items in a common space. Such approaches can also be extended to a multi-relational setting by using context to influence the embeddings. Co-embedding also offers a natural approach to “zero shot” learning: whenever a new item is encountered, its embedding can be used to determine its associations.

However, despite their success, current co-embedding methods have drawbacks. Beyond special cases, current formulations of co-embedding are not convex, and existing approaches rely on local training methods (often alternating descent) to acquire the embeddings. A consequence is that the results are not easily repeatable, since every detail of the training algorithm can, in principle, affect the result. A related drawback is that the problem specification is no longer decoupled from the details of the implementation, which can prevent end users, who otherwise understand the specifications, from successfully deploying the technology.

In this paper we offer a unified perspective on co-embedding by presenting a simple framework that expresses association problems in a common format. Second, we show how convex formulations can be generally achieved by simple rank relaxations. Importantly, the proposed reformulation can be applied to both alignment based and distance based score models. This reduction expands the range of efficient training formulations for so-called “metric learning”, which to date has only received efficient formulations for restricted cases. Next, we investigate an efficient training strategy that allows the convex formulation to scale to problems of interest. Finally, we present a few case studies that demonstrate the advantages of global versus local training.

2 Background

We first consider binary association problems between two sets $\mathcal{X}$ and $\mathcal{Y}$, which could be identical or nonidentical, finite or infinite, depending on the circumstance. The three most common association problems are:

Ranking: given $x \in \mathcal{X}$, sort the elements $y \in \mathcal{Y}$ in descending order of their association with x. This is a common approach to retrieval and recommendation problems.

Prediction: given $x \in \mathcal{X}$, enumerate those $y \in \mathcal{Y}$ that are associated with x. This is a common formulation of link prediction, tagging and multi-label classification problems.

Query answering: given a query pair (x, y), indicate whether or not x and y are associated. This is a common formulation of relation learning problems.

Although other prominent forms of association problems exist, particularly those requiring a numerical response (Bennett and Lanning 2007), we focus on discrete problems in this paper. To tackle such problems we consider the standard approach of using an *association score function* $s : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$, and, when appropriate, a *decision threshold function* $t : \mathcal{X} \rightarrow \mathbb{R}$.

Ranking: given $x \in \mathcal{X}$, sort the elements of $\mathcal{Y}$ according to the scores $s(x, y_{i_1}) \geq s(x, y_{i_2}) \geq \cdots$.

Prediction: given $x \in \mathcal{X}$, enumerate the elements $y \in \mathcal{Y}$ that satisfy $s(x, y) > t(x)$.

Query answering: given (x, y) return $\text{sign}(s(x, y) - t(x))$.

Although $\mathcal{Y}$ is normally considered to be finite, which supports a simple view of ranking and prediction, it need not be: zero-shot problems consider unobserved y elements.

2.1 Co-embedding

A prominent approach to representing association score and decision threshold functions is based on *co-embedding*. The idea is to start with some initial representation of the data as feature vectors; that is, let $\boldsymbol{\phi}(x) \in \mathbb{R}^m$ denote the initial representation of $x \in \mathcal{X}$ and let $\boldsymbol{\psi}(y) \in \mathbb{R}^n$ denote the initial representation of $y \in \mathcal{Y}$.[1] Then objects from $\mathcal{X}$ and $\mathcal{Y}$ are mapped to finite dimensional vectors in a common embedding space. The simplest (and still most common) form of such map is a parametric linear map that computes the embedding $\boldsymbol{\phi}(x) \mapsto \mathbf{u}(x) \in \mathbb{R}^d$ via

$$\mathbf{u}(x) = U\boldsymbol{\phi}(x) \text{ for some } U \in \mathbb{R}^{d\times m}. \tag{1}$$

and the embedding $\boldsymbol{\psi}(y) \mapsto \mathbf{v}(y) \in \mathbb{R}^d$ via

$$\mathbf{v}(y) = V\boldsymbol{\psi}(y) \text{ for some } V \in \mathbb{R}^{d\times n}. \tag{2}$$

Given such an embedding, there are two standard models for expressing the association between x and y.

The *alignment model* uses score and threshold functions:

$$\begin{aligned} s(x,y) &= \langle \mathbf{u}(x), \mathbf{v}(y) \rangle = \boldsymbol{\phi}(x)'U'V\boldsymbol{\psi}(y) & (3)\\ t(x) &= \langle \mathbf{u}(x), \mathbf{u}_0 \rangle \quad = \boldsymbol{\phi}(x)'U'\mathbf{u}_0, & (4) \end{aligned}$$

where the threshold is based on a direct embedding $\mathbf{u}_0$ of a null object. This approach is common in many areas, including image tagging (Weston, Bengio, and Usunier 2011), multi-label classification (Guo and Schuurmans 2011), and link prediction (Bleakley, Biau, and Vert 2007).

The *distance model* uses score and threshold functions:

$$\begin{aligned} s(x,y) &= -\|\mathbf{u}(x) - \mathbf{v}(y)\|^2 = -\|U\boldsymbol{\phi}(x) - V\boldsymbol{\psi}(y)\|^2 & (5)\\ t(x) &= -\|\mathbf{u}(x) - \mathbf{u}_0\|^2 \quad = -\|U\boldsymbol{\phi}(x) - \mathbf{u}_0\|^2, & (6) \end{aligned}$$

where again the decision threshold function is usually based on a direct embedding $\mathbf{u}_0$ of a null object. This model underlies work on "metric learning" (Globerson et al. 2007; Weinberger and Saul 2009), however it has also been used in the area of multi-relation learning (Sutskever and Hinton 2008), with renewed interest (Bordes et al. 2013; 2011).

Interestingly, most work has adopted one of these two models without comparing their behavior. Some recent work in multi-relational learning has started to consider the relative capabilities of these representations (Socher et al. 2013).

2.2 Evaluating Score Functions on Data

Association models are most often learned from large data collections, where training examples come in the form of positive or negative associations between pairs of objects (x, y), sometimes called "must link" and "must not link" constraints respectively (Chopra, Hadsell, and LeCun 2005). Let E denote the set of "must link" pairs, let $\bar{E}$ denote the set of "must not link" pairs, let $S = E \cup \bar{E}$, and let E^0 denote the set of remaining pairs. That is, $E \cup \bar{E} \cup E^0$ form a partition of $\mathcal{X} \times \mathcal{Y}$. The sets E and $\bar{E}$ are presumed to be finite, although obviously E^0 need not be. For a given object $x \in \mathcal{X}$, we let $Y(x) = \{y : (x, y) \in E\}$ and $\bar{Y}(x) = \{\bar{y} : (x, \bar{y}) \in \bar{E}\}$. For sets Y, we use $|Y|$ to denote cardinality. The nature of the training set can vary between settings. For example, in link prediction and tagging, observations are often only positive "must link" pairs; whereas, in multi-label classification one often assumes that a complete set of link/no-link information over $\mathcal{Y}$ is provided for each x given in the training set (hence assuming $\mathcal{Y}$ is finite). Ranking and retrieval problems usually fall between these two extremes, with unobserved positive links primarily assumed to be negative pairs.

How such data is to be used to train the score function is determined by how one wishes to evaluate the result.

Ranking: In ranking, performance has most often been assessed by the AUC (Joachims 2002; Menon and Elkan 2011; Cortes and Mohri 2003). For a given x, the AUC of s is given by

$$\frac{1}{|Y(x)|}\frac{1}{|\bar{Y}(x)|}\sum_{y\in Y(x)}\sum_{\bar{y}\in\bar{Y}(x)} 1(s(x,y) > s(x,\bar{y})), \tag{7}$$

where $1(\xi)$ denotes the indicator function that returns 1 when ξ is true, 0 otherwise. More recently the ordered weighted average (OWA) family of ranking error functions has become preferred (Usunier, Buffoni, and Gallinari 2009). OWA generalizes AUC by allowing emphasis to be shifted to ranking errors near the top of the list, through the introduction of penalties $\boldsymbol{\alpha} \geq 0$ such that $\boldsymbol{\alpha}'\mathbf{1} = 1$ and $\alpha_1 \geq \alpha_2 \geq \cdots$. For a given x, the OWA is defined by

$$\sum_{y\in Y(x)}\sum_{\bar{y}\in\bar{Y}(x)} \alpha_{\pi(x,\bar{y})} 1(s(x,y) \leq s(x,\bar{y})), \tag{8}$$

where $\pi(x, \bar{y})$ denotes the position of $\bar{y}$ in the list sorted by $s(x, \bar{y}_1) \geq s(x, \bar{y}_2) \geq \cdots$.

Query answering: For query answering, performance is most often assessed by pointwise prediction error, given by

$$\sum_{y\in Y(x)} 1(s(x,y) \leq t(x)) + \sum_{\bar{y}\in\bar{Y}(x)} 1(s(x,\bar{y}) > t(x)). \tag{9}$$

[1] See Appendix A for a discussion of feature representations.

Prediction: There are many performance measures used to evaluate prediction performance (Sebastiani 2002; Tsoumakas, Katakis, and Vlahavas 2009). Pointwise prediction error is common, but it is known to be inappropriate in scenarios like extreme class imbalance (Joachims 2005; Menon and Elkan 2011), where it favors the trivial classifier that always predicts the most common label. Other standard performance measures are the precision, recall and F1 measure (macro or micro averaged) (Sebastiani 2002; Tsoumakas, Katakis, and Vlahavas 2009). Here we propose a useful generalization of pointwise prediction error that also provides a useful foundation for formulating later training algorithms: The idea is to introduce an OWA error measure for *prediction* instead of ranking. For a given x, this new OWA-prediction error is defined by

$$\sum_{y\in Y(x)} \alpha_{\sigma(x,y)} 1(s(x,y)\le t(x)) + \sum_{\bar{y}\in \bar{Y}(x)} \alpha_{\pi(x,\bar{y})} 1(s(x,\bar{y}) > t(x)), \qquad (10)$$

where $\sigma(x,y)$ denotes the position of y in the list sorted by $s(x,y_1)\le s(x,y_2)\le\cdots$, and $\pi(x,\bar{y})$ denotes the position of $\bar{y}$ in the list sorted by $s(x,\bar{y}_1)\ge s(x,\bar{y}_2)\ge\cdots$. The exact match error is achieved by setting $\boldsymbol{\alpha} = \mathbf{1}_1$ (i.e., all 0s except a 1 in the first position), whereas the pointwise prediction error (9) is achieved by setting $\boldsymbol{\alpha} = \mathbf{1}$.

2.3 Training Score Functions

Given a target task, a standard approach to training, arising from work on classification, is to minimize a *convex upper bound* on the performance measure of interest (Tsochantaridis et al. 2005; Joachims 2005).

For example, for *ranking*, using a convex upper bound on OWA loss has proved to provide state of the art results (Usunier, Buffoni, and Gallinari 2009; Weston, Bengio, and Usunier 2011). Using co-embedding, the training problem is

$$\min_{U,V} \sum_{x\in S}\sum_{y\in Y(x)}\sum_{\bar{y}\in\bar{Y}(x)} \alpha_{\pi(x,\bar{y})} L(s(x,y)-s(x,\bar{y})), \qquad (11)$$

where $L(s(x,y)-s(x,\bar{y})) \ge 1(s(x,y)\le s(x,\bar{y}))$ for a convex and non-increasing loss function L. Here the parameters U and V appear in the score model, either (3) or (5).

For *prediction*, recent improvements in multi-label classification and tagging have resulted from the use of so called calibrated losses (Fuernkranz et al. 2008; Guo and Schuurmans 2011). Interestingly, these losses are both convex upper bounds on (10) for different choices of $\boldsymbol{\alpha}$ (not previously realized). For example, the first approach uses $\boldsymbol{\alpha} = a\mathbf{1}$ to upper bound on pointwise error (9), while the second uses $\boldsymbol{\alpha} = \mathbf{1}_1$ to achieve an upper bound on exact match error. The resulting training problem can be formulated

$$\min_{U,V,\mathbf{u}_0} \sum_{x\in S}\sum_{y\in Y(x)} \alpha_{\sigma(x,y)} L(s(x,y)-t(x)) + \sum_{\bar{y}\in\bar{Y}(x)} \alpha_{\pi(x,\bar{y})} L(t(x)-s(x,\bar{y})), \qquad (12)$$

where $L(s(x,y)-t(x)) \ge 1(s(x,y)\le t(x))$ and $L(t(x)-s(x,\bar{y})) \ge 1(s(x,\bar{y}) > t(x))$ for a convex and non-increasing loss function L. Here the additional parameter $\mathbf{u}_0$ appears in the threshold model, either (4) or (6).

Unfortunately, even though convex loss functions are common in co-embedding approaches, they do not make the training problems (11) and (12) convex. For the alignment model (3) non-convexity arises from the bilinear interaction between U and V, whereas the nonlinearity of the distance model (5) creates non-convexity when composed with the loss. Therefore, it is currently standard practice in co-embedding to resort to local optimization algorithms with no guarantee of solution quality. The most popular choice is alternating descent in the alignment model, since the problems are convex in U given V, and vice versa. Even then, the distance model does not become convex even in single parameters, and local descent is used (Sutskever and Hinton 2008; Hinton and Paccanaro 2002).

3 Convex Relaxations

We now introduce the main formulations we use. Our goal is to first demonstrate that the previous training formulations (11) and (12) can be re-expressed in a convex form, subject to a relaxation of the implicit rank constraint. Interestingly, the convex reformulation extends to the distance based score model (5) as well as the alignment based score model (3), after an initial change of variables.

3.1 Alignment Score Model

First, for the alignment model (3) it is straightforward to observe that the score function can be re-parameterized as

$$s(x,y) \quad = \quad \boldsymbol{\phi}(x)' M \boldsymbol{\psi}(y) \qquad (13)$$

for a matrix variable $M = U'V \in \mathbb{R}^{m\times n}$. This simple change of variable allows the problems (11) and (12) to be expressed equivalently as minimization over M subject to the constraint that $\mathrm{rank}(M)\le d$. Since rank is not convex, we introduce a relaxation and replace rank with the trace norm of M.[2] Since we assumed the loss function in (11) and (12) was convex, a linear parameterization of the score function (13) coupled with replacing the rank constraint by trace norm regularization leads to a convex formulation of the training problems (11) and (12) respectively. In particular, (11) becomes minimizing the following over M

$$\sum_{x\in S}\sum_{y\in Y(x)}\sum_{\bar{y}\in\bar{Y}(x)} \alpha_{\pi(x,\bar{y})} L(s(x,y)-s(x,\bar{y})) + \lambda\|M\|_{\mathrm{tr}}, \qquad (14)$$

where we have introduced a regularization parameter λ, which allows the desired rank to be enforced by a suitable choice (Cai, Candes, and Shen 2008). Similarly, for prediction training, (12) becomes

$$\min_{M,\mathbf{m}} \sum_{x\in S}\sum_{y\in Y(x)} \alpha_{\sigma(x,y)} L(s(x,y)-t(x)) + \sum_{\bar{y}\in\bar{Y}(x)} \alpha_{\pi(x,\bar{y})} L(t(x)-s(x,\bar{y})) + \lambda\|M\|_{\mathrm{tr}}, \qquad (15)$$

which is jointly convex in the optimization variables M and $\mathbf{m} = U'\mathbf{u}_0$ using the model (3) and (4). Although these reformulations are not surprising, below we discuss how the resulting optimization problems can be solved efficiently.

[2] The trace norm is known to be the tightest convex approximation to rank, in that it is the bi-conjugate of the rank function over the spectral-norm unit sphere (Recht, Fazel, and Parrilo 2010).

3.2 Distance Score Model

It is more interesting to observe that a similar reformulation allows the *distance based* score model to also be trained with a convex minimization. The key is to achieve a linear parameterization of the distance model (5) and (6) through a reformulation and change of variables. In particular, note

$$
\begin{aligned}
-s(x,y) &= \|U\boldsymbol{\phi}(x) - V\boldsymbol{\psi}(y)\|^2 && (16)\\
&= \begin{bmatrix} \boldsymbol{\phi}(x) \\ -\boldsymbol{\psi}(y) \end{bmatrix}' \begin{bmatrix} U'U & U'V \\ V'U & V'V \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}(x) \\ -\boldsymbol{\psi}(y) \end{bmatrix} && (17)\\
&= \begin{bmatrix} \boldsymbol{\phi}(x) \\ -\boldsymbol{\psi}(y) \end{bmatrix}' M \begin{bmatrix} \boldsymbol{\phi}(x) \\ -\boldsymbol{\psi}(y) \end{bmatrix}, && (18)
\end{aligned}
$$

for a square matrix variable $M \in \mathbb{R}^{(m+n)\times(m+n)}$. Importantly, this parameterization of the distance based model is exact and linear in the parameters. The new parameter M however must satisfy a few constraints to be a valid expression of (18): namely, that $M \succeq 0$ and that $\operatorname{rank}(M) \leq d$. The first constraint is convex, but the latter is not, hence we again relax the rank constraint with trace norm regularization, as above. Interestingly, the final training formulations can be written as above, whether training for a ranking problem (14) or training for a prediction problem (15).

An important advantage that the distance based reformulation (18) holds over the alignment based reformulation (13) is that (18) allows an effective way to encode side information. For example, if prior information is available that allows one to specify linear distance constraints between elements $y \in \mathcal{Y}$, then these same constraints can be imposed on the learned embedding while maintaining convexity. In particular, let $\tilde{M}$ denote the lower right $n \times n$ block of M. If one would like to impose the constraint that object y_1 is closer to y_2 than y_3, i.e., $dist(y_1, y_2) < dist(y_1, y_3)$ (say, based on prior knowledge), then this can be directly enforced in the joint embedding submatrix $\tilde{M}$ via the linear constraint

$$
\begin{aligned}
&(\boldsymbol{\psi}(y_1) - \boldsymbol{\psi}(y_2))' \tilde{M} (\boldsymbol{\psi}(y_1) - \boldsymbol{\psi}(y_2)) && (19)\\
&< \quad (\boldsymbol{\psi}(y_1) - \boldsymbol{\psi}(y_3))' \tilde{M} (\boldsymbol{\psi}(y_1) - \boldsymbol{\psi}(y_3)) . && (20)
\end{aligned}
$$

Encoding similar information is not straightforward in the alignment representation (13) without losing convexity.

3.3 Efficient Training Algorithm

Let us write the training problem as

$$\min_M F(M) + \lambda \|M\|_{\mathrm{tr}}, \tag{21}$$

where F denotes the convex training objective of interest. Significant recent progress has been made in developing efficient algorithms for solving such problems (Dudik, Harchaoui, and Malick 2012). Early approaches were based on alternating direction methods that exploited variational representations of the trace norm via, for example

$$\|M\|_{\mathrm{tr}} = \frac{1}{2} \min_{\Omega \succeq 0} \operatorname{tr}(M'\Omega^{-1}M) + \operatorname{tr}(\Omega). \tag{22}$$

Given such a characterization, and alternating direction strategy can successively optimize M and Ω, exploiting the fact that Ω will have a closed form update (Argyriou, Evgeniou, and Pontil 2008; Grave, Obozinski, and Bach 2011). Unfortunately, such methods do not scale well to large problems because a full factorization must be computed after each iteration. Another prominent strategy has been to exploit a simple projection operator, singular value thresholding (Cai, Candes, and Shen 2008), in a proximal gradient descent algorithm (Ji and Ye 2009). Unfortunately, once again scaling is hampered by the requirement of computing a full singular value decomposition (SVD) in each iterate.

A far more scalable approach has recently been developed based on a coordinate descent. Here the idea is to keep a factored representation A and B such that $M = AB'$, where the search begins with thin A and B matrices and incrementally grows them (Dudik, Harchaoui, and Malick 2012). The benefit of this approach is that only the top singular vector pair is required on each iteration, which is a significant savings over requiring the full SVD. A useful improvement is the recent strategy of (Zhang, Yu, and Schuurmans 2012), which combines the approach of (Dudik, Harchaoui, and Malick 2012) with an earlier method of (Srebro, Rennie, and Jaakkola 2004). Here the idea is to start with thin matrices A and B as before, but locally optimize these matrices by replacing the trace norm of M with a well known identity

$$\|M\|_{\mathrm{tr}} = \min_{A,B:AB'=M} \frac{1}{2}(\|A\|_F^2 + \|B\|_F^2) \tag{23}$$

(Srebro, Rennie, and Jaakkola 2004). The key is to escape a local minimum when the local optimization terminates: here the strategy of (Dudik, Harchaoui, and Malick 2012) is used to escape by generating a column to add to A and B. In particular, to escape local minima one need only solve

$$\max_{\mathbf{a},\mathbf{b}:\|\mathbf{a}\|\leq 1, \|\mathbf{b}\|\leq 1} -\mathbf{a}'\nabla F(M)\mathbf{b} \tag{24}$$

to recover a new column $\mathbf{a}$ and $\mathbf{b}$ to add to A and B respectively, subject to a small line search

$$\min_{\mu\geq 0, \nu\geq 0} F(\mu M + \nu \mathbf{a}\mathbf{b}') + \lambda(\mu c + \nu) \tag{25}$$

for scalar μ and ν, where $c = \frac{1}{2}(\|A\|_F^2 + \|B\|_F^2)$ at the current iterate. The solution to (24), can be efficiently computed via the leading left and right singular vector pair of $-\nabla F(M)$. This method is quite effective (Zhang, Yu, and Schuurmans 2012), often requiring only a handful of outer escapes to produce an optimal M in our experiments.

4 Multi-relational Extension

Often an association problem involves additional context that determines the relationships between objects x and y. Such context can be side information, or specify which of an alternative set of relations is of interest. To accommodate this extension, it is common to introduce a third set of objects $\mathcal{Z}$. (Obviously, more sets can be introduced.)

A typical form of training data still consists of "must link" and "must not link" tuples $(x, y, z) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$. (The problem is now implicitly a hyper-graph.) Let E denote the set of positive "must link" tuples, let $\bar{E}$ denote the set of negative "must not link" tuples, let $S = E \cup \bar{E}$, and let E^0 denote the set of remaining tuples. The sets E and $\bar{E}$ are presumed to be finite. For a given pair (x, z), we let $Y(x, z) = \{y : (x, y, z) \in E\}$ and $\bar{Y}(x, z) = \{\bar{y} : (x, \bar{y}, z) \in \bar{E}\}$.

Such an extension can easily be handled in the framework of score functions. In particular, one can extend the concept of an association score to now hold between three objects via $s(x, y, z)$. Standard problems can still be posed.

Ranking: given (x, z) sort the elements of $\mathcal{Y}$ according to the scores $s(x, y_{i_1}, z) \geq s(x, y_{i_2}, z) \geq \cdots$.

Prediction: given (x, z) enumerate $y \in \mathcal{Y}$ that satisfy $s(x, y, z) > t(x, z)$ for a threshold function $t(x, z)$.

Query answering: given (x, y, z) return $\text{sign}(s(x, y, z) - t(x, z))$.

The embedding framework can be extended to handle such additional objects by also mapping z to a latent representation from an initial feature representation $\boldsymbol{\xi}(z) \in \mathbb{R}^p$.

Alignment Score Model The linear alignment based model (13) can be easily extended by expanding the matrix M to a three-way tensor T, allowing a general alignment score function to be expressed

$$s(x, y, z) \;\; = \;\; \sum_{ijk} T_{ijk} \boldsymbol{\phi}(x)_i \boldsymbol{\psi}(y)_j \boldsymbol{\xi}(z)_k \qquad (26)$$

which is still linear in the parameter tensor T. Such a parameterization will maintain convexity of the previous formulations. However, tensor variables introduce two problems in the context of co-embedding. First, there is no longer a simple notion of rank, nor a simple convex regularization strategy that can effectively approximate rank. Second, the tensor variable can become quite large if the initial feature dimensions m, n and p are large. Some current work ignores this issue and uses a full tensor (Socher et al. 2013; Jenatton et al. 2012), but others have found success by working with compressed representations (Nickel, Tresp, and Kriegel 2011; Gantner et al. 2010; Rendle and Schmidt-Thieme 2009). Below we will consider a compact linear representation used by (Rendle and Schmidt-Thieme 2009), which decomposes T into the repeated sum of two base matrices N and P, such that $T_{ijk} = N_{ij} + P_{kj}$. Convex co-embedding can be recovered with such a representation, but controlling the rank of N and P through trace norm regularization.

Distance Score Model Beginning with the work of (Hinton and Paccanaro 2002; Sutskever and Hinton 2008), extensions of the distance based score model (5) have been a popular approach to multi-relational learning. Several recent works have adopted similar forms of score models, including (Bordes et al. 2013; 2011). Unfortunately, none of these representations admit an equivalent linear tensor form, and we leave the exploration of alternative representations for multi-relational distance models for future work.

5 Case Study: Multi-label Prediction

To investigate the efficacy of convex embedding, we conducted an initial experiment on multi-label classification with the multi-label data sets shown in Table 1. In each case, we used 1000 examples for training and the rest for testing (except *Emotion* where we used a $\frac{2}{3}, \frac{1}{3}$ train-test split), repeating 10 times for different random splits. In particular, we used the alignment score model (13) and a smoothed version (28) of the large margin multi-label loss (27), which has given state of the art results (Guo and Schuurmans 2011):

$$\sum_{x\in S} \max_{y\in Y(x)} L(m(x,y)) + \max_{\bar{y}\in \bar{Y}(x)} L(\bar{m}(x,\bar{y})) \qquad (27)$$

$$\leq \sum_{x\in S} \operatorname*{softmax}_{y\in Y(x)} \tilde{L}(m(x,y)) + \operatorname*{softmax}_{\bar{y}\in \bar{Y}(x)} \tilde{L}(\bar{m}(x,\bar{y})), (28)$$

where $m(x,y) = s(x,y) - t(x)$; $\bar{m}(x,\bar{y}) = t(x) - s(x,\bar{y})$; $L(m) = (1-m)_+$; $\tilde{L}(m) = \frac{1}{4}(2-m)_+^2$ if $0 \leq m \leq 2$, $(1-m)_+$ otherwise; and $\text{softmax}_{y\in Y} f(y) = \ln \sum_{y\in Y} \exp(f(y))$. (Note that (27) follows from the loss in (15) using $\boldsymbol{\alpha} = \mathbf{1}_1$.) We also added a squared Frobenius norm regularizer on M with the same weight λ.

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

Table 1: Data sets properties for multi-label experiments.

	Corel	Emot.	Media.	Scene	Yeast
CVX time	6.0s	0.3s	10.6s	3.4s	3.6s
ALT time	9.2s	3.0s	497.6s	19.5s	8.0s
CVX obj	4014	1060	3996	2593	3635
ALT obj	4014	1060	3996	2593	3635
ALT0 obj	4022	1077	4126	2603	3637
CVX err	7%	29%	11%	18%	46%
ALT err	7%	29%	11%	18%	46%
ALT0 err	7%	31%	14%	18%	51%
λ	0.3	0.45	0.2	3.0	1.0
CVX rank	19	4	3	4	3

Table 2: Multi-label results averaged over 10 splits: time in seconds; average objective value over 100 random initializations (ALT0 indicates initializing from 0); pointwise test error; regularization parameter and rank of CVX solution.

The aim of this study is to compare the global training method developed above (CVX), which uses a convex parameterization (M and $\mathbf{m}$), against a conventional alternating descent strategy (ALT) that uses the standard factored parameterization ($U'V = M$ and $U'\mathbf{u}_0 = \mathbf{m}$). To ensure a fair comparison, we first run the global method to extract the rank of M, then fixed the dimensions of U and V to match.

The results of this experiment, given in Table 2, are surprising in two respects. First, under random initialization, we found that the local optimizer, ALT, achieves the global objective in all the data splits on all data sets in this setting. Consequently, the same training objectives and test errors were observed for both global and local training. Evidently there are no local minima in the problem formulation (15) using loss (28) with squared Frobenius norm regularization, even when using the factored parameterization $U'V = M$

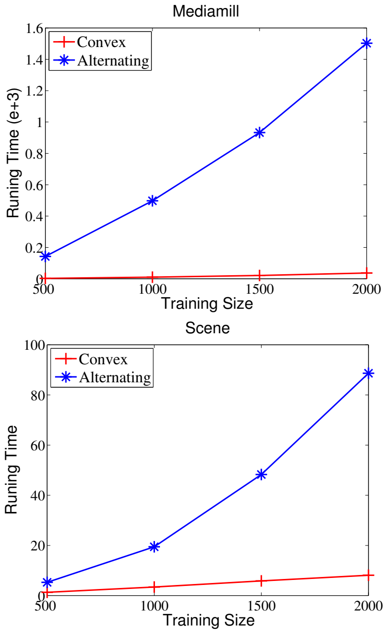

Figure 1: Multi-label run time comparison (seconds).

and $U'\mathbf{u}_0 = \mathbf{m}$ (although we have no proof to support such a claim). An additional investigation reveals that there are non-optimal critical points in the local objective, as shown by initializing ALT with all zeros; see Table 2.

The second outcome is that the global method can be significantly faster than alternating minimization. To further investigate the scaling properties of these methods we conducted further experiments on *Mediamill* and *Scene* using increasing training sizes. The results, given in Figure 1, show that CVX is increasingly more efficient than ALT. Despite its convenience, ALT can be an expensive algorithm in practice. Overall, these results suggest that convex minimization can be a more efficient alternative in many problems where alternating descent remains prominent.

6 Case Study: Tag Recommendation

Next, we undertook a study on a multi-relational problem: solving Task 2 of the 2009 ECML/PKDD Discovery Challenge. This problem considers three sets of entities—users, items, and tags—where each user has labeled a subset of the items with relevant tags. The goal is to predict the tags the users will assign to other items. Here we let $\mathcal{X}$ denote the set of users, $\mathcal{Z}$ the set of items, and $\mathcal{Y}$ the set of tags respectively; and used the feature representations $\boldsymbol{\phi}(x) = \mathbf{1}_x$, $\boldsymbol{\psi}(y) = \mathbf{1}_y$ and $\boldsymbol{\xi}(z) = \mathbf{1}_z$ in the tensor model (26).[3] The training examples are provided in a data tensor E, such that $E(x, y, z) = 1$ indicates that tag y is among the tags user x has assigned to item z; $E(x, y, z) = -1$ indicates that tag y is not among those user x assigned to item z; and $E(x, y, z) = 0$ denotes an unknown element. The goal is to predict unknown values subject to a constraint that at most five tags can be active for any (user, item) pair.

The winner of this challenge (Rendle and Schmidt-Thieme 2009) used a co-embedding model in the non-convex form outlined above, hence they only considered local training. Here, we investigate whether a convex formulation can improve on such an approach, using the Challenge data provided by BibSonomy. Following (Jäschke et al. 2008) we exploit the *core at level* 10 subsample, which reduces the data set to 109 unique users, 192 unique items and 229 unique tags.

Prediction Following (Rendle and Schmidt-Thieme 2009), we rank the tags that each user assigns to an item. Given a learned score function s, the top five tags y are predicted from a given user-item pair (x, z) via

$$\hat{E}(x, y, z) = \begin{cases} 1 & \text{if } s(x, y, z) \text{ in top 5 values of } s(x, :, z) \\ -1 & \text{otherwise.} \end{cases}$$

Experimental Settings We parameterize the tensor with the pairwise interaction model (Rendle and Schmidt-Thieme 2010; Chen et al. 2013), which uses the decomposition

$$s(x, y, z) \quad = \quad T_{xyz} \quad = \quad N_{x,y} + P_{z,y} \quad \forall x, y, z. \quad (29)$$

Following (Rendle and Schmidt-Thieme 2009), we use the ranking logistic loss function for learning N and P in the formulation (14), but replace their low rank assumptions on N and P with a trace norm relaxation

$$Reg(N, P) \quad = \quad \lambda_1 \|N\|_{tr} + \lambda_2 \|P\|_{tr}. \qquad (30)$$

We also include a Frobenius norm regularizer on N and P, following (Rendle and Schmidt-Thieme 2009).

The aim of this study is, again, to compare the global training method developed above (CVX), which uses the convex parameterization (N and P), against a conventional alternating descent strategy (ALT) that uses a factored parameterization ($U'V = N$ and $Q'R = P$). Below, we apply a common regularization parameter $\lambda = \lambda_1 = \lambda_2$ to the trace and squared Frobenius norm regularizers, and consider the rank returned by CVX as well as the hard rank choices $d \in \{32, 64, 128, 256\}$.

Experimental Results The results of this study are shown in Table 3 below. The first four columns report the settings used: the training method, the shared regularization parameter λ, the rank of N (d_1), and the rank of P (d_2). The final three columns report the outcomes: the final objective value obtained, the value of the per-instance averaged $F1$ measure on the test data (which is the evaluation criterion of the Discovery Challenge), and the training time (in minutes).

The table is also organized into four vertical blocks. The top block provides a controlled comparison between the global training method developed in this paper, CVX, and alternating minimization, ALT. In this block, the global method is first trained using the fixed regularization parameter λ, after which the rank of its solutions are recovered, $d_1 = \mathrm{rank}(N)$ and $d_2 = \mathrm{rank}(P)$. These are then used to determine the dimensions of the matrices $U'V = N$ and

[3] Such indicator representations are discussed in Appendix A.

Method	λ	d_1	d_2	objective	$F1$	time
CVX	10	59	73	42	**0.42**	41
ALT	10	59	73	42	**0.42**	980
ALT0	10	59	73	1402	0.08	6
ALT1	10	59	73	150	0.32	880
ALT	1e-4	32	32	3.5	0.32	582
ALT	1e-4	64	64	3.5	0.34	597
ALT	1e-4	128	128	3.5	0.36	627
ALT	1e-4	256	256	3.5	0.36	669
ALT	5e-5	32	32	3.5	0.33	589
ALT	5e-5	64	64	3.5	0.32	594
ALT	5e-5	128	128	3.5	0.34	619
ALT	5e-5	256	256	3.5	0.34	690
ALT	0	32	32	3.5	0.32	583
ALT	0	64	64	3.5	0.33	593
ALT	0	128	128	3.5	0.33	634
ALT	0	256	256	3.5	0.31	688

Table 3: Tag recommendation results. All methods were initialized randomly, except ALT0 indicates initializing from all 0s, and ALT1 indicates initializing from all 1s.

$Q'R = P$ used by ALT. The second and third block show the results for ALT using the fixed parameter values (d_1, d_2 and λ) that were used in the award winning approach of (Rendle and Schmidt-Thieme 2009). Finally, the fourth block, shows the results for ALT without any regularization, but imposing only rank constraints.

There are a number of interesting conclusions one can draw from these results. First, it can be seen that both CVX and ALT with the parameter values shown in the top block achieve the best $F1$ value among all methods, even surpassing the result quality of the award winning parameterization on this data set.

More interestingly, we see that, once again, ALT with random initialization achieves the same result as CVX when controlling for rank and regularization, albeit with significantly greater computational cost. This result suggests that the introduction of trace norm regularization has somehow eliminated local minima from the problem once again, although we have no proof to support such a conclusion. Indeed, by initializing ALT with all 0s or all 1s one can see again that convergence to non-optimal critical points is obtained; such points are avoided by CVX.

7 Conclusion

We have investigated a general approach to co-embedding that unifies alignment based and distance based score models. Based on this unification, we provided a general convex formulation of both models by replacing the intractable rank constraint with a trace norm regularization. To achieve scalable training for these models, we adopted a recent hybrid training strategy that combines an outer "boosting" loop with inner smooth optimization. The resulting training procedure is more efficient than alternating descent while yielding global instead of local solutions. In one case study (tag recommendation), the result was improved generalization performance over current state of the art local training methods, whereas in the other case study (multi-label prediction) we observed that global optimization achieves the local result, but with significant time savings over naive alternation.

There are many directions for future work. One direction is to investigate other linear compressions of tensor representations that allow greater freedom to trade off space versus expressiveness. Another direction is to investigate alternative, tighter approximations of rank when the target dimensionality is pre-specified. Finally, the possibility for exploiting the distance model to express prior side information on y targets via tractable distance constraints in the embedding space has yet to be properly explored.

A Additional Discussion on Features

Note that co-embedding presumes that the objects $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ were all assigned initial feature representations, $\boldsymbol{\phi}(x) \in \mathbb{R}^m$ and $\boldsymbol{\psi}(y) \in \mathbb{R}^n$. The nature of these initial representations play an important role in determining what generalizations can or cannot be easily captured.

One extreme but common form is a simple indicator $\boldsymbol{\phi}(x) = \mathbf{1}_x$ where $\mathbf{1}_x$ is (conceptually) a vector of all zeros with a single 1 in the position corresponding to $x \in \mathcal{X}$. Such a representation explicitly enumerates $\mathcal{X}$, presuming it is finite. Although indicators have many obvious shortcomings, they remain common. For example, work on community identification from link structure (Newman 2010) is based on indicators. Similarly, most work on multi-label prediction uses label indicators $\mathbf{1}_y$ when no prior information is encoded between labels y. Indicators do not provide information that supports direct generalization between objects, nor do they support out of sample prediction. Note that the embeddings $\mathbf{v}(y) = V\mathbf{1}_y = V_{:y}$ assign a separate embedding vector $V_{:y}$ to y independently of the other elements of $\mathcal{Y}$, which can be onerous to store if the sets are large.

Recently, there has been renewed interest in endowing objects with meaningful *property based* features, or "attributes" in recent computer vision research (Akata et al. 2013; Farhadi et al. 2009), link prediction (Bleakley, Biau, and Vert 2007; Menon and Elkan 2011) and recommendation (Gantner et al. 2010). Attributes allow generalization between objects that is based on prior knowledge, even if an object has not been seen in the training data. In the framework of co-embedding, this is particularly intuitive: a new object, say y, that has not been seen during training can still be embedded in the latent space. If y's feature representation $\boldsymbol{\psi}(y)$ is similar to other objects from $\mathcal{Y}$ seen in the training data, then y's embedding $\mathbf{v}(y) = V\boldsymbol{\psi}(y)$ should also be similar, hence y will exhibit similar scores $s(x, y)$ from x. This allows the prospect of zero-shot learning where one can predict x's association with a target label y not seen during training. Similarly, an attribute based feature representation $\boldsymbol{\phi}(x)$ allows out of sample prediction for objects x not seen during training; a standard goal in supervised learning.

These same points continue to apply in the multi-relational setting. For example, if one wishes for meaningful generalizations between contexts z, or for zero-shot transfer to novel z contexts, then here too it is imperative that the initial feature representation $\boldsymbol{\xi}(z)$ be property based.

References

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

Argyriou, A.; Evgeniou, T.; and Pontil, M. 2008. Convex multi-task feature learning. *Machine Learning* 73(3):243–272.

Bennett, J., and Lanning, S. 2007. The Netflix Prize. In *Proceedings of KDD Cup and Workshop 2007*.

Bleakley, K.; Biau, G.; and Vert, J. 2007. Supervised reconstruction of biological networks with local models. In *International Conference on Intelligent Systems for Molecular Biology (ISMB)*.

Bordes, A.; Weston, J.; Collobert, R.; and Bengio, Y. 2011. Learning structured embeddings of knowledge bases. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *Neural Information Processing Systems (NIPS)*.

Cai, J.; Candes, E.; and Shen, Z. 2008. A singular value thresholding algorithm for matrix completion. *SIAM J Optim* 20:1956–1982.

Chen, S.; Lyu, M. R.; King, I.; and Xu, Z. 2013. Exact and stable recovery of pairwise interaction tensors. In *Advances in Neural Information Processing Systems (NIPS)*.

Chopra, S.; Hadsell, R.; and LeCun, Y. 2005. Learning a similarity metric discriminatively, with application to face verification. In *IEEE Conf. Computer Vision and Patt. Recogn. (CVPR)*, 539–546.

Cortes, C., and Mohri, M. 2003. AUC optimization vs. error rate minimization. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*.

Dudik, M.; Harchaoui, Z.; and Malick, J. 2012. Lifted coordinate descent for learning with trace-norm regularization. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Farhadi, A.; Endres, I.; Hoiem, D.; and Forsyth, D. 2009. Describing objects by their attributes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Fuernkranz, J.; Huellermeier, E.; Mencia, E.; and Brinker, K. 2008. Multilabel classification via calibrated label ranking. *Machine Learning* 73(2).

Gantner, Z.; Drumond, L.; Freudenthaler, C.; Rendle, S.; and Schmidt-Thieme, L. 2010. Learning attribute-to-feature mappings for cold-start recommendations. In *Proceedings of the IEEE Conference on Data Mining (ICDM)*, 176–185.

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

Grave, E.; Obozinski, G.; and Bach, F. 2011. Trace lasso: a trace norm regularization for correlated designs. In *Advances in Neural Information Processing Systems (NIPS)*.

Guo, Y., and Schuurmans, D. 2011. Adaptive large margin training for multilabel classification. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Hinton, G., and Paccanaro, A. 2002. Learning hierarchical structures with linear relational embedding. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*.

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

Jenatton, R.; Roux, N. L.; Bordes, A.; and Obozinski, G. 2012. A latent factor model for highly multi-relational data. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*.

Ji, S., and Ye, J. 2009. An accelerated gradient method for trace norm minimization. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Joachims, T. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the ACM SIGKDD Conferece on Knowledge Discovery and Data Mining*.

Joachims, T. 2005. A support vector method for multivariate performance measures. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Menon, A., and Elkan, C. 2011. Link prediction via matrix factorization. In *European Conference on Machine Learning (ECML)*.

Newman, M. 2010. *Networks: An Introduction*. Oxford.

Nickel, M.; Tresp, V.; and Kriegel, H. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Recht, B.; Fazel, M.; and Parrilo, P. 2010. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review* 52:471–501.

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

Rendle, S., and Schmidt-Thieme, L. 2010. Pairwise interaction tensor factorization for personalized tag recommendation. In *International Conference on Web Search and Data Mining*, 81–90.

Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*.

Sebastiani, F. 2002. Machine learning in automated text categorization. *ACM Computing Surveys* 34(1):1–47.

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

Srebro, N.; Rennie, J.; and Jaakkola, T. 2004. Large-margin matrix factorization. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*.

Sutskever, I., and Hinton, G. 2008. Using matrices to model symbolic relationship. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*.

Tsochantaridis, I.; Hofmann, T.; Joachims, T.; and Altun, Y. 2005. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research* 6:1453–1484.

Tsoumakas, G.; Katakis, I.; and Vlahavas, I. 2009. Mining multi-label data. In *Data Mining and Knowledge Discovery Handbook, 2nd edition. Springer*.

Usunier, N.; Buffoni, D.; and Gallinari, P. 2009. Ranking with ordered weighted pairwise classification. In *Proceedings of the International Conference on Machine Learning (ICML)*.

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

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

Weston, J.; Bengio, S.; and Usunier, N. 2011. WSABIE: Scaling up to large vocabulary image annotation. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

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