# Solving Association Problems with Convex Co-embedding

by

Farzaneh Mirzazadeh

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Computing Science

University of Alberta

© Farzaneh Mirzazadeh, 2017

# Abstract

Co-embedding is the process of mapping elements from multiple sets into a common latent space, which can be exploited to infer element-wise associations by considering the geometric proximity of their embeddings. Such an approach underlies the state of the art for link prediction, relation learning, multi-label tagging, relevance retrieval and ranking. This dissertation provides contributions to the study of co-embedding for solving association problems.

First, a unifying view for solving association problems with co-embedding is presented, which covers both alignment-based and distance-based models. Although current approaches rely on local training methods applied to non-convex formulations, I demonstrate how general convex formulations can be achieved for co-embedding. I then empirically compare convex versus non-convex formulations of the training problem under an alignment model. Surprisingly, the empirical results reveal that, in most cases, the two are equivalent.

Second, the connection between metric learning and co-embedding is investigated. I show that heterogeneous metric learning can be cast as distance-based co-embedding, and propose a scalable algorithm for solving the training problem globally. The co-embedding framework allows metric learning to be applied to a wide range of association problems—including link prediction, relation learning, multi-label tagging and ranking. I investigate the relation between the standard non-convex training formulation and the proposed convex reformulation of heterogeneous metric learning, both empirically and analytically. Again, it is discovered that under certain conditions, the objective values achieved by the two approaches are identical. I develop a formal characterization of the conditions under which this equality holds.

Finally, a constrained form of co-embedding is proposed for structured output prediction. A key bottleneck in structured output prediction is the need for inference during training and testing, usually requiring some form of dynamic programming. Rather than using approximate inference or tailoring a specialized inference method for a particular structure I instead pre-compile prediction constraints directly into the learned representation. By eliminating the need for explicit inference a more scalable approach to structured output prediction can be achieved, particularly at test time. I demonstrate the idea for hierarchical multi-label prediction under subsumption and mutual exclusion constraints, where a relationship to maximum margin structured output prediction can be established. Experiments demonstrate that the benefits of structured output training can still be realized even after inference has been eliminated.

*To my beloved parents.*

# Acknowledgment

First of all, I am deeply grateful to my advisor, Professor Dale Schuurmans, for all I learned from him or because of him, about machine learning, good research, life and myself. It has been a great privilege to conduct research and complete apprenticeship under the supervision of both a highly masterful, knowledgeable, and enthusiastic researcher, and a very smart, influential and kind person. His insightful feedback often made the struggles of a PhD all worth it. I very much appreciated his willingness to listen to his students and understand them. In particular, I am very thankful to all the time he devoted to training me in our meetings, reviewing my proposals and improving my presentations and documents. I will definitely miss those advisory meetings.

I should also thank Professor Russell Greiner, my co-advisor, for all his support and for always being kind to me. Among all other things, he gave me an exceptional leadership opportunity to guide younger students and interns of his group, which I greatly benefited from. I would also like to thank him for the warm atmosphere he creates in his research group that gives the students the sense of belonging.

Next, I would like to greatly thank my wonderful PhD examining committee: Dr. Richard Zemel of the University of Toronto, Dr. Michael Bowling, Dr. Csaba Szepesvari, and Dr. Joerg Sander for their insightful comments, and for carefully reviewing my thesis and providing thorough, valuable, and inspiring feedback. My candidacy exam and my defence sessions meant a lot to me.

In addition to my advisor, I would like to thank NSERC, Alberta Innovates, the Government of Alberta, the University of Alberta, and the AICML for funding my PhD. This research would not have been possible without their generous support.

Additionally, I would like to thank Dr. Guohui Lin, who had a highly positive

role in the first two years of my PhD by giving me exceptional academic freedom, supporting my scholarship applications, financially supporting my research for two summers, and finally, facilitating my transfer to the machine learning group by recommending me to Dr. Schuurmans.

I am very grateful for our collaborations with Dr. Martha White and Dr. András György. I learned a lot from our discussions. Thanks to both of them for expressing interest to the project both at the beginning and throughout.

I would also like to thank my colleagues and friends in machine learning group, specially Ken Dwyer, Leah Hackman, and Martha White. In particular, Ken helped me with my NSERC application, Leah with several of my presentations, and Martha by agreeing to answer my machine learning questions when I was a complete beginner. Whenever I approached Martha with my questions, she helped me with open arms.

I appreciate the support of Samaneh Eskandari, Homa Foroughi, and Hosna Jabbari, specially in the last year of my PhD, when I was working on my thesis remotely. They did many favors to me in the town and campus. Without their assistance, the last year of my PhD could not have passed so smoothly.

A special thanks to my family! In particular, I am deeply grateful to my parents for their great support, kindness and for their patience with me and their other children, in the years that we have not been around. My mom was a great long-distance support in the ups and downs of my PhD, and my dad had a significant role in encouraging me to start my PhD, at the beginning and to wrap it up, at the end, both of which I really appreciate. I hope I can go back to them and make up for my long absence.

Lastly, a very special thanks to my husband, Babak, for all his support, for the cheerful moments he made for us in these years and for remaining patient with my strong engagement with my PhD. At the times when I doubted the path, he was the one who did not let me give up. Thank you Babak!

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

Many machine learning sub-communities have converged on a common approach of *co-embedding* to tackle machine learning problems. The idea is to first embed elements from multiple sets into a common low dimensional Euclidean space and then use Euclidean geometry to infer associations between elements. The target problems that can be addressed by such an approach are diverse, ranging from link prediction (Yamanishi, 2008) to question answering (Bordes et al., 2014).

In this thesis, I focus on studying the power of co-embedding to solve such *association problems*, where elements from a number of sets are associated with one another. Examples include ranking, multilabel tagging, multiclass classification, and link prediction. A dominant and common approach for solving this class of problems is to learn an intermediate *association score* given target association information. For example, multiclass training of Crammer and Singer (2001) attempts to learn intermediate association score functions so that the true label of each training example receives higher association with that example (by a margin) than the other labels. Similarly, multilabel prediction (Fürnkranz et al., 2008) involves learning an intermediate association score as well as threshold scores so that labels with association scores above the threshold coincide with the correct labels for a given example. Finally, for label ranking (Hüllermeier et al., 2008), one attempts to learn an association function that gives similar rankings on labels as that of the true labels. Not surprisingly, to be useful, the learned models should also generalize well on unseen data.

Despite the varied history of association problems, co-embedding approaches

solve these problems in a unique fashion by providing a geometric basis for defining association scores. In particular, one can exploit the *alignment* (i.e., inner product) between embedding vectors to determine the association strength; or, alternatively, the *Euclidean distance* between embedding vectors can be used for this purpose. Yet other distance models can be considered based on the $L_1$ norm (Chopra et al., 2005; Bordes et al., 2011; Wang et al., 2014) or the $L_\infty$ norm. Co-embedding approaches now provide the state of the art in a wide range of applications, achieving improved association quality. Furthermore, co-embedding can provide additional insight by revealing relationships between items in a common space (Globerson et al., 2007). Such approaches can also be extended to a *multi-relational* setting by considering items from more than two sets, for example in query adaptive item recommendation, where users, queries and items are associated (Weston et al., 2012). Remarkably, by sharing feature representations between target items, co-embedding also offers a natural approach to *zero shot* learning, where assignments to previously unseen labels are queried at test time (Li et al., 2003; Palatucci et al., 2009).

Distance based co-embedding is closely related to *metric learning* (Xing et al., 2002; Kulis, 2013), where a distance function between data instances is learned to help simplify a target task. The case where data instances belong to different sets is referred to as *heterogeneous metric learning* (Zhai et al., 2013). It is notable that the application of metric learning to co-embedding expands the range of problems that can be addressed by metric learning to all association problems. Moreover, efficient formulations and fast computational strategies for co-embedding directly lead to advances in heterogeneous metric learning, which to date has only received efficient formulations for restricted cases.

Co-embedding is not only a powerful approach for tackling standard association problems, but also for solving *structured association problems* over sets that have additional special structure. For example, suppose we wish to annotate images with tags “animal”, “flower”, “cat”, etc., where any number of tags can be assigned to each image, but the tags also form a hierarchy such that any “cat” is an “animal” while nothing can be both an “animal” and a “flower”. Approaching this problem from the perspective of distance based co-embedding, one would embed both the

images and tags into a joint latent space, where images that fit the description of a tag category are embedded nearby. However, to enforce the hierarchy and mutual exclusion constraints, the Euclidean geometry of embeddings can also be exploited further. Let the decision region for each tag be modeled with a Euclidean ball centered at the embedding point of the tag, so that any image is tagged as, say, "cat" if and only if it is embedded inside the Euclidean ball corresponding to the "cat" tag. Then enforcing hierarchical structure becomes straightforward: By requiring that first, *the Euclidean ball corresponding to the "cat" object must lie inside the Euclidean ball corresponding to the "animal" object*, and second, *the Euclidean balls corresponding to "animal" and "flower" objects must be disjoint*, one can guarantee that the hierarchical and mutual exclusion constraints are enforced.

Structured association problems are usually tackled by *structured output prediction* methods (Taskar, 2004; Tsochantaridis et al., 2005), which require explicit inference to be performed over joint label predictions, usually in the form of a dynamic program. However, rather than use approximate inference or tailor a specialized inference method for a particular structure, a co-embedding approach makes it possible to insert prediction constraints directly into the learned representation. By eliminating the need for explicit inference, particularly at test time, a more scalable approach to structured output prediction can be achieved. This property is essential for time-sensitive user-facing applications.

Despite their success, current co-embedding methods do have some 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 dissertation, I first offer a unified view of co-embedding by presenting a simple framework that expresses association problems in a common format. Within this general framework, I then demonstrate how a convex training formulation can

be achieved by relaxing the low rank constraint on the embeddings. Importantly, the proposed reformulation can be applied to both alignment based and distance based score models. In the experimental analysis, I evaluate a global training algorithm in different case studies, where it is observed that the training objective values achieved by local and global solvers are often identical.

Next, I further investigate the topic of co-embedding within the framework of convex heterogeneous metric learning. Effective training algorithms in this case require an efficient approach to imposing a semidefinite matrix constraint. For this purpose, I propose a particular algorithmic strategy that is both scalable and correct, providing a proof of convergence to a globally optimal solution. In addition, I empirically investigate the relation between the solutions provided by local and global solvers.

Finally, I tackle the association problem of structured multilabel prediction under implication and mutual exclusion constraints. The main result is to demonstrate how inference can be eliminated from structured output prediction by imposing convex constraints on the learned representations that encode the prior knowledge about the label relationships. That is, the intuition underlying the aforementioned image tagging example (with "cats", "animals", and "flowers") is developed in a principled way to guarantee consistency of label assignments with simple logical constraints. The outcome is a useful model in which a relationship to maximum margin structured output prediction can be established. Experiments demonstrate that the benefits of structured output training can still be realized, even after inference has been eliminated.

## 1.1 Contributions

Key contributions of this dissertation are the following.

- A unified view of association problems is developed that includes link prediction in graphs, multilabel classification, ranking, and applications including knowledge graph completion, image tagging, question answering, and recommendation.

- A unified view of co-embedding approaches to association problems is developed that encompasses both distance based and alignment based co-embedding.
- A tractable training procedure for alignment based co-embedding methods is developed by
  - formulating a convex training problem,
  - identifying a scalable training algorithm,
  - making an interesting empirical observation about the relation between the locally and globally optimal trained models.
- A further understanding of distance based co-embedding, also known as metric learning, is achieved by
  - offering a way to solve association problems via metric learning by relating distance based co-embedding methods to metric learning,
  - establishing a convex training formulation,
  - developing a scalable training algorithm with a proof of convergence,
  - formally characterizing the conditions under which local training methods applied to the standard non-convex formulation are equivalent to the proposed convex reformulation of heterogeneous metric learning.
- An important new observation is made that inference can be completely eliminated from structured multi-label classification by embedding the logical relationships between labels directly into the score model.
- A concrete demonstration of this structure pre-compilation idea is provided for multilabel prediction models, where it is shown that implication and mutual exclusion relationships can be easily embedded in the score model while maintaining convexity in model parameters.
- A novel convex approach to structured multi-label prediction is proposed.

## 1.2 Publication Notes

This research has been published in three peer reviewed publications. The material in Chapter 3 was presented at the Twenty-Eighth Annual Conference on Artificial Intelligence (AAAI) in 2014 (Mirzazadeh et al., 2014); the content of Chapter 4 was published presented at the European Conference on Machine Learning (ECML) in 2015 (Mirzazadeh et al., 2015b); and the results of Chapter 5 were presented in Neural Information Processing Systems (NIPS) in 2015 (Mirzazadeh et al., 2015a).

# Chapter 2

# Background

Euclidean co-embedding considers the simple, but effective, approach of mapping items from multiple sets into a common low dimensional Euclidean space. Once so embedded, simple Euclidean geometry can be used to solve many types of association problems, as illustrated in Figure 2.1. Co-embedding is sometimes referred to as *joint embedding* (Bengio and Weston, 2011) or *semantic embedding* (Norouzi et al., 2013) and underlies many useful formulations in machine learning. For example, Yamanishi (2008) embeds nodes of a heterogeneous graph to support link prediction, Bordes et al. (2014) use co-embedding of questions and answers to rank appropriate answers to a query for retrieval and recommendation, and Rendle et al. (2009) embed users, items, and tags for user-specific tag recommendation.

In the following, we formally define co-embedding, but for clarity focus on the case where two sets are embedded into a common space. The extension to more than two sets is straightforward.

## 2.1 Definition

The process of co-embedding begins with an initial representation of data given as feature vectors; that is, we 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 $y \in \mathcal{Y}$. Then objects from $\mathcal{X}$ and $\mathcal{Y}$ are mapped into finite dimensional vectors in a common embedding space using a function of their feature representations. The simplest and still most common form of such mapping is a *parametric linear map* that computes the

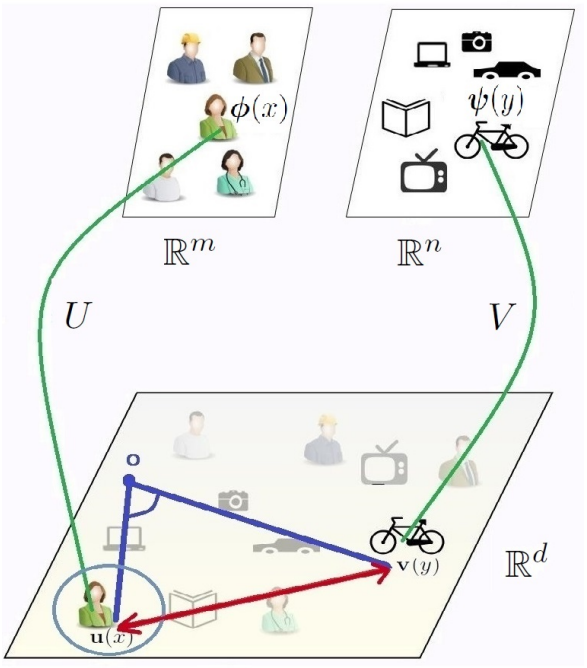


Figure 2.1: An illustration of co-embedding. Here a user $x$ with features $\phi(x) \in \mathbb{R}^m$ and an item $y$ with features $\psi(y) \in \mathbb{R}^n$ are mapped into a single $\mathbb{R}^d$ space.

embedding $\boldsymbol{\phi}(x) \mapsto \mathbf{u}(x) \in \mathbb{R}^d$ via

$$\mathbf{u}(x) = U\boldsymbol{\phi}(x) \text{ for some } U \in \mathbb{R}^{d \times m}, \tag{2.1}$$

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

$$\mathbf{v}(y) = V\boldsymbol{\psi}(y) \text{ for some } V \in \mathbb{R}^{d \times n}, \tag{2.2}$$

where $\mathbf{u}(x)$ and $\mathbf{v}(y)$ are embedding vectors of objects $x$ and $y$ respectively, and $U$ and $V$ are parameters; see Figure 2.1.

Although co-embedding can be viewed as a stand-alone preprocessing procedure that optimizes a structure preserving objective function independent of the end task—for example by minimizing the distance distortion—such a task independent approach is suboptimal. Instead, to obtain a useful task-specific embedding without substantial manual design, the co-embedding parameters $U$ and $V$ are normally optimized to minimize a task specific objective. We will discuss this in more detail in Section 3.1 of Chapter 3, where we show how co-embedding can be used to solve association problems.

## 2.2 Initial Representation

The nature of the initial representations, $\boldsymbol{\phi}(x)$ and $\boldsymbol{\psi}(y)$, play a major role in determining what generalizations can or cannot be easily captured. Recently, exploiting rich features learned via *deep neural networks* has become a popular approach for learning such representations. For instance, Frome et al. (2013), for the purpose of visual-semantic co-embedding, employ the deep convolutional neural network of Krizhevsky et al. (2012) to obtain initial visual features. Also, a well-known two-layer architecture, the skip-gram model (Word2Vec) (Mikolov et al., 2013), has been used to obtain the initial features representations for language modeling.

Another particularly simple form of initial representation is the indicator vector

$$\boldsymbol{\phi}(x) \;=\; \mathbf{1}_x, \tag{2.3}$$

where $\mathbf{1}_x$ is, conceptually, a vector of all zeros except for a single 1 in the position corresponding to $x \in \mathcal{X}$. Such a representation explicitly enumerates elements of a finite set $\mathcal{X}$. Although indicators have obvious shortcomings, they are common in practice. For example, work on community identification from the link structure of a graph is based on indicators (Newman, 2010). Also, the tags in image annotation tasks are often represented with indicators (Weston et al., 2011). Similarly, most work on multilabel prediction use label indicators when no prior knowledge is encoded about labels $y$. 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 does not support direct generalization between objects, does not support out-of-sample prediction, and can be onerous to store if the sets are large.

### 2.2.1 Zero Shot Learning and Out of Sample Prediction

Recently, there has been renewed interest in endowing objects with meaningful *property based* features, or "attributes" in computer vision research (Al-Halah et al., 2016; Akata et al., 2013; Farhadi et al., 2009); see Figure 2.2 for an example. Property based features are also common in link prediction (Bleakley et al., 2007; Menon and Elkan, 2011; Gong et al., 2014) and recommender systems (Gantner et al.,

polar bear

black: no
white: yes
brown: no
stripes: no
eats fish: yes

zebra

black: yes
white: yes
brown: no
stripes: yes
eats fish: no

Figure 2.2: Endowing label objects with attribute representations enables the prospect of zero shot learning (figure from Deng et al. (2014)).

2010; Pazzani and Billsus, 2007). Property based features like these allow *generalization* between objects based on prior knowledge, even if an object has not been seen in the training data. In particular, attribute based features for a label $y \in \mathcal{Y}$ allow the prospect of *zero-shot learning* where one can predict an object $x$'s association with a target label $y$ that was not seen during training[1] (Li et al., 2003; Larochelle et al., 2008a; Palatucci et al., 2009; Socher et al., 2013c; Ba et al., 2015; Xian et al., 2016; Vinyals et al., 2016). Similarly, a property based feature representation $\boldsymbol{\phi}(x)$ for an element $x \in \mathcal{X}$ allows *out of sample* prediction for objects $x$ not seen during training; a standard goal in supervised learning.

In the framework of co-embedding, these issues are 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 geometric relationships to a given $x$.

## 2.3 Number of Sets

Thus far I have focused on the case of embedding items from two sets, but many applications involve associating items from different numbers of sets.

[1] In the language of the recommender system literature, attributes are referred to as "content-based features" (Pazzani and Billsus, 2007) and zero shot learning is referred to as "cold start recommendation". (Schein et al., 2002).

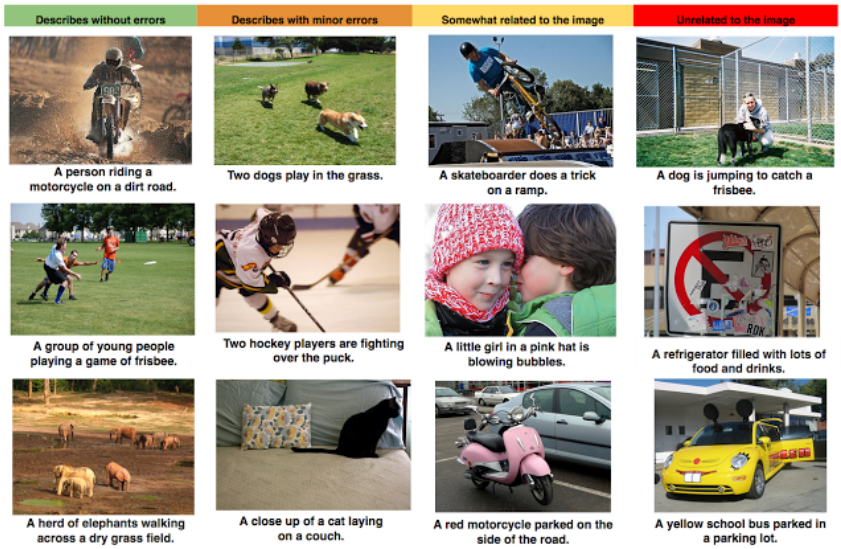


Figure 2.3: A sample of image descriptions from the Google Research page.

### 2.3.1 Two-set co-embedding

For the most common case of associating items between two sets, typical tasks include image annotation (with zero shot learning), item recommendation to users (collaborative filtering) and multi-modal representation learning.

In natural language processing, Bordes et al. (2014) co-embed questions and answers to retrieve appropriate answers to a query, while Bordes et al. (2012) embed words and senses for word sense disambiguation. In computer vision, Weston et al. (2010) and Akata et al. (2013) embed images and tags for image tagging. Kiros et al. (2014) embed images and sentences for image retrieval, as well as image description retrieval and generation; see Figure 2.3 for an illustration. Recently, Vendrov et al. (2016) jointly embed text and images for the task of image-caption retrieval and hierarchy prediction. In recommender systems, (Rendle et al., 2009) co-embed users and items. Finally, Yamanishi (2008) embeds nodes of a heterogeneous graph for link prediction. Notably, while not recognized before, the majority of standard multi-label prediction methods (eg (Guo and Schuurmans, 2011)) can be viewed as the co-embedding of data examples and labels into a pre-prediction space.

Another class of methods addresses multi-modal representation learning. For

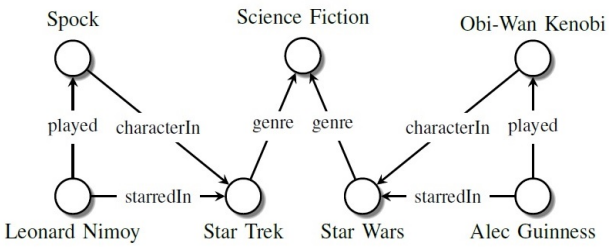


Figure 2.4: A sample knowledge graph where nodes represent entities and edge labels represent type of relations (figure from Nickel et al. (2016)).

example, Ngiam et al. (2011) use deep networks to learn features for two modalities, demonstrating that cross modal feature learning can improve the features learned for each modality individually (e.g., audio and video).

### 2.3.2 Three-set or More Co-embedding

While nothing conceptually limits the number of sets that can be associated, co-embedding of more than three sets is uncommon. Three set co-embedding is mainly useful when a third criteria (context, side information) is added to, say, a recommendation scenario. For example, in a personalized tag recommendation for webpages, a model that assumes each person has a different preference for assigning tags can be expressed by a three-way co-embedding (Rendle and Schmidt-Thieme, 2010). Similarly, Weston et al. (2012) consider personalized query-based item retrieval, where they embed users, items and queries. Later in the process, a three-way dependence might be decomposed to a number of two-way dependencies for the sake of simplicity. In addition to recommendation, three-set co-embedding can also be beneficial in semantic analysis; for example Globerson et al. (2007) embed documents, words and authors for semantic document analysis.

A particularly important application of three-set co-embedding is the representation and completion of knowledge graphs. Knowledge graphs model facts about the world in the form of entities and relationships between them, using a triple, (head, relation, tail), to annotate each edge; see Figure 2.4 for an illustration. Large knowledge graphs, such as Wordnet (Miller, 1995), Freebase (Bollacker et al., 2008), and the Google Knowledge Graph, provide useful sources of knowledge

for question answering. Several works have therefore focused on the representation and completion of such graphs (Bordes et al., 2011; Nickel et al., 2011; Socher et al., 2013a; Nickel et al., 2016).

A co-embedding approach to knowledge graph representation can be obtained by embedding each head, relation, and tail into a joint space. Such a unified representation can flexibly encode structural and symbolic information from a knowledge base to support new uses. For example, co-embedding of knowledge graphs has allowed the integration of knowledge bases by exploiting recent machine learning methods for prediction and retrieval tasks (Bordes et al., 2011). Sometimes only the entities at the head and tail of a relation are embedded, and the relation is represented as a transformation or matrix multiplication; for example as in (Bordes et al., 2013).

Whenever items from more than two sets are embedded, there is a question of how best to aggregate their geometric proximity. Depending on the type of proximity used, different approaches have been considered. For distance based proximities, whenever three or more points are to be related, their pair-wise proximities are typically aggregated. Although some novel notions of three-way distance have been proposed in the literature, see for example (Joly and Le Calvé, 1995), these have rarely been explicitly used for association tasks. For proximities expressed by inner products, an extension to three-way relations can naturally be handled by using tensors to model 3-way interactions; see Section 3.2.3 of Chapter 3 for a more detailed treatment.

### 2.3.3 Single-set Embedding

A special case occurs when elements of a *single set* are mapped into a low dimensional space, reducing the problem to conventional *Euclidean embedding*. In standard embedding, the entities to be mapped are considered to belong to the same set; i.e., possessing the same type or modality. Many supervised learning models are based on single-set embedding. For instance, Vert and Yamanishi (2004) and Menon and Elkan (2011) develop techniques to embed nodes from a homogeneous graph into a low dimensional space, which can be used to support supervised link

prediction in domains ranging from social networks and protein-protein interaction networks to co-authorship networks. Chopra et al. (2005) embed images of faces in a low dimensional space to support face recognition. In fact, most metric learning methods also fall in this class (Kulis, 2013).

Another class of embedding models are used to learn continuous representations for discrete structures. The best known example is embedding words in a low dimensional space to capture their semantics, which is a classic unsupervised learning approach used in text retrieval. In particular, the skip-gram method of Mikolov et al. (2013) (also known as Word2Vec) embeds words in a low dimensional space such that the context of each word (i.e., the window of surrounding words) is preserved. Shaw and Jebara (2009) embed nodes of a graph to preserve structure in support of graph compression and representation learning. Recently, the Node2Vec method of Grover and Leskovec (2016) adapted the skip-gram model to graphs, which they apply to link prediction and multi-label classification.

It should be noted that embedding is a mature subject in both the machine learning and theoretical computer science literatures. For example, a large body of work explores embedding for the purpose of dimensionality reduction in an unsupervised setting, such as Principal Component Analysis (PCA) (Hastie et al., 2009a), Local Linear Embedding (LLE) (Roweis and Saul, 2000), and Metric Multidimensional Scaling (MDS) (Cox and Cox, 2000). Another major subject covered in the theoretical field of metric embedding investigates the problem of embedding a weighted graph satisfying the triangle inequality (i.e., a metric) in a Euclidean space while (approximately) preserving the weights (Sidiropoulos, 2008). These works lie outside the scope of this thesis.

# Chapter 3

# Solving Association Problems with Alignment Based Co-embedding

In this chapter, we first describe our observation that a large group of problems tackled independently by different machine learning subcommunities are in fact very similar in nature. We unify these under the title of *association problems*. Next, we describe our second observation that many existing solutions to this class of problems are also very similar: they all employ a form of co-embedding. Then, we highlight that existing work typically exploits local optimization techniques to train co-embedding models; we instead propose convex re-formulations for training. While similar convex relaxations apply to both alignment and distance models, in this chapter we focus on alignment models. We defer the study of distance based co-embedding to Chapter 4, where we explore it in the context of metric learning.

After setting up the framework and formulating training of alignment based models, we identify an efficient algorithm that makes training computationally possible. Finally, we evaluate the performance of the method on real data for a number of interesting case studies: namely, multilabel prediction and tag recommendation. In the first application, which is a two set co-embedding problem, a model is to be learned from labeled training data to assign a suitable subset of labels to a new data example. In the second application, given a tensor (i.e. a 3-way array with one element for each (user, item, tag) triple) and two sets, one of known and another of unknown elements, the task is to predict the unknown elements. In other words, the task is to predict which of the candidate tags would the corresponding user assign

to the corresponding item.

The key contributions of this chapter are the following:

**Contribution 1** *A unified view of association problems is proposed.*

**Contribution 2** *A unified approach to solving association problems based on co-embedding is proposed that captures typical strategies.*

**Contribution 3** *A convex reformulation of training for co-embedding models is developed.*

**Contribution 4** *An appropriate training algorithm for alignment-based co-embedding is identified.*

**Contribution 5** *We make a notable empirical observation that, under random initialization, solutions to the convex and non-convex formulations of training for alignment models are the same.*

## 3.1 Score-based Association Learning

Associating elements of sets is a fundamental problem in applications as diverse as ranking, retrieval, recommendation, link prediction, relation learning, tagging, and multilabel 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 elements 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; multilabel *tagging* can be expressed as predicting which subset of a set of label elements are associated with a given query object; *link prediction* involves determining which elements from a set are related to elements from another set; and so on. These problems can be extended to a *multi-relational* setting by introducing a third criterion as context or side-information to the associations. For example, a user dependent query answering.

In this section, we offer a unified perspective on co-embedding by presenting a simple framework that expresses association problems in a common format.

### 3.1.1 A Unified View of Association Problems

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

**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 directed link prediction, tagging and multilabel 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 for instance (Bennett and Lanning, 2007), we focus on *discrete problems* in this dissertation.

In association problems that consider two sets, observations are typically *dyads*, i.e. pairs with one element from each of the sets (Hofmann et al., 1998; Hoff, 2005; Meeds et al., 2006; Menon and Elkan, 2010a,b). The extension to problems where more than two sets are considered is discussed in Section 3.2.3.

### 3.1.2 Score-based Solutions

To tackle association problems, several solutions have been proposed in the machine learning literature. A natural and common approach is to use an intermediate function—called a score or utility function—to facilitate association learning (Fürnkranz et al., 2008). In particular, we consider using an *association score function*

$$s : \mathcal{X} \times \mathcal{Y} \to \mathbb{R},$$

and, when appropriate a *decision threshold function*

$$t : \mathcal{X} \to \mathbb{R}.$$

Many models preset the threshold value instead of parameterizing and learning it. In some other models (such as (Socher et al., 2013a)), the threshold is selected based on validation data as a post processing step after training. However, a parametrized threshold that is learned jointly with other model parameters can adapt to the data distribution. In Chapter 4 we show that two threshold functions, one for each set, can be used to generalize the proposed framework to a more symmetric model.

Given score and threshold functions, the test phase of an association problem is summarized as follows.

**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 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.

### 3.1.3 Co-embedding for Defining the Association Score

Suppose linear maps $U$ and $V$ jointly embed elements of $\mathcal{X}$ and $\mathcal{Y}$ respectively to a common space. Given such a co-embedding, there are two standard models for expressing an association score.

The *alignment model* uses the score and threshold functions:

$$s(x, y) = \langle \mathbf{u}(x), \mathbf{v}(y) \rangle = \boldsymbol{\phi}(x)^\top U^\top V \boldsymbol{\psi}(y) \tag{3.1}$$

$$t(x) = \langle \mathbf{u}(x), \mathbf{u}_0 \rangle = \boldsymbol{\phi}(x)^\top U^\top \mathbf{u}_0, \tag{3.2}$$

where $U^\top$ denotes the transpose of a matrix $U$ and 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 et al., 2011), multilabel classification (Guo and Schuurmans, 2011), and link prediction (Bleakley et al., 2007).

The *distance model* uses the score and threshold functions:

$$s(x, y) = -\|\mathbf{u}(x) - \mathbf{v}(y)\|^2 = -\|U\boldsymbol{\phi}(x) - V\boldsymbol{\psi}(y)\|^2 \tag{3.3}$$

$$t(x) = -\|\mathbf{u}(x) - \mathbf{u}_0\|^2 \quad = -\|U\boldsymbol{\phi}(x) - \mathbf{u}_0\|^2, \tag{3.4}$$

where again the decision threshold function is usually based on a direct embedding, $\mathbf{u}_0$, of a null object. This latter model underlies work on "metric learning" (Globerson et al., 2007; Weinberger and Saul, 2009), and has also been used in the area of multi-relation learning (Sutskever and Hinton, 2008), with renewed interest (Bordes et al., 2013, 2011). Chapter 4 studies such distance models in more detail.

Interestingly, most work has adopted one of these two models without considering the other, although some recent work in multi-relational learning has started to relate these representations (Socher et al., 2013a).

### 3.1.4 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 et al., 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 different settings. For example, in link prediction and tagging, observations are often only positive "must link" pairs; whereas, in multilabel 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 to use such data 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; Cortes and Mohri, 2003; Menon and Elkan, 2011). For a given $x$, the AUC of a score function $s$ is given by

$$\frac{1}{|Y(x)|}\frac{1}{|\bar{Y}(x)|}\sum_{y\in Y(x)}\sum_{\bar{y}\in\bar{Y}(x)} 1\Big(s(x,y) > s(x,\bar{y})\Big), \tag{3.5}$$

where $\mathbf{1}(\xi)$ denotes the indicator function that returns $1$ when $\xi$ is true, $0$ otherwise. More recently the ordered weighted average (OWA) family of ranking error functions has become preferred (Usunier et al., 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 $\boldsymbol{\alpha}^\top \mathbf{1} = 1$ and $\alpha_1 \geq \alpha_2 \geq \cdots$. For a given $x$, the OWA is defined by

$$\sum_{y\in Y(x)}\sum_{\bar{y}\in\bar{Y}(x)} \alpha_{\pi(x,\bar{y})} 1\Big(s(x,y) \leq s(x,\bar{y})\Big), \tag{3.6}$$

where $\pi(x,\bar{y})$ denotes the position of $\bar{y}$ in the list sorted by

$$s(x,\bar{y}_1) \geq s(x,\bar{y}_2) \geq \cdots .$$

To better understand the effect of the ordered weighting, we can first investigate the two extreme cases. One is when the weight given to the top ranking element is not more than any other element, where ordered weighting reduces to an average. The other extreme case is when the total weight is concentrated on the top ranking element only, where ordered weighting reduces to finding a maximum. All other cases are something in between. By changing the distribution of ordered weights, relative concentration or attention on the higher ranking elements can be tuned. For example one can imagine an OWA weighting that top element receives $\frac{2}{3}$ of attention and the second top element $\frac{1}{3}$. It is not hard to play with weights to come up with other reasonable weightings that distribute attention differently. This is useful in retrieval applications, because often being accurate at the top is very important there.

**Query answering**: For query answering, performance is most often assessed by the point-wise prediction error, given by

$$\sum_{y \in Y(x)} 1\Big(s(x,y) \leq t(x)\Big) + \sum_{\bar{y} \in \bar{Y}(x)} 1\Big(s(x,\bar{y}) > t(x)\Big). \tag{3.7}$$

The first summation in (3.7) counts the false negative predictions: i.e. the number of queries with true values being positive, but predicted negative. Similarly the second summation takes care of counting false positives.

**Prediction**: There are many performance measures used to evaluate prediction performance (Sebastiani, 2002; Tsoumakas et al., 2009). Point-wise 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 et al., 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

$$\sum_{y \in Y(x)} \alpha_{\sigma(x,y)} 1\Big(s(x,y) \leq t(x)\Big) + \sum_{\bar{y} \in \bar{Y}(x)} \alpha_{\pi(x,\bar{y})} 1\Big(s(x,\bar{y}) > t(x)\Big), \tag{3.8}$$

in which $\alpha_i$ is the $i$th element of a vector $\boldsymbol{\alpha}$ with length $len(\boldsymbol{\alpha}) = |Y(x)| + |\bar{Y}(x)|$ and $\sigma(x,y)$ and $\pi(x,\bar{y})$ are defined as follows. The function $\sigma(x,y)$ gives the position of $y$ in the list of scores $s(x,y_1) \leq s(x,y_2) \leq \cdots$ sorted in ascending order among $y_1, y_2, \cdots \in Y(x)$. The function $\pi(x,\bar{y})$ gives the position of $\bar{y}$ in the list of scores $s(x,\bar{y}_1) \geq s(x,\bar{y}_2) \geq \cdots$ sorted in descending order among $\bar{y}_1, \bar{y}_2, \cdots \in \bar{Y}(x)$.

An upper bound on 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 (3.7) is achieved by setting $\boldsymbol{\alpha} = \mathbf{1}$.

**Proposition 1** *An upper bound on the exact match error is achieved by setting* $\boldsymbol{\alpha} = \mathbf{1}_1$ *in the OWA-prediction error* (3.8)*, i.e.*

$$1\Big(y \in Y(x),\, \bar{y} \in \bar{Y}(x),\, \big(s(x,y) < t(x)\big) \vee \big(t(x) \leq s(x,\bar{y})\big)\Big) \leq \sum_{y \in Y(x)} \alpha_{\sigma(x,y)}\; 1\Big(s(x,y) \leq t(x)\Big) + \sum_{\bar{y} \in \bar{Y}(x)} \alpha_{\pi(x,\bar{y})}\; 1\Big(s(x,\bar{y}) > t(x)\Big). \quad (3.9)$$

See Appendix A.1 for a proof.

### 3.1.5 Training the Score Function

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 et al., 2009; Weston et al., 2011). In our co-embedding framework, the training problem is

$$\min_{U,V} \sum_{x \in S} \sum_{y \in Y(x)} \sum_{\bar{y} \in \bar{Y}(x)} \alpha_{\pi(x,\bar{y})} L\Big(s(x,y) - s(x,\bar{y})\Big), \quad (3.10)$$

where $L\Big(s(x,y) - s(x,\bar{y})\Big) \geq 1\Big(s(x,y) \leq s(x,\bar{y})\Big)$ for a convex and non-increasing loss function $L$. Here the parameters $U$ and $V$ appear in the score model, either (3.1) or (3.3).

For *prediction*, recent improvements in multilabel classification and tagging have resulted from the use of so-called calibrated losses (Fürnkranz et al., 2008; Guo and Schuurmans, 2011). Interestingly, these losses are both convex upper bounds on (3.8) for different choices of $\boldsymbol{\alpha}$ (not previously realized). For example, the first approach uses $\boldsymbol{\alpha} = a\mathbf{1}$ to upper bound point-wise error (3.7), 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 as

$$\min_{U,V,\mathbf{u}_0} \sum_{x \in S} \sum_{y \in Y(x)} \alpha_{\sigma(x,y)} L\Big(s(x,y) - t(x)\Big) + \sum_{x \in S} \sum_{\bar{y} \in \bar{Y}(x)} \alpha_{\pi(x,\bar{y})} L\Big(t(x) - s(x,\bar{y})\Big), \quad (3.11)$$

where

$$L\Big(s(x,y) - t(x)\Big) \geq 1\Big(s(x,y) \leq t(x)\Big)$$

and

$$L\Big(t(x) - s(x,\bar{y})\Big) \geq 1\Big(s(x,\bar{y}) > t(x)\Big)$$

for a convex and non-increasing loss function $L$. Here the parameter $\mathbf{u}_0$ appears in the threshold model, either (3.2) or (3.4).

Unfortunately, even though convex loss functions are common in co-embedding approaches, they do not make the training problems (3.10) and (3.11) convex. For the alignment model (3.1), non-convexity arises from the bilinear interaction between $U$ and $V$, whereas the nonlinearity of the distance model (3.3) 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).

## 3.2 Alignment-based Co-embedding for Association Learning

We now introduce the main formulation we consider in this chapter. Our goal is to first demonstrate that the previous training formulations (3.10) and (3.11) 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 (3.3) as well as the alignment based score model (3.1), after an initial change of variables. In this section, we focus on the alignment score model only.

### 3.2.1 Convex Relaxations for Alignment Model

For the alignment model (3.1), one can re-parametrize the score function as

$$s_M(x,y) \quad = \quad \boldsymbol{\phi}(x)^\top M \boldsymbol{\psi}(y) \tag{3.12}$$

for a matrix variable $M = U^\top V \in \mathbb{R}^{m \times n}$. This simple change of variables allows the problems (3.10) and (3.11) to be expressed equivalently as minimization over $M$ subject to the constraint that $\mathrm{rank}(M) \leq d$. Since rank is not convex, we introduce a relaxation and replace rank with a trace norm regularization of $M$.[1]

Since we assumed the loss function in (3.10) and (3.11) was convex, a linear parametrization of the score function $s$ (3.12) coupled with replacing the rank constraint by trace norm regularization leads to a convex formulation of the training problems (3.10) and (3.11) respectively. In particular, (3.10) becomes minimizing the following over $M$

$$\sum_{x \in S} \sum_{y \in Y(x)} \sum_{\bar{y} \in \bar{Y}(x)} \alpha_{\pi(x,\bar{y})} L\Big(s_M(x,y) - s_M(x,\bar{y})\Big) + \lambda \|M\|_{\mathrm{tr}}, \tag{3.13}$$

where we have introduced a regularization parameter $\lambda$, which allows the desired rank to be achieved by a suitable choice (Cai et al., 2008).

Similarly, for prediction, (3.11) becomes

$$\begin{aligned} \min_{M, \mathbf{m}} & \sum_{x \in S} \sum_{y \in Y(x)} \alpha_{\sigma(x,y)} L\Big(s_M(x,y) - t_{\mathbf{m}}(x)\Big) \\ & \sum_{\bar{y} \in \bar{Y}(x)} \alpha_{\pi(x,\bar{y})} L\Big(t_{\mathbf{m}}(x) - s_M(x,\bar{y})\Big) + \lambda \|M\|_{\mathrm{tr}}, \end{aligned} \tag{3.14}$$

which is jointly convex in the optimization variables $M$ and $\mathbf{m} = U^\top \mathbf{u}_0$ using the model (3.1) and (3.2). Although these reformulations are not surprising, below we discuss how the resulting optimization problems can be solved efficiently.

### 3.2.2 Efficient Training Algorithm

Let us write the training problem as

$$\min_M F(M) + \lambda \|M\|_{\mathrm{tr}}, \tag{3.15}$$

where $F$ denotes the convex training objective of interest. Significant recent progress has been made in developing efficient algorithms for solving such problems (Dudik

[1] 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 et al., 2010).

et al., 2012). Early approaches were based on alternating direction methods that exploited variational representations of the trace norm via, for example

$$\|M\|_{\mathrm{tr}} = \frac{1}{2} \min_{\Omega \succeq 0} \mathrm{tr}(M^\top \Omega^{-1} M) + \mathrm{tr}(\Omega). \tag{3.16}$$

Given such a characterization, an alternating direction strategy can successively optimize $M$ and $\Omega$, exploiting the fact that $\Omega$ will have a closed form update (Argyriou et al., 2008; Grave et al., 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 et al., 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 $B$ of $M$ such that

$$M = AB^\top,$$

where the search begins with thin $A$ and $B$ matrices and incrementally grows them (Dudik et al., 2012). The benefit of this approach is that only the top singular vector pair is required on each iteration, which is a significant saving over requiring the full SVD.

A useful improvement is the recent strategy of Zhang et al. (2012), which combines the approach of Dudik et al. (2012) with an earlier method of Srebro et al. (2004). Here the idea is to start with thin factors $A$ and $B$ as before, but locally optimize these matrices by replacing the trace norm of $M$ with a well known identity

$$\|M\|_{\mathrm{tr}} = \min_{A,B:AB^\top = M} \frac{1}{2}\Big(\|A\|_F^2 + \|B\|_F^2\Big) \tag{3.17}$$

(Srebro et al., 2004). The key is to escape a local minimum when the local optimization terminates: here the strategy of (Dudik et al., 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

$$\max_{\mathbf{a},\mathbf{b}:\|\mathbf{a}\|\leq 1,\|\mathbf{b}\|\leq 1} -\mathbf{a}^\top \nabla F(M)\mathbf{b} \tag{3.18}$$

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

$$\min_{\mu\geq 0,\nu\geq 0} F\big(\mu M + \nu \mathbf{a}\mathbf{b}^\top\big) + \lambda(\mu c + \nu) \tag{3.19}$$

for scalar $\mu$ and $\nu$, where $c = \frac{1}{2}(\|A\|_F^2 + \|B\|_F^2)$ at the current iterate. The solution to (3.18) can be efficiently computed via the leading left and right singular vector pair of $-\nabla F(M)$. This method, which is actually a boosting approach, is quite effective (Zhang et al., 2012), often requiring only a handful of outer escapes to produce an optimal $M$ in our experiments. For reduction to boosting and in particular Adaboost see page 82, Example 4.2 of Yu (2013).

### 3.2.3 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. Section 2.3.2 provides some examples on multi-relational association learning problems.

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 $t(x, z)$.

**Query answering**: given $(x, y, z)$ return 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 $\boldsymbol{\xi}(z) \in \mathbb{R}^p$.

The linear alignment based model (3.12) can be easily extended by expanding the matrix $M$ to a three-way tensor $T$, allowing a general alignment score function to be expressed

$$s_T(x, y, z) \quad = \quad \sum_{ijk} T_{ijk} \boldsymbol{\phi}(x)_i \boldsymbol{\psi}(y)_j \boldsymbol{\xi}(z)_k \tag{3.20}$$

which is still linear in the parameter tensor $T$.

Such a parametrization 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., 2013a; Jenatton et al., 2012), but others have found success by working with compressed representations (Nickel et al., 2011; Gantner et al., 2010; Rendle and Schmidt-Thieme, 2009).

In Section 3.4 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.

## 3.3 Case Study: Multilabel Prediction

To investigate the efficacy of convex embedding, we conducted an initial experiment on multilabel classification with the multilabel data sets shown in Table 3.1. In each case, we used 1000 examples for training and the rest for testing (except

| 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 3.1: Data properties for co-embedding experiments for multilabel prediction. $1000$ examples used for training and the rest for testing ($2/3$-$1/3$ split for Emotion).

*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 (3.12) and a smoothed version (3.22) of the large margin multilabel loss in (3.21), which gave state of the art results (Guo and Schuurmans, 2011):

$$\sum_{x \in S} \max_{y \in Y(x)} L\big(m(x,y)\big) + \max_{\bar{y} \in \bar{Y}(x)} L\big(\bar{m}\big(x,\bar{y}\big)\big) \tag{3.21}$$

$$\leq \sum_{x \in S} \operatorname*{softmax}_{y \in Y(x)} \tilde{L}\big(m(x,y)\big) + \operatorname*{softmax}_{\bar{y} \in \bar{Y}(x)} \tilde{L}\big(\bar{m}(x,\bar{y})\big), \tag{3.22}$$

where

$$m(x,y) = s(x,y) - t(x), \quad \bar{m}(x,\bar{y}) = t(x) - s(x,\bar{y}),$$

$$L(m) = (1-m)_+; \quad \tilde{L}(m) = \begin{cases} \frac{1}{4}(2-m)_+^2 & \text{if } 0 \leq m \leq 2 \\ (1-m)_+ & \text{otherwise,} \end{cases}$$

and

$$\operatorname*{softmax}_{y \in Y} f(y) = \ln \sum_{y \in Y} \exp\big(f(y)\big).$$

(Note that (3.21) follows from the loss in (3.14) using $\boldsymbol{\alpha} = \mathbf{1}_1$.)

The aim of this study is to compare the global training method developed above (CVX), which uses a convex parametrization ($M$ and $\mathbf{m}$), against a conventional alternating descent strategy (ALT) that uses the standard factored parametrization ($U^\top V = M$ and $U^\top \mathbf{u}_0 = \mathbf{m}$). 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. For a regularization parameter $\lambda$, we regularize the trace norm of $M$ in the convex

| | 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% |
| $\lambda$ | 0.3 | 0.45 | 0.2 | 3.0 | 1.0 |
| CVX rank | 19 | 4 | 3 | 4 | 3 |

Table 3.2: Multilabel prediction 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.

parametrization as $\lambda\|M\|_{tr}$, while the squared Frobenius norm of $U$ and $V$ in the factored form as $\frac{\lambda}{2}(\|U\|_F^2 + \|V\|_F^2)$. Recall from (3.17) that

$$\|M\|_{\mathrm{tr}} = \min_{U,V:\, U^\top V = M} \frac{1}{2}\Big(\|U\|_F^2 + \|V\|_F^2\Big).$$

The results of this experiment, given in Table 3.2, are surprising in two respects. First, under random initializations, we found that the local optimizer, ALT, achieves the global objective in all the data splits on all data sets for all 100 initializations 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 (3.14) using loss (3.22) with squared Frobenius norm regularization, even when using the factored parametrization $U^\top V = M$ and $U^\top \mathbf{u}_0 = \mathbf{m}$. 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 3.2.

## 3.4 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. No explicit features are available. 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 (3.20). 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.

For prediction, following Rendle and Schmidt-Thieme (2009), we rank the tags that each user assigns to an item. Given a score function $s$, the top five tags $y$ are predicted for 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 parametrize the tensor with the pairwise interaction model (Rendle and Schmidt-Thieme, 2010; Chen et al., 2013b), which uses the decomposition

$$s(x, y, z) \;=\; T_{xyz} \;=\; N_{x,y} + P_{z,y} \quad \forall x, y, z. \tag{3.23}$$

Following (Rendle and Schmidt-Thieme, 2009), we use the ranking logistic loss function for learning $N$ and $P$ in the formulation (3.13), but replace their low rank assumptions on $N$ and $P$ with a trace norm relaxation

$$Reg(N, P) \;=\; \lambda_1 \|N\|_{tr} + \lambda_2 \|P\|_{tr}. \tag{3.24}$$

The aim of this study is, again, to compare the global training method developed above (CVX), which uses the convex parametrization ($N$ and $P$), against a

| Method | $\lambda$ | $d_1$ | $d_2$ | obj | $F1$ | 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 3.3: Tag recommendation results. All methods were initialized randomly, except ALT0 indicates initializing from all 0s, and ALT1 indicates initializing from all 1s.

conventional alternating descent strategy (ALT) that uses a factored parametrization ($U^\top V = N$ and $Q^\top R = P$). We also include a Frobenius norm regularizer on $U$, $V$, $Q$, and $R$ following (Rendle and Schmidt-Thieme, 2009).

$$Reg(U, V, Q, R) = \frac{\lambda_1}{2}(\|U\|_F^2 + \|V\|_F^2) + \frac{\lambda_2}{2}(\|Q\|_F^2 + \|R\|_F^2) \tag{3.25}$$

Equation (3.17) explains the relation between regularizers used in the two settings (3.24) and (3.25). 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 $d_1, d_2 \in \{32, 64, 128, 256\}$.

**Experimental Results** The results of this study are shown in Table 3.3 below. The first four columns report the settings used: the training method, the shared regularization parameter $\lambda$, the rank of $N$ and the rank of $P$. 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 section, CVX, an alternating minimization, ALT. In this block, the global method is first trained using the fixed regularization parameter $\lambda$, 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 $U^\top V = N$ and $Q^\top R = P$ used by ALT. The second and third block show the results for ALT using the fixed parameter values ($d_1$, $d_2$ and $\lambda$) 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 parametrization 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. This result suggests that one or a combination of the introduction of trace norm regularization, the parametrization, or the optimization method has somehow eliminated local minima from the problem once again. 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.

## 3.5 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 alignment 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 terms of the training objective value achieved with local and global optimization

strategies, in our experiments for all random initializations of local optimization, for both experiments local and global objective values have been equal. However, there have been specific initializations of local training, that led to non-optimal critical points.

# Chapter 4

# Scalable Metric Learning for Distance Based Co-embedding

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 data with a single modality, metric learning has been well explored (Xing et al., 2002; Globerson and Roweis, 2005; Davis et al., 2007; Weinberger and 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 and Xing, 2013).

Concurrently, there is a growing literature that tackles *co-embedding problems*, where *multiple* sets or modalities are embedded into a common space so that their elements could be associated. Current approaches to these problems are mainly based on deep neural networks (Ngiam et al., 2011; Srivastava and Salakhutdinov, 2012; Socher et al., 2013b; Frome et al., 2013) and simpler non-convex objectives (Chopra et al., 2005; Larochelle et al., 2008b; 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 score

used for distance based co-embedding is the squared Euclidean distance metric.

In this chapter, we study distance based co-embedding and demonstrate that it can be cast as metric learning. Once formalized, this connection allows metric learning methods to be applied to the wide class of association problems such as link prediction, multilabel and multiclass 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 chapter then investigates optimization strategies for metric learning that guarantee convergence to a global optimum. Typically, metric learning approaches are expressed with convex formulations subject to a semi-definite constraint over a matrix variable, $C \succeq 0$. Standard attempts to solve such a convex constrained problem suffer from scalability issues. An alternative approach that is gaining popularity works with a low-rank factorization $Q$ instead, implicitly maintains positive semi-definiteness through $C = QQ^\top$ (Burer and 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 local minimum $Q^*$ provides a *global* solution $C^* = Q^*{Q^*}^\top$, as long as the objective is smooth and convex *in* $C$. This result is often neglected in the metric learning literature. However, as discussed in this chapter, it can be directly used to perform a single local search and achieve global results to metric learning.

Moreover, by using this result, we have been able to develop a fast iterative approach to metric learning that improves previous approaches (Journée et al., 2010; Zhang et al., 2012). Next, we empirically compare the run time of the proposed and original versions of the algorithm on three examples. This chapter then concludes with an empirical investigation of two distance based co-embedding tasks: multilabel classification and tagging. In these tasks, we first train co-embedding models both in a convex form in the presence of a semidefinite constraint and also the non-

convex factored form. For different local minima achieved from local optimization applied to the the non-convex formulations, we illustrate the changes in geometrical spread of local minima over different ranks of $Q$. The outcome graphs 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. We then evaluate the performance of the approach on these case studies.

The main contributions of this chapter are the following:

**Contribution 6** *The relationship between association problems and metric learning is demonstrated.*

**Contribution 7** *A convex training formulation for distance based co-embedding, and hence for heterogeneous metric learning, is developed.*

**Contribution 8** *We illustrate how the distribution of local minima in the non-convex factored formulation of metric learning is affected by increasing rank.*

**Contribution 9** *A scalable iterative algorithm for training a smooth convex objective function subject to a semi-definite constraint is developed.*

**Contribution 10** *A proof of convergence is provided for the proposed algorithm.*

**Contribution 11** *The conditions under which the non-convex training formulation yields globally optimal solutions are identified.*

## 4.1 Preliminaries: Metric Learning

The goal of metric learning is to learn a distance function between data instances that helps solve prediction problems. 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.). To obtain task-specific distances without extensive manual design, supervised metric learning attempts to exploit task-specific information to guide the learning process.

Suppose one has a sample of $t$ observations, $\mathbf{x}_i \in \mathcal{X}$, and a feature map $\phi$ where $\phi : \mathcal{X} \to \mathbb{R}^n$. Then a training matrix $\phi(X) = \big[\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_t)\big] \in \mathbb{R}^{n \times t}$ can be obtained by applying $\phi$ to each of the original data points.[1] 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

$$d_C(\mathbf{x}_1, \mathbf{x}_2) = \Big(\phi(\mathbf{x}_1) - \phi(\mathbf{x}_2)\Big)^\top C \Big(\phi(\mathbf{x}_1) - \phi(\mathbf{x}_2)\Big) \tag{4.1}$$

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 task dependent information that should influence the learning and improve it compared to an unsupervised disance 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:[2]

$$\min_{C \succeq 0, C \in \mathcal{C}} \sum_i L_i\big(\phi(X)^\top C \phi(X)\big) + \beta \, \text{reg(C)}. \tag{4.2}$$

For example, in large margin nearest neighbor learning (Weinberger and Saul, 2009), one might want to minimize

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

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 label different than $\mathbf{x}_i$.

Although supervised metric learning has typically been used for classification, one can apply it to other settings where distances between data points are useful, like kernel regression or ranking. Interestingly, the applicability of metric learning can be extended well beyond the framework (4.2) by additionally observing that *co-embedding* elements from different sets can be expressed as a *joint* metric learning problem.

[1] Throughout the document we extend functions $\mathbb{R} \to \mathbb{R}$ to vectors or matrices element-wise.

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

## 4.2 Distance-based Co-embedding as Metric Learning

Recall the distance based co-embedding framework from Chapter 3: we are given two sets of data objects $\mathcal{X}$ and $\mathcal{Y}$ and wish to map the elements $\mathbf{x} \in \mathcal{X}$ and $\mathbf{y} \in \mathcal{Y}$ from each set into a common Euclidean space. Importantly, the modality of data in $\mathcal{X}$ and $\mathcal{Y}$ and the number of features in the initial representation of elements from these sets could be different. The association score $s(x, y)$ is then computed based on the Euclidean distances between co-embedding vectors. In other words, the closer two objects are in embedding space, the more associated they are considered. Based on the association score and decision thresholds, the final outputs are determined (which could be predictions, answers to queries, or a ranking over items of one of the sets). To provide adaptive decision thresholds, when required, a dummy element is also embedded from each set as a distance keeper.

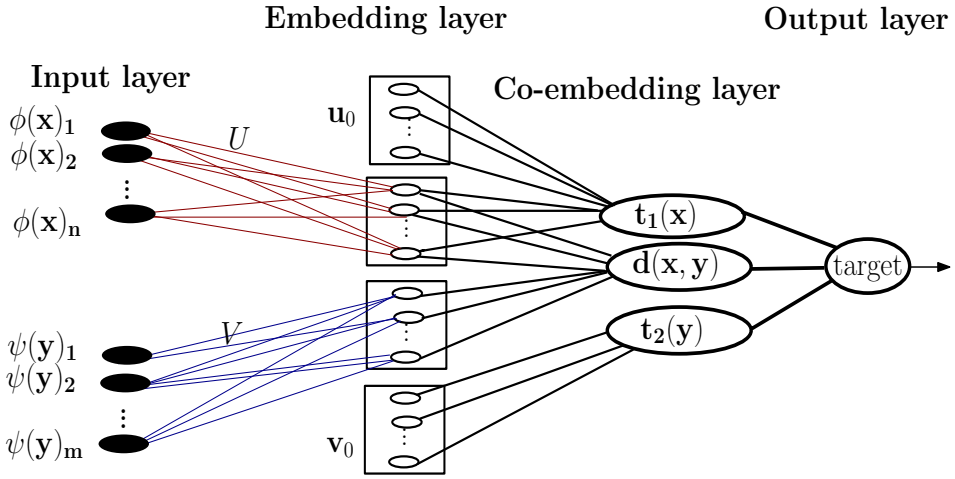


Figure 4.1: A neural network view of co-embedding

Figure 4.1 depicts this set-up for prediction or query answering as a neural network. 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 func-

tion, $d(\mathbf{x}, \mathbf{y})$ that holds the distance between embedding vectors $\mathbf{u}(\mathbf{x})$ and $\mathbf{v}(\mathbf{y})$ (outputs of the previous layer), via

$$d(\mathbf{x}, \mathbf{y}) = -s(x, y) = \|\mathbf{u}(\mathbf{x}) - \mathbf{v}(\mathbf{y})\|^2, \tag{4.3}$$

Here, association score $s(x, y)$ equals $-d(\mathbf{x}, \mathbf{y})$.

In addition to the distance function, two decision thresholds, $t_1(\mathbf{x})$ and $t_2(\mathbf{y})$ are computed in co-embedding layer, via

$$t_1(\mathbf{x}) = \|\mathbf{u}(\mathbf{x}) - \mathbf{u}_0\|^2, \tag{4.4}$$

$$t_2(\mathbf{y}) = \|\mathbf{v}(\mathbf{y}) - \mathbf{v}_0\|^2. \tag{4.5}$$

Function $t_1(\mathbf{x})$ models the association threshold that an element $x \in \mathcal{X}$ uses. Similarly $t_2(\mathbf{y})$ holds association threshold that an element $y \in \mathcal{Y}$ uses. Data dependent modeling of thresholds, can increase the expressiveness of the model. To ensure that the threshold functions return valid distances, particularly to avoid negative values, we simply model them as a distance in first place, which is nonnegative by definition. In particular, $t_1(\mathbf{x})$, is modeled as the distance between the embedding point $\mathbf{u}(\mathbf{x})$ and a reference point $\mathbf{u}_0$, where a single $\mathbf{u}_0$ is used for every $x \in \mathcal{X}$. Here, $\mathbf{u}_0$ could be viewed as the embedding of a null object. Note that $\mathbf{u}_0$ is in turn a parameter to be learned. Similarly, $t_2(\mathbf{y})$ is the distance between $\mathbf{v}(\mathbf{y})$ and a reference point $\mathbf{v}_0$, where a single $\mathbf{v}_0$ is used for every $y \in \mathcal{Y}$.

The output layer nonlinearly combines the association scores and thresholds to predict targets. For example, in a multilabel classification problem, given an element $\mathbf{x} \in \mathcal{X}$, its association to each $\mathbf{y} \in \mathcal{Y}$ can be determined via:

$$\text{label}(\mathbf{y}|\mathbf{x}) = \text{sign}\big(t_1(\mathbf{x}) - d(\mathbf{x}, \mathbf{y})\big). \tag{4.6}$$

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

$$\text{label}(\mathbf{x}, \mathbf{y}) = \text{sign}\Big(\min(t_1(\mathbf{x}), t_2(\mathbf{y})) - d(\mathbf{x}, \mathbf{y})\Big), \tag{4.7}$$

and so on.[3] As shown in Figure 4.1, in this neural network with two hidden layers, the trainable parameters, $U$, $V$, $\mathbf{u}_0$ and $\mathbf{v}_0$, appear in the first layer only.

[3]Intuitive examples of the application of the above prediction formulations could be provided in

Although the relationship to metric learning might not be obvious, it is useful to observe that the quantities in (4.3) can be expressed in terms of a symmetric semidefinite matrix $C$ (4.8), where $C \in \mathbb{R}^{p \times p}$, $p = n + m + 2$.

$$
\begin{aligned}
C &= \begin{bmatrix} U^\top U & V^\top U & \mathbf{u}_0^\top U & \mathbf{v}_0^\top U \\ U^\top V & V^\top V & \mathbf{u}_0^\top V & \mathbf{v}_0^\top V \\ U^\top \mathbf{u}_0 & V^\top \mathbf{u}_0 & \mathbf{u}_0^\top \mathbf{u}_0 & \mathbf{v}_0^\top \mathbf{u}_0 \\ U^\top \mathbf{v}_0 & V^\top \mathbf{v}_0 & \mathbf{u}_0^\top \mathbf{v}_0 & \mathbf{v}_0^\top \mathbf{v}_0 \end{bmatrix} \\
&= \begin{bmatrix} U\ V\ \mathbf{u}_0\ \mathbf{v}_0 \end{bmatrix}^\top \begin{bmatrix} U\ V\ \mathbf{u}_0\ \mathbf{v}_0 \end{bmatrix} \qquad (4.8)
\end{aligned}
$$

The recovery formulations are presented as:

$$dist(\mathbf{x}, \mathbf{y}) = \big[\phi(\mathbf{x}), -\psi(\mathbf{y}),\ 0,\ 0\big]\ C\ \big[\phi(\mathbf{x}), -\psi(\mathbf{y}),\ 0,\ 0\big]^\top, \qquad (4.9)$$

$$t_1(\mathbf{x}) = \big[\phi(\mathbf{x}),\ \ 0,\ \ -1,\ \ 0\big]\ C\ \big[\phi(\mathbf{x}),\ \ 0,\ \ -1,\ \ 0\big]^\top, \qquad (4.10)$$

$$t_2(\mathbf{y}) = \big[0,\ -\psi(\mathbf{y}),\ \ 0,\ -1\big]\ C\ \big[0,\ -\psi(\mathbf{y}),\ 0,\ -1\big]^\top. \qquad (4.11)$$

This yields a novel distance function with mutually consistent threshold representation, all linear in $C$.

**Modeling comparison** An important advantage that the distance based reformulation (4.8) holds over the alignment based co-embedding reformulation (3.12) of Chapter 3, is that (4.8) allows an effective way to encode side information about elements of each set (in addition to information about elements from different sets) with price of using more parameters. 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 $\tilde{C}$ denote the $m \times m$ submatrix of $C$ in (4.8) corresponding to $V^\top V$. If one would like to impose the constraint that object $y_1$ is closer to

the context of modeling customer purchase decisions or modeling social relationships. For example, (4.6) could be useful to model a customer's purchase decision. By this model, each customer buys any item whose distance in the co-embedding space is close enough for the threshold he/she puts for buying stuff. On the other hand, 4.7 could be useful in a model for predicting friendship relationships. By this model, a friendship relationship between $x$ and $y$ is formed if their distance in co-embedding space is small enough for the personal thresholds each of them put for forming their friendships.

$y_2$ than $y_3$, i.e., $d(y_1, y_2) < d(y_1, y_3)$ (say, based on prior knowledge), then this can be directly enforced in the joint embedding submatrix $\tilde{C}$ via the linear constraint in (4.12) below:

$$\begin{aligned} &\Big(\boldsymbol{\psi}(y_1) - \boldsymbol{\psi}(y_2)\Big)^\top \tilde{C} \Big(\boldsymbol{\psi}(y_1) - \boldsymbol{\psi}(y_2)\Big) \\ &< \Big(\boldsymbol{\psi}(y_1) - \boldsymbol{\psi}(y_3)\Big)^\top \tilde{C} \Big(\boldsymbol{\psi}(y_1) - \boldsymbol{\psi}(y_3)\Big). \end{aligned} \tag{4.12}$$

In Chapter 5, we exploit this idea to impose structure on prediction models, in particular in structured multi-label prediction.

While, alignment models can encode between-set prior knowledge in a similar way:

$$\boldsymbol{\phi}(x_1)^\top M \boldsymbol{\psi}(y_1) < \boldsymbol{\phi}(x_1)^\top M \boldsymbol{\psi}(y_2), \tag{4.13}$$

encoding in-set information is not straightforward in the alignment representation (3.12) without losing convexity.

On the other hand, alignment models enjoy smaller number of parameters $O(mn)$ compared to distance models $O(m^2+n^2+mn)$. Hence learning an alignment models is supposed to need less data.

Finally, the semantic of score in alignment models and distance models are different. Alignment models tend to match angles while distance models tend to lower distances between associated items.

**Convex heterogeneous metric learning framework** Finally, based on the new representation proposed, one can extend the general metric learning framework (4.2) 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 $\mathbf{f}(X, Y)$ as,

$$\mathbf{f}(X, Y) = \big[\phi(X)^\top, -\widehat{\psi}(Y)^\top, -\mathbf{1}, -\mathbf{1}\big]^\top \in \mathbb{R}^{t \times (n+m+2)}, \tag{4.14}$$

where $\mathbf{1}$ denotes an all-one vector (of dimension $t$ in this case), we propose to find the matrix $C$ by solving

$$\min_{C \in \mathbb{R}^{p \times p}, C \succeq 0} \sum_i L_i\big(\mathbf{f}(X, Y)^\top C\, \mathbf{f}(X, Y)\big) + \beta\, \mathrm{reg}(C)\ . \tag{4.15}$$

Similar to the alignment model of Chapter 3, using general convex loss functions $L_i$ in (4.15) makes the training loss function totally convex in the optimization parameter $C$.

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, due to an outer optimization over a dual variable $\boldsymbol{\alpha}$, while (4.15) leads to a convex optimization problem when the losses $L_i$ and the regularizer are convex and $\beta \geq 0$.

**Regularization** Regularization is an important and standard consideration in metric learning, since the risk of overfitting is ever present. We select the most widely used regularizer, the Frobenius norm, which, interestingly, if applied to the factors yields the trace norm regularizer on $C$,

$$\|U\|_F^2 + \|V\|_F^2 + \|\mathbf{u}_0\|_F^2 + \|\mathbf{v}_0\|_F^2 = \mathrm{tr}(C) = \|C\|_{\mathrm{tr}}, \tag{4.16}$$

where the trace norm $\|.\|_{tr}$ (also known as nuclear norm) of a matrix is the sum of its singular values. For a square matrix, the trace $\mathrm{tr}(.)$ is the sum of the elements on its main diagonal. Crucially, the equality (4.16) allows one to optimize over $C$ directly without considering the implicit $U$, $V$, $\mathbf{u}_0$ or $\mathbf{v}_0$ components. 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 non-smooth trace norm simplifies to $\|C\|_{\mathrm{tr}} = \mathrm{tr}(C)$, a smooth function which allows efficient optimization.

## 4.3 Algorithm

In this section we propose a scalable training algorithm, the Iterative Local Algorithm (ILA). After presenting the goal, we describe the rough idea and then proceed to the formal statement of the algorithm and the theory behind it. The section concludes with the proof of convergence of the ILA algorithm.

### 4.3.1 The Goal

Given the convex training problem (4.15) and the regularizer in (4.16), the immediate question is how to efficiently solve it.

First note that, using the loss formulation

$$L(C) = \sum_i L_i\Big( \mathbf{f}(X,Y)^\top C \mathbf{f}(X,Y) \Big),$$

and the common regularizer $\mathrm{tr}(C)$, the training objective can be written as

$$\min_{C \in \mathbb{R}^{p \times p}, C \succeq 0} f(C) \qquad \text{where } f(C) = L(C) + \beta \,\mathrm{tr}(C). \tag{4.17}$$

One way to encode the semidefinite constraint is via a change of variable $C = QQ^\top$:

$$\min_{Q \in \mathbb{R}^{p \times d}} f(QQ^\top) = \min_{Q \in \mathbb{R}^{p \times d}} L(QQ^\top) + \beta \,\mathrm{tr}(QQ^\top). \tag{4.18}$$

This optimization, however, becomes non-convex in $Q$. The reason is that convexity of a function $L$ is not preserved with respect to the variable $Q$ after composing with a quadratic function of that variable, i.e. $QQ^\top$.

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 (4.18) with $\mathrm{rank}(Q) < d$, then $QQ^\top$ is a global optimum of (4.17). This is useful, since once conditions are satisfied, enables one to apply a single local search and find the globally optimal solution. In what follows, $C^*$ will denote an optimum of (4.17) 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 a consequence of the following, more general result.

### 4.3.2 General idea

Finding global minima of a convex function (such as the bowl-shaped function in Figure 4.2 (left)) is straightforward, since any local minimum of a convex function

Figure 4.2: Convex optimization (left) versus non-convex optimization (right) for metric learning or distance based co-embedding

must also be a global minimum. In particular, for a smooth function, one can simply apply gradient descent or any efficient local minimization method that can find a local minimum. By contrast, finding a global minimum of a non-convex function is a hard problem in general.

Recently, however, there has been renewed interest in developing algorithms that can efficiently find global minima for certain non-convex objectives (see Figure 4.2 (right)) , and some recent advances have been achieved. A specific example is optimizing the non-convex objective function in (4.18). Here, I first explain the basic idea of how Journée et al. (2010) solve such a problem globally, then explain a proposed improvement. For the purpose of these explanations, I exploit the toy graphs in Figure 4.2.

Suppose one seeks the global minimum of a non-convex function with the form shown on the right of Figure 4.2. Journée et al. (2010) suggest an iterative approach where, starting from an initial point, a local search is performed until a critical point is reached (depicted by the red dot shown in right graph of Figure 4.2). For the problems of interest, we will establish that, for any critical point on the boundary of the rank constraint that is not a global minimizer, a descent direction is guaranteed to be available and easily recoverable. Notably, computing this descent direction will not require computing the Hessian, which is usually too expensive even to store. Once a direction has been identified, the process of escaping the current point consists of adding a suitable column to the factor $Q$ of $C$, as shown in Figure 4.3.

By repeatedly escaping boundary saddle points, a critical point will eventually be reached where a descent direction is no longer available; in such a case, for the

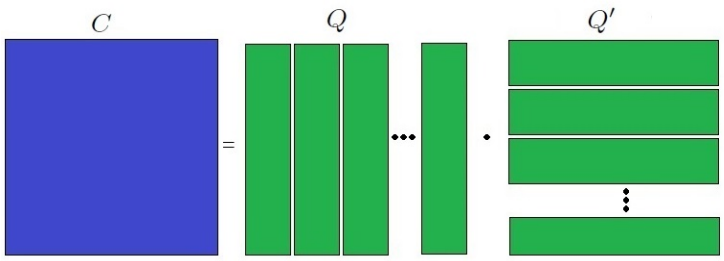


Figure 4.3: Iteratively inserting columns to factors $Q$ of a positive semi-definite matrix $C$, where $C = QQ^\top$.

specific objectives we consider below, Journée et al. (2010) have proved that such a point must be a global minimum. The main drawback with this strategy is that each local search for a critical point can be expensive, and the number of such searches can be large.

Therefore, to reduce the overall number of iterations, a natural idea is to add more than one column to $Q$ in each iteration; in particular, we consider doubling the number of columns added in successive updates. Similar to binary search, such a strategy is intended to reduce the overall number of column expansions needed to find the target solution from linear to logarithmic, while simultaneously exploiting the fact that the local optimization is more efficient when the $Q$ matrix has fewer columns. In the next section, I therefore develop a strategy for generating a guaranteed descent direction that consists of $k$ columns. We will be able to detect when a sufficient number of columns has been added so that further descent is no longer possible, and a global minimum found. Section 4.3.3 formalizes these statements and proves them.

An important technicality we consider is that determining whether a critical point is a saddle point or a local minimum depends on the domain of the function. For example, a point in the domain $A = \{M | \operatorname{rank}(M) \leq r\}$ could achieve a local minimum but at the same time be a saddle point in a larger domain $B$ where $B = \{M | \operatorname{rank}(M) \leq r + 1\} \supset A$. Such a case only happens if the saddle point is located on the boundary of $A$, which corresponds to the constraint being active; that is, the rank of the matrix is exactly $r$ and not smaller; see Figure 4.4 for

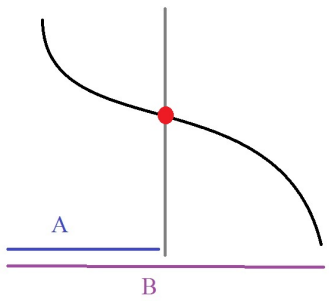


Figure 4.4: Local minimum of the inner domain $A$ happens to be a saddle point of the outer domain $B$ for $A \subset B$

an illustration. It is important to differentiate between these two types of critical points in the upcoming argument. For example, although a local descent search initiated from a randomly generated point is unlikely to settle on an interior saddle point, our experiments show that local descents often converge to boundary points that are saddles in a larger domain, such as illustrated in Figure 4.4.

### 4.3.3 Formal statement

In this section, I first establish some of the key technical claims that are required before proposing the specific algorithm.

**Proposition 1** *Consider any local solution $Q$ of the optimization problem* (4.18)*, i.e. a $Q$ such that $\nabla L(QQ^\top)Q + \beta Q = 0$. Let $\mathbf{u}_1, ..., \mathbf{u}_k$ be the eigenvectors corresponding to the top $k$* **positive** *eigenvalues $\lambda_1, ..., \lambda_k$ of $-\nabla L(C) - \beta I$, for $C = QQ^\top$. Then, if $C$ is not a solution to* (4.17)*, it follows that*

1. *$k > 0$ (i.e. for a non-optimal solution $C$ of* (4.17)*, at least one eigenvalue of $-\nabla f(C)$ is positive),*

2. *Eigenvectors $\mathbf{u}_1, ..., \mathbf{u}_k$ are orthogonal to $Q$, yielding $Q_k = [Q\ \mathbf{u}_1\ ...\ \mathbf{u}_k]$ such that $C_k = Q_k Q_k^\top = C + \sum_{i=1}^k \mathbf{u}_i \mathbf{u}_i^\top$ satisfies $\mathrm{rank}(C_k) = \mathrm{rank}(C) + k$; and*

3. *the descent direction $\sum_{i=1}^k \mathbf{u}_i \mathbf{u}_i^\top$ is the solution to*

$$\underset{\substack{\|\mathbf{u}_i\| \leq 1, i=1,...,k \\ \mathbf{u}_i^\top \mathbf{u}_j = 0, i \neq j,\ \mathbf{u}_i \neq 0}}{\operatorname{argmin}} \left\langle -\nabla L(C) - \beta I, \textstyle\sum_{i=1}^k \mathbf{u}_i \mathbf{u}_i^\top \right\rangle. \tag{4.19}$$

See Appendix B.1 for the proof of the proposition.

Proposition 1 has a simple interpretation: Part 1 introduces a certificate of global optimality for the optimization problem in (4.17), which will be used to design a stopping criterion for the algorithm.

Part 2 identifies a strategy for selecting an initial point for the local optimization of Problem (4.18) provided that the rank has been increased after encountering a critical point. This is needed to restart the local search for the next critical point after reaching a rank-constrained critical point in the current iteration. The new initial point is generated by appending the new columns $\mathbf{u}_1$ ... $\mathbf{u}_k$ (eigenvectors) to the current local solution $Q$.

Part 3 shows that the proposed direction $\sum_{i=1}^{k} \mathbf{u}_i \mathbf{u}_i^\top$ is in fact, a descent direction of the objective (4.17) at point $C = QQ^\top$. In other words, taking a sufficiently small step from $C$ in the proposed direction is guaranteed to decrease the value of the objective (4.17), verifying that progress is made toward a global minimum at every iteration (if the local search is performed properly).

**Corollary 1** *Let $Q \in \mathbb{R}^{p\times d}$. If*

*(i) $Q$ is a local minimum of $f(QQ^\top)$ with $\operatorname{rank}(Q) < d$, or*

*(ii) $Q$ is a critical point of $f(QQ^\top)$ with $\operatorname{rank}(Q) = p$,*

*then $QQ^\top$ is a solution of* (4.17)*.*

Corollary 1 characterizes the conditions on the rank of a local solution that imply it is globally optimal. The optimization can be halted as soon as one of these conditions is fulfilled; see Appendix B.2 for a proof.

To efficiently solve (4.17), we now propose the Iterative Local Algorithm (ILA) shown in Algorithm 1. ILA iteratively adds groups of columns to an initially empty $Q$ and performs a local optimization over $Q \in \mathbb{R}^{p\times d}$ until convergence; see Figure 4.3 for an illustration.

The main advantage of this approach over simply setting $d = p$ is that good initial points are generated, and incrementally growing $d$ optimizes over much smaller $Q$ variables. Furthermore, one expects that when the number of columns $d$ of $Q_{init}$

**Algorithm 1** Iterative local algorithm (ILA)

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

is at least $d^*$, ILA will find 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, similar 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 $i$th 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 $d^*$.[4]

[4] 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.

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 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 immediately implies ILA solves (4.17) when the local optimizer avoids interior 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* (4.17) *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 at most $\log(p)$ iterations when the local optimizer avoids interior 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 an interior saddle point. 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 this is possible, because there is no guarantee that the rank of $Q$ is not reduced in the local optimization step. If the rank reduction happens and $Q$ is a local optimum, $QQ^\top$ 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 interior saddle points have not been considered in previous papers. On the contrary, below we show that that ILA is still consistent under mild technical conditions on $L$, even if the local optimizer can get trapped at interior saddle points.

**Proposition 2** *Suppose that $f$ is $\nu$-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 $C \succeq 0$. A local optimizer in line 12, i.e. an optimizer that returns a critical point $Q$ such that $\nabla f(QQ^\top)Q = 0$, the matrix $QQ^\top$ in ILA converges to the globally optimal solution of* (4.17).
(See Appendix B.3 for a proof.)

## 4.4 Empirical Computational Efficiency

To compare the exponential versus linear rank expansion strategies for ILA we first consider a standard metric learning problem. In this experiment, 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

$$\min_{C\succeq 0} \|X^\top X - X^\top C X\|_F^2 + \beta \operatorname{tr}(C) \tag{4.20}$$

with a regularization parameter $\beta = 0.5$.

Figure 4.5 compares the run times of the linear versus exponential expansion strategies, both of which optimize over $Q$ of increasing width rather than $C = QQ^\top$. Both methods used the same local optimizer but differed in how many new columns were 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. 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.

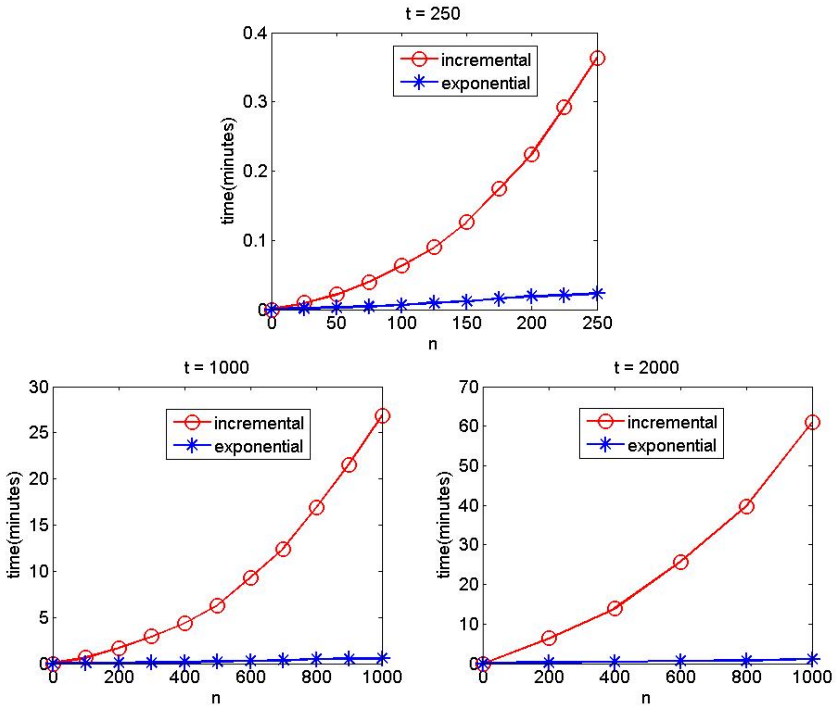


Figure 4.5: Comparing the run time in minutes (y-axis) of linear versus exponential strategies in ILA as data dimension (x-axis) is increased. Top shows $t = 250$, bottom left shows $t = 1000$, and bottom right shows $t = 2000$.

## 4.5 Case Study: Multilabel Prediction

Next, we evaluated ILA on the challenging setting of multilabel classification with real data, as in Section 3.3. Recall that, in this setting one can view the labels themselves as objects to be co-embedded with data instances; given such an embedding, the multilabel 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 $\mathbf{y}$.

We can cast multilabel learning as an equivalent *metric learning* problem, where one learns the matrix $C$. Following the development in Section 4.2 (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 $Q = \big[U,\ V,\ \mathbf{u}_0\big] \in \mathbb{R}^{p \times d}$, where $p = n + m + 1$. Then, as in (4.8), the co-embedding problem of optimizing $U$, $V$ and $\mathbf{u}_0$ can be equivalently expressed as a metric learning problem of optimizing the matrix $C = QQ^\top \in \mathbb{R}^{p \times p}$.

**Training objective** To develop a novel metric learning based approach to multilabel 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 adapt the convex large margin loss (3.22) used for this purpose in Section (3.3) as below.

$$\min_{C \succeq 0} \beta \operatorname{tr}(C) + \sum_{x \in \mathcal{X}} \Big[ \operatorname*{softmax}_{y \in \mathcal{Y}(\mathbf{x})} \tilde{L}\big(d_C(\mathbf{x}, \mathbf{y}) - t_C(\mathbf{x})\big) + \operatorname*{softmax}_{\bar{\mathbf{y}} \in \bar{\mathcal{Y}}(\mathbf{x})} \tilde{L}\big(t_C(\mathbf{x}) - d_C(\mathbf{x}, \bar{\mathbf{y}})\big) \Big], \quad (4.21)$$

where $\operatorname{softmax}$ is defined as

$$\operatorname*{softmax}_{y \in \mathcal{Y}}(z_y) = \ln \textstyle\sum_{y \in \mathcal{Y}} \exp(z_y),$$

and we have

$$t_C(x) = \Big[\phi(\mathbf{x}),\, \mathbf{0},\, -1\Big]\; C\; \Big[\phi(\mathbf{x}),\, \mathbf{0},\, -1\Big]^\top,$$
$$d_C(\mathbf{x}, \mathbf{y}) = \Big[\phi(\mathbf{x}),\, -\psi(\mathbf{y}),\, 0\Big]\; C\; \Big[\phi(\mathbf{x}),\, -\psi(\mathbf{y}),\, 0\Big]^\top.$$

**Results** Here we use $\mathcal{Y}(\mathbf{x}) \subset \mathcal{Y}$ 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 the non-convex form of (4.21) used in Line 12 of Algorithm 1, we use Frobenius norm regularization on the co-embedding parameters $U$, $V$ and $\mathbf{u}_0$, which was shown in Section 4.2 to yield trace regularization of $C$,

$$\|U\|_F^2 + \|V\|_F^2 + \|\mathbf{u}_0\|_2^2 = \mathrm{tr}(U^\top U) + \mathrm{tr}(V^\top V) + \mathbf{u}_0^\top \mathbf{u}_0 = \mathrm{tr}(C).$$

We investigate the behavior of ILA on the multilabel classification data sets that we summarized in Table 3.1 of Section 3.3. To establish the suitability of metric learning for multilabel classification, we evaluated test performance using three commonly used criteria for multilabel classification: the Hamming score (Table 4.1), micro averaged F1 measure (Table 4.2) and macro averaged F1 measure (Table 4.3). We chose $\beta$ by cross-validation over $\{1, 0.5, 0.1, 0.05, 0.01, 0.005\}$. Next, 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., 2009b); 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 4.1–4.3 are averaged over 10 splits and demonstrate comparable performance to the best competitors consistently in all three criteria for all data sets.

Next, to illustrate the distribution of objective values reached at local minima, as the rank of $Q$ is changed, we ran local optimization from 1000 random initializations of $Q$ at successive values $d = r$ of the number of columns of $Q$, using $\beta = 1$. The objective values at the local optima we observed are plotted in Figure 4.6 as a

| | 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 4.1: Comparison of ILA with competitors in terms of Hamming score, showing average over 10 splits ± standard deviation.

| | 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 4.2: Comparison of ILA with competitors in terms of Micro F1, showing average over 10 splits ± standard deviation.

function of $d$. Notably, all local minima achieved for $d = r$, are not larger in value than local minima achieved for the relaxed problem defined by $d = r + 1$, for any $r$ investigated. Moreover, as expected from the theory, the local optimizer always achieves the globally optimal value when $d \geq d^*$. 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 $d \geq d^*$, 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 4.6, which traps the optimization in a saddle point. Overall, empirically and theoretically, we find that ILA quickly finds global solutions for the multilabel objective, while typically producing good solutions before $d = d^*$.

| | 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 4.3: Comparison of ILA with competitors in terms of Macro F1, showing average over 10 splits ± standard deviation.

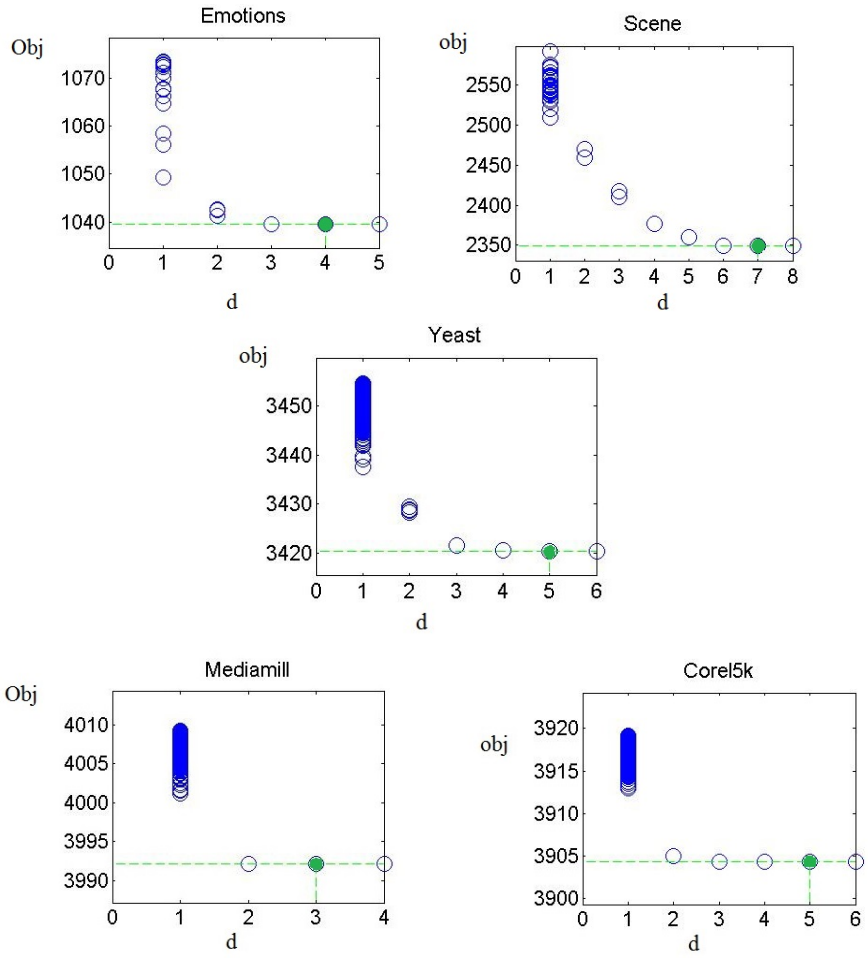


Figure 4.6: Illustrating the distribution of training objective values at locally optimal solutions. The plots show training objective values achieved by local optimization shown given 1000 initializations of $Q$ for different number of columns $d$ of $Q \in \mathbb{R}^{p \times d}$. For small $d$ a 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 $d = d^*$, where $d^* = [4, 7, 5, 3, 5]$ respectively.

## 4.6 Case Study: Tag Recommendation

Again, we explored Task 2 of the 2009 ECML/PKDD Discovery Challenge: a multi-relational problem involving users, items and tags. Recall the learning scenario from Section 3.4: users have tagged subsets of the items and the task is to recommend tags to them for other items. Again the training data is given in form of a tensor $E$, where $E(x, y, z) = 1$ indicates that $x$ has tagged $z$ with $y$, $E(x, y, z) = -1$ indicates that $y$ is not a tag of $z$ according to $x$, and $E(x, y, z) = 0$ 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.

**Training Objective** 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, we summarize the three-way interaction between a user, item and tag by the sum of 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, for a given user-item pair $(x, z)$, tags can be predicted via

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

The training problem can be expressed as metric learning by exploiting a construction reminiscent of Section 4.2: the embedding vectors can conceptually be stacked in matrix factor $Q = \begin{bmatrix} \sigma, & \tau, & \rho \end{bmatrix}^\top$, to define a matrix $C = QQ^\top$. To learn $C$, we use the same loss $L$ proposed by Rendle and Schmidt-Thieme (2009) and used in Section 3.4, regularized by the Frobenius norm over $\sigma$, $\tau$ and $\rho$ (which again corresponds to trace regularization of $C$), yielding the convex training problem

$$\min_{C \succeq 0} \quad \beta \operatorname{tr}(C) + \sum_{x,z} \sum_{y \in tag(x,z)} \sum_{\bar{y} \notin tag(x,z)} L\big(d_C(x, z, \bar{y}) - d_C(x, z, y)\big). \tag{4.22}$$

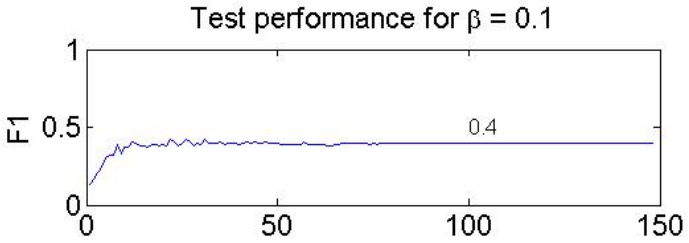


Figure 4.7: F1 measure achieved by ILA on test data with an increasing number of columns. Optimal rank is 84 in this case.

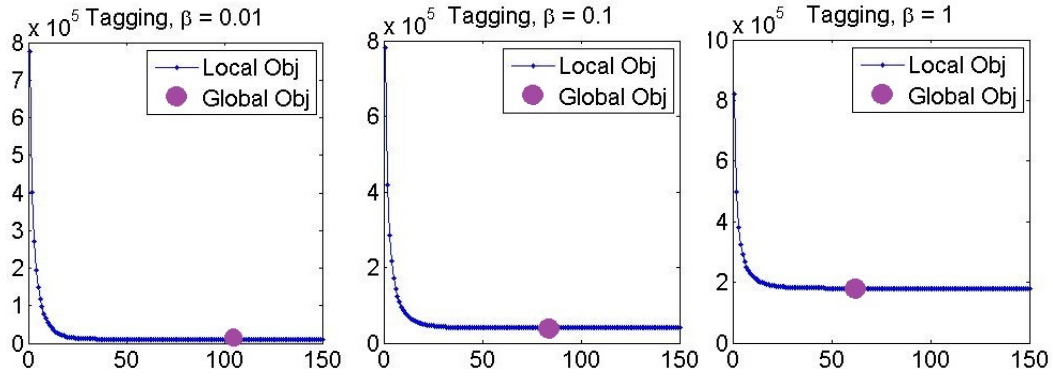


Figure 4.8: Training objectives for $\beta \in \{0.01, 0.1, 1\}$ as a function of the rank of $C$, where the optimal ranks are $105$, $84$ and $62$ respectively.

**Results** To establish the suitability of metric learning for multi-relational prediction, we evaluated the test performance achieved on the down-sampled data of Section 3.4 with $109$, $192$, $229$ unique users, items, and tags respectively. Figure 4.7 shows that ILA efficiently approaches state of the art F1 performance of $0.42$ reported in Section 3.4.

Next, we 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 4.8 shows that although the optimal rank can be larger in this scenario, the properties of the local solutions become even more apparent: 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.

## 4.7 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.

# Chapter 5

# Eliminating Inference from Structured Multilabel Prediction

Structured output prediction has been an important topic in machine learning. Many prediction problems involve complex structures, such as predicting parse trees for sentences (Taskar, 2004), predicting sequence labelings for language and genomic data (Bakir et al., 2007), or predicting multilabel taggings for documents and images (Deng et al., 2010, 2014; Joachims, 1999; Lewis et al., 2004).

Initial breakthroughs in this area arose from tractable discriminative training methods—conditional random fields (Lafferty et al., 2001; Sun, 2014) and structured large margin training (Srikumar and Manning, 2014; Taskar et al., 2003; Tsochantaridis et al., 2005)—that compare complete output configurations against given target structures rather than simply learning to predict each component in isolation.

More recently, search based approaches that exploit sequential prediction methods have also proved effective for structured prediction (Doppa et al., 2012; Daume and Langford, 2009; Li et al., 2013a; Weiss and Taskar, 2013). Despite the improvements contributed by these approaches, the need to conduct inference or search over complex outputs both during the training and testing phase proves to be a significant bottleneck in practice.

In this chapter, we investigate an alternative approach to structured output prediction based on co-embedding that eliminates the need for inference or search at test time. The idea is to shift the burden of coordinating predictions to the train-

ing phase, by pre-compiling constraints in the learned representation that ensure prediction relationships are satisfied. The primary benefit of this approach is that prediction cost can be significantly reduced without sacrificing the desired coordination of structured output components. Since prediction phase is the recurring step in learning systems and typically requires quick if not real time response, reducing the prediction time would be beneficial in practice.

We demonstrate the proposed approach concretely for the problem of *multilabel classification* with hierarchical and mutual exclusion constraints on output labels (Deng et al., 2014). Multi-label classification is an important subfield of structured output prediction where multiple labels must be assigned to a single object that respect semantic relationships such as subsumption, mutual exclusion or weak forms of correlation. The problem is of growing importance as larger tag sets are being used to annotate images and documents on the Web. Research on multi-label classification has focused on how to improve independent label classification (Joachims, 1999) by incorporating dependence information between labels, distinguished by whether they exploit known relationships between the labels or have to infer or adapt to such relationships without explicit prior knowledge.

In the latter case, many works have developed tailored training losses for multilabel prediction that penalize joint prediction behavior (Crammer and Singer, 2003; Dembczyński et al., 2012, 2013a; Elisseeff and Weston, 2001; Mencía and Fürnkranz, 2008; Guo and Schuurmans, 2011; Tsoumakas et al., 2009) without assuming any specific form of prior knowledge. Li et al. (2013b) use Restricted Boltzmann Machines (RBMs) to infer high order relations between labels in the context of image segmentation. More recently, several works have focused on coping with large label spaces by using low dimensional projections to label subspaces (Bi and Kwok, 2013; Chen et al., 2013a; Chen and Lin, 2012; Cissé et al., 2013; Hsu et al., 2009; Kapoor et al., 2012; Lin et al., 2014). Belanger and McCallum (2016) exploit a deep architecture to capture dependencies between labels that leads to intractable graphical models, and perform structure learning by automatically learning features of the structured output. Other works have focused on exploiting weak forms of prior knowledge expressed as similarity information between labels that can be

obtained from auxiliary sources (Akata et al., 2013; Hariharan et al., 2012; Ji et al., 2010).

Unfortunately, none of these approaches strictly enforce prior logical relationships between label predictions. By contrast, other research has sought to exploit known prior relationships between labels. The most prominent such approaches have been to exploit generative or conditional graphical models over the label set (Dembczynski et al., 2010), (Jin and Ghahramani, 2002; Kae et al., 2013), and (Ueda and Saito, 2002). Unfortunately, the graphical model structures that can be imposed are limited to junction trees with small treewidth (Dembczynski et al., 2010). When general structure is possible the score function would be limited to discrete and sub-modular functions, so that inference can performed tractably via efficient graph cut or other algorithmic approaches (Kohli and Torr, 2007; Tarlow et al., 2011; Kolmogorov and Zabih, 2002). Here, the definition of submodularity requires the label set to be a totally ordered set. This condition is not generally applicable (Li and Huber, 2017). Other graphical models require approximation (Jancsary et al., 2013; Marchand et al., 2014; Petterson and Caetano, 2011). Other work, using output kernels, has also been shown able to model complex relationships between labels (Dinuzzo and Fukumizu, 2011; Kadri et al., 2013) but is hampered by an intractable pre-image problem[1] at test time, unless the kernels are restricted and special losses are used (Guo and Schuurmans, 2013).

In this chapter, we focus on tractable methods and consider the scenario where a set of logical label relationships is given *a priori*; in particular, implication and mutual exclusion relationships that arise naturally in document and image tagging scenarios. These relationships have been the subject of extensive work on multilabel prediction, where it is known that if the implication/subsumption relationships form a tree (Rousu et al., 2006) or a directed acyclic graph (Bi and Kwok, 2012, 2011; Deng et al., 2014) then efficient dynamic programming algorithms can be developed for tractable inference during training and testing, while for general pairwise models (Weston et al., 2010, 2011) approximate inference is required. The

[1]Pre-image problem in the context of output kernels is the problem of mapping back from the kernel to the output set.

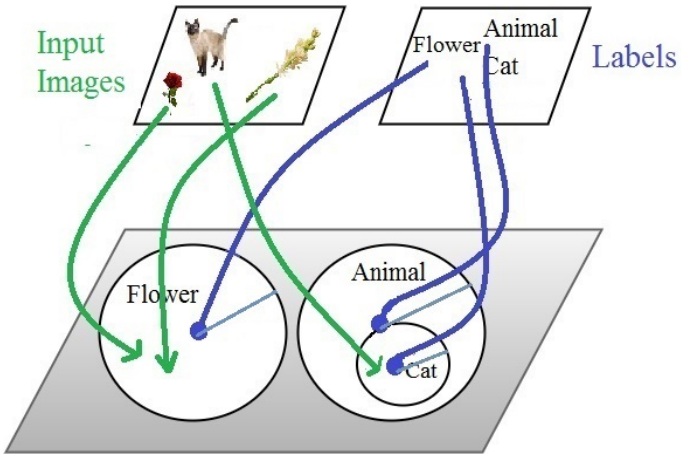


Figure 5.1: Embedding constraints for multilabel prediction: A Venn diagram is formed in embedding space over labels to impose hierarchy and exclusion constraints.

main contribution of this chapter is to show how these relationships can be enforced without the need for dynamic programming. The idea is to impose label relationships as constraints on the underlying score model during training so that a trivial labeling algorithm can be employed at test time, a process that can be viewed as pre-compiling inference during the training phase.

The focus of literature has been on many other relevant topics not addressed by this chapter including learning from incomplete labelings (Gentile and Orabona, 2014; Wu et al., 2011; Xu et al., 2013; Yu et al., 2014), exploiting hierarchies for multiclass rather than multilabel prediction (Bengio et al., 2010; Deng et al., 2011; Gao and Koller, 2011; Weinberger and Chapelle, 2008), exploiting multimodal data to improve prediction (Frome et al., 2013; Socher et al., 2013a), deriving generalization bounds for structured output and multilabel prediction problems (Giguère et al., 2013; London et al., 2013; Punyakanok et al., 2005), and investigating the consistency of multilabel losses (Dembczyński et al., 2013b; Gao and Zhou, 2013). Another interesting large margin approach that applies multilabel prediction over structured outputs is given by (Lampert, 2011). On the other hand, it might worth mentioning that following the proposed method in Mirzazadeh et al. (2015a), a related method is also proposed in Vendrov et al. (2016) independently via a co-embedding-based approach for hierarchical multilabel prediction.

Recall the intuition underlying the proposed method described in Chapter 1, which was based on considering a multi-label image tagging problem with labels "animal", "flower", and "cat" tags. The idea is to ensure that the decision regions for the labels are constrained to match a conceptual Venn diagram that expresses the desired logical constraints between the labels, in terms of their inclusion and exclusion relationships. See Figure 5.1 for a demonstration of the idea.

The main contributions of this chapter are the following.

**Contribution 12** *We observe that inference in structured output prediction can be completely eliminated by embedding the logical relationships between labels directly into the score model.*

**Contribution 13** *As a concrete demonstration of this idea for multi-label prediction models, we show that implication and mutual exclusion relationships can be easily embedded in the score model, ensuring the constraints hold over the whole domain while maintaining convexity in model parameters.*

**Contribution 14** *We show that inference is not necessary, either at train or test time, for structured multi-label classification, even when logical relationships between labels are strictly enforced.*

**Contribution 15** *By exploiting these ideas, we show that the efficiency of the resulting structured multi-label predictor can be dramatically improved.*

## 5.1 Preliminaries

### 5.1.1 Structured Output Prediction

We consider a standard prediction model where a score function $s$,

$$s : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$$

with parameters $\theta$ is used to determine the prediction for a given input $\mathbf{x}$ via

$$\hat{\mathbf{y}} \quad = \quad \arg\max_{\mathbf{y} \in \mathcal{Y}} s(\mathbf{x}, \mathbf{y}). \tag{5.1}$$

Here $\mathbf{y}$ is a *configuration* of assignments over a set of components (that might depend on $\mathbf{x}$). Since $\mathcal{Y}$ is a combinatorial set, (5.1) cannot usually be solved by enumeration; some structure is required for efficient prediction. For example, $s$ might decompose as $s(\mathbf{x}, \mathbf{y}) = \sum_{c \in \mathcal{C}} s(\mathbf{x}, \mathbf{y}_c)$ over a set of cliques $\mathcal{C}$ that form a junction tree, where $\mathbf{y}_c$ denotes the portion of $\mathbf{y}$ covered by clique $c$. Furthermore, $\mathcal{Y}$ might encode constraints to aid tractability, such as $\mathbf{y}$ forming a consistent matching in a bipartite graph, or a consistent parse tree (Taskar, 2004). The key practical requirement is that $s$ and $\mathcal{Y}$ allow an efficient solution to (5.1). The operation of maximizing or summing over all $\mathbf{y} \in \mathcal{Y}$ is referred to as *inference*, and usually involves a dynamic programming step tailored to the specific structure encoded by $s$ and $\mathcal{Y}$.

For supervised learning one attempts to infer a useful score function given a set of $t$ training pairs $(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), ..., (\mathbf{x}_t, \mathbf{y}_t)$ that specify the correct output associated with each input. The training phase for conditional random fields (Lafferty et al., 2001) and structured large margin learning (below with margin scaling) (Taskar et al., 2003; Tsochantaridis et al., 2005; Lampert, 2011) can both be expressed as optimizations over the score model parameters $\theta$ respectively:

$$\min_{\theta \in \Theta} \quad r(\theta) + \sum_{i=1}^{t} \log \Big( \sum_{\mathbf{y} \in \mathcal{Y}} \exp(s_\theta(\mathbf{x}_i, \mathbf{y})) \Big) - s_\theta(\mathbf{x}_i, \mathbf{y}_i) \tag{5.2}$$

$$\min_{\theta \in \Theta} \quad r(\theta) + \sum_{i=1}^{t} \max_{\mathbf{y} \in \mathcal{Y}} \Big( \Delta(\mathbf{y}, \mathbf{y}_i) + s_\theta(\mathbf{x}_i, \mathbf{y}) \Big) - s_\theta(\mathbf{x}_i, \mathbf{y}_i), \tag{5.3}$$

where $r(\theta)$ is a suitable regularizer over $\theta \in \Theta$. Equations (5.1), (5.2) and (5.3) suggest that inference over $\mathbf{y} \in \mathcal{Y}$ is required at each stage of training and testing, which typically raise scaling challenges. However, our goal is to show this is not necessarily the case whether to compute the training loss or (sub)gradient at an example at training time, or to compute a prediction at test time.

### 5.1.2 Structured Multilabel Prediction

To demonstrate how inference might be avoided, consider the special case of multi-label prediction with label constraints. Multi-label prediction specializes the previous set up by assuming $\mathbf{y}$ is a Boolean assignment to a fixed set of variables

$y_1, \ldots, y_\ell$, where each label is assigned 1 (true) or 0 (false), i.e.

$$\mathbf{y} = (y_1, y_2, ..., y_\ell), \quad y_k \in \{0, 1\}.$$

As noted, an extensive literature that has investigated various structural assumptions on the score function to enable tractable prediction. For simplicity we adopt the factored form that has been reconsidered in recent work (Deng et al., 2014; Hariharan et al., 2012) (and originally (Joachims, 1999)):

$$s(\mathbf{x}, \mathbf{y}) = \sum_k s(\mathbf{x}, y_k).$$

This form allows (5.1) to be simplified to

$$\hat{\mathbf{y}} \quad = \quad \arg\max_{\mathbf{y} \in \mathcal{Y}} \sum_k s(\mathbf{x}, y_k) \quad = \quad \arg\max_{\mathbf{y} \in \mathcal{Y}} \sum_k y_k s_k(\mathbf{x}) \tag{5.4}$$

where

$$s_k(\mathbf{x}) = s(\mathbf{x}, y_k = 1) - s(\mathbf{x}, y_k = 0)$$

gives the decision function associated with label $y_k \in \{0, 1\}$. That is, based on (5.4), if the constraints in $\mathcal{Y}$ were ignored, one would have the relationship

$$\hat{y_k} = 1 \Leftrightarrow s_k(\mathbf{x}) \geq 0.$$

The constraints in $\mathcal{Y}$ play an important role however: Deng et al. (2014) show that imposing prior implications and mutual exclusions as constraints in $\mathcal{Y}$ yields state of the art accuracy results for image tagging on the ILSVRC corpus (Russakovsky et al., 2015). This result was achieved in (Deng et al., 2014) by developing a novel and rather sophisticated dynamic program that can efficiently solve (5.4) under these constraints. Here we show how such a dynamic program can be eliminated.

## 5.2 Inserting Label Constraints into the Representation

The main contribution of this chapter is to observe that inference can be completely eliminated by embedding the logical relationships between labels directly into the

score model. In particular, we give a concrete demonstration of this idea for multilabel prediction models, showing that implication and mutual exclusion relationships can be easily embedded in the score model, ensuring the constraints hold over the whole domain while maintaining convexity in model parameters.

Consider the two common forms of logical relationships between labels: implication and mutual exclusion. For *implication* one would like to enforce relationships of the form $y_1 \Rightarrow y_2$, meaning that whenever the label $y_1$ is set to 1 (true) then the label $y_2$ must also be set to 1 (true). For *mutual exclusion* one would like to enforce relationships of the form $\neg y_1 \vee \neg y_2$, meaning that at least one of the labels $y_1$ and $y_2$ must be set to 0 (false) (i.e., not both can be simultaneously true). These constraints arise naturally in multilabel classification, where label sets are increasingly large and embody semantic relationships between categories (Bi and Kwok, 2012; Deng et al., 2014; Weston et al., 2011). For example, images can be tagged with labels "dog", "cat" and "Siamese" where "Siamese" implies "cat", while "dog" and "cat" are mutually exclusive (but an image could depict neither). These implication and mutual exclusion constraints constitute the "HEX" constraints considered in (Deng et al., 2014).

Our goal is to express the logical relationships between label assignments as constraints on the score function that hold universally over all $\mathbf{x} \in \mathcal{X}$. In particular, using the decomposed representation (5.4), the desired label relationships correspond to the following constraints

$$\text{Implication} \quad y_1 \Rightarrow y_2: \qquad s_1(\mathbf{x}) \geq -\delta \quad \Rightarrow \quad s_2(\mathbf{x}) \geq \delta \qquad \forall \mathbf{x} \in \mathcal{X} \quad (5.5)$$

$$\text{Mutual exclusion } \neg y_1 \vee \neg y_2: \qquad s_1(\mathbf{x}) < -\delta \quad \text{or} \quad s_2(\mathbf{x}) < -\delta \quad \forall \mathbf{x} \in \mathcal{X} \quad (5.6)$$

where we have introduced the additional margin quantity $\delta \geq 0$ for subsequent large margin training.

### 5.2.1 Score Model

The first key consideration is representing the score function in a manner that allows the desired relationships to be expressed. Unfortunately, the standard linear form $s(\mathbf{x}, \mathbf{y}) = \langle \theta, f(\mathbf{x}, \mathbf{y}) \rangle$ cannot allow the needed constraints to be enforced over

all $x \in \mathcal{X}$ without further restricting the form of the feature representation $f$, however, this is a constraint one needs to avoid since it rules out most natural feature representations.

More specifically, consider a standard set up where there is a mapping $f(\mathbf{x}, y_k)$ that produces a feature representation for an input-label pair $(\mathbf{x}, y_k)$. For clarity, we additionally make the standard assumption that the inputs and outputs each have independent feature representations (Hariharan et al., 2012), hence $f(\mathbf{x}, y_k)$ could be expressed as

$$f(\mathbf{x}, y_k) = \phi(\mathbf{x}) \otimes \psi_k$$

for an input feature map $\phi$ and label feature representation $\psi_k$ , where $\otimes$ is the Kronecker product. In this case, a score function expressed in bilinear form (in feature representations) has the form

$$s_k(\mathbf{x}) = \phi(\mathbf{x})^\top A \psi_k + b^\top \phi(\mathbf{x}) + c^\top \psi_k + d$$

for parameters $\theta = (A, b, c, d)$.

Unfortunately, such a score function does not allow $s_k(\mathbf{x}) \geq 0$ to be expressed over all $\mathbf{x} \in \mathcal{X}$ without either assuming $A = 0$ and $b = 0$, or special structure in $\phi$. The inability to express universal constraints on $s_k(\mathbf{x})$ that hold over all $\mathbf{x}$ is likely why such an approach has not been previously proposed in the literature.

To overcome this restriction we consider a more general scoring model that extends the standard bi-linear form to a form that is linear in the parameters but *quadratic* in the feature representations: This is a key step that allows the constraints to be embedded while retaining linearity in the score model parameters. In particular, we consider

$$-s_k(\mathbf{x}) = \begin{bmatrix} \phi(\mathbf{x}) \\ \psi_k \\ 1 \end{bmatrix}^\top \begin{bmatrix} P & A & \mathbf{b} \\ A^\top & Q & \mathbf{c} \\ \mathbf{b}^\top & \mathbf{c}^\top & r \end{bmatrix} \begin{bmatrix} \phi(\mathbf{x}) \\ \psi_k \\ 1 \end{bmatrix} \tag{5.7}$$

for

$$\theta = \begin{bmatrix} P & A & \mathbf{b} \\ A^\top & Q & \mathbf{c} \\ \mathbf{b}^\top & \mathbf{c}^\top & r \end{bmatrix}. \tag{5.8}$$

Here $\theta = \theta^\top$ and $s_k$ is linear in $\theta$ for each $k$. The benefit of a quadratic form in the features is that it allows constraints over $\mathbf{x} \in \mathcal{X}$ to be easily imposed on label scores via convex constraints on $\theta$.

**Lemma 1** *If $\theta \succeq 0$ then $-s_k(\mathbf{x}) = \|U\phi(\mathbf{x}) + \mathbf{u} - V\psi_k\|^2$ for some $U$, $V$ and $\mathbf{u}$.*

(See Appendix C.1 for a proof.)

The representation (5.7) generalizes both the standard bi-linear (alignment-based) and distance-based models. The standard bi-linear model is achieved by $P = 0$, $Q = 0$ and $r = 0$. By Lemma 1, the semidefinite assumption $\theta \succeq 0$ also yields a model that has a co-embedding interpretation: the feature representations $\phi(\mathbf{x})$ and $\psi_k$ are both mapped (linearly) into a common Euclidean space where the score is determined by the squared distance between the embedded vectors (with an additional offset $\mathbf{u}$).

To aid the presentation below we simplify this model a bit further. Set $\mathbf{b} = 0$ and observe that (5.8) reduces to

$$
\begin{aligned}
s_k(\mathbf{x}) &= \gamma_k - \begin{bmatrix} \phi(\mathbf{x}) \\ \psi_k \end{bmatrix}^\top \begin{bmatrix} P & A \\ A^\top & Q \end{bmatrix} \begin{bmatrix} \phi(\mathbf{x}) \\ \psi_k \end{bmatrix} && (5.9)\\
&= \gamma_k - \|U\phi(\mathbf{x}) - V\psi_k\|^2 && (5.10)\\
&= \gamma_k - \|\mu(\mathbf{x}) - \nu(\mathbf{y})\|^2, && (5.11)
\end{aligned}
$$

where the term $\gamma_k = -r - 2\mathbf{c}^\top\psi_k = -\mathbf{u}^\top\mathbf{u} - 2\mathbf{u}^\top V\psi_k$. can be interpreted as specifying a $\mathbf{y}$-dependent decision threshold over the squared distances and the form (5.11) provides a convenient shorthand that focuses on the squared distance between the embedding vectors $\mu(\mathbf{x})$ and $\nu(\mathbf{y})$ for $\mathbf{x}$ and $\mathbf{y}$ respectively. (Decision thresholds in distance-based co-embedding models are introduced in Section 4.2. For example, see Figure 4.1 .)

In particular, we modify the parametrization to $\theta = \{\gamma_k\}_{k=1}^{\ell} \cup \{\theta_{PAQ}\}$ such that $\theta_{PAQ}$ denotes the matrix of parameters in (5.9). Importantly, (5.9) remains linear in the new parametrization, which is essential for obtaining a convex training formulation. Lemma 1 can then be modified accordingly for a similar convex constraint on $\theta$.

**Lemma 2** *If $\theta_{PAQ} \succeq 0$ then there exist $U$ and $V$ such that for all labels $k$ and $l$*

$$s_k(\mathbf{x}) = \gamma_k - \|U\phi(\mathbf{x}) - V\psi_k\|^2 \quad (5.12)$$

$$\psi_k^\top Q\psi_k - \psi_k^\top Q\psi_l - \psi_l^\top Q\psi_k + \psi_l^\top Q\psi_l = \|V\psi_k - V\psi_l\|^2. \quad (5.13)$$

(See Appendix C.2 for a proof.)

This representation now allows us to embed the desired label relationships as simple convex constraints on the score model parameters $\theta$.

### 5.2.2 Implication Constraints

**Theorem 1** *Assume the quadratic-linear score model* (5.9) *and $\theta_{PAQ} \succeq 0$. Then for any $\delta \geq 0$ and $\alpha > 0$, the implication constraint in* (5.5) *is implied for all $\mathbf{x} \in \mathcal{X}$ by:*

$$\gamma_1 + \delta + (1+\alpha)\big(\psi_1^\top Q\psi_1 - \psi_1^\top Q\psi_2 - \psi_2^\top Q\psi_1 + \psi_2^\top Q\psi_2\big) \leq \gamma_2 - \delta \quad (5.14)$$

$$\left(\tfrac{\alpha}{2}\right)^2 \big(\psi_1^\top Q\psi_1 - \psi_1^\top Q\psi_2 - \psi_2^\top Q\psi_1 + \psi_2^\top Q\psi_2\big) \geq \gamma_1 + \delta. \quad (5.15)$$

(See Appendix C.3 for a proof.)

An illustration of the geometric interpretation for implication constraints is shown in Figure 5.2 (left). Implications constraints guarantee that the region in the embedding space assigned to the implying variable is inside the region assigned to the implied variable. The margin $\delta$ tends to prevent the boundary of regions from getting too close. In order to guarantee implication, for the implying label, the margin has to be outside of the region, while for the implied variable the margin has to be inside.

### 5.2.3 Mutual Exclusion Constraints

**Theorem 2** *Assume the quadratic-linear score model* (5.9) *and $\theta_{PAQ} \succeq 0$. Then for any $\delta \geq 0$ the mutual exclusion constraint in* (5.6) *is implied for all $\mathbf{x} \in \mathcal{X}$ by:*

$$\tfrac{1}{2}\big(\psi_1^\top Q\psi_1 - \psi_1^\top Q\psi_2 - \psi_2^\top Q\psi_1 + \psi_2^\top Q\psi_2\big) > \gamma_1 + \gamma_2 + 2\delta. \quad (5.16)$$

(See Appendix C.4 for a proof.)

Figure 5.2: Implication constraints (left) guarantee that the region assigned to an implying variable is inside the region assigned to the implied variable. Mutual exclusion constraints (right) guarantee that regions assigned to the two variables are mutually exclusive.

Importantly, once $\theta_{PAQ} \succeq 0$ is imposed, the other constraints in Theorems 1 and 2 are all *linear* in the parameters $Q$ and $\boldsymbol{\gamma}$.

An illustration of the geometric interpretation of mutual exclusive constraints appears in Figure 5.2 (right). Mutual exclusion constraints guarantee that the regions in embedding space assigned to mutually exclusive variables are in turn mutually exclusive. A margin of confidence guarantees that the two regions do not get too close. For both regions the margin (shown in red) must be outside the boundary to guarantee the desirable structure.

## 5.3 Properties

We now establish that the above constraints on the parameters in (5.9) achieve the desired properties. In particular, we show that given the constraints, inference can be removed both from the prediction problem (5.4) and from structured large margin training (5.3).

### 5.3.1 Prediction Equivalence

First note that the decision of whether a label $y_k$ is associated with $\mathbf{x}$ can be determined by

$$\begin{aligned} s(\mathbf{x}, y_k = 1) \geq s(\mathbf{x}, y_k = 0) \quad &\Leftrightarrow \quad \max_{y_k \in \{0,1\}} y_k s_k(\mathbf{x}) \geq 0 \\ &\Leftrightarrow \quad \arg\max_{y_k \in \{0,1\}} y_k s_k(\mathbf{x}) = 1. \end{aligned} \tag{5.17}$$

Consider joint assignments $\mathbf{y} = (y_1, ..., y_l) \in \{0,1\}^l$ and let $\mathcal{Y}$ denote the set of joint assignments that are consistent with a set of implication and mutual exclusion constraints. (It is assumed the constraints are satisfiable; that is, $\mathcal{Y}$ is not the empty set.) Then the optimal joint assignment for a given $\mathbf{x}$ can be specified by $\arg\max_{\mathbf{y}\in\mathcal{Y}} \sum_{k=1}^{l} y_k s_k(\mathbf{x})$.

**Proposition 1** *If the constraint set $\mathcal{Y}$ imposes the implication and mutual exclusion constraints in* (5.5) *and* (5.6) *(and is nonempty), and the score function $s$ satisfies the corresponding constraints for some $\delta > 0$, then*

$$\max_{\mathbf{y}\in\mathcal{Y}} \sum_{k=1}^{l} y_k s_k(\mathbf{x}) \quad = \quad \sum_{k=1}^{l} \max_{y_k} y_k s_k(\mathbf{x}) \tag{5.18}$$

(See Appendix C.5 for a proof.)

Since the feasible set $\mathcal{Y}$ embodies non-trivial constraints over assignment vectors in (5.18), interchanging maximization with summation is not normally justified. However, Proposition 1 establishes that, if the score model also satisfies its respective constraints (e.g., as established in the previous section), then maximization and summation *can* be interchanged, and inference over predicted labelings can be replaced by greedy componentwise labeling, while preserving equivalence.

### 5.3.2 Re-expressing Large Margin Structured Output Training

Given a target joint assignment over labels $\mathbf{t} = (t_1, ..., t_l) \in \{0,1\}^l$, and using the score model (5.9), the standard structured output large margin training loss (5.3) can then be written as

$$\sum_i \max_{\mathbf{y}\in\mathcal{Y}} \Delta(\mathbf{y}, \mathbf{t}_i) + \sum_{k=1}^{l} s(\mathbf{x}_i, y_k) - s(\mathbf{x}_i, t_{ik}) =$$

$$\sum_i \max_{\mathbf{y}\in\mathcal{Y}} \Delta(\mathbf{y}, \mathbf{t}_i) + \sum_{k=1}^{l} (y_k - t_{ik}) s_k(\mathbf{x}_i), \tag{5.19}$$

using the simplified score function representation such that $t_{ik}$ denotes the $k$-th label of the $i$-th training example. If we furthermore make the standard assumption that $\Delta(\mathbf{y}, \mathbf{t}_i)$ decomposes as $\Delta(\mathbf{y}, \mathbf{t}_i) = \sum_{k=1}^{l} \delta_k(y_k, t_{ik})$, the loss can be simplified to

$$\sum_i \max_{\mathbf{y}\in\mathcal{Y}} \sum_{k=1}^{l} \delta_k(y_k, t_{ik}) + (y_k - t_{ik}) s_k(\mathbf{x}_i). \tag{5.20}$$

Note also that since $y_k \in \{0,1\}$ and $t_{ik} \in \{0,1\}$ the margin functions $\delta_k$ typically have the form $\delta_k(0,0) = \delta_k(1,1) = 0$ and $\delta_k(0,1) = \delta_{k01}$ and $\delta_k(1,0) = \delta_{k10}$ for constants $\delta_{k01}$ and $\delta_{k10}$, which for simplicity we will assume are equal, $\delta_{k01} = \delta_{k10} = \delta$ for all $k$ (although label specific margins might be possible). This is the same $\delta$ used in the constraints (5.5) and (5.6).

The difficulty in computing this loss is that it apparently requires an exponential search over $\mathbf{y}$. When this exponential search can be avoided, it is normally avoided by developing a dynamic program. Instead, we can now see that the search over $\mathbf{y}$ can now be eliminated.

**Proposition 2** *If the score function $s$ satisfies the implication and mutual exclusion constraints in* (5.5) *and* (5.6)*, then*

$$\sum_i \max_{\mathbf{y}\in\mathcal{Y}} \sum_{k=1}^{l} \delta(y_k, t_{ik}) + (y_k - t_{ik}) s_k(\mathbf{x}_i)$$
$$= \sum_i \sum_{k=1}^{l} \max_{y_k} \delta(y_k, t_{ik}) + (y_k - t_{ik}) s_k(\mathbf{x}_i). \qquad (5.21)$$

(See Appendix C.6 for a proof.)

Similar to Section 5.3.1, Proposition 2 demonstrates that if the constraints (5.5) and (5.6) are satisfied by the score model $s$, then structured large margin training (5.3) reduces to independent labelwise training under the standard hinge loss, while preserving equivalence. That is, once again, inference can be entirely removed from consideration, although label coordination is still being considered in the constraints on $s$.

## 5.4 Efficient Implementation

Even though Section 5.2 achieves the primary goal of demonstrating how desired label relationships can be embedded as convex constraints on score model parameters, the linear-quadratic representation (5.9) unfortunately does not allow convenient scaling: the number of parameters in $\theta_{PAQ}$ (5.9) is $\binom{n+\ell}{2}$ (accounting for symmetry), which is quadratic in the number of features, $n$, in $\phi$ and the number of

labels, $\ell$. Such a large optimization variable is not practical for most applications, where $n$ and $\ell$ can be quite large. The semidefinite constraint $\theta_{PAQ} \succeq 0$ can also be costly in practice. Therefore, to obtain scalable training we require some further refinement.

In our experiments below we obtained a scalable training procedure by exploiting trace norm regularization on $\theta_{PAQ}$ to reduce its rank. The key benefit of trace norm regularization is that efficient solution methods exist that work with a low rank factorization of the matrix variable while automatically ensuring positive semidefiniteness and still guaranteeing global optimality (Haeffele et al., 2014; Journée et al., 2010). Therefore, we conducted the main optimization in terms of a smaller matrix variable $B$ such that $BB^\top = \theta_{PAQ}$. As shown in Chapter 4, provided that $B$ has sufficient rank, then any local solution is globally optimal based on works of (Haeffele et al., 2014; Journée et al., 2010). Second, to cope with the constraints, we employed an augmented Lagrangian method (Nocedal and Wright, 2006) that increasingly penalizes constraint violations, but otherwise allows simple unconstrained optimization. All optimizations for smooth problems were performed using LBFGS and non-smooth problems were solved using a bundle method (Mäkelä, 2003).

## 5.5 Experimental Evaluation

To evaluate the proposed approach, we conducted experiments on multilabel text classification data that has a natural hierarchy defined over the label set. In particular, we investigated three multilabel text classification data sets, that have hierarchical label sets with mutual exclusion constraints, and repeatable train/test splits. The data sets are Enron, WIPO and Reuters, obtained from `https://sites.google.com/site/hrsvmproject/datasets-hier` ; see Table 5.1 for details. Some preprocessing was performed on the label relations to ensure consistency with our assumptions. In particular, all implications were added to each instance to ensure consistency with the hierarchy, while mutual exclusions were defined between siblings whenever this did not create a contradiction.

| Dataset | Features | Labels | Depth | # Train | # Test | Reference |
|---|---|---|---|---|---|---|
| Enron | 1001 | 57 | 4 | 988 | 660 | (Klimt and Yang, 2004) |
| Wipo | 74435 | 183 | 5 | 1352 | 358 | (Rousu et al., 2006) |
| Reuters | 47235 | 103 | 5 | 3000 | 3000 | (Lewis et al., 2004) |

Table 5.1: Data set properties for constrained co-embedding experiments to precompile inference into representation

We conducted experiments to compare the effects of replacing inference with the constraints outlined in Section 5.2, using the score model (5.9). For comparison, we trained using the structured large margin formulation (5.3), and trained under a multilabel prediction loss without inference, but both including then excluding the constraints. For the multilabel training loss we used the smoothed calibrated separation ranking loss (3.22) of Chapter 3. In each case, the regularization parameter was simply set to 1. For inference, we implemented the inference algorithm outlined in (Deng et al., 2014).

| % test error | Enron | WIPO | Reuters |
|---|---|---|---|
| unconstrained | 12.4 | 21.0 | 27.1 |
| constrained | 9.8 | 2.6 | 4.0 |
| inference | 6.8 | 2.7 | 29.3 |

| test time (sec) | Enron | WIPO | Reuters |
|---|---|---|---|
| unconstrained | 0.054 | 0.070 | 0.60 |
| constrained | 0.054 | 0.070 | 0.60 |
| inference | 0.481 | 0.389 | 5.20 |

Table 5.2: Test set prediction error in percent (top); Test set prediction time in Seconds (bottom)

The results are given in Table 5.2, showing both the test set prediction error (using labelwise prediction error, i.e. Hamming loss) and the test prediction times. As expected, one can see benefits from incorporating known relationships between the labels when training a predictor. In each case, the addition of constraints leads to a significant improvement in test prediction error, versus training without any constraints or inference added. Training with inference (i.e., classical structured large margin training) still proves to be an effective training method overall, in one case improving the results over the constrained approach, providing an example where

the proposed constraints are sufficient but not necessary for imposing relations between labels. In two other case inference only falls slightly behind the constraint method. The key difference between the approach using constraints versus that using inference is in terms of the time it takes to produce predictions on test examples. Using inference to make test set predictions clearly takes significantly longer than applying labelwise predictions from either a constrained or unconstrained model, as shown in the right subtable of Table 5.2.

## 5.6 Conclusion

We have demonstrated a novel approach to structured multilabel prediction where inference is replaced with constraints on the score model. On multilabel text classification data, the proposed approach does appear to be able to achieve competitive generalization results, while reducing the time needed to make predictions at test time. In cases where logical relationships are known to hold between the labels, using either inference or imposing constraints on the score model appears to yield benefits over generic training approaches that ignore the prior knowledge.

# Chapter 6

# Conclusions

This thesis has studied the power of co-embedding in solving standard and structured association problems. The main advantage of this approach stems from its simplicity and intuitiveness, due primarily to its geometric basis. Even when other approaches are available, models with a geometric interpretation allow practitioners to exploit their intuition during the design process, even while maintaining a sound theoretical basis. The combination of intuition and theory can greatly simplify the design of a learning system and make it more understandable.

Co-embedding offers a novel perspective on classical problems such as classification. In the classical view of classification, input examples are embedded into a target (output) space, where each candidate class has a pre-embedded representative. The dimension of the embedding space is typically set to be equal to the number of candidate classes, and the association score is often computed with an alignment model. Predictions are then made by returning the label (or labels) whose embedding has an inner product with the example embedding that is highest (or exceeds some threshold). Alternatively, a co-embedding approach to classification considers embedding both the inputs and outputs into a common latent space, where the embeddings are learned from data. In other words, none of the input or output sides have pre-embedded representatives. Additionally, the computation of an association score is not necessarily limited to only using inner product; on the contrary, both an alignment model and a distance model are equally applicable. This more general perspective makes it easier to envision extensions to novel classification scenarios, such as zero shot learning (e.g. by learning an embedding function

that can map the representation of a new label into the latent space to coherently associate input examples to the new class).

## 6.1 Summary

Through these investigations, we faced a series of questions. Initially, we addressed the following question:

- Are there common general tractable solutions for problems such as *supervised link prediction in graphs* and *multilabel classification*?

We found that the answer to this question is affirmative. From a particular perspective, not only are these problems similar, they are similar to other problems, such as ranking, prediction, and query answering. We unified these under the title of association problems, and found that standard approaches are also very similar, in that they share a basic strategy of co-embedding. That is, to associate elements between sets, these approaches first joint embed items from the sets into a common low dimensional space, and then use geometry to associate them.

To incorporate geometry, an association score can be based on inner product or Euclidean distance. We established a connection between metric learning and distance based score models. We noted that common training formulations of co-embedding are non-convex, but showed how these can be reformulation to convex forms by relaxing the rank constraint with the trace norm. We then developed scalable training algorithms for co-embedding models, leading to tractable learning paradigm.

During the empirical evaluation of co-embedding models in different case studies, we made the surprising observation that the training objectives achieved by the non-convex and convex formulations are often identical. This observation lead to the next major question that we addressed in this dissertation:

- Is there *any theoretical basis* for observing *the equality of objective values in local and global* optimization to train co-embedding models?

We obtained a positive answer to this question as well. By adapting existing theoretical results to our setting, we established that, under certain conditions, training is guaranteed to reach the same objective, whether using the convex or non-convex formulations. This result led to the design of a scalable and correct computational strategy for training co-embedding models.

We finally addressed the following question:

- Suppose some *structure* is known to underlie a specific association problem. Is there a way to perform co-embedding while ensuring that the structure holds in the output, *without adding computational overhead to the prediction phase*?

The answer to this question was also found to be positive. For structured multilabel prediction, we demonstrated that a constrained form of co-embedding could be performed, where prior structure is pre-compiled via convex constraints in the training phase. The basic idea is to express the desired structure in a Venn diagram so that so that geometric objects (such as Euclidean balls) can be embedded in the latent space to express various constraints, such as implications and mutual exclusions. In this way, the burden of imposing structure is transferred entirely to the training phase. The advantage of this approach is that the training phase need only be performed once, while structure preserving prediction can be efficiently performed for each test example with no computational overhead added. In many real applications, this is a beneficial trade-off.

## 6.2 Limitations

One limitation of the convex co-embedding framework presented in this dissertation is that only linear maps are expressed in a convex form. It is not obvious that this restriction can be lifted while ensuring efficient global training. However, if one does not care about global optimality, nonlinear mappings can be immediately applied.

A potential limitation of the constrained optimization approach we developed for structured output co-embedding is that, so far, we have only considered a max-

imum margin structured output training objective. It is not immediately obvious whether this approach can be extended to conditional random fields, for example. We also only demonstrated the approach for implication and mutual exclusion constraints in multi-label prediction; extending the approach to more general structured output prediction problems remains future work. A second limiting assumption here concerns the large margin training, since it assumes the loss is decomposable. Extensions to cases where this assumption does not hold would be a valuable achievement if possible.

## 6.3 Research Directions

A number of follow-up directions are suggested by this research.

- Other linear compressions of tensor representations for three-set association problems need to be investigated, seeking alternatives from existing approaches that allow greater freedom to trade off space versus expressiveness.

- Other distance functions need to be investigated for distance based association models, particularly based on the $L_1$ and $L_\infty$ norms, to see if other properties, such as sparsity, might be beneficial.

- Alternative, tighter approximations of rank need to be investigated for when the target dimensionality is pre-specified.

- The proposed approach to structured association learning needs to be extended to more general structures, potentially by combining the method with search based prediction methods.

- A particularly achievable extension is the prediction when the output vector is known to be maximally sparse, in particular when it has cardinality one. The approach in Chapter 5 could be directly applied by embedding mutually exclusive classes.

- In the same line as the last suggestion, sparse representation learning and Cardinality Restricted Boltzmann Machines (Swersky et al., 2012) are other

interesting generalization to study.

- Strategies for coping with missing labels need to be investigated.
- Efficient data structures need to be exploited to reduce prediction time to sub-linear in the cardinality of the embedded sets.
- The rate of convergence for the proposed algorithm, ILA, needs to be analyzed in detail, for given local optimization algorithms.
- Unsupervised models of co-embedding need to be developed, which would support novel forms of co-clustering.
- In the applied terms, the application of the unstructured and structured co-embedding methods proposed in this document, to huge datasets such as for learning knowledge graph relations, image captioning, large scale ranking, and question answering would be useful.
- Empirical study of the semantic meaning of translation vectors in the co-embedding space would be another interesting direction to pursue.
- The possibility of adapting co-embedding to regression problems needs to be investigated.
- The applicability of co-embedding to different learning paradigms, such as reinforcement learning, are yet to be considered. Interestingly, embedding is already proved highly successful in reinforcement learning, but not scalable enough (Bowling et al., 2005). The proposed advances in scalable co-embedding could potentially be useful here.

Finding answers to some of these questions can hopefully open new directions of research.

# Bibliography


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)*.

Al-Halah, Z., Tapaswi, M., and Stiefelhagen, R. (2016). Recovering the missing link: Predicting class-attribute associations for unsupervised zero-shot learning. 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.

Ba, J., Swersky, K., Fidler, S., and Salakhutdinov, R. (2015). Predicting deep zero-shot convolutional neural networks using textual descriptions. In *IEEE International Conference on Computer Vision (ICCV)*.

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

Bakir, G., Hofmann, T., Schölkopf, B., Smola, A., Taskar, B., and Vishwanathan, S. (2007). *Predicting Structured Data*. MIT Press.

Belanger, D. and McCallum, A. (2016). Structured prediction energy networks. In *International Conference on Machine Learning (ICML)*.

Bengio, S. and Weston, J. (2011). Joint embedding for item association. CA Patent App. CA 2,786,727.

Bengio, S., Weston, J., and Grangier, D. (2010). Label embedding trees for large multi-class tasks. In *Neural Information Processing Systems (NIPS)*.

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

Bi, W. and Kwok, J. (2011). Multi-label classification on tree- and DAG-structured hierarchies. In *International Conference on Machine Learning (ICML)*.

Bi, W. and Kwok, J. (2012). Mandatory leaf node prediction in hierarchical multilabel classification. In *Neural Information Processing Systems (NIPS)*.

Bi, W. and Kwok, J. (2013). Efficient multi-label classification with many labels. In *International Conference on Machine Learning (ICML)*.

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)*.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In *ACM SIGMOD international conference on Management of data*. ACM.

Bordes, A., Glorot, X., Weston, J., and Bengio, Y. (2012). Joint learning of words and meaning representations for open-text semantic parsing. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

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)*.

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

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

Bowling, M., Ghodsi, A., and Wilkinson, D. (2005). Action respecting embedding. In *International Conference on Machine Learning (ICML)*.

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

Cai, J., Candes, E., and Shen, Z. (2008). A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20:1956–1982.

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

Chen, M., Zheng, A., and Weinberger, K. (2013a). Fast image tagging. In *International Conference on Machine Learning (ICML)*.

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

Chen, Y. and Lin, H. (2012). Feature-aware label space dimension reduction for multi-label classification. In *Neural Information Processing Systems (NIPS)*.

Cheng, L. (2013). Riemannian similarity learning. In *Internatal Conference on Machine Learning (ICML)*.

Chopra, S., Hadsell, R., and LeCun, Y. (2005). Learning a similarity metric discriminatively, with application to face verification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Cissé, M., Usunier, N., Artieres, T., and Gallinari, P. (2013). Robust bloom filters for large multilabel classification tasks. In *Neural Information Processing Systems (NIPS)*.

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

Cox, T. F. and Cox, M. A. (2000). *Multidimensional scaling*. CRC press.

Crammer, K. and Singer, Y. (2001). On the algorithmic interpretation of multiclass kernel-based vector machines. *Journal of Machine Learning Research (JMLR)*, 2.

Crammer, K. and Singer, Y. (2003). A family of additive online algorithms for category ranking. *Journal of Machine Learning Research (JMLR)*, 3:1025–1058.

Daume, H. and Langford, J. (2009). Search-based structured prediction. *Machine Learning*, 75:297–325.

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

Dembczynski, K., Cheng, W., and Hüllermeier, E. (2010). Bayes optimal multilabel classification via probabilistic classifier chains. In *International Conference on Machine Learning (ICML)*.

Dembczyński, K., Jachnik, A., Kotłlowski, W., Waegeman, W., and Hüllermeier, E. (2013a). Optimizing the F-measure in multi-label classification: plug-in rule approach versus structured loss minimization. In *International Conference on Machine Learning (ICML)*.

Dembczyński, K., Kotłlowski, W., and Hüllermeier, E. (2013b). Optimizing the F-measure in multi-label classification: plug-in rule approach versus structured loss minimization. In *International Conference on Machine Learning (ICML)*.

Dembczyński, K., Waegeman, W., Cheng, W., and Hüllermeier, E. (2012). On label dependence and loss minimization in multi-label classification. *Machine Learning*, 88(1):5–45.

Deng, J., Berg, A., Li, K., and Li, F. (2010). What does classifying more than 10,000 image categories tell us? In *European Conference on Computer Vision (ECCV)*.

Deng, J., Ding, N., Jia, Y., Frome, A., Murphy, K., Bengio, S., Li, Y., Neven, H., and Adam, H. (2014). Large-scale object classification using label relation graphs. In *Proceedings ECCV*.

Deng, J., Satheesh, S., Berg, A., and Fei-Fei, L. (2011). Fast and balanced: Efficient label tree learning for large scale object recognition. In *Neural Information Processing Systems (NIPS)*.

Dinuzzo, F. and Fukumizu, K. (2011). Learning low-rank output kernels. *Journal of Machine Learning Research*, 20:181–196.

Doppa, J., Fern, A., and Tadepalli, P. (2012). Output space search for structured prediction. In *International Conference on Machine Learning (ICML)*.

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

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

Elisseeff, A. and Weston, J. (2001). A kernel method for multi-labelled classification. In *Neural Information Processing Systems (NIPS)*.

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

Frome, A., Corrado, G., Shlens, J., Bengio, S., Dean, J., Ranzato, M., and Mikolov, T. (2013). DeViSE: A deep visial-semantic embedding model. In *Neural Information Processing Systems (NIPS)*.

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

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

Gao, T. and Koller, D. (2011). Discriminative learning of relaxed hierarchy for large-scale visual recognition. In *International Conference on Computer Vision (ICCV)*.

Gao, W. and Zhou, Z. (2013). On the consistency of multi-label learning. *Artificial Intelligence*, 199-200:22–44.

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

Gentile, C. and Orabona, F. (2014). On multilabel classification and ranking with bandit feedback. *Journal of Machine Learning Research*, 15:2451–2487.

Giguère, S., Laviolette, F., Marchand, M., and Sylla, K. (2013). Risk bounds and learning algorithms for the regression approach to structured output prediction. In *International Conference on Machine Learning (ICML)*.

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

Globerson, A. and Roweis, S. T. (2005). Metric learning by collapsing classes. In *Neural Information Processing Systems (NIPS)*.

Gong, N. Z., Talwalkar, A., Mackey, L., Huang, L., Shin, E. C. R., Stefanov, E., Shi, E. R., and Song, D. (2014). Joint link prediction and attribute inference using a social-attribute network. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(2):27.

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

Grover, A. and Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*.

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

Guo, Y. and Schuurmans, D. (2013). Multi-label classification with output kernels. In *European Conference on Machine Learning (ECML)*.

Haeffele, B., Vidal, R., and Young, E. (2014). Structured low-rank matrix factorization: Optimality, algorithm, and applications to image processing. In *International Conference on Machine Learning (ICML)*.

Hariharan, B., Vishwanathan, S., and Varma, M. (2012). Efficient max-margin multi-label classification with applications to zero-shot learning. *Machine Learning*, 88:127–155.

Hastie, T., Tibshirani, R., and Friedman, J. (2009a). *The Elements of Statistical Learning*. Springer, 2nd edition.

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

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

Hoff, P. D. (2005). Bilinear mixed-effects models for dyadic data. *Journal of the american Statistical association*, 100(469):286–295.

Hofmann, T., Puzicha, J., and Jordan, M. I. (1998). Learning from dyadic data. In *Neural Information Processing Systems (NIPS)*.

Hsu, D., Kakade, S., Langford, J., and Zhang, T. (2009). Multi-label prediction via compressed sensing. In *Neural Information Processing Systems (NIPS)*.

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

Hüllermeier, E., Fürnkranz, J., Cheng, W., and Brinker, K. (2008). Label ranking by learning pairwise preferences. *Artif. Intell.*, 172(16-17):1897–1916.

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

Jancsary, J., Nowozin, S., and Rother, C. (2013). Learning convex QP relaxations for structured prediction. In *International Conference on Machine Learning (ICML)*.

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 *Neural Information Processing Systems (NIPS)*.

Ji, S., Tang, L., Yu, S., and Ye, J. (2010). A shared-subspace learning framework for multi-label classification. *ACM Transactions on Knowledge Discovery from Data*, 4(2).

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

Jin, R. and Ghahramani, Z. (2002). Learning with multiple labels. In *Neural Information Processing Systems (NIPS)*.

Joachims, T. (1999). Transductive inference for text classification using support vector machines. In *International Conference on Machine Learning (ICML)*.

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

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

Joly, S. and Le Calvé, G. (1995). Three-way distances. *Journal of Classification*, 12(2):191–205.

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

Kadri, H., Ghavamzadeh, M., and Preux, P. (2013). A generalized kernel approach to structured output learning. In *International Conference on Machine Learning (ICML)*.

Kae, A., Sohn, K., Lee, H., and Learned-Miller, E. (2013). Augmenting CRFs with Boltzmann machine shape priors for image labeling. In *Proceedings CVPR*.

Kapoor, A., Jain, P., and Vishwanathan, R. (2012). Multilabel classification using Bayesian compressed sensing. In *Neural Information Processing Systems (NIPS)*.

Kiros, R., Salakhutdinov, R., and Zemel, R. (2014). Multimodal neural language models. In *International Conference on Machine Learning (ICML)*.

Klimt, B. and Yang, Y. (2004). The enron corpus: A new dataset for email classification research. In *European Conference on Machine Learning (ECML)*.

Kohli, P. and Torr, P. H. (2007). Dynamic graph cuts for efficient inference in markov random fields. *IEEE transactions on pattern analysis and machine intelligence (PAMI)*, 29(12):2079–2088.

Kolmogorov, V. and Zabih, R. (2002). What energy functions can be minimized via graph cuts? In *European Conference on Computer Vision (ECCV)*, ECCV '02.

Krizhevsky, A., Sutskever, I., and Hinton, G. (2012). Imagenet classification with deep convolutional neural networks. In *Neural Information Processing Systems (NIPS)*.

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

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

Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning (ICML)*.

Lampert, C. (2011). Maximum margin multi-label structured prediction. In *Neural Information Processing Systems (NIPS)*.

Larochelle, H., Erhan, D., and Bengio, Y. (2008a). Zero-data learning of new tasks. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 646–651.

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

Lewis, D., Yang, Y., Rose, T., and Li, F. (2004). RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397.

Li, F., Fergus, R., and Perona, P. (2003). A bayesian approach to unsupervised one-shot learning of object categories. In *IEEE International Conference on Computer Vision (ICCV)*.

Li, M. and Huber, D. (2017). Guaranteed parameter estimation for discrete energy minimization. *arXiv preprint arXiv:1701.03151*.

Li, Q., Wang, J., Wipf, D., and Tu, Z. (2013a). Fixed-point model for structured prediction. In *International Conference on Machine Learning (ICML)*.

Li, Y., Tarlow, D., and Zemel, R. (2013b). Exploring compositional high order pattern potentials for structured output learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Lin, Z., Ding, G., Hu, M., and Wang, J. (2014). Multi-label classification via feature-aware implicit label space encoding. In *International Conference on Machine Learning (ICML)*.

London, B., Huang, B., Taskar, B., and Getoor, L. (2013). Collective stability in structured prediction: Generalization on one example. In *International Conference on Machine Learning (ICML)*.

Mäkelä, M. (2003). Multiobjective proximal bundle method for nonconvex nonsmooth optimization: Fortran subroutine MPBNGC 2.0. Technical report, U. of Jyväskykä.

Marchand, M., Su, H., Morvant, E., Rousu, J., and Shawe-Taylor, J. (2014). Multilabel structured output learning with random spanning trees of max-margin Markov networks. In *Neural Information Processing Systems (NIPS)*.

Meeds, E., Ghahramani, Z., Neal, R. M., and Roweis, S. T. (2006). Modeling dyadic data with binary latent factors. In *Neural Information Processing Systems (NIPS)*.

Mencía, E. and Fürnkranz, J. (2008). Efficient pairwise multi-label classification for large-scale problems in the legal domain. In *European Conference on Machine Learning (ECML)*.

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

Menon, A. K. and Elkan, C. (2010a). A log-linear model with latent features for dyadic prediction. In *2010, IEEE International Conference on Data Mining (ICDM)*.

Menon, A. K. and Elkan, C. (2010b). Predicting labels for dyadic data. *Data Min. Knowl. Discov.*, 21(2):327–343.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Neural Information Processing Systems (NIPS)*.

Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Mirzazadeh, F., Guo, Y., and Schuurmans, D. (2014). Convex co-embedding. In *AAAI Conference on Artificial Intelligence*.

Mirzazadeh, F., Ravanbakhsh, S., Ding, N., and Schuurmans, D. (2015a). Embedding inference for structured multilabel prediction. In *Neural Information Processing Systems (NIPS)*.

Mirzazadeh, F., White, M., Gyorgy, A., and Schuurmans, D. (2015b). Scalable metric learning for co-embedding. In *European Conference on Machine Learning (ECML)*.

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

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

Nickel, M., Murphy, K., Tresp, V., and Gabrilovich, E. (2016). A review of relational machine learning for knowledge graphs. *IEEE*, 104(1):11–33.

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

Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization*. Springer, New York, NY, USA, 2nd edition.

Norouzi, M., Mikolov, T., Bengio, S., Singer, Y., Shlens, J., Frome, A., Corrado, G., and Dean, J. (2013). Zero-shot learning by convex combination of semantic embeddings. *CoRR*, abs/1312.5650.

Palatucci, M., Pomerleau, D., Hinton, G., and Mitchell, T. (2009). Zero-shot learning with semantic output codes. In *Neural Information Processing Systems (NIPS)*.

Pazzani, M. J. and Billsus, D. (2007). The adaptive web. chapter Content-based Recommendation Systems, pages 325–341. Springer-Verlag, Berlin, Heidelberg.

Petterson, J. and Caetano, T. (2011). Submodular multi-label learning. In *Neural Information Processing Systems (NIPS)*.

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

Punyakanok, V., Roth, D., tau Yih, W., and Zimak, D. (2005). Learning and inference over constrained output. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

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

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

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*, pages 81–90.

Rousu, J., Saunders, C., Szedmak, S., and Shawe-Taylor, J. (2006). Kernel-based learning of hierarchical multilabel classification models. *Journal of Machine Learning Research*, 7:1601–1626.

Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

Schein, A. I., Popescul, A., Ungar, L. H., and Pennock, D. M. (2002). Methods and metrics for cold-start recommendations. In *International ACM SIGIR conference on Research and development in information retrieval*. ACM.

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

Shaw, B. and Jebara, T. (2009). Structure preserving embedding. In *International Conference on Machine Learning (ICML)*, page 118.

Sidiropoulos, A. (2008). *Computational Metric Embeddings*. Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science.

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

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

Socher, R., Ganjoo, M., Manning, C., and Ng, A. (2013c). Zero-shot learning through cross-modal transfer. In *Neural Information Processing Systems (NIPS)*.

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

Srikumar, V. and Manning, C. (2014). Learning distributed representations for structured output prediction. In *Neural Information Processing Systems (NIPS)*.

Srivastava, N. and Salakhutdinov, R. (2012). Multimodal learning with deep Boltzmann machines. In *Neural Information Processing Systems (NIPS)*.

Sun, X. (2014). Structure regularization for structured prediction. In *Neural Information Processing Systems (NIPS)*.

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

Swersky, K., Sutskever, I., Tarlow, D., Zemel, R. S., Salakhutdinov, R. R., and Adams, R. P. (2012). Cardinality restricted boltzmann machines. In *Neural Information Processing Systems (NIPS)*.

Tarlow, D., Givoni, I., Zemel, R., and Frey, B. (2011). Graph cuts is a max-product algorithm. In *Uncertainty in Artificial Intelligence (UAI)*.

Taskar, B. (2004). *Learning Structured Prediction Models: A Large Margin Approach*. PhD thesis, Stanford.

Taskar, B., Guestrin, C., and Koller, D. (2003). Max-margin Markov networks. 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 (JMLR)*, 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*.

Ueda, N. and Saito, K. (2002). Parametric mixture models for multi-labeled text. In *Neural Information Processing Systems (NIPS)*.

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

Vendrov, I., Kiros, R., Fidler, S., and Urtasun, R. (2016). Order-embeddings of images and language. In *International Conference on Learning Representations (ICLR)*.

Vert, J. and Yamanishi, Y. (2004). Supervised graph inference. In *Neural Information Processing Systems (NIPS)*.

Vinyals, O., Blundell, C., Lillicrap, T. P., Kavukcuoglu, K., and Wierstra, D. (2016). Matching networks for one shot learning. In *Neural Information Processing Systems (NIPS)*.

Wang, H., Nie, F., and Huang, H. (2014). Robust distance metric learning via simultaneous l1-norm minimization and maximization. In *International Conference on Machine Learning (ICML)*.

Weinberger, K. and Chapelle, O. (2008). Large margin taxonomy embedding for document categorization. In *Neural Information Processing Systems (NIPS)*.

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

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

Weiss, D. and Taskar, B. (2013). Learning adaptive value of information for structured prediction. In *Neural Information Processing Systems (NIPS)*.

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 *International Joint Conference on Artificial Intelligence (IJCAI)*.

Weston, J., Wang, C., Weiss, R. J., and Berenzweig, A. (2012). Latent collaborative retrieval. In *International Conference on Machine Learning (ICML)*.

Wu, L., Jin, R., and Jain, A. (2011). Tag completion for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(3):716–727.

Xian, Y., Akata, Z., Sharma, G., Nguyen, Q., Hein, M., and Schiele, B. (2016). Latent embeddings for zero-shot classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

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

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

Xu, M., Jin, R., and Zhou, Z.-H. (2013). Speedup matrix completion with side information: Application to multi-label learning. In *Neural Information Processing Systems (NIPS)*.

Yamanishi, Y. (2008). Supervised bipartite graph inference. In *Neural Information Processing Systems (NIPS)*, pages 1841–1848.

Yu, H.-F., Jain, P., Kar, P., and Dhillon, I. (2014). Large-scale multi-label learning with missing labels. In *International Conference on Machine Learning (ICML)*.

Yu, Y. (2013). *Fast Gradient Methods for Structured Sparsity*. PhD thesis, University of Alberta.

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

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

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

# Appendix A

# Proofs for Chapter 3

## A.1 Proof of Proposition 1

We are considering the case when $\boldsymbol{\alpha} = [1, 0, ..., 0]^\top$, which means the loss is determined only by the first entry in each sorted list. For a particular $x$, let $y_1$ be the $y \in Y(x)$ that has the lowest score $s(x, y)$, which will be the first y in the list $\sigma(x, y)$. Also let $\bar{y}_1$ be the $\bar{y} \in \bar{Y}(x)$ that has the highest score $s(x, \bar{y})$, which will be the first $\bar{y}$ in the list $\pi(x, \bar{y})$. The loss then becomes

$$1\Big(s(x, y_1) \leq t(x)\Big) + 1\Big(t(x) > s(x, \bar{y}_1)\Big).$$

Observe that this loss would be zero only if $s(x, y_1) > t(x)$ and $s(x, \bar{y}_1) < t(x)$. However, if that were true, then by construction we would have:

$$s(x, y) > t(x), \qquad \forall y \in Y(x) \tag{A.1}$$

$$s(x, \bar{y}) \leq t(x), \qquad \forall \bar{y} \in \bar{Y}(x) \tag{A.2}$$

that is, the loss would only be zero if there was an exact match.

In other words, for a particular example, the first term indicates the presence of a false negative prediction and the second term indicates the presence of false positive prediction. If the right hand side of (3.9) is zero, then neither any false positive nor any false negative exists , i.e. exact match error must be zero. ■

# Appendix B

# Proofs for Chapter 4

## B.1 Proof of Proposition 1

**Part 1:** First, form the Lagrangian of (4.17), given by $L(C) + \beta \operatorname{tr}(C) - \operatorname{tr}(SC)$ with $S \succeq 0$, and consider the necessary KKT conditions:

$$S = \nabla L(C) + \beta I, \quad S \succeq 0, \quad C \succeq 0, \quad SC = 0. \tag{B.1}$$

The problem is strictly feasible, since $C = I$ is a strictly feasible point; therefore, Slater' s condition holds and (B.1) is also 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 $0 = \nabla L(QQ^\top)Q + \beta Q = SQ$. 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 $-\lambda_i \mathbf{u}_i^\top Q = (\mathbf{u}_i^\top S^\top)Q = \mathbf{u}_i^\top (S^\top Q) = \mathbf{u}_i^\top \mathbf{0} = \mathbf{0}$ since $\mathbf{u}_i$ is an eigenvector of $S$ and $S$ is symmetric.

**Part 3:** To optimize the inner product (4.19), introduce Lagrange multipliers $\xi_i > 0$ for the norm constraints. Since $-S$ is symmetric, we can re-express the inner objective as

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

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 $\lambda_i = 2\xi_i > 0$. ∎

## B.2 Proof of Corollary 1

First assume condition (i) holds and argue by contradiction. Assume $QQ^\top$ is not a global optimum of (4.17), and let $\mathbf{u}_1 \in \mathbb{R}^p$ be as defined in Proposition 1. Then, $f(QQ^\top + \beta \mathbf{u}_1 \mathbf{u}_1^\top) < f(QQ^\top)$ for a sufficiently small $\beta > 0$. Furthermore, since $\mathrm{rank}(Q) < d$, 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 $\lim_{\alpha \to 0} \widehat{Q}_\alpha V^\top = QVV^\top = Q$. Moreover, since $\mathbf{u}_1$ is orthogonal to the columns of $Q$, it is also orthogonal to the columns of $QV$, so $\widehat{Q}_\alpha V (\widehat{Q}_\alpha V)^\top = QV(QV)^\top + \alpha^2 \mathbf{u}_1 \mathbf{u}_1^\top = QQ^\top + \beta \mathbf{u}_1 \mathbf{u}_1^\top$. Therefore, $f(\widehat{Q}_\alpha \widehat{Q}_\alpha^\top) = f(QQ^\top + \beta \mathbf{u}_1 \mathbf{u}_1^\top) < f(QQ^\top)$ 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 $f(QQ^\top)$, $\nabla f(QQ^\top)Q = 0$. Since $Q$ has rank $p$, the null-space of $\nabla f(QQ^\top)$ is of dimension $p$, yielding that $\nabla f(QQ^\top) = 0$. Since $QQ^\top \succeq 0$ and $f$ is convex, $C = QQ^\top$ is an optimum of (4.17). ■

## B.3 Proof of Proposition 2

Let $Q_m$ and $U_m$ denote the matrix $Q$ and $U$ in ILA when line 10 is executed the $m$th 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 $C_m = Q_m Q_m^\top$ and $C_{init,m} = Q_{init,m} Q_{init,m}^\top = a_m C_m + b_m U_m U_m^\top$.

If $C_m$ is not a global optimum of (4.17), 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 $\|Q_m\|_F^2 = \mathrm{tr}(C_m) \leq f(C_0)$, 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 (4.17) by verifying the KKT conditions (B.1) 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 $\nabla f(C_m) C_m = \nabla f(QQ^\top) QQ^\top = 0$. 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:

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

for any $\hat{b} \geq 0$, where the last equality holds since $U_m U_m^\top$ 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

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

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

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

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. ■

## B.4 An Auxiliary Lemma

This lemma is used in Appendix B.3.

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

$$f(C + S) \leq f(C) + \operatorname{tr}(S^\top \nabla f(C)) + \frac{\nu}{2}\rho(S)^2\,. \tag{B.3}$$

*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 $h^\top(\eta) = \mathrm{tr}(S^\top \nabla f(C + \eta S))$ for any $\eta \in (0, 1)$. Then

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

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

■

# Appendix C

# Proofs for Chapter 5

## C.1 Proof of Lemma 1

First expand (5.8), obtaining

$$-s_k(\mathbf{x}) = \phi(x)^\top P\phi(x) + 2\phi(x)^\top A\psi_k + 2\mathbf{b}^\top\phi(x) + \psi_k^\top Q\psi_k + 2\mathbf{c}^\top\psi_k + r.$$

Since $\theta \succeq 0$ there must exist $U$, $V$ and $\mathbf{u}$ such that

$$\theta = [U^\top, -V^\top, \mathbf{u}]^\top[U^\top, -V^\top, \mathbf{u}]$$

, where $U^\top U = P$, $U^\top V = -A$, $U^\top\mathbf{u} = \mathbf{b}$, $V^\top V = Q$, $V^\top\mathbf{u} = -\mathbf{c}$, and $\mathbf{u}^\top\mathbf{u} = r$.
A simple substitution and rearrangement shows the claim. ■

## C.2 Proof of Lemma 2

Similar to Lemma 1, since $\theta_{PAQ} \succeq 0$, there exist $U$ and $V$ such that

$$\theta_{PAQ} = [U^\top, -V^\top]^\top[U^\top, -V^\top]$$

where

$$U^\top U = P, \quad V^\top V = Q, \quad U^\top V = -A.$$

Expanding (5.9) and substituting gives (5.12).
For (5.13) note

$$\psi_k^\top Q\psi_k - \psi_k^\top Q\psi_l - \psi_l^\top Q\psi_k + \psi_l^\top Q\psi_l = (\psi_k - \psi_l)^\top Q(\psi_k - \psi_l).$$

Expanding $Q$ gives

$$\begin{aligned}(\psi_k - \psi_l)^\top Q(\psi_k - \psi_l) &= (\psi_k - \psi_l)^\top V^\top V(\psi_k - \psi_l) \\ &= \|V\psi_k - V\psi_l\|^2. \blacksquare\end{aligned}$$

## C.3 Proof of Theorem 1

First, since $\theta_{PAQ} \succeq 0$ we have the relationship (5.13), which implies that there must exist vectors $\nu_1 = V\psi_1$ and $\nu_2 = V\psi_2$ such that $\psi_1^\top Q\psi_1 - \psi_1^\top Q\psi_2 - \psi_2^\top Q\psi_1 + \psi_2^\top Q\psi_2 = \|\nu_1 - \nu_2\|^2$. Therefore, the constraints (5.14) and (5.15) can be equivalently re-expressed as

$$\gamma_1 + \delta + (1+\alpha)\|\nu_1 - \nu_2\|^2 \;\leq\; \gamma_2 - \delta \tag{C.1}$$

$$\left(\tfrac{\alpha}{2}\right)^2 \|\nu_1 - \nu_2\|^2 \;\geq\; \gamma_1 + \delta \tag{C.2}$$

with respect to these vectors. Next let $\mu(\mathbf{x}) := U\phi(x)$ (which exists by (5.12)) and observe that

$$\begin{aligned}\|\mu(\mathbf{x}) - \nu_2\|^2 &= \|\mu(\mathbf{x}) - \nu_1 + \nu_1 - \nu_2\|^2 \\ &= \|\mu(\mathbf{x}) - \nu_1\|^2 + \|\nu_1 - \nu_2\|^2 + 2\langle\mu(\mathbf{x}) - \nu_1, \nu_1 - \nu_2\rangle \qquad \text{(C.3)}\end{aligned}$$

Consider two cases.

*Case 1:* $2\langle\mu(\mathbf{x}) - \nu_1, \nu_1 - \nu_2\rangle > \alpha\|\nu_1 - \nu_2\|^2$. In this case, by the Cauchy Schwarz inequality we have

$$2\|\mu(\mathbf{x}) - \nu_1\|\|\nu_1 - \nu_2\| \geq 2\langle\mu(\mathbf{x}) - \nu_1, \nu_1 - \nu_2\rangle > \alpha\|\nu_1 - \nu_2\|^2,$$

which implies $\|\mu(\mathbf{x}) - \nu_1\| > \frac{\alpha}{2}\|\nu_1 - \nu_2\|$, hence

$$\|\mu(\mathbf{x}) - \nu_1\|^2 > \left(\frac{\alpha}{2}\right)^2 \|\nu_1 - \nu_2\|^2 \geq \gamma_1 + \delta$$

by constraint (C.2). But this implies that $s_1(\mathbf{x}) < -\delta$ therefore it does not matter what value $s_2(\mathbf{x})$ has.

*Case 2:* $2\langle\mu(\mathbf{x}) - \nu_1, \nu_1 - \nu_2\rangle \leq \alpha\|\nu_1 - \nu_2\|^2$. In this case, assume that $s_1(\mathbf{x}) \geq -\delta$, i.e. $\|\mu(\mathbf{x}) - \nu_1\|^2 \leq \gamma_1 + \delta$, otherwise it does not matter what value $s_2(\mathbf{x})$ has. Then from (C.3) it follows that

$$\|\mu(\mathbf{x}) - \nu_2\|^2 \leq \|\mu(\mathbf{x}) - \nu_1\|^2 + (1+\alpha)\|\nu_1 - \nu_2\|^2 \leq \gamma_1 + \delta + (1+\alpha)\|\nu_1 - \nu_2\|^2 \leq \gamma_2 - \delta$$

by constraint (C.1). But this implies that $s_2(\mathbf{x}) \geq \delta$, hence the implication is enforced. ■

## C.4 Proof of Theorem 2

As before, since $\theta_{PAQ} \succeq 0$ we have the relationship (5.13), which implies that there must exist vectors $\nu_1 = V\psi_1$ and $\nu_2 = V\psi_2$ such that

$$\psi_1^\top Q\psi_1 - \psi_1^\top Q\psi_2 - \psi_2^\top Q\psi_1 + \psi_2^\top Q\psi_2 = \|\nu_1 - \nu_2\|^2.$$

Observe that the constraint (5.16) can then be equivalently expressed as

$$\tfrac{1}{2}\|\nu_1 - \nu_2\|^2 \quad > \quad \gamma_1 + \gamma_2 + 2\delta, \tag{C.4}$$

and observe that

$$\begin{aligned} \|\nu_1 - \nu_2\|^2 &= \|\nu_1 - \mu(\mathbf{x}) + \mu(\mathbf{x}) - \nu_2\|^2 \\ &= \|\nu_1 - \mu(\mathbf{x})\|^2 + \|\mu(\mathbf{x}) - \nu_2\|^2 + 2\langle \nu_1 - \mu(\mathbf{x}), \mu(\mathbf{x}) - \nu_2\rangle. \end{aligned} \tag{C.5}$$

using $\mu(\mathbf{x}) := U\phi(x)$ as before (which exists by (5.12)).

Therefore

$$\|\mu(\mathbf{x}) - \nu_1\|^2 + \|\mu(\mathbf{x}) - \nu_2\|^2 = \|\nu_1 - \nu_2\|^2 - 2\langle \nu_1 - \mu(\mathbf{x}), \mu(\mathbf{x}) - \nu_2\rangle \tag{C.6}$$

$$= \|(\nu_1 - \mu(\mathbf{x})) + (\mu(\mathbf{x}) - \nu_2)\|^2 - 2\langle \nu_1 - \mu(\mathbf{x}), \mu(\mathbf{x}) - \nu_2\rangle \tag{C.7}$$

$$\geq \tfrac{1}{2}\|(\nu_1 - \mu(\mathbf{x})) + (\mu(\mathbf{x}) - \nu_2)\|^2 \tag{C.8}$$

$$= \tfrac{1}{2}\|\nu_1 - \nu_2\|^2. \tag{C.9}$$

(To prove the inequality (C.8) observe that, since $0 \leq \frac{1}{2}\|a - b\|^2$, we must have $\langle a, b\rangle \leq \frac{1}{2}\|a\|^2 + \frac{1}{2}\|b\|^2$, hence $2\langle a, b\rangle \leq \frac{1}{2}\|a\|^2 + \frac{1}{2}\|b\|^2 + \langle a, b\rangle = \frac{1}{2}\|a+b\|^2$, which establishes $-2\langle a, b\rangle \geq -\frac{1}{2}\|a + b\|^2$. The inequality (C.8) then follows simply by setting $a = \nu_1 - \mu(\mathbf{x})$ and $b = \mu(\mathbf{x}) - \nu_2$.)

Now combining (C.9) with the constraint (C.4) implies that $\|\mu(\mathbf{x}) - \nu_1\|^2 + \|\mu(\mathbf{x}) - \nu_2\|^2 \geq \frac{1}{2}\|\nu_1 - \nu_2\|^2 > \gamma_1 + \gamma_2 + 2\delta$, therefore one of $\|\mu(\mathbf{x}) - \nu_1\|^2 > \gamma_1 + \delta$ or $\|\mu(\mathbf{x}) - \nu_2\|^2 > \gamma_2 + \delta$ must hold, hence at least one of $s_1(\mathbf{x}) < -\delta$ or $s_2(\mathbf{x}) < -\delta$ must hold. Therefore, the mutual exclusion is enforced. ■

## C.5 Proof of Proposition 1

First observe that

$$\max_{\mathbf{y}\in\mathcal{Y}} \sum_{k=1}^{l} y_k s_k(\mathbf{x}) \;\leq\; \max_{\mathbf{y}} \sum_{k=1}^{l} y_k s_k(\mathbf{x}) \;=\; \sum_{k=1}^{l} \max_{y_k} y_k s_k(\mathbf{x}) \tag{C.10}$$

so making local classifications for each label gives an upper bound. However, if the score function satisfies the constraints, then the concatenation of the local label decisions $\mathbf{y} = (y_1, ..., y_l)$ must be jointly feasible; that is, $\mathbf{y} \in \mathcal{Y}$. In particular, for the implication $y_1 \Rightarrow y_2$ the score constraint (5.5) ensures that if $s_1(\mathbf{x}) > 0 \geq -\delta$ (implying $1 = \arg\max_{y_1} y_1 s_1(\mathbf{x})$) then it must follow that $s_2(\mathbf{x}) \geq \delta$, hence $s_2(\mathbf{x}) > 0$ (implying $1 = \arg\max_{y_2} y_2 s_2(\mathbf{x})$). Similarly, for the mutual exclusion $\neg y_1 \vee \neg y_2$ the score constraint (5.6) ensures $\min(s_1(\mathbf{x}), s_2(\mathbf{x})) < -\delta \leq 0$, hence if $s_1(\mathbf{x}) > 0 \geq -\delta$ (implying $1 = \arg\max_{y_1} y_1 s_1(\mathbf{x})$) then it must follow that $s_2(\mathbf{x}) < -\delta \leq 0$ (implying $0 = \arg\max_{y_2} y_2 s_2(\mathbf{x})$), and vice versa. Therefore, since the maximizer $\mathbf{y}$ of (C.10) is feasible, we actually have that the leftmost term in (C.10) is equal to the rightmost. ∎

## C.6 Proof of Proposition 2

For a given $\mathbf{x}$ and $\mathbf{t} \in \mathcal{Y}$, let $f_k(y) = \delta(y, t_k) + (y - t_k)s_k(\mathbf{x})$, hence

$$y_k = \arg \max_{y\in\{0,1\}} f_k(y).$$

It is easy to show that

$$1 \in \arg \max_{y\in\{0,1\}} f_k(y) \iff s_k(\mathbf{x}) \geq t_k\delta - (1 - t_k)\delta, \tag{C.11}$$

which can be verified by checking the two cases, $t_k = 0$ and $t_k = 1$. When $t_k = 0$ we have $f_k(0) = 0$ and $f_k(1) = \delta + s(\mathbf{x})$, therefore $1 = y_k \in \arg\max_{y\in\{0,1\}} f_k(y)$ iff $\delta + s(\mathbf{x}) \geq 0$. Similarly, when $t_k = 1$ we have $f_k(0) = \delta - s(\mathbf{x})$ and $f_k(1) = 0$, therefore $1 = y_k \in \arg\max_{y\in\{0,1\}} f_k(y)$ iff $\delta - s(\mathbf{x}) \leq 0$.
Combining these two conditions yields (C.11).

Next, we verify that if the score constraints hold, then the logical constraints over $\mathbf{y}$ are automatically satisfied even by locally assigning $y_k$, which implies the optimal joint assignment is feasible, i.e. $\mathbf{y} \in \mathcal{Y}$, establishing the claim.

**Implication** In particular, for the implication $y_1 \Rightarrow y_2$, it is assumed that $t_1 \Rightarrow t_2$ in the target labeling and also that score constraints hold, ensuring $s_1(\mathbf{x}) \geq -\delta \Rightarrow s_2(\mathbf{x}) \geq \delta$.

Consider the cases over possible assignments to $t_1$ and $t_2$:

If $t_1 = 0$ and $t_2 = 0$ then $y_1 = 1 \Rightarrow f_1(1) \geq f_1(0) \Rightarrow \delta + s_1(\mathbf{x}) \geq 0 \Rightarrow s_1(\mathbf{x}) \geq -\delta \Rightarrow s_2(\mathbf{x}) \geq \delta$ (by assumption) $\Rightarrow s_2(\mathbf{x}) \geq -\delta \Rightarrow \delta + s_2(\mathbf{x}) \geq 0 \Rightarrow f_2(1) \geq f_2(0) \Rightarrow y_2 = 1$.

If $t_1 = 0$ and $t_2 = 1$ then $y_1 = 1 \Rightarrow f_1(1) \geq f_1(0) \Rightarrow \delta + s_1(\mathbf{x}) \geq 0 \Rightarrow s_1(\mathbf{x}) \geq -\delta \Rightarrow s_2(\mathbf{x}) \geq \delta$ (by assumption) $\Rightarrow 0 \geq \delta - s_2(\mathbf{x}) \Rightarrow f_2(1) \geq f_2(0) \Rightarrow y_2 = 1$ (tight case).

The case $t_1 = 1$ and $t_2 = 0$ cannot happen by the assumption that $\mathbf{t} \in \mathcal{Y}$.

If $t_1 = 1$ and $t_2 = 1$ then $y_1 = 1 \Rightarrow f_1(1) \geq f_1(0) \Rightarrow 0 \geq \delta - s_1(\mathbf{x}) \Rightarrow s_1(\mathbf{x}) \geq -\delta \Rightarrow s_2(\mathbf{x}) \geq \delta$ (by assumption) $\Rightarrow 0 \geq \delta - s_2(\mathbf{x}) \Rightarrow f_2(1) \geq f_2(0) \Rightarrow y_2 = 1$.

**Mutual Exclusion** Similarly, for the mutual exclusion $\neg y_1 \vee \neg y_2$, it is assumed that $\neg t_1 \vee \neg t_2$ in the target labeling and also that the score constraints hold, ensuring $\min(s_1(\mathbf{x}),\, s_2(\mathbf{x})) < -\delta$.

Consider the cases over possible assignments to $t_1$ and $t_2$:

If $t_1 = 0$ and $t_2 = 0$ then $y_1 = 1$ and $y_2 = 1$ implies that $s_1(\mathbf{x}) \geq -\delta$ and $s_2(\mathbf{x}) \geq -\delta$, which contradicts the constraint that $\min(s_1(\mathbf{x}),\, s_2(\mathbf{x})) < -\delta$ (tight case).

If $t_1 = 0$ and $t_2 = 1$ then $y_1 = 1$ and $y_2 = 1$ implies that $s_1(\mathbf{x}) \geq -\delta$ and $s_2(\mathbf{x}) \geq \delta$, which contradicts the same constraint.

If $t_1 = 1$ and $t_2 = 0$ then $y_1 = 1$ and $y_2 = 1$ implies that $s_1(\mathbf{x}) \geq \delta$ and $s_2(\mathbf{x}) \geq -\delta$, which again contradicts the same constraint.

The case $t_1 = 1$ and $t_2 = 1$ cannot happen by the assumption that $\mathbf{t} \in \mathcal{Y}$.

Therefore, since the concatenation, $\mathbf{y}$, of the independent maximizers of (5.21) is feasible, i.e. $\mathbf{y} \in \mathcal{Y}$, we have that the rightmost term in (5.21) equals the leftmost.

■