Large-scale Machine Learning over Graphs

Hanxiao Liu

November 2017

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Yiming Yang (Chair), Carnegie Mellon University
Jaime G. Carbonell, Carnegie Mellon University
J. Zico Kolter, Carnegie Mellon University
Karen Simonyan, DeepMind

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.

Keywords: Machine learning, spectral graph theory, optimization, semi-supervised learning, graph-based learning, time-series forecasting, evolution, neural architecture search

Abstract

Graphs are ubiquitous in statistical modeling and a broad range of machine learning applications. Examples are social networks, natural language dependency structures, latent interrelationships among tasks, and neural network topologies. Despite their versatility in representing structured data, how to fuse the information from heterogeneous and/or dynamically evolving graphs poses a grand challenge to existing machine learning theory and optimization algorithms. Furthermore, efficient graph topology optimization is another important but unsolved problem which entails searching over a combinatorially large discrete space. In this thesis, we address these open challenges in several complementary aspects:

In §1 we focus on a novel framework for fusing multiple heterogeneous graphs into a single homogeneous graph, on which learning tasks can be conveniently carried out in a principled manner. We also propose a new approach to impose analogical structures among heterogeneous nodes, which offers a theoretical unification of several representative models along with improved generalization.

In §2 we focus on graph induction problems in the context of graph-based semi-supervised learning. We start with a nonparametric method that is able to recover the optimal latent label diffusion pattern over the graph, and then generalize label diffusion processes as graph convolution operations whose filter weights are induced from data residing on the non-Euclidean manifold.

In §3 we extend the scope of our modeling from static graphs to dynamic graphs. Specifically, we develop an online algorithm for multi-task learning with provable sublinear regret bound, where a latent graph of task interdependencies is dynamically inferred on-the-fly. We also look at time-series forecasting tasks, showing that the explicitly modeling of the graph dependencies among temporally evolving variables can improve the prediction accuracy.

In §4 we formulate neural architecture search as a graph topology optimization problem. We present a simple yet efficient evolutionary algorithm that automatically identifies high-performing architectures based on a novel hierarchical representation scheme, where smaller operations are automatically discovered and reused as building blocks to form larger ones. The learned architecture achieves highly-competitive performance on ImageNet against the state-of-the-art, outperforming a large number of modern convolutional neural networks that were designed by hand.

Contents

List of Figures

List of Tables

Chapter 1

Learning over Heterogeneous Graphs

1.1 Learning across Graphs via Graph Product (Completed work, ICML'15, ICML'16)

Cross-graph Relational Learning (CGRL) refers to the problem of predicting the strengths or labels of multi-relational tuples of heterogeneous object types, through the joint inference over multiple graphs which specify the internal connections among each type of objects. CGRL is an open challenge in machine learning due to the daunting number of all possible tuples to deal with when the numbers of nodes in multiple graphs are large, and because the labeled training instances are extremely sparse as typical. Existing methods such as tensor factorization or tensor-kernel machines do not work well because of the lack of convex formulation for the optimization of CGRL models, the poor scalability of the algorithms in handling combinatorial numbers of tuples, and/or the non-transductive nature of the learning methods which limits their ability to leverage unlabeled data in training.

In this section, we propose a novel framework which formulates CGRL as a convex optimization problem, enables transductive learning using both labeled and unlabeled tuples, and offers a scalable algorithm that guarantees the optimal solution and enjoys a linear time complexity with respect to the sizes of input graphs. Experiments over citation networks and compound-protein interactions show the proposed method successfully scaled to large cross-graph inference problems, and outperformed other representative approaches significantly.

1.1.1 Motivation

Many important problems in multi-source relational learning could be cast as joint learning over multiple graphs about how heterogeneous types of objects interact with each other. In literature data analysis, for example, publication records provide rich information about how authors collaborate with each other in a co-authoring graph, how papers are linked in citation networks, how keywords are related via ontology, and so on. The challenging question is about how to combine such heterogeneous information in individual graphs for the labeling or scoring of the multi-relational associations in tuples like `(author,paper,keyword)`, given some observed instances of such tuples as the labeled training set. Automated labeling or scoring of unobserved

tuples allows us to discover who have been active in the literature on what areas of research, and to predict who would become influential in which areas in the future. In protein data analysis, as another example, a graph of proteins with pairwise sequence similarities is often jointly studied with a graph of chemical compounds with their structural similarities for the discovery of interesting patterns in `(compound,protein)` pairs. We call the prediction problem in both examples *cross-graph learning of multi-relational associations*, or simply *cross-graph relational learning* (CGRL), where the multi-relational associations are defined by the tuples of heterogeneous types of objects, and each object type has its own graph with type-specific relational structure as a part of the provided data. The task is to predict the labels or the scores of unobserved multi-relational tuples, conditioned on a relatively small set of labeled instances.

CGRL is an open challenge in machine learning for several reasons. Firstly, the number of multi-relational tuples grows combinatorially in the numbers of individual graphs and the number of nodes in each graph. How to make cross-graph inference computationally tractable for large graphs is a tough challenge. Secondly, how to combine the internal structures or relations in individual graphs for joint inference in a principled manner is an open question. Thirdly, supervised information (labeled instances) is typically extremely sparse in CGRL due to the very large number of all possible combinations of heterogeneous objects in individual graphs. Consequently, the success of cross-graph learning crucially depends on effectively leveraging the massively available unlabeled tuples (and the latent relations among them) in addition to the labeled training data. In other words, how to make the learning transductive is crucial for the true success of CGRL. Research on transdcutive CGRL has been quite limited, to our knowledge.

Existing approaches in CGRL or CGRL-related areas can be outlined as those using tensors or graph-regularized tensors, and kernel machines that combine multiple kernels.

Tensor methods have been commonly used for combining multi-source evidence of the interactions among multiple types of objects [54, 74, 81] as the combined evidence can be naturally represented as tuples. However, most of the tensor methods do not explicitly model the internal graph structure for each type of objects, although some of those methods implicitly leverage such information via graph-based regularization terms in their objective function that encourage similar objects within each graph to share similar latent factors [18, 71]. A major weakness in such tensor methods is the lack of convexity in their models, which leads to ill-posed optimization problems particularly in high-order scenarios. It has also been observed that tensor factorization models suffer from label-sparsity issue, which is typically severe in CGRL.

Kernel machines have been widely studied for supervised classifiers, where a kernel matrix corresponds to a similarity graph among a single type of objects. Multiple kernels can be combined, for example, by taking the tensor product of each individual kernel matrix, which results in a desired kernel matrix among cross-graph multi-relational tuples. The idea has been explored in relational learning combined with SVMs [10], perceptions [7] or Gaussian process [104] for two types of objects and is generalizable to the multi-type scenario of CGRL. Although being generic, the complexity of such kernel-based methods grows exponentially in the number of individual kernels (graphs) and the size of each individual graph. As a result, kernel machines suffer from poor scalability in general. In addition, kernel machines are purely supervised (not for transductive learning), i.e., they cannot leverage the massive number of available non-observed tuples induced from individual graphs and the latent connections among them. Those limitations make existing kernel methods less powerful for solving the CGRL problem in large scale and

under severely data-sparse conditions.

We propose a novel framework for CGRL which can be characterized as follows: (i) It uses graph products to map heterogeneous sources of information and the link structures in individual graphs onto a single *homogeneous* graph; (ii) It provides a convex formulation and approximation of the CGRL problem that ensure robust optimization and efficient computation; and (iii) It enables transductive learning in the form of label propagation over the induced homogeneous graph so that the massively available non-observed tuples and the latent connections among them can play an important role in effectively addressing the label-sparsity issue.

The proposed framework is most related to [64], where the authors formulated graph products for learning the edges of a bipartite graph. Our new framework is fundamentally different in two aspects. First, our new formulation and algorithms allow the number of individual graphs to be greater than two, while method in [64] is only applicable to two graphs. Secondly, the algorithms in [64] suffer from cubic complexity over the graphs sizes (quadratic by using a non-convex approximation), while our new algorithm enjoys both the convexity of the formulation and the low time complexity which is linear over the graph sizes.

Our method also shares the high-level goal with Statistical Relational Learning (SRL) [38] and Inductive Logic Programming (ILP) [61] in terms of multirelational learning. However, both of our problem setting and formulation differ substantially from existing SRL/ILP approaches focusing on first-order logic and/or probabilistic reasoning over graphical models.

1.1.2 The Proposed Framework

Notations

We are given J heterogeneous graphs where the j-th graph contains n_j vertices and is associated with an adjacency matrix $G^{(j)} \in \mathbb{R}^{n_j \times n_j}$. We use i_j to index the i_j-th vertex of graph j, and use a tuple $(i_1, \ldots, i_J)$ to index each multi-relation across the J graphs. The system predictions over all possible $\prod_{j=1}^{J} n_j$ multi-relations is summarized in an order-J tensor $f \in \mathbb{R}^{n_1 \times \cdots \times n_J}$, where $f_{i_1,i_2,\ldots,i_J}$ corresponds to the prediction about tuple $(i_1, \ldots, i_J)$.

Denote by $\otimes$ the Kronecker (Tensor) product. We use $\bigotimes_{j=1}^{J} x_j$ (or simply $\bigotimes_j x_j$) as the shorthand for $x_1 \otimes \cdots \otimes x_J$. Denote by $\times_j$ the j-mode product between tensors. We refer the readers to [54] for a thorough introduction about tensor mode product.

Graph Product

In a nutshell, graph product (GP) [1] is a mapping from each cross-graph multi-relation to each vertex in a new graph $\mathscr{P}$, whose edges encode similarities among the multi-relations (illustrated in Fig. 1.1). A desirable property of GP is it provides a natural reduction from the original multi-relational learning problem over *heterogeneous* information sources (Task 1.1.1) to an equivalent graph-based learning problem over a *homogeneous* graph (Task 1.1.2).

Task 1.1.1. *Given J graphs $G^{(1)}, \ldots, G^{(J)}$ with a small set of labeled multi-relations $\mathcal{O} = \{(i_1, \ldots, i_J)\}$, predict labels of the unlabeled multi-relations.*

[1] While traditional GP only applies to two graphs, we generalize it to the case of multiple graphs.

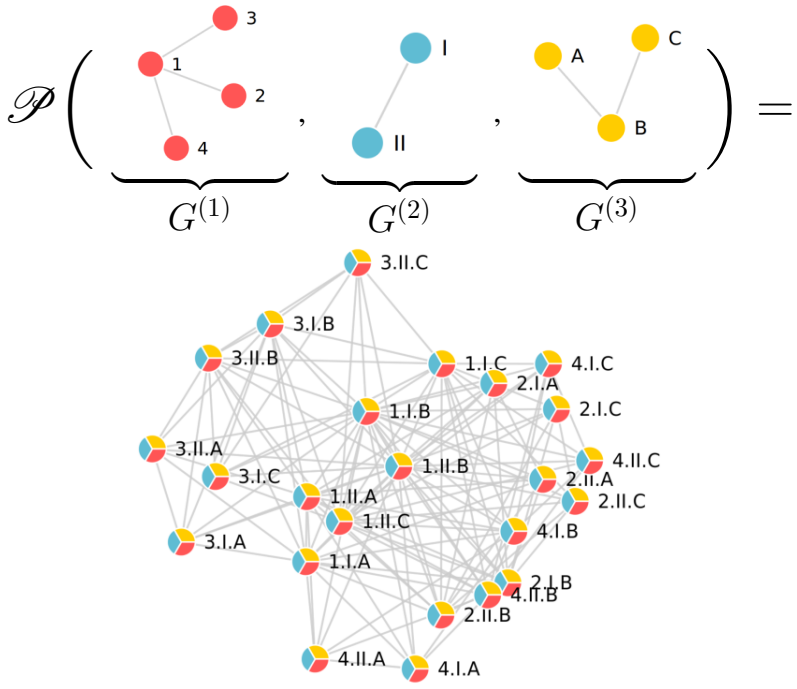

Figure 1.1: Product of three graphs $G^{(1)}$, $G^{(2)}$ and $G^{(3)}$. Each vertex in the resulting product graph $\mathscr{P}\big(G^{(1)}, G^{(2)}, G^{(3)}\big)$ corresponds to a multi-relation across the original graphs. For instance, vertex `3.II.B` in $\mathscr{P}$ corresponds to multi-relation `(3,II,B)` across $G^{(1)}$, $G^{(2)}$ and $G^{(3)}$.

Task 1.1.2. *Given the product graph $\mathscr{P}\big(G^{(1)}, \ldots, G^{(J)}\big)$ with a small set of labeled vertices $\mathcal{O} = \{(i_1, \ldots, i_J)\}$, predict labels of its unlabeled vertices.*

Spectral Graph Product

We define a parametric family of GP operators named the spectral graph product (SGP), which is of particular interest as it subsumes the well-known Tensor GP and Cartesian GP (Table 1.1), is well behaved (Theorem 1.1.1) and allows efficient optimization routines (Section 1.1.3).

Let $\lambda_{i_j}^{(j)}$ and $v_{i_j}^{(j)}$ be the i_j-th eigenvalue and eigenvector for the graph j, respectively. We construct SGP by defining the eigensystem of its adjacency matrix based on the provided J heterogeneous eigensystems of $G^{(1)}, \ldots, G^{(J)}$.

Definition 1.1.1. *The SGP of $G^{(1)}, \ldots, G^{(J)}$ is a graph consisting of $\prod_j n_j$ vertices, with its adjacency matrix $\mathscr{P}_\kappa := \mathscr{P}_\kappa\left(G^{(1)}, \ldots, G^{(J)}\right)$ defined by the following eigensystem*

$$\left\{\kappa\big(\lambda_{i_1}^{(1)}, \ldots, \lambda_{i_J}^{(J)}\big), \bigotimes_j v_{i_j}^{(j)}\right\}_{i_1,\ldots,i_J} \tag{1.1}$$

where κ is a pre-specified nonnegative nondecreasing function over $\lambda_{i_j}^{(j)}, \forall j = 1, 2, \ldots, J$.

In other words, the $(i_1, \ldots, i_J)$-th eigenvalue of $\mathscr{P}_\kappa$ is defined by coupling the $\lambda_{i_1}^{(1)}, \ldots, \lambda_{i_J}^{(J)}$ with function κ, and the $(i_1, \ldots, i_J)$-th eigenvector of $\mathscr{P}_\kappa$ is defined by coupling $v_{i_1}^{(1)}, \ldots, v_{i_J}^{(J)}$

via tensor (outer) product.

Remark 1.1.1. *If each individual* $\{v_{i_j}^{(j)}\}_{i_j=1}^{n_j}$ *forms an orthogonal basis in* $\mathbb{R}^{n_j}$, $\forall j \in 1, \ldots, J$, *then* $\{\bigotimes_j v_{i_j}^{(j)}\}_{i_1,\ldots,i_J}$ *forms an orthogonal basis in* $\mathbb{R}^{\prod_{j=1}^J n_j}$.

In the following example we introduce two special kinds of SGPs, assuming $J = 2$ for brevity. Higher-order cases are later summarized in Table 1.1.

Example 1.1.1. *Tensor GP defines* $\kappa(\lambda_{i_1}, \lambda_{i_2}) = \lambda_{i_1}\lambda_{i_2}$, *and is equivalent to Kronecker product:* $\mathscr{P}_{\text{Tensor}}\big(G^{(1)}, G^{(2)}\big) = \sum_{i_1,i_2}(\lambda_{i_1}\lambda_{i_2})\big(v_{i_1}^{(1)} \otimes v_{i_2}^{(2)}\big)\big(v_{i_1}^{(1)} \otimes v_{i_2}^{(2)}\big)^\top \equiv G^{(1)} \otimes G^{(2)}$.

Cartesian GP defines $\kappa(\lambda_{i_1}, \lambda_{i_2}) = \lambda_{i_1} + \lambda_{i_2}$, *and is equivalent to the Kronecker sum:* $\mathscr{P}_{\text{Cartesian}}\big(G^{(1)}, G^{(2)}\big) = \sum_{i_1,i_2}(\lambda_{i_1} + \lambda_{i_2})\big(v_{i_1}^{(1)} \otimes v_{i_2}^{(2)}\big)\big(v_{i_1}^{(1)} \otimes v_{i_2}^{(2)}\big)^\top \equiv G^{(1)} \oplus G^{(2)}$.

SGP Type	$\kappa\big(\lambda_{i_1}^{(1)}, \cdots, \lambda_{i_J}^{(J)}\big)$	$[\mathscr{P}_\kappa]_{(i_1,\cdots i_J),(i'_1,\cdots i'_J)}$
Tensor	$\prod_j \lambda_{i_j}^{(j)}$	$\prod_j G_{i_j,i'_j}^{(j)}$
Cartesian	$\sum_j \lambda_{i_j}^{(j)}$	$\sum_j G_{i_j,i'_j}^{(j)} \prod_{j'\neq j} \delta_{i_{j'}=i'_{j'}}$

Table 1.1: Tensor GP and Cartesian GP in higher-orders.

While Tensor GP and Cartesian GP provide mechanisms to associate multiple graphs in a multiplicative/additive manner, more complex cross-graph association patterns can be modeled by specifying κ. E.g., $\kappa(\lambda_{i_1}, \lambda_{i_2}, \lambda_{i_3}) = \lambda_{i_1}\lambda_{i_2} + \lambda_{i_2}\lambda_{i_3} + \lambda_{i_3}\lambda_{i_1}$ indicates pairwise associations are allowed among three graphs, but no triple-wise association is allowed as term $\lambda_{i_1}\lambda_{i_2}\lambda_{i_3}$ is not involved. Including higher order polynomials in κ amounts to incorporating higher-order associations among the graphs, which can be achieved by simply exponentiating κ.

Since what the product graph $\mathscr{P}$ offers is essentially a similarity measure among multi-relations, shuffling the order of input graphs $G^{(1)}, \ldots, G^{(J)}$ should not affect $\mathscr{P}$'s topological structure. For SGP, this property is guaranteed by the following theorem:

Theorem 1.1.1 (The Commutative Property)**.** *SGP is commutative (up to graph isomorphism) if* κ *is commutative.*

We omit the proof. The theorem suggests the SGP family is well-behaved as long as κ is commutative, which is true for both Tensor and Cartesian GPs as both multiplication and addition operations are order-insensitive.

Optimization Objective

It is often more convenient to equivalently write tensor f as a multi-linear map. E.g., when $J = 2$, tensor (matrix) $f \in \mathbb{R}^{n_1 \times n_2}$ defines a bilinear map from $\mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$ to $\mathbb{R}$ via $f(x_1, x_2) := x_1^\top f x_2$ and we have $f_{i_1,i_2} = f(e_{i_1}, e_{i_2})$. Such equivalence is analogous to high-order cases where f defines a multi-linear map from $\mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_J}$ to $\mathbb{R}$.

To carry out transductive learning over $\mathscr{P}_\kappa$ (Task 1.1.2), we inject the structure of the product graph into f via a Gaussian random fields prior [107]. The negative log-likelihood of the prior

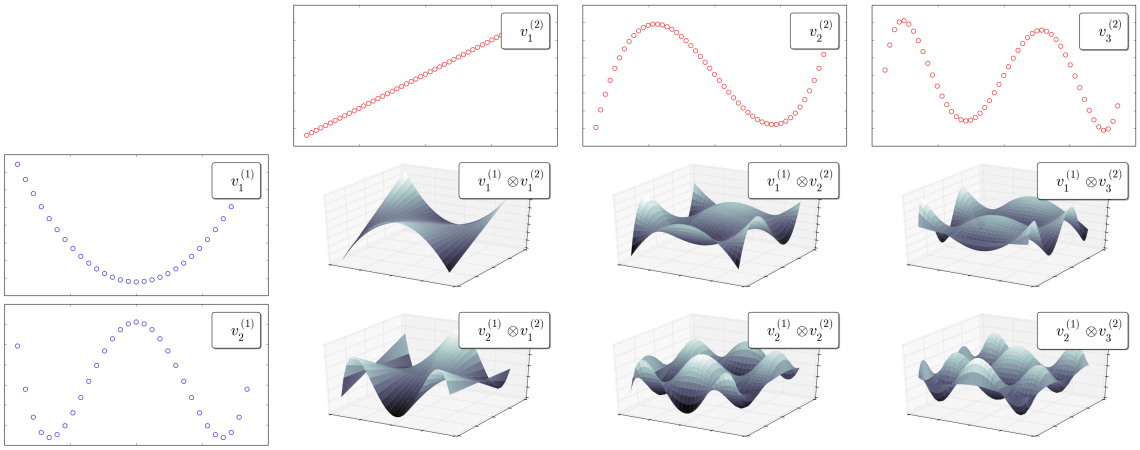

Figure 1.2: An illustration of the eigenvectors of $G^{(1)}$, $G^{(2)}$ and $\mathscr{P}\big(G^{(1)}, G^{(2)}\big)$. The leading nontrivial eigenvectors of $G^{(1)}$ and $G^{(2)}$ are denoted by blue and red curves, respectively. The induced leading nontrivial eigenvectors of $\mathscr{P}\big(G^{(1)}, G^{(2)}\big)$ are illustrated in 3D. If $G^{(1)}$ and $G^{(2)}$ are symmetrically normalized, their eigenvectors (corresponding to eigenvectors of the graph Laplacian) will be ordered by smoothness w.r.t. the graph structures. As a result, eigenvectors of $\mathscr{P}\big(G^{(1)}, G^{(2)}\big)$ will also be ordered by smoothness.

$-\log p\left(f \mid \mathscr{P}_\kappa\right)$ is the same (up to constant) as the following squared semi-norm

$$\|f\|_{\mathscr{P}_\kappa}^2 = vec(f)^\top \mathscr{P}_\kappa^{-1} vec(f) = \sum_{i_1, i_2, \ldots, i_J} \frac{f\big(v_{i_1}^{(1)}, \ldots, v_{i_J}^{(J)}\big)^2}{\kappa\big(\lambda_{i_1}^{(1)}, \ldots, \lambda_{i_J}^{(J)}\big)} \tag{1.2}$$

Our optimization objective is therefore defined as

$$\min_{f \in \mathbb{R}^{n_1 \times \cdots \times n_J}} \ \ell_{\mathcal{O}}(f) + \frac{\gamma}{2}\|f\|_{\mathscr{P}_\kappa}^2 \tag{1.3}$$

where $\ell_{\mathcal{O}}(\cdot)$ is a loss function to be defined later (Section 1.1.4), $\mathcal{O}$ is the set of training tuples, and γ is a tuning parameter controlling the strength of graph regularization.

1.1.3 Approximation

The computational bottleneck for optimization (1.3) lies in evaluating $\|f\|_{\mathscr{P}_\kappa}^2$ and its first-order derivative, due to the extremely large size of $\mathscr{P}_\kappa$. In the following, we first identify the computation bottleneck of using the exact formulation, based on which we propose our convex approximation scheme that reduces the time complexity of evaluating the semi-norm $\|f\|_{\mathscr{P}_\kappa}^2$ from $O\Big(\big(\sum_j n_j\big)\big(\prod_j n_j\big)\Big)$ to $O\big(\prod_j d_j\big)$, where $d_j \ll n_j$ for $j = 1, \ldots, J$.

Complexity of the Exact Formulation

The brute-force evaluation of $\|f\|_{\mathscr{P}_\kappa}^2$ according to (1.2) costs $O\big(\big(\prod_j n_j\big)^2\big)$, as one has to evaluate $O\big(\prod_j n_j\big)$ terms inside the summation where each term costs $O\big(\prod_j n_j\big)$. However, redun-

dancies exist and the minimum complexity for the exact evaluation is given as follows

Proposition 1.1.1. *The exact evaluation of semi-norm $\|f\|_{\mathscr{P}_\kappa}$ takes $O\big(\big(\sum_j n_j\big)\big(\prod_j n_j\big)\big)$ flops.*

Proof. Notice that the collection of all numerators in (1.2), namely $\big[f\big(v_{i_1}^{(1)},\ldots,v_{i_J}^{(J)}\big)\big]_{i_1,\cdots,i_J}$, is a tensor in $\mathbb{R}^{n_1\times\cdots\times n_J}$ that can be precomputed via

$$\Big(\big(f\times_1 V^{(1)}\big)\times_2 V^{(2)}\Big)\cdots\times_J V^{(J)} \tag{1.4}$$

where $\times_j$ stands for the j-mode product between a tensor in $\mathbb{R}^{n_1\times\cdots\times n_j\times\cdots\times n_J}$ and $V^{(j)}\in\mathbb{R}^{n_j\times n_j}$. The conclusion follows as the j-th mode product in (1.4) takes $O\big(n_j\prod_j n_j\big)$ flops, and one has to do this for each $j=1,\ldots,J$. When $J=2$, (1.4) reduces to the multiplication of three matrices ${V^{(1)}}^\top f V^{(2)}$ at the complexity of $O\left((n_1+n_2)n_1n_2\right)$. □

Approximation via Tucker Form

Equation (1.4) implies the key for complexity reduction is to reduce the cost of the j-mode multiplications $\cdot\times_j V^{(j)}$. Such multiplication costs $O\big(n_j\prod_j n_j\big)$ in general, but can be carried out more efficiently if f is structured.

Our solution is twofold: First, we include only the top-d_j eigenvectors in $V^{(j)}$ for each graph $G^{(i)}$, where $d_j\ll n_j$. Hence each $V^{(j)}$ becomes a thin matrix in $\mathbb{R}^{n_j\times d_j}$. Second, we restrict tensor f to be within the linear span of the top $\prod_{j=1}^J d_j$ eigenvectors of the product graph $\mathscr{P}_\kappa$

$$f=\sum_{k_1,\cdots,k_J=1}^{d_1,\cdots,d_J}\alpha_{k_1,\cdots,k_J}\bigotimes_j v_{k_j}^{(j)} \tag{1.5}$$

$$=\alpha\times_1 V^{(1)}\times_2 V^{(2)}\times_3\cdots\times_J V^{(J)} \tag{1.6}$$

The combination coefficients $\alpha\in\mathbb{R}^{d_1\times\cdots\times d_J}$ is known as the core tensor of Tucker decomposition. In the case where $J=2$, the above is equivalent to saying $f\in\mathbb{R}^{n_1\times n_2}$ is a low-rank matrix parametrized by $\alpha\in\mathbb{R}^{d_1\times d_2}$ such that $f=\sum_{k_1,k_2}\alpha_{k_1,k_2}v_{k_1}^{(1)}{v_{k_2}^{(2)}}^\top=V^{(1)}\alpha{V^{(2)}}^\top$.

Combining (1.5) with the orthogonality property of eigenvectors leads to the fact that $f\big(v_{k_1}^{(1)},\ldots,v_{k_J}^{(J)}\big)=\alpha_{k_1,\cdots,k_J}$. To see this for $J=2$, notice $f\big(v_{k_1}^{(1)},v_{k_2}^{(2)}\big)={v_{k_1}^{(1)}}^\top f v_{k_1}^{(2)}={v_{k_1}^{(1)}}^\top V^{(1)}\alpha{V^{(2)}}^\top v_{k_1}^{(2)}=e_{k_1}^\top\alpha e_{k_2}=\alpha_{k_1,k_2}$. Therefore the semi-norm in (1.2) can be simplified as

$$\|f\|_{\mathscr{P}_\kappa}^2=\|\alpha\|_{\mathscr{P}_\kappa}^2=\sum_{k_1,\ldots,k_J=1}^{d_1,\cdots,d_J}\frac{\alpha_{k_1,\cdots,k_J}^2}{\kappa\big(\lambda_{k_1}^{(1)},\ldots,\lambda_{k_J}^{(J)}\big)} \tag{1.7}$$

Comparing (1.7) with (1.2), the number of inside-summation terms is reduced from $O\big(\prod_j n_j\big)$ to $O\big(\prod_j d_j\big)$ where $d_j\ll n_j$. In addition, the cost for evaluating each term inside summation is reduced from $O\big(\prod_j n_j\big)$ to $O(1)$.

Denote by $V_{i_j}^{(j)}\in\mathbb{R}^{d_j}$ the i_j-th row of $V^{(j)}$, we obtain the following optimization by replacing f with α in (1.3)

$$\begin{aligned}\min_{\alpha\in\mathbb{R}^{d_1\times\cdots\times d_J}}\ &\ell_{\mathcal{O}}(f)+\frac{\gamma}{2}\|\alpha\|_{\mathscr{P}_\kappa}^2\\ \text{s.t.}\quad &f=\alpha\times_1 V^{(1)}\times_2\cdots\times_J V^{(J)}\end{aligned} \tag{1.8}$$

Optimization above has intuitive interpretations. In principle, it is natural to emphasis bases in f that are "smooth" w.r.t. the manifold structure of $\mathscr{P}_\kappa$, and de-emphasis those that are "non-smooth" in order to obtain a parsimonious hypothesis with strong generalization ability. We claim this is exactly the role of regularizer (1.7). To see this, note any nonsmooth basis $\bigotimes_j v_{k_j}^{(j)}$ of $\mathscr{P}_\kappa$ is likely to be associated with small a eigenvalue $\kappa\big(\lambda_{k_1}^{(1)}, \dots, \lambda_{k_J}^{(J)}\big)$ (illustrated in Fig. 1.2). The conclusion follows by noticing that $\alpha_{k_1,\dots,k_J}$ is essentially the activation strength of $\bigotimes_j v_{k_j}^{(j)}$ in f (implied by (1.5)), and that (1.7) is going to give any $\alpha_{k_1,\dots,k_J}$ associated with a small $\kappa\big(\lambda_{k_1}^{(1)}, \dots, \lambda_{k_J}^{(J)}\big)$ a stronger penalty.

(1.8) is a convex optimization problem over α with any convex $\ell_{\mathcal{O}}(\cdot)$. Spectral approximation techniques for graph-based learning has been found successful in standard classification tasks [33], which are special cases under our framework when $J = 1$. We introduce this technique for multi-relational learning, which is particularly desirable as the complexity reduction will be much more significant for high-order cases $(J >= 2)$.

While f in (1.5) is assumed to be in the Tucker form, other low-rank tensor representation schemes are potentially applicable. E.g., the Candecomp/Parafac (CP) form that further restricts α to be diagonal, which is more aggressive but substantially less expressive. The Tensor-Train decomposition [77] offers an alternative representation scheme in the middle of Tucker and CP, but the resulting optimization problem will suffer from non-convexity.

1.1.4 Optimization

We define $\ell_{\mathcal{O}}(f)$ to be the ranking ℓ_2-hinge loss

$$\ell_{\mathcal{O}}(f) = \frac{\sum_{\substack{(i_1,\dots,i_J)\in\mathcal{O}\\(i'_1,\dots,i'_J)\in\bar{\mathcal{O}}}} \Big(f_{i_1\dots i_J} - f_{i'_1\dots i'_J}\Big)_+^2}{|\mathcal{O}\times\bar{\mathcal{O}}|} \tag{1.9}$$

where $(x)_+ = \max(0, 1-x)$, $\bar{\mathcal{O}}$ is the complement of $\mathcal{O}$ w.r.t. all possible multi-relations. Eq. (1.9) encourages the valid tuples in our training set $\mathcal{O}$ to be ranked higher than those corrupted ones in $\bar{\mathcal{O}}$, and is known to be a surrogate of AUC.

We use stochastic gradient descent for optimization as $|\mathcal{O}|$ is usually large. In each iteration, a random valid multirelation $(i_1,\dots,i_J)$ is uniformly drawn from $\mathcal{O}$, a random corrupted multirelation $(i'_1,\dots,i'_J)$ is uniformly drawn from $\bar{\mathcal{O}}$. Each noisy gradient is computed as

$$\nabla_\alpha = \frac{\partial \ell_{\mathcal{O}}}{\partial f}\left(\frac{\partial f_{i_1,\dots,i_J}}{\partial \alpha} - \frac{\partial f_{i'_1,\dots,i'_J}}{\partial \alpha}\right) + \gamma\alpha \oslash \kappa \tag{1.10}$$

where we abuse the notation by defining $\kappa \in \mathbb{R}^{d_1\times\cdots\times d_J}$, $\kappa_{k_1,\dots,k_J} := \kappa\big(\lambda_{k_1}^{(1)}, \dots, \lambda_{k_J}^{(J)}\big)$; $\oslash$ is the element-wise division between tensors. The gradient w.r.t. α in (1.10) is

$$\frac{\partial f_{i_1,\dots,i_J}}{\partial \alpha} = \frac{\partial\big(\alpha \times_1 V_{i_1}^{(1)} \times_2 \cdots \times_J V_{i_J}^{(J)}\big)}{\partial \alpha} \tag{1.11}$$

$$= \bigotimes_j V_{i_j}^{(j)} \quad \in \mathbb{R}^{d_1\times\dots d_J} \tag{1.12}$$

Each SGD iteration costs $O\big(\prod_j d_j\big)$ flops, which is independent from $n_1, n_2, \ldots, n_J$. After obtaining the solution $\hat{\alpha}(\kappa)$ of optimization (1.8) for any given SGP $\mathscr{P}_\kappa$, our final predictions in $\hat{f}(\kappa)$ can be recovered via (1.5).

Following AdaGrad [28], we allow adaptive step sizes for each element in α. That is, in the t-th iteration we use $\eta^{(t)}_{k_1,\ldots,k_J} = \eta_0 \Big/ \Big[\sum_{\tau=1}^{t} \nabla^{(\tau)}_{\alpha_{k_1,\ldots,k_J}}{}^2\Big]^{\frac{1}{2}}$ as the step size for $\alpha_{k_1,\ldots,k_J}$, where $\big\{\nabla^{(\tau)}_{\alpha_{k_1,\ldots,k_J}}\big\}_{\tau=0}^{t}$ are historical gradients associated with $\alpha_{k_1,\ldots,k_J}$ and η_0 is the initial step size (set to be 1). The strategy is particularly efficient with highly redundant gradients, which is our case where the gradient is a regularized rank-2 tensor, according to (1.10) and (1.12).

In practice (especially for large J), the computation cost of tensor operations involving $\bigotimes_{j=1}^{J} V^{(j)}_{i_j} \in \mathbb{R}^{d_1,\ldots,d_J}$ is not ignorable even if $d_1, d_2, \ldots, d_J$ are small. Fortunately, such medium-sized tensor operations in our algorithm are highly parallelable over GPU.

1.1.5 Experiments

Datasets

We evaluate our method on real-world data in two different domains: the Enzyme dataset [101] for compound-protein interaction and the DBLP dataset of scientific publication records. Fig. 1.3 illustrates their heterogeneous objects and relational structures.

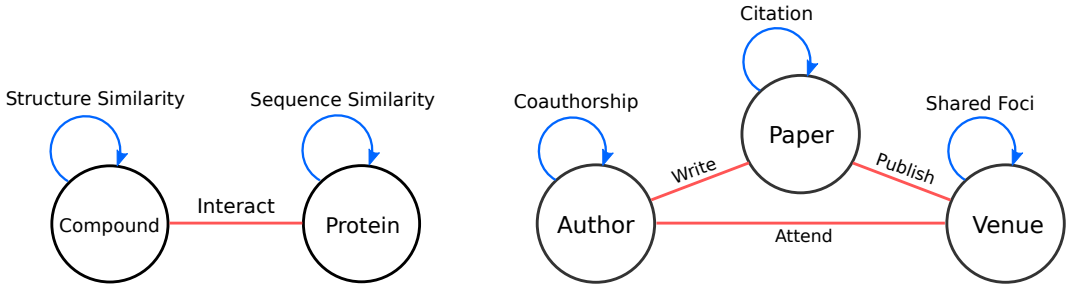

Figure 1.3: The heterogeneous types of objects (the circles) and the relational structures in the Enzyme (left) and DBLP (right) data sets. The blue edges represent the within-graph relations and the red edges represent the cross-graph interactions. The corresponding tuples in Enzyme is in the form of `(Compound,Protein)`, and in DBLP is in the form of `(Author,Paper,Venue)`.

The Enzyme dataset has been used for modeling and predicting drug-target interactions, which contains a graph of 445 chemical compounds (drugs) and a graph of 664 proteins (targets). The prediction task is to label the unknown compound-protein interactions based on both the graph structures and a small set of 2,926 known interactions. The graph of compounds is constructed based on the SIMCOMP score [42], and the graph of proteins is constructed based on the normalized SmithWaterman score [86]. While both graphs are provided in the dense form, we converted them into sparse kNN graphs where each vertex is connected with its top 1% neighbors.

As for the DBLP dataset, we use a subset of 34,340 DBLP publication records in the domain of Artificial Intelligence [91], from which 3 graphs are constructed as:

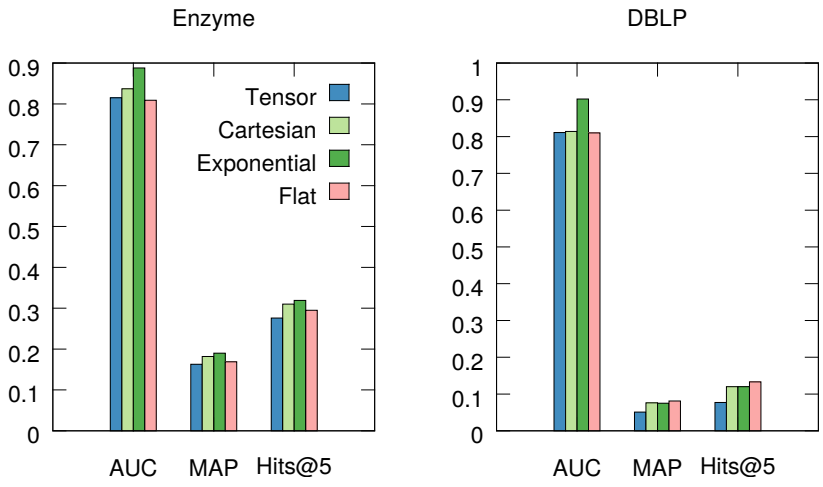

Figure 1.4: Performance of TOP with different SGPs.

- For the author graph ($G^{(1)}$) we draw an edge between two authors if they have coauthored an overlapping set of papers, and remove the isolated authors using a DFS algorithm. We then obtain a symmetric kNN graph by connecting each author with her top 0.5% nearest neighbors using the count of co-authored papers as the proximity measure. The resulting graph has 5,517 vertices with 17 links per vertex on average.
- For the paper graph ($G^{(2)}$) we connect two papers if both of them cite another paper, or are cited by another paper. Like $G^{(1)}$, we remove isolated papers using DFS and construct a symmetric 0.5%-NN graph. To measure the similarity of any given pair of papers, we represent each paper as a bag-of-citations and compute their cosine similarity. The resulted graph has 11,879 vertices and has an average degree of 50.
- For the venue graph ($G^{(3)}$) we connect two venues if they share similar research focus. The venue-venue similarity is measured by the total number of cross-citations in between, normalized by the size of the two venues involved. The symmetric venue graph has 22 vertices and an average degree of 7.

Tuples in the form of `(Author,Paper,Venue)` are extracted from the publication records, and there are 15,514 tuples (cross-graph interactions) after preprocessing.

Methods for Comparison

- Transductive Learning over Product Graph (**TOP**).
  The proposed method. We explore different choices of κ's for parametrizing the spectral graph product as in Table 1.1.5.

- Tensor Factorization (**TF**) and Graph-regularized TF (**GRTF**). In TF we factorize $f \in \mathbb{R}^{n_1 \times \cdots \times n_J}$ as a set of dimensionality-reduced latent factors $C^{d_1, \times \cdots \times d_J}$, $U_1^{n_1 \times d_1}, \ldots, U_J \in \mathbb{R}^{n_J \times d_J}$. In GRTF, we further enhanced the traditional TF by adding graph regularizations

Name	$\kappa(x, y)$ $(J = 2)$	$\kappa(x, y, z)$ $(J = 3)$
Tensor	xy	xyz
Cartesian	$x + y$	$x + y + z$
Exponential	e^{x+y}	$e^{xy+yz+xz}$
Flat	1	1

to the objective function, which enforce the model to be aware of the context information in $G^{(j)}$'s [18, 71];

- One-class Nearest Neighbor (**NN**). We score each tuple $(i_1, \ldots, i_J)$ in the test set with $\hat{f}(i_1, \ldots, i_J) = \max_{(i'_1,\ldots,i'_J)\in\mathcal{O}} \prod_{j=1}^{J} G_{i_j i'_j}$. That is, we assume the tuple-tuple similarity can be factorized as the product of vertex-level similarities across different graphs. We experimented with several other similarity measures and empirically found the multiplicative similarity leads to the best overall performance. Note it does not rely on the presence of any negative examples.
- Ranking Support Vector Machines [51] (**RSVM**). For the task of completing the missing paper in `(Author,?,Venue)`, we use a Learning-to-Rank strategy by treating `(Author,Venue)` as the query and `Paper` as the document to be retrieved. The query feature is constructed by concatenating the eigen-features of `Author` and `Venue`, where we define the eigen-feature of vertex i_j in graph j as $V_{i_j}^{(j)} \in \mathbb{R}^{d_j}$. The feature for each query-document pair is obtained by taking the tensor product of the query feature and document eigen-feature.
- Low-rank Tensor Kernel Machines (**LTKM**). While traditional tensor-based kernel construction methods for tuples suffer from poor scalability. We propose to speedup by replacing each individual kernel with its low-rank approximation before tensor product, leading to a low-rank kernel of tuples which allows more efficient optimization routines.

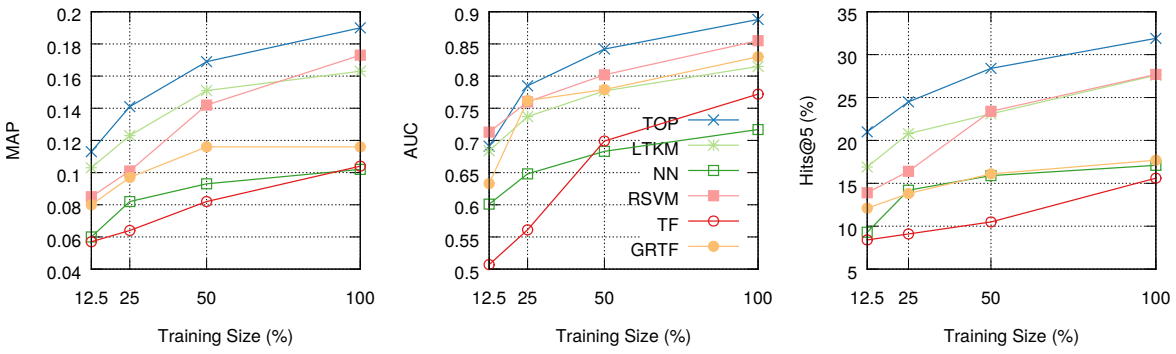

Figure 1.5: Test-set performance of different methods on Enzyme.

For fair comparison, TF, GRTF, RSVM and LTKM use exactly the same loss as that for TOP, i.e. e.q. (1.9). All algorithms are trained using a mini-batched stochastic gradient descent. We use the same eigensystems (eigenvectors and eigenvalues) of the $G^{(j)}$'s as the input for TOP, RSVM and LTKM. The number of top-eigenvalues/eigenvectors d_j for graph j is chosen such

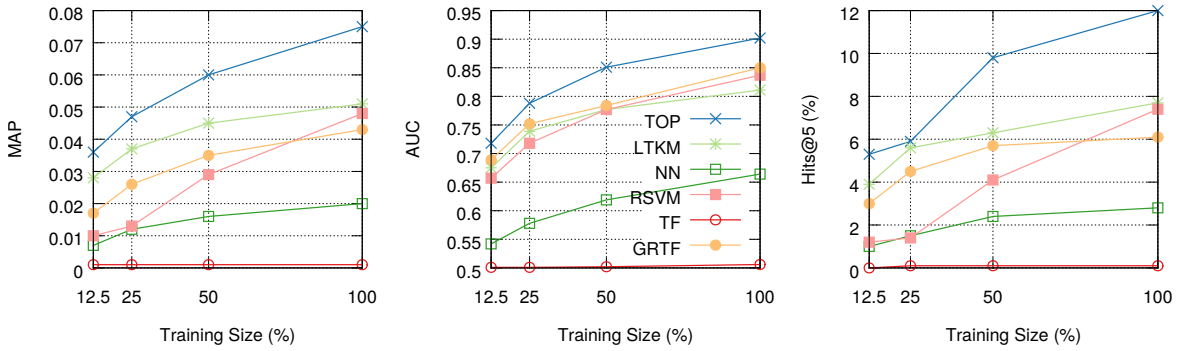

Figure 1.6: Test-set performance of different methods on DBLP.

that $\lambda_1^{(j)}, \ldots, \lambda_{d_j}^{(j)}$ approximately cover 80% of the total spectral energy of $G^{(j)}$. Under this criterion, we use $d_1 = 1,281$, $d_2 = 2,170$, $d_3 = 6$ for DBLP; $d_1 = 150$, $d_2 = 159$ for Enzyme.

Experiment Setups

For both datasets, we randomly sample one third of known interactions for training (denoted by $\mathcal{O}$), one third for validation and use the remaining ones for testing. Known interactions in the test set, denoted by $\mathcal{T}$, are treated as positive examples. All tuples not in $\mathcal{T}$, denoted by $\bar{\mathcal{T}}$, are treated as negative. Tuples present in $\mathcal{O}$ are removed from $\bar{\mathcal{T}}$ to avoid misleading results [15].

We measure algorithm performance on Enzyme based on the quality of inferred target proteins given each compound, namely by the ability of completing `(Compound,?)`. For DBLP, the performance is measured by the quality of inferred papers given author and venue, namely by the ability of completing `(Author,?,Venue)`. We use Mean Average Prevision (MAP), Area Under the Curve (AUC) and Hits at Top 5 (Hits@5) as our evaluation metrics.

Results

Fig. 1.4 compares the results of TOP with various parameterizations of the spectral graph product (SGP). Among those, Exponential κ works better on average.

Figs. 1.5 and 1.6 show the main results, comparing TOP (with Exponential κ) with other representative baselines. Clearly, TOP outperforms all the other methods on both datasets in all the evaluation metrics of MAP [2], AUC and Hit@5.

Fig. 1.7 shows the performance curves of TOP on Enzyme over different model sizes (by varying the d_j's). With a relatively small model size compared with using the full spectrum, TOP's performance converges to the optimal point.

1.2 Leveraging Analogical Structures (Completed work, ICML'17)

Large-scale multi-relational embedding refers to the task of learning the latent representations for entities and relations in large knowledge graphs. An effective and scalable solution for this prob-

[2]MAP scores for random guessing are 0.014 on Enzyme and 0.00072 on DBLP, respectively.

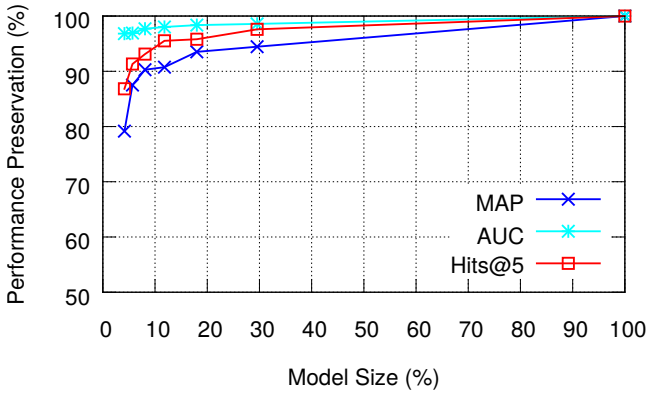

Figure 1.7: Performance of TOP v.s. model size on Enzyme.

lem is crucial for the true success of knowledge-based inference in a broad range of applications. This section proposes a novel framework for optimizing the latent representations with respect to the *analogical* properties of the embedded entities and relations. By formulating the learning objective in a differentiable fashion, our model enjoys both theoretical power and computational scalability, and significantly outperformed a large number of representative baselines on benchmark datasets. The model further offers an elegant unification of several well-known methods in relational embedding, which can be proven to be special instantiations of our framework.

1.2.1 Motivation

Multi-relational embedding, or knowledge graph embedding, is the task of finding the latent representations of entities and relations for better inference over knowledge graphs. This problem has become increasingly important in recent machine learning due to the broad range of important applications of large-scale knowledge bases, such as Freebase [12], DBpedia [5] and Google's Knowledge Graph [85], including question-answering [34], information retrieval [25] and natural language processing [35].

A knowledge base (KB) typically stores factual information as subject-relation-object triplets. The collection of such triplets forms a directed graph whose nodes are entities and whose edges are the relations among entities. Real-world knowledge graph is both extremely large and highly incomplete by nature [69]. How can we use the observed triplets in an incomplete graph to induce the unobserved triples in the graph presents a tough challenge for machine learning research.

Various statistical relational learning methods [38, 75] have been proposed for this task, among which vector-space embedding models are most particular due to their advantageous performance and scalability [15]. The key idea in those approaches is to find dimensionality reduced representations for both the entities and the relations, and hence force the models to generalize during the course of compression. Representative models of this kind include tensor factorization [74, 85], neural tensor networks [21, 88], translation-based models [15, 63, 96], bilinear models and its variants [94, 102], pathwise methods [41], embeddings based on holographic representations [76], and product graphs that utilizes additional site information for the predictions of unseen edges in a semi-supervised manner [64, 65]. Learning the embeddings of

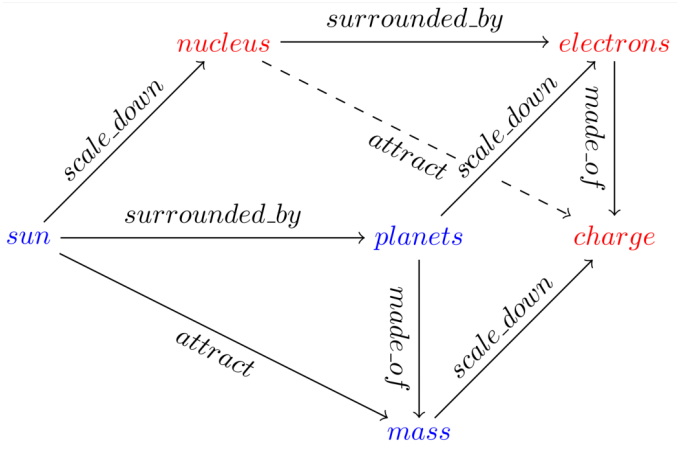

Figure 1.8: Commutative diagram for the analogy between the Solar System (blue) and the Rutherford-Bohr Model (red) (atom system). By viewing the atom system as a "miniature" of the solar system (via the $scale\_down$ relation), one is able to complete missing facts (triplets) about the latter by mirroring the facts about the former. The analogy is built upon three basic analogical structures (parallelograms): "sun is to $planets$ as $nucleus$ is to $electrons$", "sun is to $mass$ as $nucleus$ is to $charge$" and "$planets$ are to $mass$ as $eletrons$ are to $charge$".

entities and relations can be viewed as a knowledge induction process, as those induced latent representations can be used to make inference about new triplets that have not been seen before.

Despite the substantial efforts and great successes so far in the research on multi-relational embedding, one important aspect is missing, i.e., to study the solutions of the problem from the *analogical inference* point of view, by which we mean to rigorously define the desirable analogical properties for multi-relational embedding of entities and relations, and to provide algorithmic solution for optimizing the embeddings w.r.t. the analogical properties. We argue that analogical inference is particularly desirable for knowledge base completion, since for instance if system A (a subset of entities and relations) is analogous to system B (another subset of entities and relations), then the unobserved triples in B could be inferred by mirroring their counterparts in A. Figure 1.8 uses a toy example to illustrate the intuition, where system A corresponds to the solar system with three concepts (entities), and system B corresponds the atom system with another three concepts. An analogy exists between the two systems because B is a "miniature" of A. As a result, knowing how the entities are related to each other in system A allows us to make inference about how the entities are related to each other in system B by analogy.

Although *analogical reasoning* was an active research topic in classic AI (artificial intelligence), early computational models mainly focused on non-differentiable rule-based reasoning [31, 37, 95], which can hardly scale to very large KBs such as Freebase or Google's Knowledge Graph. How to leverage the intuition of analogical reasoning via statistical inference for automated embedding of very large knowledge graphs has not been studied so far, to our knowledge.

It is worth mentioning that analogical structures have been observed in the output of several word/entity embedding models [68, 78]. However, those observations stopped there as merely empirical observations. Can we mathematically formulate the desirable analogical structures and leverage them in our objective functions to improve multi-relational embedding? In this case, can

we develop new algorithms for tractable inference for the embedding of very large knowledge graphs? These questions present a fundamental challenge which has not been addressed by existing work, and answering these questions are the main contributions we aim in this section. We name this open challenge as the *analogical inference* problem, for the distinction from rule-based *analogical reasoning* in classic AI.

Our specific novel contributions are the following:

1. A new framework that, for the first time, explicitly models analogical structures in multi-relational embedding, and state-of-the-art performance on benchmark datasets;
2. The algorithmic solution for conducting analogical inference in a differentiable manner, whose implementation is as scalable as the fastest known relational embedding algorithms;
3. The theoretical insights on how our framework provides a unified view of several representative methods as its special (and restricted) cases, and why the generalization of such cases lead to the advantageous performance of our method as empirically observed.

1.2.2 The Proposed Framework

Analogical reasoning is known to play a central role in human induction about knowledge [37, 44, 45, 70]. Here we provide a mathematical formulation of the analogical structures of interest in multi-relational embedding in a latent semantic space, to support algorithmic inference about the embeddings of entities and relations in a knowledge graph.

Notations

Let $\mathcal{E}$ and $\mathcal{R}$ be the space of all entities and their relations. A knowledge base $\mathcal{K}$ is a collection of triplets $(s, r, o) \in \mathcal{K}$ where $s \in \mathcal{E}, o \in \mathcal{E}, r \in \mathcal{R}$ stand for the subject, object and their relation, respectively. Denote by $v \in \mathbb{R}^{|\mathcal{E}| \times m}$ a look-up table where $v_e \in \mathbb{R}^m$ is the vector embedding for entity e, and denote by tensor $W \in \mathbb{R}^{|\mathcal{R}| \times m \times m}$ another look-up table where $W_r \in \mathbb{R}^{m \times m}$ is the matrix embedding for relation r. Both v and W are to be learned from $\mathcal{K}$.

Relations as Linear Maps

We formulate each relation r as a linear map that, for any given $(s, r, o) \in \mathcal{K}$, transforms the subject s from its original position in the vector space to somewhere near the object o. In other words, we expect the latent representations for any valid (s, r, o) to satisfy

$$v_s^\top W_r \approx v_o^\top \tag{1.13}$$

The degree of satisfaction in the approximated form of (1.13) can be quantified using the inner product of $v_s^\top W_r$ and v_o. That is, we define a bilinear score function as:

$$\phi(s, r, o) = \langle v_s^\top W_r, v_o \rangle = v_s^\top W_r v_o \tag{1.14}$$

Our goal is to learn v and W such that $\phi(s, r, o)$ gives high scores to valid triples, and low scores to the invalid ones. In contrast to some previous models [15] where relations are modeled as additive translating operators, namely $v_s + w_r \approx v_o$, the multiplicative formulation in (1.13) offers

a natural analogy to the first-order logic where each relation is treated as a predicate operator over input arguments (subject and object in our case). Clearly, the linear transformation defined by a matrix is a richer operator than the additive transformation defined by a vector. Multiplicative models are also found to substantially outperform additive models empirically [74, 102].

Normal Transformations

Instead using arbitrary matrices to implement linear maps, a particular family of matrices has been studied for "well-behaved" linear maps. This family is named as the *normal matrices*.

Definition 1.2.1 (Normal Matrix)**.** *A real matrix A is normal if and only if $A^\top A = AA^\top$.*

Normal matrices have nice theoretical properties which are often desirable form relational modeling, e.g., they are unitarily diagonalizable and hence can be conveniently analyzed by the spectral theorem [29]. Representative members of the normal family include:

- Symmetric Matrices for which $W_r W_r^\top = W_r^\top W_r = W_r^2$. These includes all diagonal matrices and positive semi-definite matrices, and the symmetry implies $\phi(s, r, o) = \phi(o, r, s)$. They are suitable for modeling symmetric relations such as $is\_identical$.
- Skew-/Anti-symmetric Matrices for which $W_r W_r^\top = W_r^\top W_r = -W_r^2$, which implies $\phi(s, r, o) = -\phi(o, r, s)$. These matrices are suitable for modeling asymmetric relations such as is_parent_of.
- Rotation Matrices for which $W_r W_r^\top = W_r^\top W_r = I_m$, which suggests that the relation r is invertible as W_r^{-1} always exists. Rotation matrices are suitable for modeling 1-to-1 relationships (bijections).
- Circulant Matrices [39], which have been implicitly used in recent work on holographic representations [76]. These matrices are usually related to the learning of latent representations in the Fourier domain.

In the remaining parts, we denote all the real normal matrices in $\mathbb{R}^{m\times m}$ as $\mathcal{N}_m(\mathbb{R})$.

Analogical Structures

Consider the famous example in the word embedding literature [68, 78], for the following entities and relations among them:

"man is to $king$ as $woman$ is to $queen$"

In an abstract notion we denote the entities by a (as man) , b (as $king$), c (as $woman$) and d (as $queen$), and the relations by r (as $crown$) and r' (as $male \mapsto female$), respectively. These give us the subject-relation-object triplets as follows:

$$a \xrightarrow{r} b, \quad c \xrightarrow{r} d, \quad a \xrightarrow{r'} c, \quad b \xrightarrow{r'} d \tag{1.15}$$

For multi-relational embeddings, r and r' are members of $\mathcal{R}$ and are modeled as linear maps.

The relational maps in (1.15) can be visualized using a commutative diagram [2, 17] from the Category Theory, as shown in Figure 1.9, where each node denotes an entity and each edge

denotes a linear map that transforms one entity to the other. We also refer to such a diagram as a "parallelogram" to highlight its particular *algebraic structure*[3].

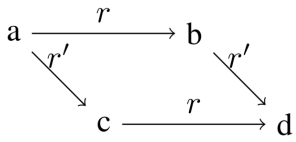

Figure 1.9: Parallelogram diagram for the analogy of "a is to b as c is to d", where each edge denotes a linear map.

The parallelogram in Figure 1.9 represents a very basic analogical structure which could be informative for the inference about unknown facts (triplets). To get a sense about why analogies would help in the inference about unobserved facts, we notice that for entities a, b, c, d which form an analogical structure in our example, the parallelogram structure is fully determined by symmetry. This means that if we know $a \xrightarrow{r} b$ and $a \xrightarrow{r'} c$, then we can induce the remaining triplets of $c \xrightarrow{r} d$ and $b \xrightarrow{r'} d$. In other words, understanding the relation between man and $king$ helps us to fill up the unknown relation between $woman$ and $queen$.

Analogical structures are not limited to parallelograms, of course, though parallelograms often serve as the building blocks for more complex analogical structures. As an example, in Figure 1.8 of §1.2.1 we show a compound analogical structure in the form of a triangular prism, for mirroring the correspondent entities/relations between the atom system and the solar system. Formally define the desirable analogical structures in a computationally tractable objective for optimization is the key for solving our problem, which we will introduce next.

Commutative Constraint for Linear Maps

Although it is tempting to explore all potentially interesting parallelograms in the modeling of analogical structure, it is computationally intractable to examine the entire powerset of entities as the candidate space of analogical structures. A more reasonable strategy is to identify some desirable properties of the analogical structures we want to model, and use those properties as constraints for reducing the candidate space.

An desirable property of the linear maps we want is that all the directed paths with the same starting node and end node form the *compositional equivalence*. Denoting by "$\circ$" the composition operator between two relations, the parallelogram in Figure 1.9 contains two equivalent compositions as:

$$r \circ r' = r' \circ r \tag{1.16}$$

which means that a is connected to d via either path. We call this the *commutativity* property of the linear maps, which is a necessary condition for forming commutative parallelograms and therefore the corresponding analogical structures. Yet another example is given by Figure 1.8, where sun can traverse to $charge$ along multiple alternative paths of length three, implying the commutativity of relations $surrounded\_by$, $made\_of$, $scale\_down$.

[3]Notice that this is different from parallelograms in the geometric sense because each edge here is a linear map instead of the difference between two nodes in the vector space.

The composition of two relations (linear maps) is naturally implemented via matrix multiplication [41, 102], hence equation (1.16) indicates

$$W_{r\circ r'} = W_r W_{r'} = W_{r'} W_r \tag{1.17}$$

One may further require the commutative constraint (1.17) to be satisfied for any pair of relations in $\mathcal{R}$ because they may be simultaneously present in the same commutative parallelogram for certain subsets of entities. In this case, we say the relations in $\mathcal{R}$ form a commuting family.

It is worth mentioning that $\mathcal{N}_m(\mathbb{R})$ is not closed under matrix multiplication. As the result, the composition rule in eq. (1.17) may not always yield a legal new relation—$W_{r\circ r'}$ may no longer be normal. However, any commuting family in $\mathcal{N}_m(\mathbb{R})$ is indeed closed under multiplication. This explains the necessity of having a commuting family of relations from an alternative perspective.

The Optimization Objective

The generic goal for multi-relational embedding is to find entity and relation representations such that positive triples labeled as $y = +1$ receive higher score than the negative triples labeled as $y = -1$. This can be formulated as

$$\min_{v,W} \ \mathbb{E}_{s,r,o,y\sim\mathcal{D}} \ \ell\left(\phi_{v,W}(s,r,o), y\right) \tag{1.18}$$

where $\phi_{v,W}(s,r,o) = v_s^\top W_r v_o$ is our score function based on the embeddings, ℓ is our loss function, and $\mathcal{D}$ is the data distribution constructed based on the training set $\mathcal{K}$.

To impose analogical structures among the representations, we in addition require the linear maps associated with relations to form a commuting family of normal matrices.

This gives us the objective function for ANALOGY:

$$\min_{v,W} \ \mathbb{E}_{s,r,o,y\sim\mathcal{D}} \ \ell\left(\phi_{v,W}(s,r,o), y\right) \tag{1.19}$$

$$\text{s.t.} \ \ W_r W_r^\top = W_r^\top W_r \quad \forall r \in \mathcal{R} \tag{1.20}$$

$$W_r W_{r'} = W_{r'} W_r \quad \forall r, r' \in \mathcal{R} \tag{1.21}$$

where constraints (1.20) and (1.21) are corresponding to the normality and commutativity requirements, respectively. Such a constrained optimization may appear to be computationally expensive at the first glance. In §1.2.3, however, we will recast it as a simple lightweight problem for which each SGD update can be carried out efficiently in $O(m)$ time.

1.2.3 Optimization

The constrained optimization (1.19) is computationally challenging due to the large number of model parameters in tensor W, the matrix normality constraints, and the quadratic number of pairwise commutative constraints in (1.21).

Interestingly, by exploiting the special properties of commuting normal matrices, we will show in Corollary 1.2.2.1 that ANALOGY can be alternatively solved via an another formulation of substantially lower complexity. Our findings are based on the following lemma and theorem:

Lemma 1.2.1. *[98] For any real normal matrix A, there exists a real orthogonal matrix Q and a block-diagonal matrix B such that $A = QBQ^\top$, where each diagonal block of B is either*

1. *A real valued scalar.*
2. *A 2-dimensional real matrix in the form of $\begin{bmatrix} x & -y \\ y & x \end{bmatrix}$, where both x, y are real scalars.*

The lemma suggests any real normal matrix can be block-diagonalized into an almost-diagonal canonical form. In the following, we use $\mathcal{B}_m^n$ to denote all $m \times m$ almost-diagonal matrices with $n < m$ real scalars on the diagonal.

Theorem 1.2.2. *(Adapted from [40]) If a set of real normal matrices $A_1, A_2, ...$ form a commuting family, namely*

$$A_i A_j = A_j A_i \; \forall i, j \tag{1.22}$$

then they can be block-diagonalized by the same real orthogonal basis Q.

The theorem implies that the set of dense relational matrices $\{W_r\}_{r\in\mathcal{R}}$, if mutually commutative, can always be *simultaneously block-diagonalized* into another set of sparse almost-diagonal matrices $\{B_r\}_{r\in\mathcal{R}}$.

Corollary 1.2.2.1. *For any given solution (v^*, W^*) of optimization (1.19), there always exists an alternative set of embeddings (u^*, B^*) such that $\phi_{v^*,W^*}(s, r, o) \equiv \phi_{u^*,B^*}(s, r, o)$, $\forall(s, r, o)$, and (u^*, B^*) is given by the solution of:*

$$\min_{u,B} \; \mathbb{E}_{s,r,o,y\sim\mathcal{D}} \; \ell\left(\phi_{u,B}(s, r, o), y\right) \tag{1.23}$$

$$B_r \in \mathcal{B}_m^n \quad \forall r \in \mathcal{R} \tag{1.24}$$

The corollary offers a equivalent but highly efficient formulation for ANALOGY.

proof sketch. With commutative constraints, there must exist some orthogonal matrix Q such that $W_r = QB_rQ^\top$, $B_r \in \mathcal{B}_m^n$, $\forall r \in \mathcal{R}$. We can plug-in these expressions into optimization (1.19) and let $u = vQ$, obtaining

$$\phi_{v,W}(s, r, o) = v_s^\top W_r v_o \tag{1.25}$$

$$= v_s^\top Q B_r Q^\top v_o \tag{1.26}$$

$$= u_s^\top B_r u_o = \phi_{u,B}(s, r, o) \tag{1.27}$$

In addition, it is not hard to verify that constraints (1.20) and (1.21) are automatically satisfied by exploiting the facts that Q is orthogonal and $\mathcal{B}_m^n$ is a commutative normal family. □

Constraints (1.24) in the alternative optimization problem can be handled by simply binding together the coefficients within each of those 2×2 blocks in B_r. Note each B_r consists of only m free parameters, allowing efficient evaluation of the gradient w.r.t. any given triple in $O(m)$.

1.2.4 On Unifying Representative Methods

Here we provide a unified view of several embedding models [76, 94, 102], by showing that they are restricted versions under our framework, hence are implicitly imposing analogical properties. This explains their strong empirical performance as compared to other baselines (§1.2.5).

DistMult

DistMult [102] embeds both entities and relations as vectors, and defines the score function as

$$\phi(s, r, o) = \langle v_s, v_r, v_o \rangle \tag{1.28}$$

$$\text{where } v_s, v_r, v_o \in \mathbb{R}^m, \forall s, r, o \tag{1.29}$$

where $\langle \cdot, \cdot, \cdot \rangle$ denotes the generalized inner product.

Proposition 1.2.1. *DistMult embeddings can be fully recovered by ANALOGY embeddings when* $n = m$.

Proof. This is trivial to verify as the score function (1.29) can be rewritten as $\phi(s, r, o) = v_s^\top B_r v_o$ where B_r is a diagonal matrix given by $B_r = \text{diag}(v_r)$. □

Entity analogies are encouraged in DistMult as the diagonal matrices $\text{diag}(v_r)$'s are both normal and mutually commutative. However, DistMult is restricted to model symmetric relations only, since $\phi(s, r, o) \equiv \phi(o, r, s)$.

Complex Embeddings (ComplEx)

ComplEx [94] extends the embeddings to the complex domain $\mathbb{C}$, which defines

$$\phi(s, r, o) = \Re\left(\langle v_s, v_r, \overline{v_o} \rangle\right) \tag{1.30}$$

$$\text{where } v_s, v_r, v_o \in \mathbb{C}^m, \forall s, r, o \tag{1.31}$$

where $\overline{x}$ denotes the complex conjugate of x.

Proposition 1.2.2. *ComplEx embeddings of embedding size* m *can be fully recovered by ANALOGY embeddings of embedding size* $2m$ *when* $n = 0$.

Proof. Let $\Re(x)$ and $\Im(x)$ be the real and imaginary parts of any complex vector x. We recast ϕ in (1.30) as

$$\phi(r, s, o) = + \left\langle \Re(v_r), \Re(v_s), \Re(v_o) \right\rangle \tag{1.32}$$

$$+ \left\langle \Re(v_r), \Im(v_s), \Im(v_o) \right\rangle \tag{1.33}$$

$$+ \left\langle \Im(v_r), \Re(v_s), \Im(v_o) \right\rangle \tag{1.34}$$

$$- \left\langle \Im(v_r), \Im(v_s), \Re(v_o) \right\rangle = {v'_s}^\top B_r v'_o \tag{1.35}$$

The last equality is obtained via a change of variables: For any complex entity embedding $v \in \mathbb{C}^m$, we define a new real embedding $v' \in \mathbb{R}^{2m}$ such that

$$\begin{cases} (v')_{2k} & = \Re(v)_k \\ (v')_{2k-1} & = \Im(v)_k \end{cases} \quad \forall k = 1, 2, \dots m \tag{1.36}$$

The corresponding B_r is a block-diagonal matrix in $\mathcal{B}^0_{2m}$ with its k-th block given by

$$\begin{bmatrix} \Re(v_r)_k & -\Im(v_r)_k \\ \Im(v_r)_k & \Re(v_r)_k \end{bmatrix} \tag{1.37}$$

□

Holographic Embeddings (HolE)

HolE [76] defines the score function as

$$\phi(s, r, o) = \langle v_r, v_s * v_o \rangle \tag{1.38}$$

$$\text{where } \; v_s, v_r, v_o \in \mathbb{R}^m, \forall s, r, o \tag{1.39}$$

where the association of s and o is implemented via circular correlation denoted by $*$. This formulation is motivated by the holographic reduced representation [79].

To relate HolE with ANALOGY, we rewrite (1.39) in a bilinear form with a circulant matrix $C(v_r)$ in the middle

$$\phi(r, s, o) = v_s^\top C(v_r) v_o \tag{1.40}$$

where entries of a circulant matrix are defined as

$$C(x) = \begin{bmatrix} x_1 & x_m & \cdots & x_3 & x_2 \\ x_2 & x_1 & x_m & & x_3 \\ \vdots & x_2 & x_1 & \ddots & \vdots \\ x_{m-1} & & \ddots & \ddots & x_m \\ x_m & x_{m-1} & \cdots & x_2 & x_1 \end{bmatrix} \tag{1.41}$$

It is not hard to verify that circulant matrices are normal and commute [39], hence entity analogies are encouraged in HolE, for which optimization (1.19) reduces to an unconstrained problem as equalities (1.20) and (1.21) are automatically satisfied when all W_r's are circulant.

We further reveal the equivalence between HolE and ComplEx with minor relaxation:

Proposition 1.2.3. *HolE embeddings can be obtained via the following score function*

$$\phi(s, r, o) = \Re\left(\langle v_s, v_r, \overline{v_o} \rangle\right) \tag{1.42}$$

$$\textit{where } \; v_s, v_r, v_o \in \mathfrak{F}(\mathbb{R}^m), \forall s, r, o \tag{1.43}$$

where $\mathfrak{F}(\mathbb{R}^m)$ denotes the image of $\mathbb{R}^m$ in $\mathbb{C}^m$ through the Discrete Fourier Transform (DFT). In particular, the above reduces to ComplEx by relaxing $\mathfrak{F}(\mathbb{R}^m)$ to $\mathbb{C}^m$.

Proof. Let $\mathfrak{F}$ be the DFT operator defined by $\mathfrak{F}(x) = Fx$ where $F \in \mathbb{C}^{m \times m}$ is called the Fourier basis of DFT. A well-known property for circulant matrices is that any $C(x)$ can always be diagonalized by F, and its eigenvalues are given by Fx [39].

Hence the score function can be further recast as

$$\phi(r, s, o) = v_s^\top F^{-1} \operatorname{diag}(F v_r) F v_o \tag{1.44}$$

$$= \frac{1}{m} \overline{(F v_s)}^\top \operatorname{diag}(F v_r)(F v_o) \tag{1.45}$$

$$= \frac{1}{m} \langle \overline{\mathfrak{F}(v_s)}, \mathfrak{F}(v_r), \mathfrak{F}(v_o) \rangle \tag{1.46}$$

$$= \Re\left[\frac{1}{m} \langle \overline{\mathfrak{F}(v_s)}, \mathfrak{F}(v_r), \mathfrak{F}(v_o) \rangle \right] \tag{1.47}$$

Let $v'_s = \overline{\mathfrak{F}(v_s)}, v'_o = \overline{\mathfrak{F}(v_o)}$ and $v'_r = \frac{1}{m}\mathfrak{F}(v_r)$, we obtain exactly the same score function as used in ComplEx

$$\phi(s, r, o) = \Re\left(\langle v'_s, v'_r, \overline{v'_o}\rangle\right) \tag{1.48}$$

(1.48) is equivalent to (1.30) apart from an additional constraint that v'_s, v'_r, v'_o are the image of $\mathbb{R}$ in the Fourier domain. □

1.2.5 Experiments

Datasets

We evaluate ANALOGY and the baselines over two benchmark datasets for multi-relational embedding released by previous work [15], namely a subset of Freebase (FB15K) for generic facts and WordNet (WN18) for lexical relationships between words.

The dataset statistics are summarized in Table 1.2.

Dataset	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{R}\rvert$	#train	#valid	#test
FB15K	14,951	1,345	483,142	50,000	59,071
WN18	40,943	18	141,442	5,000	5,000

Table 1.2: Dataset statistics for FB15K and WN18.

Baselines

We compare the performance of ANALOGY against a variety types of multi-relational embedding models developed in recent years. Those models can be categorized as:

- Translation-based models where relations are modeled as translation operators in the embedding space, including TransE [15] and its variants TransH [96], TransR [63], TransD [48], STransE [73] and RTransE [36].
- Multi-relational latent factor models including LFM [47] and RESCAL [74] based collective matrix factorization.
- Models involving neural network components such as neural tensor networks [88] and PTransE-RNN [63], where RNN stands for recurrent neural networks.
- Pathwise models including three different variants of PTransE [62] which extend TransE by explicitly taking into account indirect connections (relational paths) between entities.
- Models subsumed under our proposed framework (§1.2.4), including DistMult [102] based simple multiplicative interactions, ComplEx [94] using complex coefficients and HolE [76] based on holographic representations. Those models are implicitly leveraging analogical structures per our previous analysis.
- Models enhanced by external information. We use Node+LinkFeat (NLF) [93] as a representative example, which leverages textual mentions derived from the ClueWeb corpus.

Evaluation Metrics

Following the literature of multi-relational embedding, we use the conventional metrics of Hits@k and Mean Reciprocal Rank (MRR) which evaluate each system-produced ranked list for each test instance and average the scores over all ranked lists for the entire test set of instances.

The two metrics would be flawed for the *negative instances* created in the test phase as a ranked list may contain some positive instances in the training and validation sets [15]. A recommended remedy, which we followed, is to remove all training- and validation-set triples from all ranked lists during testing. We use "filt." and "raw" to indicate the evaluation metrics with or without filtering, respectively.

In the first set of our experiments, we used on Hits@k with k=10, which has been reported for most methods in the literature. We also provide additional results of ANALOGY and a subset of representative baseline methods using MRR, Hits@1 and Hits@3, to enable the comparison with the methods whose published results are in those metrics.

Implementation Details

Loss Function: We use the logistic loss for ANALOGY throughout all experiments, namely $\ell(\phi(s,r,o),y) = -\log\sigma(y\phi(s,r,o))$, where σ is the sigmoid activation function. We empirically found this simple loss function to perform reasonably well as compared to more sophisticated ranking loss functions.

Asynchronous AdaGrad: Our C++ implementation[4] runs over a CPU, as ANALOGY only requires lightweight linear algebra routines. We use asynchronous stochastic gradient descent (SGD) for optimization, where the gradients w.r.t. different mini-batches are simultaneously evaluated in multiple threads, and the gradient updates for the shared model parameters are carried out without synchronization. While being efficient, asynchronous SGD causes little performance drop when parameters associated with different mini-batches are mutually disjoint with a high probability [80]. The learning rate is adapted based on historical gradients as in AdaGrad [28].

Creation of Negative Samples: Since only valid triples (positive instances) are explicitly given in the training set, invalid triples (negative instances) need to be artificially created. Specifically, for every positive example (s,r,o), we generate three negative instances (s',r,o), (s,r',o), (s,r,o') by corrupting s, r, o with random entities/relations $s' \in \mathcal{E}$, $r' \in \mathcal{R}$, $o' \in \mathcal{E}$. The union of all positive and negative instances defines our data distribution $\mathcal{D}$ for SGD updates.

Model Selection: We conducted a grid search to find the hyperparameters of ANALOGY which maximize the filtered MRR on the validation set, by enumerating all combinations of the embedding size $m \in \{100, 150, 200\}$, ℓ_2 weight decay factor $\lambda \in \{10^{-1}, 10^{-2}, 10^{-3}\}$ of model coefficients v and W, and the ratio of negative over positive samples $\alpha \in \{3, 6\}$. The resulting hyperparameters for the WN18 dataset are $m = 200, \lambda = 10^{-2}, \alpha = 3$, and those for the FB15K dataset are $m = 200, \lambda = 10^{-3}, \alpha = 6$. The number of scalars on the diagonal of each B_r is always set to be $\frac{m}{2}$. We set the initial learning rate to be 0.1 for both datasets and adjust it using AdaGrad during optimization. All models are trained for 500 epochs.

[4]Code available at https://github.com/quark0/ANALOGY.

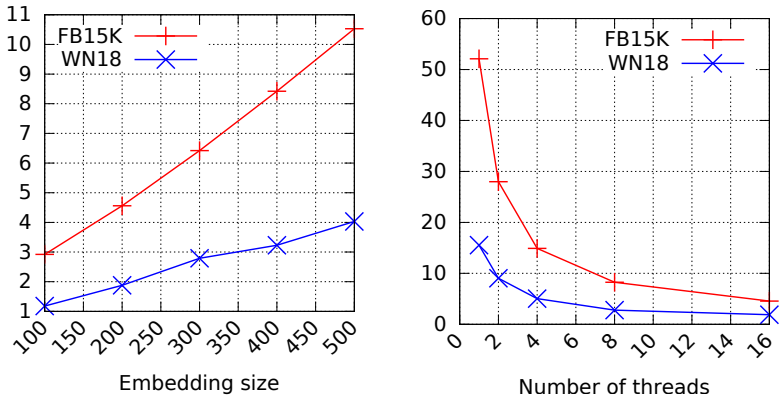

Figure 1.10: CPU run time per epoch (secs) of ANALOGY. The left figure shows the run time over increasing embedding sizes with 16 CPU threads; The right figure shows the run time over increasing number of CPU threads with embedding size 200.

Results

Table 1.3 compares the Hits@10 score of ANALOGY with that of 23 competing methods using the published scores for these methods in the literature on the WN18 and FB15K datasets. For the methods not having both scores, the missing slots are indicated by "–". The best score on each dataset is marked in the bold face; if the differences among the top second or third scores are not statistically significant from the top one, then these scores are also bold faced. We used one-sample proportion test [103] at the 5% p-value level for testing the statistical significances[5].

Table 1.4 compares the methods (including ours) whose results in additional metrics are available. The usage of the bold faces is the same as those in Table 1.3.

In both tables, ANALOGY performs either the best or the 2nd best which is in the equivalent class with the best score in each case according statistical significance test. Specifically, on the harder FB15K dataset in Table 1.3, which has a very large number of relations, our model outperforms all baseline methods. These results provide a good evidence for the effective modeling of analogical structures in our approach. We are pleased to see in Table 1.4 that ANALOGY outperforms DistMult, ComplEx and HolE in all the metrics, as the latter three can be viewed as more constrained versions of our method (as discussed in (§1.2.4)). Furthermore, our assertion on HolE for being a special case of ComplEx (§1.2.4) is justified in the same table by the fact that the performance of HolE is dominated by ComplEx.

In Figure 1.10 we show the empirical scalability of ANALOGY, which not only completes one epoch in a few seconds on both datasets, but also scales linearly in the size of the embedding problem. As compared to single-threaded AdaGrad, our asynchronous AdaGrad over 16 CPU threads offers 11.4x and 8.3x speedup on FB15K and WN18, respectively, on a single commercial desktop.

[5]Note proportion tests only apply to performance scores as proportions, including Hits@k, but are not applicable to non-proportional scores such as MRR. Hence we only conducted the proportion tests on the Hits@k scores.

1.3 Concluding Remarks

In the first part of this chapter we presented a novel convex optimization framework for transductive CGRL and a scalable algorithmic solution with guaranteed global optimum and a time complexity that does not depend on the sizes of input graphs. Our experiments on multi-graph data sets provide strong evidence for the superior power of the proposed approach in modeling cross-graph inference and large-scale optimization.

In the second part, we presented a novel framework for explicitly modeling analogical structures in multi-relational embedding, along with a differentiable objective function and a linear-time inference algorithm for large-scale embedding of knowledge graphs. The proposed approach obtains the state-of-the-art results on two popular benchmark datasets, outperforming a large number of strong baselines in most cases. Although we only focused on the multi-relational inference for knowledge-base embedding, we believe that analogical structures exist in many other machine learning problems beyond the scope of this section. We hope this work shed light on a broad range of important problems where scalable inference for analogical analysis would make an impact, such as machine translation and image captioning (both problems require modeling cross-domain analogies). We leave these interesting topics as our future work.

Table 1.3: Hits@10 (filt.) of all models on WN18 and FB15K categories into three groups: (i) 19 baselines without modeling analogies; (ii) 3 baselines and our proposed ANALOGY which implicitly or explicitly enforce analogical properties over the induced embeddings (see §1.2.4); (iii) One baseline relying on large external data resources in addition to the provided training set.

Models	WN18	FB15K
Unstructured [15]	38.2	6.3
RESCAL [74]	52.8	44.1
NTN [88]	66.1	41.4
SME [14]	74.1	41.3
SE [13]	80.5	39.8
LFM [47]	81.6	33.1
TransH [96]	86.7	64.4
TransE [15]	89.2	47.1
TransR [63]	92.0	68.7
TKRL [99]	–	73.4
RTransE [36]	–	76.2
TransD [48]	92.2	77.3
CTransR [63]	92.3	70.2
KG2E [43]	93.2	74.0
STransE [73]	93.4	79.7
DistMult [102]	93.6	82.4
TransSparse [49]	93.9	78.3
PTransE-MUL [62]	–	77.7
PTransE-RNN [62]	–	82.2
PTransE-ADD [62]	–	84.6
NLF (with external corpus) [93]	*94.3*	*87.0*
ComplEx [94]	**94.7**	84.0
HolE [76]	**94.9**	73.9
Our ANALOGY	**94.7**	**85.4**

Table 1.4: MRR and Hits@{1,3} of a subset of representative models on WN18 and FB15K. The performance scores of TransE and REACAL are cf. the results published in [94] and [76], respectively.

	WN18				FB15			
Models	MRR (filt.)	MRR (raw)	Hits@1 (filt.)	Hits@3 (filt.)	MRR (filt.)	MRR (raw)	Hits@1 (filt.)	Hits@3 (filt.)
RESCAL [74]	89.0	60.3	84.2	90.4	35.4	18.9	23.5	40.9
TransE [15]	45.4	33.5	8.9	82.3	38.0	22.1	23.1	47.2
DistMult [102]	82.2	53.2	72.8	91.4	65.4	24.2	54.6	73.3
HolE [76]	93.8	61.6	**93.0**	**94.5**	52.4	23.2	40.2	61.3
ComplEx [94]	94.1	58.7	**93.6**	**94.5**	69.2	24.2	59.9	75.9
Our ANALOGY	**94.2**	**65.7**	**93.9**	**94.4**	**72.5**	**25.3**	**64.6**	**78.5**

Chapter 2

Learning with Graph Induction

2.1 Nonparametric Learning of Graph Diffusion (Completed work, AISTATS'16)

This section proposes a novel nonparametric framework for semi-supervised learning and for optimizing the Laplacian spectrum of the data manifold simultaneously. Our formulation leads to a convex optimization problem that can be efficiently solved via the bundle method, and can be interpreted as to asymptotically minimize the generalization error bound of semi-supervised learning with respect to the graph spectrum. Experiments over benchmark datasets in various domains show advantageous performance of the proposed method over strong baselines.

2.1.1 Motivation

Graph representation of data is ubiquitous in machine learning. In many scenarios, we are given a partially labeled graph with only a small number of labeled vertices, and the task is to predict the missing labels of the large number of unlabeled vertices.

With limited supervision, it is often crucial to leverage the intrinsic manifold structure of both the labeled and unlabeled vertices during the training phase. A variety of graph-based semi-supervised learning (SSL) algorithms have been proposed under this motivation, including label propagation [106], Gaussian random fields [107] and Laplacian Support Vector Machines [67]. Many of those approaches rely on the assumption that strongly connected vertices are likely to share the same labels, and fall under the manifold regularization framework [8] where the graph Laplacian [22] plays a key role.

Given a graph, the graph Laplacian characterizes how the label of each vertex diffuses (propagates) from itself to its *direct* neighbors. While the graph Laplacian in its original form may not be sufficiently expressive for modeling complex graph transduction patterns, it has been shown that a rich family of important graph transduction patterns under various assumptions, including multi-step random walk, heat diffusion [55] and von-Neumann diffusion [46], can be incorporated into SSL by transforming the spectrum[1] of the graph Laplacian with nonnegative

[1] In this section, we refer to the spectrum of a matrix as the multiset of its eigenvalues.

nondecreasing functions [52, 87, 108]. The collection of those functions are referred to as the *Spectral Transformation* (ST) family.

Despite of the expressiveness of the ST family, how to find the optimal ST for any problem in hand is an open challenge. While manual specification [52, 87] is clearly suboptimal, various approaches have been proposed to automatically find the optimal ST. Among the existing works, parametric approaches assume the optimal ST belongs some pre-specified function family (e.g. the polynomial or exponential), and then find the function hyperparameter via grid search or curve-fitting [59]. However, the fundamental question about how to choose the function family is left unanswered, and it is not clear whether commonly used parametric function families are rich enough to subsume the true optimal ST. On the other hand, a more flexible nonparametric framework based on kernel-target alignment has been studied in [108], where the optimization of ST is efficiently solved via quadratically constrained quadratic programming (QCQP). However, the target matrix itself may be unreliable as it is constructed based a very small number of observed labels, and it is not conclusive whether a better alignment score always leads to a better prediction performance.

Note all the above approaches adopt *two-step* procedures, where the optimal ST is empirically estimated in some preprocessing step before SSL is carried out (with the ST obtained in the previous step). We argue that the separation of ST optimization from SSL may result in suboptimal performance, as combining the two steps together will allow the learned ST to better adapt to the problem structure.

This section addresses the aforementioned challenge by proposing a principled optimization framework which *simultaneously* conducts SSL and finds the optimal ST for the graph Laplacian used in SSL. Starting with the natural formulation of the joint optimization, we show how it can be reformulated as an equivalent *convex* optimization problem via Lagrangian duality, and then derive an efficient algorithm using the bundle method. We refer to our new approach as *Adaptive Spectral Transform* (AST), meaning that the ST is automatically adapted to the problem in hand and its target domain.

Besides improved performance over benchmark datasets across various domains, insights are provided regarding the advantageous performance of AST by revisiting an existing theorem on SSL from a new angle. Specifically, we show that AST actually aims to asymptotically minimize the generalization error bound of SSL.

2.1.2 The Proposed Method

SSL with the Graph Laplacian

Given a graph G of m vertices, where each vertex denotes an instance and each edge encodes the affinity between a pair of instances. Suppose only a very small set $\mathcal{T}$ of l vertices has been labeled where $l \ll m$, our task is to predict the missing labels of the remaining $m - l$ vertices based on both the l labeled vertices and the intrinsic manifold structure of G.

Denote by y_i the true label and by $\mathrm{f}_i \in \mathbb{R}$ the system-estimated score for vertex i, resp. In order to leverage the labels, we hope f_i and y_i to be as close as possible for all $i \in \mathcal{T}$. Meanwhile, to leverage the large amount of unlabeled vertices, we want the scores for all (both labeled and unlabeled) vertices to be smooth w.r.t. the graph structure of G. The two desired properties entail

the following optimization problem:

$$\min_{\mathrm{f}\in\mathbb{R}^m} \quad \frac{1}{l}\sum_{i\in\mathcal{T}} \ell(\mathrm{f}_i, y_i) + \gamma \mathrm{f}^\top \mathcal{L}\mathrm{f} \tag{2.1}$$

where the first term is the empirical loss of the system-predicted scores $\mathrm{f} \in \mathbb{R}^m$, $\mathcal{L}$ in the second term is the normalized graph Laplacian matrix associated with G characterizing G's manifold structure. Specifically, denote by A the adjacency matrix of G, by D a diagonal matrix of degrees with $d_{ii} = \sum_j a_{ij}$ and by $L = D - A$ the graph Laplacian. The normalized graph Laplacian is defined as $\mathcal{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$, with its eigensystem denoted by $\{(\lambda_i, \phi_i)\}_{i=1}^m$. For convenience, we assume the eigenvalues of $\mathcal{L}$ are in the increasing order: $\lambda_1 \leq \lambda_2 \ldots \leq \lambda_m$. It is well known that the smallest eigenvalue λ_1 is always zero, and that ϕ_i's with small indices tend to be "smoother" over the data manifold than those with large indices [22].

In (2.1), the label information is encoded in the empirical loss $\ell(\mathrm{f}_i, y_i)$. E.g., one could specify $\ell(\mathrm{f}_i, y_i)$ to be $(\mathrm{f}_i - y_i)^2$. The manifold assumption is encoded in the second term (a.k.a. the manifold regularizer) involving the graph Laplacian, satisfying

$$\mathrm{f}^\top \mathcal{L}\mathrm{f} \equiv \frac{1}{2}\sum_{i\sim j} a_{ij}\left(\frac{\mathrm{f}_i}{\sqrt{d_{ii}}} - \frac{\mathrm{f}_j}{\sqrt{d_{jj}}}\right)^2 \tag{2.2}$$

$$\equiv \sum_{i=1}^m \lambda_i \langle \phi_i, \mathrm{f}\rangle^2 \tag{2.3}$$

Eq. (2.2) suggests that the regularizer essentially encourages scores f_i, f_j (normalized by the squared root of degrees) to be close when vertices i, j are strongly connected in G, namely when a_{ij} is large. An alternative perspective, as implied by (2.3), is to think of the regularizer as penalizing the projection of f onto different bases (the $\phi_i' s$) with different weights (the $\lambda_i' s$), where the smooth components in f are going to receive lighter penalty than the nonsmooth ones.

Transforming the Laplacian Spectrum

Although the graph Laplacian gives a nice characterization about how vertices in G influence their direct neighbors, it is not sufficiently expressive for modeling complex label propagation patterns, such as multi-step influence from a given vertex to its indirect neighbors and the decay of such influence. As a simple remedy to incorporate a richer family of label propagation patterns over the manifold, various methods have been proposed based on transforming the spectrum of $\mathcal{L}$ using some nonnegative nondecreasing function, known as the spectral transformation [52, 87, 108].

As an example, by taking the exponential of the Laplacian spectrum, one gets $\sum_{i=1}^m e^{\beta\lambda_i}\phi_i\phi_i^\top = e^{\beta\mathcal{L}}$ where β is a nonnegative scalar. The transformed Laplacian has a neat physical interpretation in terms of heat diffusion process, and is closely related to infinite random walk with decay over the manifold [55]. From (2.3)'s perspective, the replacement of λ_i with e^{λ_i} can be viewed as a way to exaggerate the difference in weighing the bases. That is, the nonsmooth components in f are going to receive a larger relative penalty during the optimization after the exponential transformation.

Formally, we define the *Spectral Transformation* (ST) over $\mathcal{L}$ as $\sigma(\mathcal{L}) := \sum_{i=1}^{m} \sigma(\lambda_i)\phi_i\phi_i^\top$, where $\sigma : \mathbb{R}_+ \mapsto \mathbb{R}_+$ is a nondecreasing function which transforms each Laplacian eigenvalue to a nonnegative scalar. Besides the aforementioned diffusion kernel where $\sigma(x) = e^{\beta x}$, other commonly used STs include $\sigma(x) = x + \beta$ (Gaussian field), $\sigma(x) = \frac{1}{(\alpha - x)^\beta}$ (multi-step random walk), $\sigma(x) = \left[\cos\left(\frac{\pi}{4}x\right)\right]^{-1}$ (inverse cosine) [59, 87], etc.

The ST-enhanced SSL is formulated as

$$\min_{\mathrm{f}\in\mathbb{R}^m} \quad \frac{1}{l}\sum_{i\in\mathcal{T}} \ell(\mathrm{f}_i, y_i) + \gamma \mathrm{f}^\top \sigma(\mathcal{L})\mathrm{f} \tag{2.4}$$

Adapting the Spectral Transform

The nature of SSL described in (2.4) crucially depends our choice of ST. It is a common practice to manually specify σ [52, 87] or to learn the hyperparameter of σ within a pre-specified function family [59, 84]. Both methods are suboptimal when the true σ^* lies in a broader function space.

In this section, we focus on automatically learning σ^* from data with no prior assumption on its function form. In terms of SSL, we argue it suffices to learn $\{\sigma^*(\lambda_i)\}_{i=1}^m$ instead of the analytical expression of σ^*, as the objective in (2.4) is uniquely determined by these m transformed eigenvalues. Therefore, in the following we switch our focus from the task of making σ adaptive to the equivalent task of making each $\sigma(\lambda_i)$ adaptive.

Define $\theta \in \mathbb{R}^m$ where $\theta_i := \sigma(\lambda_i)^{-1}$. We are going to focus on learning θ as notation-wise it is more convenient to work with the reciprocals. After substituting the ST σ with θ in (2.4), the optimization becomes

$$\min_{\mathrm{f}\in\mathbb{R}^m} \quad \underbrace{\frac{1}{l}\sum_{i\in\mathcal{T}} \ell(\mathrm{f}_i, y_i) + \gamma \sum_{i=1}^{m} \theta_i^{-1} \langle \phi_i, \mathrm{f}\rangle^2}_{Q(\mathrm{f};\theta)} \tag{2.5}$$

When $\theta_i = 0$, we define $\theta_i^{-1} := 0$ as its pseudo-inverse. For brevity, in the following we assume all the θ_i's are strictly positive. The singular case where some θ_i's are exactly zero will be studied specifically in Section 2.1.3.

To determine θ for (2.5), Zhu et al. [108] proposed a two-step procedure based on empirical kernel-target alignment. In the first step, an empirical estimation about θ is obtained by maximizing the alignment score between the kernel matrix implied by θ, i.e. $\sum_i \theta_i\phi_i\phi_i^\top$, and a target kernel matrix induced from a small amount of observed labels. In the second step, the estimated $\hat{\theta}$ is plugged-into the SSL objective (2.5) for learning f.

Different from existing (manual/parametric/two-step) approaches, we argue that it is beneficial to put the task of finding the optimal θ^* and the task of SSL into a unified optimization framework, as the two procedures can mutually reinforce each other, thus making θ^* more adapted to the problem structure.

It may appear straightforward to approach the aforementioned goal by minimizing (2.5) w.r.t. f and w.r.t. θ in an alternating manner. Unfortunately, the resulting optimization is non-convex, and a meaningless solution can be obtained by simply setting all the θ_i^{-1}'s to zero.

Instead, we propose to achieve this goal by solving the following optimization problem (AST)

$$\min_{\theta\in\Theta} \left(\min_{\mathrm{f}\in\mathbb{R}^m} \ \mathcal{C}(\mathrm{f},\theta) \right) + \tau\|\theta\|_1 \tag{2.6}$$

where $\mathcal{C}(\mathrm{f};\theta)$ is the SSL objective defined in (2.5), τ is a positive scalar-valued tuning parameter, and Θ denotes the set of all possible reciprocals of the transformed Laplacian spectrum

$$\begin{aligned}\Theta &= \left\{\theta : \theta_i = \sigma(\lambda_i)^{-1}, \forall i = 1, 2 \dots m, \ \sigma \text{ is a valid ST}\right\} \\ &\equiv \left\{\theta : \theta_1 \geq \theta_2, \dots \geq \theta_m \geq 0\right\}\end{aligned} \tag{2.7}$$

The second equality is derived based on the facts that (i) the λ_i's are in the increasing order (ii) σ can be *any* nonnegative nondecreasing function.

The intuition behind optimization (2.6) is that we want the optimal θ^* (and the associated optimal ST) to simultaneously satisfy the following criteria:

(a) It should tend to minimize the SSL objective (2.5). As in multiple kernel learning [6, 60], this is arguably the most natural and effective way to make θ^* adaptive to the problem structure.

(b) It should have a moderate ℓ_1-norm. Namely the "transformed" data manifold should have a moderate total effective resistance [16]. As we will see later, this additional requirement is crucial as it precludes degenerate solutions. It also makes our bundle method more efficient by sparsifying θ (Section 2.1.3).

2.1.3 Optimization

Let us present our optimization strategies for solving (2.6), starting with the following theorem

Theorem 2.1.1 (Convexity of AST)**.**
(2.6) *is a convex optimization problem over* θ.

After presenting the proof for Theorem 2.1.1 (Section 2.1.3), we propose our method to compute the gradient for (2.6)'s structured objective function in Section 2.1.3, and offer a bundle method for efficient optimization in Section 2.1.3. We will study the singular case where some θ_i's are allowed to be exactly zero in Section 2.1.3, which can be particularly useful in large-scale scenarios. The SSL subroutine for AST is discussed in Section 2.1.3.

Proof of Convexity

We proof Theorem 2.1.1 by first reformulating optimization (2.6)'s objective function into an equivalent minimax-type function via Lagrangian duality, and then showing the convexity of the equivalent optimization problem.

The Lagrangian dual for $\mathcal{C}(\mathrm{f};\theta)$ is

$$\underbrace{-\omega(-u) - \frac{1}{4\gamma}\sum_{i=1}^{m}\theta_i \langle\phi_i, u\rangle^2}_{\bar{C}(u;\theta)} \tag{2.8}$$

where $\omega(\cdot)$ is the conjugate function for $\sum_{i\in\mathcal{T}} \ell(\mathrm{f}_i, y_i)$. It is not hard to verify that the Slater's condition holds for optimization (2.5), i.e. $\min_{\mathrm{f}\in\mathbb{R}^m} \; \mathcal{C}(\mathrm{f};\theta)$, thus strong duality ensures that

$$\min_{\mathrm{f}\in\mathbb{R}^m} \mathcal{C}(\mathrm{f};\theta) = \max_{u\in\mathbb{R}^m} \; \bar{C}(u;\theta) \tag{2.9}$$

and optimization (2.6) for AST can be recast as

$$\min_{\theta\in\Theta} \; \underbrace{\left(\max_{u\in\mathbb{R}^m} \; \bar{C}(u;\theta)\right)}_{g(\theta)} + \tau\|\theta\|_1 \tag{2.10}$$

We claim the resulting equivalent problem (2.10) is convex over θ. To see this, notice that $\bar{C}(u;\theta)$ defined in (2.8) is an affine over θ for each given u, and recall that the pointwise maximum of any set of convex functions (affines) is still convex, the first structured term $g(\theta)$ in optimization (2.10), i.e. $\max_{u\in\mathbb{R}^m} \; \bar{C}(u;\theta)$, is hence convex over θ. The conclusion follows by further noticing the second term $\|\theta\|_1$ in (2.10) is also a convex function, and that Θ in (2.7) is a convex domain.

Computing the Structured Gradient

In this section, we discuss our method to compute the gradient of $g(\theta) := \max_u \bar{C}(u;\theta)$ in (2.10), denoted by $\nabla_\theta g(\theta)$, as a prerequisite for subsequent optimization algorithms. We rely on Danskin's Theorem [26] as $g(\theta)$ is the maximum of infinite number of functions:

Theorem 2.1.2 (Danskin's Theorem)**.** *If function $g(\theta)$ is in the form of $g(\theta) := \max_{u\in\mathcal{U}} \bar{C}(u;\theta)$ where $\mathcal{U}$ is a compact space and $\bar{C}(\cdot;\theta)$ is a differentiable function with $\bar{C}(u;\theta)$ and $\nabla\bar{C}(u;\theta)$ depending continuously on u and θ, then the subgradient of $g(\theta)$, i.e. $\partial_\theta g(\theta)$, is given by $\partial_\theta \bar{C}(\hat{u};\theta)$ where $\hat{u} \in \operatorname{argmax}_{u\in\mathcal{U}} \bar{C}(u;\theta)$.*

For our case $\mathcal{U} := \mathbb{R}^m$ and the subgradient $\partial_\theta g(\theta)$ can be substituted with gradient $\nabla_\theta g(\theta)$ as the function of interest is differentiable. Recall that we have assumed all θ_i's to be positive, $\bar{C}(u;\theta)$ is strictly convex over u and therefore $\hat{u} := \operatorname{argmax}_u \bar{C}(u;\theta)$ is always unique.

Suppose $\hat{u}$ is given, following Theorem 2.1.2 we have

$$\nabla_\theta g(\theta) = \nabla_\theta \bar{C}(\hat{u};\theta) = -\frac{\left(\langle\phi_1,\hat{u}\rangle^2,\ldots,\langle\phi_m,\hat{u}\rangle^2\right)^\top}{4\gamma} \tag{2.11}$$

To compute the R.H.S. of (2.11), we have to get $\hat{u}$ in advance via solving $\max_u \bar{C}(u;\theta)$. In case the conjugate function involved in $\bar{C}(u;\theta)$ is hard to work with, it is more convenient to first obtain the primal solution $\hat{\mathrm{f}}$ by solving the corresponding primal problem $\min_{\mathrm{f}} \; \mathcal{C}(\mathrm{f};\theta)$ described in (2.5), and then recover the dual solution $\hat{u}$ from $\hat{\mathrm{f}}$ via the K.K.T. condition.

According to the stationarity condition, $\hat{u}$ and $\hat{\mathrm{f}}$ must satisfy

$$\hat{u} = 2\gamma\left(\sum_{i=1}^m \theta_i^{-1}\phi_i\phi_i^\top\right)\hat{\mathrm{f}} \tag{2.12}$$

This suggests an alternative to (2.11), i.e. to compute the gradient of $g(\theta)$ directly based on the primal variable via

$$\nabla_\theta g(\theta) = -\gamma\left(\frac{\langle\phi_1,\hat{\mathrm{f}}\rangle^2}{\theta_1^2},\ldots,\frac{\langle\phi_m,\hat{\mathrm{f}}\rangle^2}{\theta_m^2}\right)^\top \tag{2.13}$$

where $\hat{\mathrm{f}} := \operatorname{argmin}_{\mathrm{f}} \mathcal{C}(\mathrm{f}, \theta)$ is obtained by applying any SSL algorithm[2] to (2.5).

Bundle Method for AST

After obtaining $\nabla_\theta g(\theta)$ according to section 2.1.3, it is straightforward to minimize the AST objective in (2.10): $g(\theta) + \gamma\|\theta\|_1$ via the subgradient method or proximal gradient method. However, both algorithms have slow convergence rate, and it can be tricky to choose a suitable step size to ensure efficient convergence.

We propose to use the bundle method for (2.10) (equivalently, (2.6)), which has been found particularly efficient in solving problems involving structured loss functions [53, 90]. Our method is a variant of bundle method for regularized risk minimization (BMRM) [92], and subsumes the semi-infinite linear programming (SILP) for large-scale multiple kernel learning [89].

The key idea is to replace the "tough" part in (2.10), i.e. $g(\theta)$, with an "easy" piecewise linear function $\tilde{g}(\theta)$ that lowerbounds the original $g(\theta)$, as shown in Figure 2.2. After the replacement, optimization (2.10) becomes

$$\min_{\theta \in \Theta} \ \tilde{g}(\theta) + \tau\|\theta\|_1 \tag{2.14}$$

We then alternate between solving the surrogate problem (2.14) and refining the lowerbound $\tilde{g}(\theta)$ until convergence. Note (2.14) is a Linear Programming (LP), as its objective function is piecewise linear and its feasible set Θ defined in (2.7) is a polyhedron.

To obtain a piecewise lowerbound $\tilde{g}(\theta)$ for $g(\theta)$, recall any convex function can be lowerbounded by its tangents. Hence it suffices to let $\tilde{g}(\theta)$ be the supremum of a set of tangents associated with historical iterations. Specifically, we define $\tilde{g}(\theta)$ at the t-th iteration as

$$\tilde{g}^{(t)}(\theta) := \max_{0 \le i \le t-1} \ g(\theta^{(i)}) + \left\langle \nabla g\big(\theta^{(i)}\big), \theta - \theta^{(i)} \right\rangle \tag{2.15}$$

where superscript "$^{(i)}$" indexes the quantity associated with the i-th iteration. It is not hard to verify that $g^{(t)}(\theta) \le g(\theta)$ always holds, and that $g^{(t)}(\theta)$ tends to better approximate $g(\theta)$ as t increases.

Details of the bundle method for AST is presented in Algorithm 1.

Scale Up by Exploiting Singularity

Now let us focus on the singular cases where some θ_i's (and their pseudo-inverse θ_i^{-1}'s) are exactly zero. This may happen in two scenarios:

(a) During the bundle method, some θ_i's are shrunk to zero after solving the LP (2.14) due to the presence of the ℓ_1-regularization over θ.

(b) Small-valued θ_i's associated with those nonsmooth ϕ_i's are truncated to be zero for the sake of scalability. This strategy will substantially reduce the parameter size of SSL, and has been successfully applied to large-scale problems [33].

[2]Many off-the-shelf SSL solvers can be easily modified for solving the primal problem (2.5).

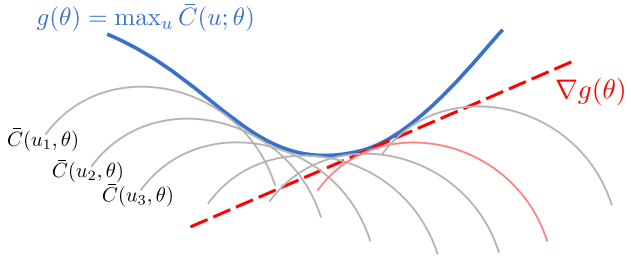

Figure 2.1: A visual interpretation of Danskin's theorem. Computing the derivative of $\nabla g(\theta)$ is equivalent to solving for $\hat{u}$ and computing the derivative of $\bar{C}(\hat{u};\theta)$.

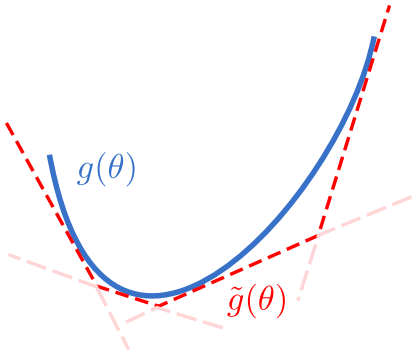

Figure 2.2: We maintain a piecewise lowerbound $\tilde{g}(\theta)$, which keeps being refined during optimization to better approximate $g(\theta)$.

In the following, we will assume $\theta_i > 0$ for $1 \le i \le k$ and $\theta_i = 0$ for $k < i \le m$, where $k \ll m$. To handle the singular case, we modify $\bar{C}(\mathrm{f};\theta)$ in (2.5) as

$$\frac{1}{l}\sum_{i\in\mathcal{T}} \ell(\mathrm{f}_i, y_i) + \gamma \sum_{1\le i\le k} \theta_i^{-1} \langle \phi_i, \mathrm{f}\rangle^2 + \sum_{k<i\le m} \mathbf{1}_{\{\langle \phi_i, \mathrm{f}\rangle = 0\}} \tag{2.16}$$

where $\mathbf{1}_{\{\cdot\}}$ equals zero if the inside-bracket condition is satisfied and equals $+\infty$ otherwise. The third term in (2.16) is crucial in that otherwise the projection of f onto ϕ_i for any $k < i \le m$ will be left unregularized and the resulting model can easily over-fit.

The solution f^* for minimizing (2.16) must lie in the span of $\{\phi_i\}_{i=1}^k$ as otherwise the indicator function will go to infinity. Let $\mathrm{f} := \sum_{1\le j\le k} \alpha_j \phi_j$. (2.16) can be reduced to consist of only k ($k \ll m$) parameters

$$\frac{1}{l}\sum_{i\in\mathcal{T}} \ell\left(e_i^\top \sum_{1\le j\le k} \alpha_j \phi_j, y_i\right) + \gamma \sum_{1\le i\le k} \theta_i^{-1}\alpha_i^2 \tag{2.17}$$

where e_i stands for the i-th unit vector in $\mathbb{R}^m$.

Applying similar analysis[3] in the previous subsections to the modified $\bar{C}(\mathrm{f};\theta)$ in (2.16), for singular cases the gradient of $g(\theta)$ during bundle method is given by

$$\nabla_\theta g(\theta) = -\gamma\left(\frac{\langle \phi_1, \hat{\mathrm{f}}\rangle^2}{\theta_1^2}, \dots \frac{\langle \phi_k, \hat{\mathrm{f}}\rangle^2}{\theta_k^2}, 0, \dots 0\right)^\top \tag{2.18}$$

$$\equiv -\gamma\left(\frac{\hat{\alpha}_1{}^2}{\theta_1^2}, \dots \frac{\hat{\alpha}_k{}^2}{\theta_k^2}, 0, \dots 0\right)^\top \tag{2.19}$$

where $\hat{\mathrm{f}}$ and $\hat{\alpha}$ are solutions for minimizing (2.16) and minimizing (2.17), respectively. Eq. (2.19) holds because $\langle \phi_i, \hat{\mathrm{f}}\rangle = \sum_{1\le j\le k} \hat{\alpha}_j \langle \phi_i, \phi_j\rangle = \hat{\alpha}_i$.

To carry out bundle method for the singular case, we need to compute $\nabla_\theta g(\theta)$ via (2.19), which requires $\hat{\alpha}$ as the solution of minimizing (2.17). Compared to solving optimization (2.5)

[3]The analysis follows Sections 2.1.3, 2.1.3 and 2.1.3. We omit the details due to the space limit.

Algorithm 1: Bundle Method for AST

Input

ϵ desired convergence accuracy
$\mathcal{L}$ normalized graph Laplacian of G
ℓ loss function based on available labels
γ tuning parameter for manifold regularization in (2.5)
τ tuning parameter for the ℓ_1-norm in (2.6)

Output

f system-inferred vertex labels
θ system-inferred reciprocals of the transformed Laplacian eigenvalues

Initialization

$t \leftarrow 0$;
/*take pseudo-inverse when necessary*/;
$\{\lambda_i, \phi_i\}_{i=1}^m \leftarrow \mathrm{eig}(\mathcal{L})$, $\left\{\theta_i^{(0)} \leftarrow \lambda_i^{-1}\right\}_{i=1}^m$;

do

/*solve (2.5) via standard SSL*/;
$\mathrm{f}^{(t)} \leftarrow \mathrm{argmin}_{\mathrm{f}\in\mathbb{R}^m} \, \mathcal{C}(\mathrm{f}; \theta^{(t)})$;
$g(\theta^{(t)}) \leftarrow \mathcal{C}(\mathrm{f}^{(t)}; \theta^{(t)})$;
/*according to (2.13)*/;
$\nabla g(\theta^{(t)}) \leftarrow -\gamma\left(\frac{\langle\phi_1, \mathrm{f}^{(t)}\rangle^2}{{\theta_1^{(t)}}^2}, \ldots, \frac{\langle\phi_m, \mathrm{f}^{(t)}\rangle^2}{{\theta_m^{(t)}}^2}\right)^\top$;
$t \leftarrow t + 1$;
/*update the piecewise-linear lowerbound*/;
$\tilde{g}^{(t)}(\theta) \leftarrow \max_{0\le i\le t-1} \; g(\theta^{(i)}) + \left\langle \nabla g(\theta^{(i)}), \theta - \theta^{(i)} \right\rangle$;
/*solve the linear programming*/;
$\theta^{(t)} \leftarrow \mathrm{argmin}_{\theta\in\{\theta|\theta_1\ge\theta_2\ge\ldots\ge\theta_m\ge 0\}} \; \tilde{g}^{(t)}(\theta) + \tau\|\theta\|_1$;

while $g(\theta^{(t-1)}) + \|\theta^{(t-1)}\|_1 - \tilde{g}^{(t)}(\theta^{(t)}) - \|\theta^{(t)}\|_1 > \epsilon$;
/*terminate when the piecewise-linear lowerbound is sufficiently close to the original function*/;

w.r.t. $\mathrm{f} \in \mathbb{R}^m$ for the non-singular case, minimizing (2.17) w.r.t. $\alpha \in \mathbb{R}^k$ can be performed much more efficiently due to the substantially reduced parameter size (recall $k \ll m$). In fact, once the top-k eigenvalues/eigenvectors $\{\lambda_j, \phi_j\}_{j=1}^k$ of $\mathcal{L}$ is obtained, the time/space complexity for both the LP subroutine and the SSL subroutine (2.17) in AST will become independent from m, which is desirable for large problems.

Solving the SSL Subroutine

Both the original and the singular AST involve solving a standard SSL problem as their intermediate subroutines, i.e. minimizing (2.5) w.r.t. f or minimizing (2.17) w.r.t. α. Here we use the later to demonstrate how existing off-the-self machine learning toolkits can be conveniently

leveraged for this purpose.

We specify $\ell(\cdot,\cdot)$ as the squared hinge loss. Besides large-margin property, its smoothness often leads to efficient optimization [20]. In this case, minimizing (2.17) can be formulated as

$$\min_{\alpha\in\mathbb{R}^k} \ \frac{1}{l}\sum_{i\in\mathcal{T}} \max\left(1-y_i e_i^\top \Phi\alpha, 0\right)^2 + \gamma\alpha^\top \mathrm{diag}(\theta_1^{-1},\theta_2^{-1},\dots\theta_k^{-1})\alpha \tag{2.20}$$

where $\Phi=[\phi_1,\phi_2,\dots\phi_k]\in\mathbb{R}^{m\times k}$. By defining $C:=(\gamma l)^{-1}$ and

$$w_j := \alpha_j\sqrt{\frac{2}{\theta_j}} \quad 1\le j\le k \tag{2.21}$$

$$x_i := \mathrm{diag}\left(\sqrt{\frac{\theta_1}{2}},\sqrt{\frac{\theta_2}{2}}\dots\sqrt{\frac{\theta_k}{2}}\right)\Phi^\top e_i \quad \forall i\in\mathcal{T} \tag{2.22}$$

optimization (2.20) can be recast as

$$\min_{w\in\mathbb{R}^k} \ C\sum_{i\in\mathcal{T}} \max\left(1-y_i\langle x_i,w\rangle,0\right)^2 + \frac{1}{2}\|w\|_2^2 \tag{2.23}$$

Note that (2.23) is the standard formulation of L2-SVM and can be efficiently solved via existing solvers such as LIBLINEAR [32]. After obtaining the solution $\hat{w}$ for (2.23), the solution $\hat{\alpha}$ for (2.20) can be easily recovered by rescaling $\hat{w}$, and then be plugged-into (2.18) to compute $\nabla_\theta g(\theta)$ required by the bundle method.

2.1.4 Theoretical Analysis

In this section we provide theoretical intuitions to justify the proposed method. We are going to show that AST can be interpreted as an automatic procedure to asymptotically minimize the SSL generalization error bound w.r.t. different STs.

Our analysis is based an existing theorem on the relationship between the generalization performance of SSL and any given (fixed) graph-Laplacian spectrum [52]. While proving the theorem is not the contribution of this section, our method provides a new angle to utilize the theorem. To the best of our knowledge, none of the previous work, including [52], have formulated or provided any algorithmic solution to *automatically* determine the optimal spectrum among all candidate spectrums in this manner (i.e. formulating and solving optimization (2.6)).

Theorem 2.1.3 (Adapted from [52])**.** *Suppose indices of the labeled vertices in $\mathcal{T}$ are sampled from $\{1,2,\dots,m\}$ uniformly at random. Let $\hat{\mathrm{f}}(\mathcal{T})$ be the system-predicted scores in $\mathbb{R}^m$ obtained via solving optimization (2.4) for any given $\mathcal{T}$, and let ℓ be a convex loss function such that $|\nabla\ell|\le b$. We have*

$$\begin{aligned}&\frac{1}{m-l}\mathbb{E}_{\mathcal{T}}\sum_{i\notin\mathcal{T}}\ell\left(\hat{\mathrm{f}}_i(\mathcal{T}),y_i\right)\\ &\le\left(\min_{\mathrm{f}\in\mathbb{R}^m}\ \frac{1}{m}\sum_{i=1}^m\ell\left(\mathrm{f}_i,y_i\right)+\gamma\mathrm{f}^\top\sigma(\mathcal{L})\mathrm{f}\right)+\frac{b^2\mathrm{tr}\left(\sigma(\mathcal{L})^{-1}\right)}{2\gamma lm}\end{aligned} \tag{2.24}$$

The L.H.S. of (2.24) stands for the empirical risk of SSL for any given ST σ.

To see the connections between AST and Theorem 2.1.3, let $\tau = \frac{b^2}{2\gamma lm}$ and recall that $\sigma(\mathcal{L}) = \sum_{i=1}^m \sigma(\lambda_i)\phi_i\phi_i^\top = \sum_{i=1}^m \theta_i^{-1}\phi_i\phi_i^\top$, we rewrite the R.H.S. of (2.24) as

$$\left(\min_{\mathrm{f}\in\mathbb{R}^m} \frac{1}{m}\sum_{i=1}^m \ell\left(\mathrm{f}_i, y_i\right) + \gamma \sum_{i=1}^m \theta_i^{-1} \left\langle \phi_i, \mathrm{f} \right\rangle^2 \right) + \tau\|\theta\|_1 \tag{2.25}$$

By comparing the AST objective function in (2.6) with (2.25), we see that AST is essentially trying to minimize a surrogate of (2.25) where the true loss $\frac{1}{m}\sum_{i=1}^m \ell\left(\mathrm{f}_i, y_i\right)$ based on all the m vertex labels is substituted by the empirical loss $\frac{1}{l}\sum_{i\in\mathcal{T}} \ell\left(\mathrm{f}_i, y_i\right)$ based on l partially observed vertex labels. The two loss functions are asymptotically equivalent as $l \to m$. This substitution is necessary since in practice it is impossible for us to access all of the m vertex labels during the training phase.

Notice there is an additional isotonic constraint $\theta_1 \geq \theta_2 \ldots \theta_m \geq 0$ for AST when minimizing the generalization error bound (2.25) w.r.t. θ, indicating AST always favours the smooth components over the non-smooth ones in the final prediction $\hat{\mathrm{f}}$.

2.1.5 Experiments

Methods for Comparison

We compare the performance of the following methods in our experiments:

(a) **SSL** is the standard SSL in (2.1) with squared hinge loss. This amounts to taking the ST in (2.4) to be the identity function $\sigma(x) = x$.

(b) **Diffusion** is the ST-enhanced SSL described in (2.4), where σ is parametrized as $\sigma(x) = e^{\beta x}$ a.k.a. the heat diffusion kernel. Prior to SSL, β is empirically estimated by maximizing the kernel alignment score [84] via grid search over $[10^{-4}, 10^4]$.

(c) **GRF** is another ST-enhanced SSL algorithm with $\sigma(x) = x+\beta$, a.k.a. the kernel of Gaussian random field. As in Diffusion, β is empirically estimated before SSL via kernel alignment over $[10^{-5}, 10^3]$.

(d) **NKTA** is nonparametric kernel-target alignment [108], a two-step procedure for ST-enhanced SSL. Prior to SSL, we find σ that maximizes the kernel alignment score without assuming its parametric form. Then, we solve (2.4) with the empirically estimated ST. We follow the formulation of [108] and solve the QCQP subroutine using SeDuMi[4].

(e) **AST** is our proposed method of Adaptive Spectral Transform. Different from the aforementioned two-step kernel alignment approaches, the optimal ST is obtained *along with* SSL by solving the convex optimization problem (2.6) via bundle method.

Experimental Settings

We compare AST against the baselines on benchmark datasets in three different domains:

[4] http://sedumi.ie.lehigh.edu/downloads

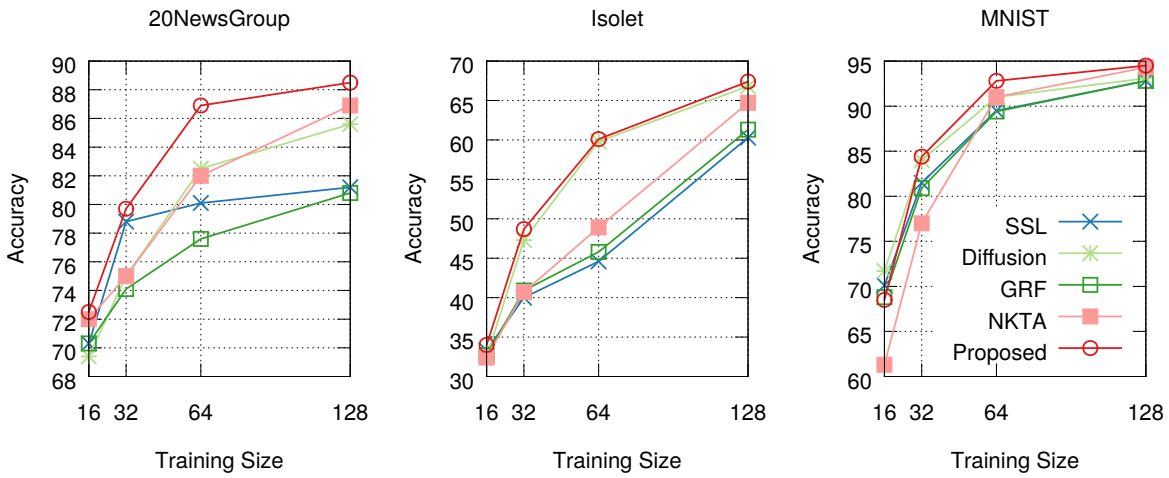

Figure 2.3: Classification accuracy on 20NewsGroup, Isolet and MNIST

1. **20NewsGroup** for document classification. We use the PC-vs-Mac subset consisting of 1,993 documents with binary labels. Following [108], a symmetrized unweighted 10-nearest neighbor (10NN) graph is constructed based on the cosine similarity between documents.
2. **Isolet** for spoken letter recognition consisting of 7,797 instances from 26 classes [5]. We construct a 10NN graph using the Euclidean distance between the audio features.
3. **MNIST** for pattern recognition of the handwritten digits. We use the full training set consisting of 60,000 images from 10 classes (digits 0-9). A 10NN graph is constructed based on the Euclidean distance among the images.

For all datasets, parameter γ for manifold regularization is fixed to be 10^{-3} for all methods as we find the results are not sensitive to the choice of γ. Instead of tuning the hyperparameter τ for our method AST, we simply fix it to be 10^{-2} across all experiments. For all datasets, only the top-50 Laplacian eigenvectors are used for SSL. For AST we use the singular version as described in Section 2.1.3 with $k = 50$.

Given a dataset of m data points, we randomly sample l labeled vertices and predict the remaining unlabeled $m - l$ vertices with methods for comparison. The training size l gradually increases from 2^4 to 2^7, and the experiment is repeated for 30 times for each given training size. The mean and standard variance of the prediction accuracy are reported.

Results

Results are presented in Figure 2.3. For all aforementioned baselines, the prediction accuracy improves and the variance tends to decrease as we gradually enlarge the training size.

First, it is evident that all ST-enhanced methods outperform the traditional SSL on average,

[5] All the algorithms for comparison can be trivially extended to the multi-class case by decomposing the original problem into multiple binary SSL tasks.

which justifies the effectiveness of allowing richer graph transduction patterns over the data manifold.

Secondly, among two-step methods based on empirical kernel-target alignment, it is evident that the nonparametric method NKTA outperforms the two parametric methods Diffusion and GRF, which justifies our previous argument that pre-specifying ST to be within some common function family is too restrictive to accurately capture the "true" graph transduction pattern.

Finally, between nonparametric methods, we observe that the performance of AST dominates NKTA over all datasets. This confirms our intuition that ST-finding and SSL are able to mutually reinforce each other during the joint optimization. The advantageous empirical performance of AST also justifies our previous theoretical analysis in Section 2.1.4.

We also notice AST yields much more stable performance than NKTA. We conjecture that NKTA might be subject to noise as it is trying to fit the target kernel matrix—a quantity induced from only a very limited amount of labels. On the other hand, AST is designed to be adaptive to the problem structure—an arguably more robust reference.

We plotted out the STs produced by different baseline methods over MNIST when $l = 128$ in Figure 2.4. Each sub-figure contains 30 curves in total corresponding to the 30 different runs. From the figure we see that while the STs produced by Diffusion and GRF are restricted to specific parametric forms, STs produced by NKTA and AST are more flexible. Figure 2.4 also shows that STs produced by AST tend to be have lower variance than those produced by NKTA, which justifies our previous stability claim about AST.

An empirical comparison of the speed of all the baseline methods is presented in Table 2.1.

Table 2.1: Speed comparison of different methods on MNIST when $l = 128$ given the top-50 eigenvalues/eigenvectors. We use convergence tolerance $\epsilon = 10^{-3}$ for AST.

Method	SSL	Diffusion	GRF	NKTA	AST
Time (secs)	0.148	0.564	0.738	24.152	2.556

2.2 Learning Graph Convolutions (On-going work)

2.3 Concluding Remarks

In the first part of this chapter, we proposed a new nonparametric framework for carrying out SSL and finding the Laplacian spectrum of the data manifold simultaneously. Different from existing two-step approaches based on manual specification or kernel-target alignment, our approach unifies both tasks into a joint optimization problem and is naturally adaptive to the problem structure. Our formulation enjoys convexity and can be efficiently solved using the bundle method. Theoretical insights are provided to show that the proposed algorithm attempts to asymptotically minimize the SSL generalization error bound w.r.t. the Laplacian spectrum. The merits of our framework are verified by its advantageous empirical performance over strong baselines.

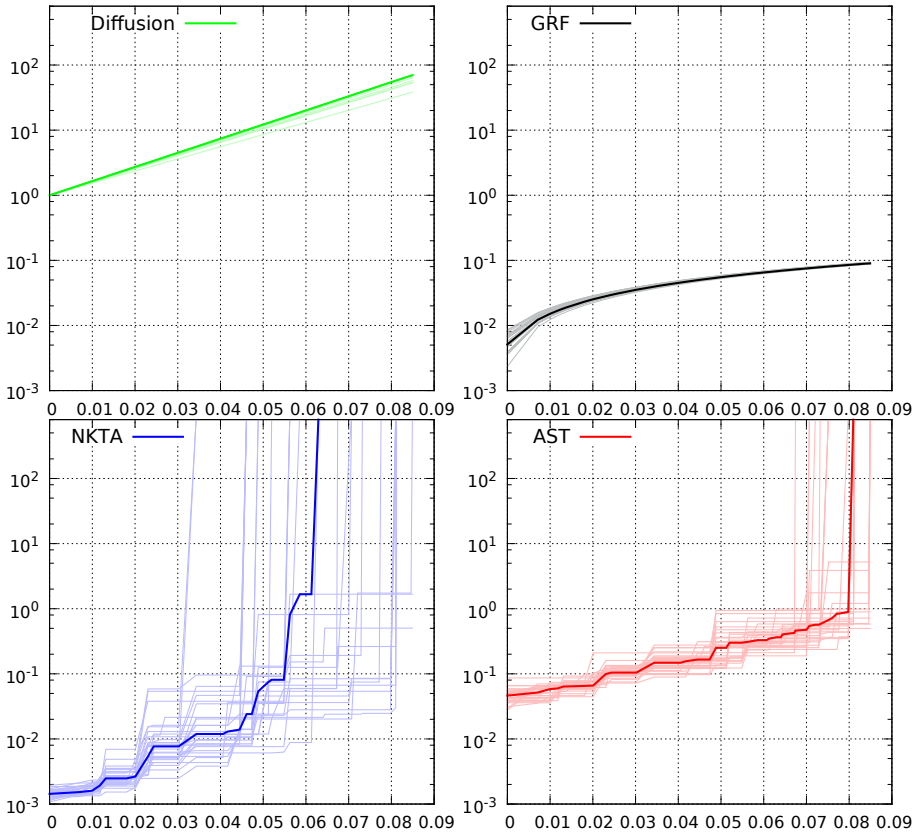

Figure 2.4: STs produced by all methods on the MNIST dataset (each sub-figure contains the results of 30 different runs), where the x-axis and y-axis (log-scale) correspond to the original spectrum λ_i's and the transformed spectrum $\sigma(\lambda_i)$'s, resp.

Chapter 3

Learning with Graph Dynamics

3.1 Online Learning of Multi-task Dependencies (Completed work, NIPS'16)

This section addresses the challenge of jointly learning both the per-task model parameters and the inter-task relationships in a multi-task online learning setting. The proposed algorithm features probabilistic interpretation, efficient updating rules and flexible modulation on whether learners focus on their specific tasks or on jointly address all tasks. The paper also proves a sub-linear regret bound as compared to the best linear predictor in hindsight. Experiments over three multitask learning benchmark datasets show advantageous performance of the proposed approach over several state-of-the-art online multi-task learning baselines.

3.1.1 Motivation

The power of joint learning in multiple tasks arises from the transfer of relevant knowledge across said tasks, especially from information-rich tasks to information-poor ones. Instead of learning individual models, multi-task methods leverage the relationships between tasks to jointly build a better model for each task. Most existing work in multi-task learning focuses on how to take advantage of these task relationships, either to share data directly [23] or to learn model parameters via cross-task regularization techniques [4, 30, 100, 105]. In a broad sense, there are two settings to learn these task relationships 1) batch learning, in which an entire training set is available to the learner 2) online learning, in which the learner sees the data in a sequential fashion. In recent years, online multi-task learning has attracted extensive research attention [1, 19, 27, 66, 82].

Following the online setting, particularly from [27, 66], at each round t, the learner *receives* a set of K observations from K tasks and *predicts* the output label for each of these observations. Subsequently, the learner receives the true labels and *updates* the model(s) as necessary. This sequence is repeated over the entire data, simulating a data stream. Our approach follows an error-driven update rule in which the model for a given task is updated only when the prediction for that task is in error. The goal of an online learner is to minimize errors compared to the full hindsight learner. The key challenge in online learning with large number of tasks is to adaptively learn the model parameters and the task relationships, which potentially change over

time. Without manageable efficient updates at each round, learning the task relationship matrix automatically may impose a severe computational burden. In other words, we need to make predictions and update the models in an efficient real time manner.

We propose an online learning framework that *efficiently* learns multiple related tasks by estimating the task relationship matrix from the data, along with the model parameters for each task. We learn the model for each task by sharing data from related task directly. Our model provides a natural way to specify the trade-off between learning the hypothesis from each task's own (possibly quite limited) data and data from multiple related tasks. We propose an iterative algorithm to learn the task parameters and the task-relationship matrix alternatively. We first describe our proposed approach under a batch setting and then extend it to the online learning paradigm. In addition, we provide a theoretical analysis for our online algorithm and show that it can achieve a sub-linear regret compared to the best linear predictor in hindsight. We evaluate our model with several state-of-the-art online learning algorithms for multiple tasks.

There are many useful application areas for online multitask learning, including optimizing financial trading, email prioritization, personalized news, and spam filtering. Consider the latter, where some spam is universal to all users (e.g. financial scams), some messages might be useful to certain affinity groups, but spam to most others (e.g. announcements of meditation classes or other special interest activities), and some may depend on evolving user interests. In spam filtering each user is a task, and shared interests and dis-interests formulate the inter-task relationship matrix. If we can learn the matrix as well as improving models from specific spam/not-spam decisions, we can perform mass customization of spam filtering, borrowing from spam/not-spam feedback from users with similar preferences. The primary contribution of this section is precisely the joint learning of inter-task relationships and its use in estimating per-task model parameters in an online setting.

Related Work

While there is considerable literature in online multitask learning, many crucial aspects remain largely unexplored. Most existing work in online multitask learning focuses on how to take advantage of task relationships. To achieve this, Lugosi et. al [66] imposed a hard constraint on the K simultaneous actions taken by the learner in the expert setting, Agarwal et. al [3] used matrix regularization, and Dekel et. al [27] proposed a global loss function, as an absolute norm, to tie together the loss values of the individual tasks. Different from existing online multi-task learning models, our paper proposes an intuitive and efficient way to learn the task relationship matrix automatically from the data, and to explicitly take into account the learned relationships during model updates.

Cavallanti et. al [19] assumes that task relationships are available *a priori*. Kshirsagar et. al [56] does the same but in a more adaptive manner. However such task-relation prior knowledge is either unavailable or infeasible to obtain for many applications especially when the number of tasks K is large [97] and/or when the manual annotation of task relationships is expensive [57]. Saha et. al [82] formulated the learning of task relationship matrix as a Bregman-divergence minimization problem w.r.t. positive definite matrices. The model suffers from high computational complexity as semi-definite programming is required when updating the task relationship matrix at each online round. We show that with a different formulation, we can obtain a similar

but much cheaper updating rule for learning the inter-task weights.

Multi-task learning has also has been studied in part under a related research topic, *Domain Adaptation* (DA) [9] under different assumptions. There are several key differences between those methods and ours: i) While DA tries to find a *single* hypothesis that works well for both the source and the target data, this section finds a hypothesis for each task by adaptively leveraging related tasks. ii) It is a typical assumption in DA that the source domains are label-rich and the target domains are label-scarce. However, we are more interested in the scenario where there is a large number of tasks with very few examples available for each task. iii) DA uses predefined uniform weights or weights induced from VC-convergence theory during training, while our method allows cross-task weights to dynamically evolve in an adaptive manner.

The most related work to ours is *Shared Hypothesis* model (*SHAMO*) from Crammer and Mansour [23], where the key idea is to use a K-means-like procedure that simultaneously clusters different tasks and learns a small pool of $m \ll K$ shared hypotheses. Specifically, each task is free to choose a hypothesis from the pool that better classifies its own data, and each hypothesis is learned from pooling together all the training data that belongs to the same cluster. A similar idea was explored by Abernathy et. al [1] under expert settings.

3.1.2 The Proposed Method

Setup

Suppose we are given K tasks where the j^{th} task is associated with N_j training examples. For brevity we consider a binary classification problem for each task, but the methods generalize to multi-class and are also applicable to regression tasks. We denote by $[N]$ the consecutive integers ranging from 1 to N. Let $\left\{(x_j^{(i)}, y_j^{(i)})\right\}_{i=1}^{N_j}$ and $L_j(w) = \frac{1}{N_j}\sum_{i\in[N_j]}\left(1 - y_j^{(i)}\langle x_j^{(i)}, w\rangle\right)_+$ be the training set and batch empirical loss for task j, respectively, where $(z)_+ = \max(0, z), x_j^{(i)} \in \mathbb{R}^d$ is the i^{th} instance from the j^{th} task and $y_j^{(i)}$ is its corresponding true label.

We start from the motivation of our formulation in Section 3.1.2, based on which we first propose a batch formulation in Section 3.1.2. Then, we extend the method to the online setting in Section 3.1.2.

Motivation

Learning tasks may be addressed independently via $w_k^* = \text{argmin}_{w_k} L_k(w_k), \forall k \in [K]$. But, when each task has limited training data, it is often beneficial to allow information sharing among the tasks, which can be achieved via the following optimization:

$$w_k^* = \text{argmin}_{w_k} \sum_{j\in[K]} \eta_{kj} L_j(w_k) \quad \forall k \in [K] \tag{3.1}$$

Beyond each task k, optimization (3.1) encourages hypothesis w_k^* to do well on the remaining $K-1$ tasks thus allowing tasks to borrow information from each other. In the extreme case where the K tasks have an identical data distribution, optimization (3.1) amounts to using $\sum_{j\in[K]} N_j$ examples for training as compared to N_k in independent learning.

The weight matrix $\boldsymbol{\eta}$ is in essence a task relationship matrix, and a prior may be manually specified according to domain knowledge about the tasks. For instance, one sets η_{kj} to a large value if task k and j share similar nature. If $\boldsymbol{\eta} = \boldsymbol{I}$, (3.1) reduces to learning tasks independently. It is clear that manual specification of $\boldsymbol{\eta}$ is feasible only when K is small. Moreover, tasks may be statistically correlated even if a domain expert is unavailable to identify an explicit relation, or if the effort required is too great. Hence, it is often desirable to automatically estimate the optimal $\boldsymbol{\eta}$ adapted to the inter-task problem structure.

We propose to learn η in a data-driven manner. For the k^{th} task, we optimize

$$w_k^*, \eta_k^* = \text{argmin}_{w_k, \eta_k \in \Theta} \sum_{j \in [K]} \eta_{kj} L_j(w_k) + \lambda r(\eta_k) \tag{3.2}$$

where Θ defines the feasible domain of η_k, and regularizer r prevents degenerate cases, e.g., where η_k becomes an all-zero vector. Optimization (3.2) shares the same underlying insight with Self-Paced Learning (SPL) [50, 58] where the algorithm automatically learns the weights over data points during training. However, the process and scope in the two methods differ fundamentally: SPL minimizes the weighted loss over *datapoints* within a single domain, while optimization (3.2) minimizes the weighted loss over multiple *tasks* across possibly heterogeneous domains.

A common choice of Θ and $r(\eta_k)$ in SPL is $\Theta = [0, 1]^K$ and $r(\eta_k) = -\|\eta_k\|_1$. There are several drawbacks of naively applying this type of settings to the multitask scenario: (i) *Unfocused update*: there is no guarantee that the k^{th} learner will put more focus on the k^{th} task itself. When task k is intrinsically difficult, η_{kk}^* could simply be set near zero and w_k^* becomes almost independent of the k^{th} task. (ii) *Weak interpretability*, the learned η_k^* may not be interpretable as it is not directly tied to any physical meanings (iii) *Lack of worst-case guarantee* in the online setting. All those issues will be addressed by our proposed model in the following.

Batch Formulation

We parametrize the aforementioned task relationship matrix $\boldsymbol{\eta} \in \mathbb{R}^{K \times K}$ as follows:

$$\boldsymbol{\eta} = \alpha \boldsymbol{I}_K + (1 - \alpha) \boldsymbol{P} \tag{3.3}$$

where $\boldsymbol{I}_K \in \mathbb{R}^{K \times K}$ is an identity matrix, $\boldsymbol{P} \in \mathbb{R}^{K \times K}$ is a *row-stochastic matrix* and α is a scalar in $[0, 1]$. Task relationship matrix $\boldsymbol{\eta}$ defined as above has the following interpretations:

1. *Concentration Factor* α quantifies the learners' "concentration" on their own tasks. Setting $\alpha = 1$ amounts to independent learning. We will see from the forthcoming Theorem 3.1.1 how to specify α to ensure the optimality of the online regret bound.
2. *Attention Matrix* $\boldsymbol{P}$ quantifies to which degree the learners are attentive to all tasks. Specifically, we define the k^{th} row of $\boldsymbol{P}$, namely $p_k \in \Delta^{K-1}$, as a probability distribution over all tasks where Δ^{K-1} stands for probability simplex. Our goal of learning data-adaptive $\boldsymbol{\eta}$ now becomes learning data-adaptive inter-task attentions in $\boldsymbol{P}$.

Common choices about $\boldsymbol{\eta}$ in several existing algorithms are special cases of (3.3). For instance, domain adaptation assumes $\alpha = 0$ and a fixed row-stochastic matrix $\boldsymbol{P}$; in multi-task

learning, we obtain the effective heuristics of specifying $\boldsymbol{\eta}$ by Cavallanti et. al. [19] when $\alpha = \frac{1}{1+K}$ and $\boldsymbol{P} = \frac{1}{K}\mathbf{1}\mathbf{1}^\top$. When there are $m \ll K$ unique distributions p_k, then the problem reduces to *SHAMO* model [23].

Equation (3.3) implies the task relationship matrix $\boldsymbol{\eta}$ is also row-stochastic, where we always reserve probability α for the k^{th} task itself as $\eta_{kk} \geq \alpha$. For each learner, the presence of α entails a trade off between learning from other tasks and concentrating on its own task.

Motivated by the above discussion, our batch formulation instantiates (3.2) as follows

$$w_k^*, p_k^* = \operatorname{argmin}_{w_k, p_k \in \Delta^{K-1}} \sum_{j \in [K]} \eta_{kj}(p_k) L_j(w_k) - \lambda \mathcal{H}(p_k) \tag{3.4}$$

$$= \operatorname{argmin}_{w_k, p_k \in \Delta^{K-1}} \mathbb{E}_{j \sim Multi(\eta_k(p_k))} L_j(w_k) - \lambda \mathcal{H}(p_k) \tag{3.5}$$

where $\mathcal{H}(p_k) = -\sum_{j \in [K]} p_{kj} \log p_{kj}$ denotes the entropy of distribution p_k. Optimization (3.4) can be viewed as to balance between minimizing the cross-task expected loss with multinomial mixture weights η_k and maximizing the entropy of attention p_k. The max-entropy regularization favours a uniform attention over all tasks and leads to analytical updating rules for p_k (therefore η_k).

Optimization (3.4) is biconvex over w_k and p_k. With $p_k^{(t)}$ fixed, solution for w_k can be obtained using off-the-shelf solvers. With $w_k^{(t)}$ fixed, solution for p_k is given in closed-form:

$$p_{kj}^{(t+1)} \propto e^{-\frac{1-\alpha}{\lambda} L_j(w_k^{(t)})} \tag{3.6}$$

The exponential rule (3.6) has an intuitive interpretation. Namely our algorithm attempts to use hypothesis $w_k^{(t)}$ obtained from the k^{th} task to classify training examples in all other tasks. Task j will be treated as related to task k if its training examples can be well classified by w_k. The intuition is that two tasks are likely to relate to each other if they share similar decision boundaries, thus merging their data pool should yield to a stronger model.

Online Formulation

In this section, we extend our batch formulation to the online setting. We assume that all tasks will be performed at each round, though the assumption can be relaxed with some added complexity to the method. At time t, the k^{th} task receives a training instance $x_k^{(t)}$, makes a prediction $\langle x_k^{(t)}, w_k^{(t)} \rangle$ and suffers a loss after $y^{(t)}$ is revealed. Our algorithm follows a error-driven update rule in which the model is updated only when a task makes a mistake.

Let $\ell_{kj}^{(t)}(w) = 1 - y_j^{(t)} \langle x_j^{(t)}, w \rangle$ if $y_j^{(t)} \langle x_j^{(t)}, w_k^{(t)} \rangle < 1$ and $\ell_{kj}(w) = 0$ otherwise. For brevity, we introduce shorthands $\ell_{kj}^{(t)} = \ell_{kj}^{(t)}(w_k^{(t)})$ and $\eta_{kj}^{(t)} = \eta_{kj}(p_k^{(t)})$.

For the k^{th} task we consider the following optimization problem at each time:

$$w_k^{(t+1)}, p_k^{(t+1)} = \operatorname*{argmin}_{w_k, p_k \in \Delta^{K-1}} C \sum_{j \in [K]} \eta_{kj}(p_k) \ell_{kj}^{(t)}(w_k) + \|w_k - w_k^{(t)}\|^2 + \lambda D_{\mathrm{KL}}\big(p_k \| p_k^{(t)}\big) \tag{3.7}$$

where $\sum_{j \in [K]} \eta_{kj}(p_k) \ell_{kj}^{(t)}(w_k) = \mathbb{E}_{j \sim Multi(\eta_k(p_k))} \ell_{kj}^{(t)}(w_k)$, and $D_{\mathrm{KL}}\big(p_k \| p_k^{(t)}\big)$ denotes the Kull-backLeibler (KL) divergence between current and previous soft-attention distributions. The presence of last two terms in (3.7) allows the model parameters to evolve smoothly over time. Optimization (3.7) is naturally analogous to the batch optimization (3.4), where the batch loss $L_j(w_k)$

is replaced by its noisy version $\ell_{kj}^{(t)}(w_k)$ at time t, and negative entropy $-\mathcal{H}(p_k) = \sum_j p_{kj} \log p_{kj}$ is replaced by $D_{\mathrm{KL}}(p_k \| p_k^{(t)})$ as known as the relative entropy. We will show the above formulation leads to analytical updating rules for both w_k and p_k, a desirable property particularly as an online algorithm.

Solution for $w_k^{(t+1)}$ conditioned on $p_k^{(t)}$ is given in closed-form by the proximal operator

$$w_k^{(t+1)} = \mathbf{prox}(w_k^{(t)}) = \mathrm{argmin}_{w_k} \; C \sum_{j \in [K]} \eta_{kj}(p_k^{(t)}) \ell_{kj}^{(t)}(w_k) + \|w_k - w_k^{(t)}\|^2 \tag{3.8}$$

$$= w_k^{(t)} + C \sum_{j : y_j^{(t)} \langle x_j^{(t)}, w_k^{(t)} \rangle < 1} \eta_{kj}(p_k^{(t)}) y_j^{(t)} x_j^{(t)} \tag{3.9}$$

Solution for $p_k^{(t+1)}$ conditioned on $w_k^{(t)}$ is also given in the closed-form which is analogous to mirror descent [72]

$$p_k^{(t+1)} = \mathrm{argmin}_{p_k \in \Delta^{K-1}} \; C(1-\alpha) \sum_{j \in [K]} p_{kj} \ell_{kj}^{(t)} + \lambda D_{\mathrm{KL}}\big(p_k \| p_k^{(t)}\big) \tag{3.10}$$

$$\implies p_{kj}^{(t+1)} \propto p_{kj}^{(t)} e^{-\frac{C(1-\alpha)}{\lambda} \ell_{kj}^{(t)}} \tag{3.11}$$

Our algorithm is "passive" in the sense that updates are carried out only when a classification error occurs, namely when $\hat{y}_k^{(t)} \neq y_k^{(t)}$. An alternative is to perform "aggressive" updates only when the active set $\{j : y_j^{(t)} \langle x_j^{(t)}, w_k^{(t)} \rangle < 1\}$ is non-empty.

3.1.3 Theoretical Analysis

Theorem 3.1.1. *$\forall k \in [K]$, let $S_k = \big\{\big(x_k^{(t)}, y_k^{(t)}\big)\big\}_{t=1}^T$ be a sequence of T examples for the k^{th} task where $x_k^{(t)} \in \mathbb{R}^d$, $y_k^{(t)} \in \{-1, +1\}$ and $\|x_k^{(t)}\|_2 \leq R$, $\forall t \in [T]$. Let C be a positive constant and let α be some predefined parameter in $[0, 1]$. Let $\{w_k^*\}_{k \in [K]}$ be any arbitrary vectors where $w_k^* \in \mathbb{R}^d$ and its hinge loss on the examples $\big(x_k^{(t)}, y_k^{(t)}\big)$ and $\big(x_j^{(t)}, y_j^{(t)}\big)_{j \neq k}$ are given by $\ell_{kk}^{(t)*} = \big(1 - y_k^{(t)} \langle x_k^{(t)}, w_k^* \rangle\big)_+$ and $\ell_{kj}^{(t)*} = \big(1 - y_j^{(t)} \langle x_j^{(t)}, w_k^* \rangle\big)_+$, respectively.*

If $\{S_k\}_{k \in [K]}$ is presented to OSMTL algorithm, then $\forall k \in [K]$ we have

$$\sum_{t \in [T]} (\ell_{kk}^{(t)} - \ell_{kk}^{(t)*}) \leq \frac{1}{2C\alpha} \|w_k^*\|^2 + \frac{(1-\alpha)T}{\alpha} \Big(\ell_{kk}^{(t)*} + \max_{j \in [K], j \neq k} \ell_{kj}^{(t)*}\Big) + \frac{CR^2T}{2\alpha} \tag{3.12}$$

Notice when $\alpha \to 1$, the above reduces to the perceptron mistake bound [83].

Corollary 3.1.1.1. *Let $\alpha = \frac{\sqrt{T}}{1+\sqrt{T}}$ and $C = \frac{1+\sqrt{T}}{T}$ in Theorem 3.1.1, we have*

$$\sum_{t \in [T]} (\ell_{kk}^{(t)} - \ell_{kk}^{(t)*}) \leq \sqrt{T} \left(\frac{1}{2} \|w_k^*\|^2 + \ell_{kk}^{(t)*} + \max_{j \in [K], j \neq k} \ell_{kj}^{(t)*} + 2R^2\right) \tag{3.13}$$

Proof. Proofs are given in the supplementary. □

Theorem 3.1.1 and Corollary 3.1.1.1 have several implications:

1. Quality of the bound depends on both $\ell_{kk}^{(t)*}$ and the maximum of $\{\ell_{kj}^{(t)*}\}_{j\in[K],j\neq k}$. In other words, the worst-case regret will be lower if the k^{th} true hypothesis w_k^* can well distinguish training examples in both the k^{th} task itself as well as those in all the other tasks.
2. Corollary 3.1.1.1 indicates the gap between the cumulative loss achieved by our algorithm and the optimal hypothesis for task k is bounded by a term growing sub-linearly in T.
3. Corollary 3.1.1.1 provides a principled way to set hyperparameters to achieve the sublinear regret bound. Specifically, recall α quantifies the self-concentration of each task. Therefore, $\alpha = \frac{\sqrt{T}}{1+\sqrt{T}} \overset{T\to\infty}{\longrightarrow} 1$ implies for large horizon it would be less necessary to rely on other tasks as available supervision for the task itself is already plenty; $C = \frac{1+\sqrt{T}}{T} \overset{T\to\infty}{\longrightarrow} 0$ suggests diminishing learning rate over the horizon length.

3.1.4 Experiments

We evaluate the performance of our algorithm under batch and online settings. All reported results in this section are averaged over 30 random runs or permutations of the training data. Unless otherwise specified, all model parameters are chosen via 5-fold cross validation.

Benchmark Datasets

We use three datasets for our experiments. Details are given below:

Landmine Detection[1] consists of 19 tasks collected from different landmine fields. Each task is a binary classification problem: landmines $(+)$ or clutter $(-)$ and each example consists of 9 features extracted from radar images with four moment-based features, three correlation-based features, one energy ratio feature and a spatial variance feature. Landmine data is collected from two different terrains: tasks 1-10 are from highly foliated regions and tasks 11-19 are from desert regions, therefore tasks naturally form two clusters. Any hypothesis learned from a task should be able to utilize the information available from other tasks belonging to the same cluster.

Spam Detection[2] We use the dataset obtained from ECML PAKDD 2006 Discovery challenge for the spam detection task. We used the task B challenge dataset which consists of labeled training data from the inboxes of 15 users. We consider each user as a single task and the goal is to build a personalized spam filter for each user. Each task is a binary classification problem: spam $(+)$ or non-spam $(-)$ and each example consists of approximately $150K$ features representing term frequency of the word occurrences. Since some spam is universal to all users (e.g. financial scams), some messages might be useful to certain affinity groups, but spam to most others. Such adaptive behavior of user's interests and dis-interests can be modeled efficiently by utilizing the data from other users to learn per-user model parameters.

Sentiment Analysis[3] We evaluated our algorithm on product reviews from amazon. The dataset contains product reviews from 24 domains. We consider each domain as a binary classi-

[1] http://www.ee.duke.edu/~lcarin/LandmineData.zip
[2] http://ecmlpkdd2006.org/challenge.html
[3] http://www.cs.jhu.edu/~mdredze/datasets/sentiment

fication task. Reviews with rating greater than 3 were labeled as positive $(+)$, those with rating smaller than 3 were labeled as negative $(-)$, reviews with rating = 3 are discarded as the sentiments were ambiguous and hard to predict. Similar to the previous dataset, each example consists of approximately $350K$ features representing term frequency of the word occurrences.

We choose 3040 examples (160 examples per task) for *landmine*, 1500 emails for *spam* (100 emails per user inbox) and 2400 reviews for *sentiment* (100 reviews per domain) for our experiments. Note that we intentionally kept the size of the training data small to drive the need for learning from other tasks, which diminishes as the training sets per task become large. Since all these datasets have a class-imbalance issue (with few $(+)$ examples as compared to $(-)$ examples), we use average Area Under the ROC Curve (AUC) as the performance measure.

Batch Setting

Since the main focus of this section is online learning, we briefly conduct an experiment on landmine detection dataset for our batch learning to demonstrate the advantages of learning from shared data. We implement two versions of our proposed algorithm with different updates: *SMTL-t* (with thresholding updates) where $p_{kj}^{(t+1)} \propto (\lambda - \ell_{kj}^{(t)})_+$[4] and *SMTL-e* (with exponential updates). We compare our *SMTL** with two standard baseline methods for our batch setting: Independent Task Learning (ITL)—learning a single model for each task and Single Task Learning (STL)—learning a single classification model for pooled data from all the tasks. In addition we compare our models with *SHAMO*, which is closest in spirit with our proposed models. We select the value for λ and α for *SMTL** and M for *SHAMO* using cross validation.

Figure 3.1 (left) shows the average AUC calculated for different training size on *landmine*. We can see that the baseline results are similar to the ones reported by Xue et. al [100]. Our proposed algorithm (*SMTL**) outperforms the other baselines but when we have very few training examples (say 20 per task), the performance of STL improves as it has more examples than the others. Since η depends on the loss incurred on the data from related tasks, this loss-based measure can be unreliable for a small training sample size. To our surprise, *SHAMO* performs worse than the other models which tells us that assuming two tasks are exactly same (in the sense of hypothesis) may be inappropriate in real-world applications. Figure 3.1 (middle & left) show the task relationship matrix η for *SMTL-t* and *SMTL-e* on *landmine* when the number of training instances is 160 per task.

Online Setting

To evaluate the performance of our algorithm in the online setting, we use all three datasets (*landmine*, *spam* and *sentiment*) and compare our proposed methods to 5 baselines. We implemented two variations of Passive-Aggressive algorithm (*PA*) [24]. *PA-ITL* learns independent model for each task and *PA-ONE* builds a single model for all the tasks. We also implemented the algorithm proposed by Dekel et. al for online multi-task learning with shared loss (*OSGL*) [27]. These three baselines do not exploit the task-relationship or the data from other tasks during model update.

[4]Our algorithm and theorem can be easily generalized to other types of updating rules by replacing $\exp$ in (3.6) with other functions. In latter cases, however, $\boldsymbol{\eta}$ may no longer have probabilistic interpretations.

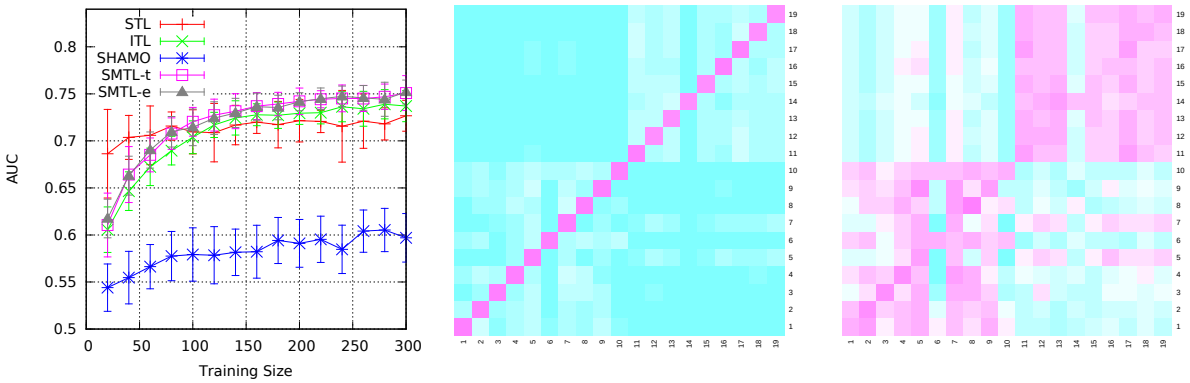

Figure 3.1: Average AUC calculated for compared models (left). A visualization of the task relationship matrix in *Landmine* learned by *SMTL-t* (middle) and *SMTL-e* (right). The probabilistic formulation of *SMTL-e* allows it to discover more interesting patterns than *SMTL-t*.

Table 3.1: Performance means and standard deviations over 30 random shuffles.

Models	**Landmine Detection**			**Spam Detection**			**Sentiment Analysis**		
	AUC	*nSV*	*Time (s)*	*AUC*	*nSV*	*Time (s)*	*AUC*	*nSV*	*Time (s)*
PA-ONE	0.5473 (0.12)	2902.9 (4.21)	0.01	0.8739 (0.01)	1455.0 (4.64)	0.16	0.7193 (0.03)	2350.7 (6.36)	0.19
PA-ITL	0.5986 (0.04)	618.1 (27.31)	0.01	0.8350 (0.01)	1499.9 (0.37)	0.16	0.7364 (0.02)	2399.9 (0.25)	0.16
OSGL	0.6482 (0.03)	740.8 (42.03)	0.01	0.9551 (0.007)	1402.6 (13.57)	0.17	0.8375 (0.02)	2369.3 (14.63)	0.17
FOML	0.6322 (0.04)	426.5 (36.91)	0.11	0.9347 (0.009)	819.8 (18.57)	1.5	0.8472 (0.02)	1356.0 (78.49)	1.20
OMTRL	0.6409 (0.05)	432.2 (123.81)	6.9	0.9343 (0.008)	840.4 (22.67)	53.6	0.7831 (0.02)	1346.2 (85.99)	128
OSMTL-t	**0.6776** (0.03)	333.6 (40.66)	0.18	0.9509 (0.007)	809.5 (19.35)	1.4	0.9354 0.01	1312.8 (79.15)	2.15
OSMTL-e	0.6404 (0.04)	458 (36.79)	0.19	**0.9596** (0.006)	804.2 (19.05)	1.3	**0.9465** (0.01)	1322.2 (80.27)	2.16

Next, we implemented two online multi-task learning related to our approach: *FOML* – initializes η with fixed weights [19], Online Multi-Task Relationship Learning (*OMTRL*) [82] – learns a task covariance matrix along with task parameters. We could not find a better way to implement the online version of the SHAMO algorithm, since the number of shared hypotheses or clusters varies over time.

Table 3.1 summarizes the performance of all the above algorithms. In addition to the AUC scores, we report the average total number of support vectors (*nSV*) and the CPU time taken for learning from one instance (*Time*). From the table, it is evident that *OSMTL** outperforms all the baselines in terms of both AUC and *nSV*. This is expected for the two default baselines (*PA-ITL* and *PA-ONE*). We believe that *PA-ONE* shows better result than *PA-ITL* in *spam* because the former learns the global information (common spam emails) that is quite dominant in spam detection problem. The update rule for *FOML* is similar to ours but using fixed weights. The

results justify our claim that making the weights adaptive leads to improved performance.

In addition to better results, our algorithm consumes less or comparable CPU time than the baselines which take into account inter-task relationships. Compared to the *OMTRL* algorithm that recomputes the task covariance matrix every iteration using expensive SVD routines, the adaptive weights in our are updated independently for each task. As specified in [82], we learn the task weight vectors for *OMTRL* separately as K independent perceptron for the first half of the training data available (*EPOCH*=0.5). *OMTRL* potentially looses half the data without learning task-relationship matrix as it depends on the quality of the task weight vectors. On the other hand, our algorithm efficiently leverages all task data by choosing the value for $\alpha = \frac{\sqrt{t}}{\sqrt{t}+1}$ for round t, and adaptively selects the update weights for the related task, based on the number of examples seen so far, as explained in Corollary 3.1.1.1.

It is evident from the table that algorithms which use loss-based update weights $\boldsymbol{\eta}$ (*OSGL*, *OSMTL**) considerably outperform the ones that do not use it (*FOML*,*OMTRL*). We believe that loss incurred per instance gives us valuable information for the algorithm to learn from that instance, as well as to evaluate the inter-dependencies among tasks. That said, task relationship information does help by learning from the related tasks' data, but we demonstrate that combining both the task relationship and the loss information can give us a better algorithm, as is evident from our experiments.

We would like to note that our proposed algorithm *OSMTL** does exceptionally better in *sentiment*, which has been used as a standard benchmark application for DA experiments in the existing literature [11]. We believe the advantageous results on *sentiment* dataset implies that even with relatively few examples, effectively knowledge transfer among the tasks/domains can be achieved by adaptively choosing the (probabilistic) inter-task relationships from the data.

3.2 Graph-Augmented Temporal Modeling (On-going work)

To be completed.

3.3 Concluding Remarks

In the first part of this chapter, we proposed a novel online multi-task learning algorithm that jointly learns the per-task hypothesis and the inter-task relationships. The key idea is based on smoothing the loss function of each task w.r.t. a probabilistic distribution over all tasks, and adaptively refining such distribution over time. In addition to closed-form updating rules, we show our method achieves the sub-linear regret bound. Effectiveness of our algorithm is empirically verified over several benchmark datasets.

Chapter 4

Learning Graph Topologies

4.1 Efficient Neural Architecture Search (On-going work)

4.1.1 Motivation

4.1.2 Asynchronous Evolution

4.1.3 Hierarchical Genetic Representation

4.1.4 Experiments

4.2 Concluding Remarks

Chapter 5

Timeline

The timeline for completion of this thesis is as follows:

- **Nov 2017–Apr 2018**: Efficient graph topology optimization.
  - *Goal*: Economical neural architecture search under a computation budget.
  - *Stretch*: Efficient search algorithms for natural/program language synthesis tasks.
- **Nov 2017–Apr 2018**: Learning with graph dynamics.
  - *Goal*: Improved spatial-temporal modeling using graph convolutional architectures.
  - *Stretch*: Better model-based reinforcement learning with improved learning of the environmental dynamics.
- **May 2018**: Thesis Writing.
- **Jun 2018**: Thesis Defense.

Bibliography

[1] Jacob Abernethy, Peter Bartlett, and Alexander Rakhlin. Multitask learning with expert advice. In *Learning Theory*, pages 484–498. Springer, 2007. 3.1.1, 3.1.1

[2] Jiří Adámek, Horst Herrlich, and George E Strecker. Abstract and concrete categories. the joy of cats. 2004. 1.2.2

[3] Alekh Agarwal, Alexander Rakhlin, and Peter Bartlett. Matrix regularization techniques for online multitask learning. *EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2008-138*, 2008. 3.1.1

[4] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008. 3.1.1

[5] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007. 1.2.1

[6] Francis R Bach, Gert RG Lanckriet, and Michael I Jordan. Multiple kernel learning, conic duality, and the smo algorithm. In *Proceedings of the twenty-first international conference on Machine learning*, page 6. ACM, 2004. a

[7] Justin Basilico and Thomas Hofmann. Unifying collaborative and content-based filtering. In *Proceedings of the twenty-first international conference on Machine learning*, page 9. ACM, 2004. 1.1.1

[8] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *The Journal of Machine Learning Research*, 7:2399–2434, 2006. 2.1.1

[9] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010. 3.1.1

[10] Asa Ben-Hur and William Stafford Noble. Kernel methods for predicting protein–protein interactions. *Bioinformatics*, 21(suppl 1):i38–i46, 2005. 1.1.1

[11] John Blitzer, Mark Dredze, Fernando Pereira, et al. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*, volume 7, pages 440–447, 2007. 3.1.4

[12] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings*

of the 2008 ACM SIGMOD international conference on Management of data, pages 1247–1250. AcM, 2008. 1.2.1

[13] Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. Learning structured embeddings of knowledge bases. In *Conference on artificial intelligence*, number EPFL-CONF-192344, 2011. 1.3

[14] Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. Joint learning of words and meaning representations for open-text semantic parsing. In *AISTATS*, volume 22, pages 127–135, 2012. 1.3

[15] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, pages 2787–2795, 2013. 1.1.5, 1.2.1, 1.2.2, 1.2.5, 1.2.5, 1.2.5, 1.3, 1.4

[16] Stephen Boyd. Convex optimization of graph laplacian eigenvalues. In *Proceedings of the International Congress of Mathematicians*, volume 3, pages 1311–1319, 2006. b

[17] Ronald Brown and Tim Porter. Category theory: an abstract setting for analogy and comparison. In *What is category theory*, volume 3, pages 257–274, 2006. 1.2.2

[18] Deng Cai, Xiaofei He, Jiawei Han, and Thomas S Huang. Graph regularized nonnegative matrix factorization for data representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(8):1548–1560, 2011. 1.1.1, 1.1.5

[19] Giovanni Cavallanti, Nicolo Cesa-Bianchi, and Claudio Gentile. Linear algorithms for online multitask classification. *The Journal of Machine Learning Research*, 11:2901–2934, 2010. 3.1.1, 3.1.1, 3.1.2, 3.1.4

[20] Kai-Wei Chang, Cho-Jui Hsieh, and Chih-Jen Lin. Coordinate descent method for large-scale l2-loss linear support vector machines. *The Journal of Machine Learning Research*, 9:1369–1398, 2008. 2.1.3

[21] Danqi Chen, Richard Socher, Christopher D Manning, and Andrew Y Ng. Learning new facts from knowledge bases with neural tensor networks and semantic word vectors. *arXiv preprint arXiv:1301.3618*, 2013. 1.2.1

[22] Fan RK Chung. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997. 2.1.1, 2.1.2

[23] Koby Crammer and Yishay Mansour. Learning multiple tasks using shared hypotheses. In *Advances in Neural Information Processing Systems*, pages 1475–1483, 2012. 3.1.1, 3.1.1, 3.1.2

[24] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online passive-aggressive algorithms. *The Journal of Machine Learning Research*, 7: 551–585, 2006. 3.1.4

[25] Jeffrey Dalton, Laura Dietz, and James Allan. Entity query feature expansion using knowledge base links. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 365–374. ACM, 2014. 1.2.1

[26] John M Danskin. The theory of max-min, with applications. *SIAM Journal on Applied*

Mathematics, 14(4):641–664, 1966. 2.1.3

[27] Ofer Dekel, Philip M Long, and Yoram Singer. Online learning of multiple tasks with a shared loss. *Journal of Machine Learning Research*, 8(10):2233–2264, 2007. 3.1.1, 3.1.1, 3.1.4

[28] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12: 2121–2159, 2011. 1.1.4, 1.2.5

[29] Nelson Dunford, Jacob T Schwartz, William G Bade, and Robert G Bartle. *Linear operators*. Wiley-interscience New York, 1971. 1.2.2

[30] Theodoros Evgeniou and Massimiliano Pontil. Regularized multi–task learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 109–117. ACM, 2004. 3.1.1

[31] Brian Falkenhainer, Kenneth D Forbus, and Dedre Gentner. The structure-mapping engine: Algorithm and examples. *Artificial intelligence*, 41(1):1–63, 1989. 1.2.1

[32] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008. 2.1.3

[33] Rob Fergus, Yair Weiss, and Antonio Torralba. Semi-supervised learning in gigantic image collections. In *Advances in neural information processing systems*, pages 522–530, 2009. 1.1.3, b

[34] David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A Kalyanpur, Adam Lally, J William Murdock, Eric Nyberg, John Prager, et al. Building watson: An overview of the deepqa project. *AI magazine*, 31(3):59–79, 2010. 1.2.1

[35] Evgeniy Gabrilovich and Shaul Markovitch. Wikipedia-based semantic interpretation for natural language processing. *Journal of Artificial Intelligence Research*, 34:443–498, 2009. 1.2.1

[36] Alberto Garcia-Duran, Antoine Bordes, and Nicolas Usunier. *Composing relationships with translations*. PhD thesis, CNRS, Heudiasyc, 2015. 1.2.5, 1.3

[37] Dedre Gentner. Structure-mapping: A theoretical framework for analogy. *Cognitive science*, 7(2):155–170, 1983. 1.2.1, 1.2.2

[38] Lise Getoor. *Introduction to statistical relational learning*. MIT press, 2007. 1.1.1, 1.2.1

[39] Robert M Gray et al. Toeplitz and circulant matrices: A review. *Foundations and Trends® in Communications and Information Theory*, 2(3):155–239, 2006. 1.2.2, 1.2.4, 1.2.4

[40] Robert Grone, Charles R Johnson, Eduardo M Sa, and Henry Wolkowicz. Normal matrices. *Linear Algebra and its Applications*, 87:213–225, 1987. 1.2.2

[41] Kelvin Guu, John Miller, and Percy Liang. Traversing knowledge graphs in vector space. *arXiv preprint arXiv:1506.01094*, 2015. 1.2.1, 1.2.2

[42] Masahiro Hattori, Yasushi Okuno, Susumu Goto, and Minoru Kanehisa. Heuristics for chemical compound matching. *Genome Informatics*, 14:144–153, 2003. 1.1.5

[43] Shizhu He, Kang Liu, Guoliang Ji, and Jun Zhao. Learning to represent knowledge graphs with gaussian embedding. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 623–632. ACM, 2015. 1.3

[44] Douglas R Hofstadter. Analogy as the core of cognition. *The analogical mind: Perspectives from cognitive science*, pages 499–538, 2001. 1.2.2

[45] Keith J Holyoak, Keith James Holyoak, and Paul Thagard. *Mental leaps: Analogy in creative thought*. MIT press, 1996. 1.2.2

[46] Takahiko Ito, Masashi Shimbo, Taku Kudo, and Yuji Matsumoto. Application of kernels to link analysis. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 586–592. ACM, 2005. 2.1.1

[47] Rodolphe Jenatton, Nicolas L Roux, Antoine Bordes, and Guillaume R Obozinski. A latent factor model for highly multi-relational data. In *Advances in Neural Information Processing Systems*, pages 3167–3175, 2012. 1.2.5, 1.3

[48] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. Knowledge graph embedding via dynamic mapping matrix. In *ACL (1)*, pages 687–696, 2015. 1.2.5, 1.3

[49] Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. Knowledge graph completion with adaptive sparse transfer matrix. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 985–991, 2016. URL http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/11982. 1.3

[50] Lu Jiang, Deyu Meng, Shoou-I Yu, Zhenzhong Lan, Shiguang Shan, and Alexander Hauptmann. Self-paced learning with diversity. In *Advances in Neural Information Processing Systems*, pages 2078–2086, 2014. 3.1.2

[51] Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM, 2002. 1.1.5

[52] Rie Johnson and Tong Zhang. Graph-based semi-supervised learning and spectral kernel design. *Information Theory, IEEE Transactions on*, 54(1):275–288, 2008. 2.1.1, 2.1.2, 2.1.2, 2.1.4, 2.1.3

[53] Marius Kloft, Ulf Brefeld, Pavel Laskov, Klaus-Robert Müller, Alexander Zien, and Sören Sonnenburg. Efficient and accurate lp-norm multiple kernel learning. In *Advances in neural information processing systems*, pages 997–1005, 2009. 2.1.3

[54] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009. 1.1.1, 1.1.2

[55] Risi Imre Kondor and John Lafferty. Diffusion kernels on graphs and other discrete input spaces. In *ICML*, volume 2, pages 315–322, 2002. 2.1.1, 2.1.2

[56] Meghana Kshirsagar, Jaime Carbonell, and Judith Klein-Seetharaman. Multisource transfer learning for host-pathogen protein interaction prediction in unlabeled tasks. In *NIPS Workshop on Machine Learning for Computational Biology*, 2013. 3.1.1

[57] Meghana Kshirsagar, Jaime Carbonell, and Judith Klein-Seetharaman. Multitask learning

for host–pathogen protein interactions. *Bioinformatics*, 29(13):i217–i226, 2013. 3.1.1

[58] M Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, pages 1189–1197, 2010. 3.1.2

[59] Jérôme Kunegis and Andreas Lommatzsch. Learning spectral graph transformations for link prediction. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 561–568. ACM, 2009. 2.1.1, 2.1.2, 2.1.2

[60] Gert RG Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I Jordan. Learning the kernel matrix with semidefinite programming. *The Journal of Machine Learning Research*, 5:27–72, 2004. a

[61] Nada Lavrac and Saso Dzeroski. Inductive logic programming. In *WLP*, pages 146–160. Springer, 1994. 1.1.1

[62] Yankai Lin, Zhiyuan Liu, Huanbo Luan, Maosong Sun, Siwei Rao, and Song Liu. Modeling relation paths for representation learning of knowledge bases. *arXiv preprint arXiv:1506.00379*, 2015. 1.2.5, 1.3

[63] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*, pages 2181–2187, 2015. 1.2.1, 1.2.5, 1.3

[64] Hanxiao Liu and Yiming Yang. Bipartite edge prediction via transductive learning over product graphs. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1880–1888, 2015. 1.1.1, 1.2.1

[65] Hanxiao Liu and Yiming Yang. Cross-graph learning of multi-relational associations. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 2235–2243, 2016. 1.2.1

[66] Gábor Lugosi, Omiros Papaspiliopoulos, and Gilles Stoltz. Online multi-task learning with hard constraints. *arXiv preprint arXiv:0902.3526*, 2009. 3.1.1, 3.1.1

[67] Stefano Melacci and Mikhail Belkin. Laplacian support vector machines trained in the primal. *The Journal of Machine Learning Research*, 12:1149–1184, 2011. 2.1.1

[68] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013. 1.2.1, 1.2.2

[69] Bonan Min, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. Distant supervision for relation extraction with an incomplete knowledge base. In *HLT-NAACL*, pages 777–782, 2013. 1.2.1

[70] Marvin Minsky. *Society of mind*. Simon and Schuster, 1988. 1.2.2

[71] Atsuhiro Narita, Kohei Hayashi, Ryota Tomioka, and Hisashi Kashima. Tensor factorization using auxiliary information. *Data Mining and Knowledge Discovery*, 25(2):298–324, 2012. 1.1.1, 1.1.5

[72] A-S Nemirovsky, D-B Yudin, and E-R Dawson. Problem complexity and method effi-

ciency in optimization. 1982. 3.1.2

[73] Dat Quoc Nguyen, Kairit Sirts, Lizhen Qu, and Mark Johnson. Stranse: a novel embedding model of entities and relationships in knowledge bases. *arXiv preprint arXiv:1606.08140*, 2016. 1.2.5, 1.3

[74] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 809–816, 2011. 1.1.1, 1.2.1, 1.2.2, 1.2.5, 1.3, 1.4

[75] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *arXiv preprint arXiv:1503.00759*, 2015. 1.2.1

[76] Maximilian Nickel, Lorenzo Rosasco, and Tomaso A. Poggio. Holographic embeddings of knowledge graphs. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 1955–1961, 2016. URL http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12484. (document), 1.2.1, 1.2.2, 1.2.4, 1.2.4, 1.2.5, 1.3, 1.4

[77] Ivan V Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33 (5):2295–2317, 2011. 1.1.3

[78] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014. 1.2.1, 1.2.2

[79] Tony A Plate. Holographic reduced representation: Distributed representation for cognitive structures. 2003. 1.2.4

[80] Benjamin Recht, Christopher Re, Stephen Wright, and Feng Niu. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pages 693–701, 2011. 1.2.5

[81] Steffen Rendle, Leandro Balby Marinho, Alexandros Nanopoulos, and Lars Schmidt-Thieme. Learning optimal ranking with tensor factorization for tag recommendation. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 727–736. ACM, 2009. 1.1.1

[82] Avishek Saha, Piyush Rai, Suresh Venkatasubramanian, and Hal Daume. Online learning of multiple tasks and their relationships. In *International Conference on Artificial Intelligence and Statistics*, pages 643–651, 2011. 3.1.1, 3.1.1, 3.1.4, 3.1.4

[83] Shai Shalev-Shwartz and Yoram Singer. Online learning: Theory, algorithms, and applications. *PhD Dissertation*, 2007. 3.1.3

[84] N Shawe-Taylor and A Kandola. On kernel target alignment. *Advances in neural information processing systems*, 14:367, 2002. 2.1.2, b

[85] Amit Singhal. Introducing the knowledge graph: things, not strings. *Official google blog*, 2012. 1.2.1, 1.2.1

[86] Temple F Smith and Michael S Waterman. Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197, 1981. 1.1.5

[87] Alexander J Smola and Risi Kondor. Kernels and regularization on graphs. In *Learning theory and kernel machines*, pages 144–158. Springer, 2003. 2.1.1, 2.1.2, 2.1.2

[88] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. In *Advances in neural information processing systems*, pages 926–934, 2013. 1.2.1, 1.2.5, 1.3

[89] Sören Sonnenburg, Gunnar Rätsch, Christin Schäfer, and Bernhard Schölkopf. Large scale multiple kernel learning. *The Journal of Machine Learning Research*, 7:1531–1565, 2006. 2.1.3

[90] Suvrit Sra, Sebastian Nowozin, and Stephen J Wright. *Optimization for machine learning*. Mit Press, 2012. 2.1.3

[91] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 990–998. ACM, 2008. 1.1.5

[92] Choon Hui Teo, SVN Vishwanthan, Alex J Smola, and Quoc V Le. Bundle methods for regularized risk minimization. *The Journal of Machine Learning Research*, 11:311–365, 2010. 2.1.3

[93] Kristina Toutanova and Danqi Chen. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pages 57–66, 2015. 1.2.5, 1.3

[94] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 2071–2080, 2016. URL http://jmlr.org/proceedings/papers/v48/trouillon16.html. (document), 1.2.1, 1.2.4, 1.2.4, 1.2.5, 1.3, 1.4

[95] Peter D Turney. The latent relation mapping engine: Algorithm and experiments. *Journal of Artificial Intelligence Research*, 33:615–655, 2008. 1.2.1

[96] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *AAAI*, pages 1112–1119. Citeseer, 2014. 1.2.1, 1.2.5, 1.3

[97] Kilian Weinberger, Anirban Dasgupta, John Langford, Alex Smola, and Josh Attenberg. Feature hashing for large scale multitask learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1113–1120. ACM, 2009. 3.1.1

[98] James Hardy Wilkinson and James Hardy Wilkinson. *The algebraic eigenvalue problem*, volume 87. Clarendon Press Oxford, 1965. 1.2.1

[99] Ruobing Xie, Zhiyuan Liu, and Maosong Sun. Representation learning of knowledge graphs with hierarchical types. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 2965–2971, 2016. 1.3

[100] Ya Xue, Xuejun Liao, Lawrence Carin, and Balaji Krishnapuram. Multi-task learning for classification with dirichlet process priors. *The Journal of Machine Learning Research*, 8:

35–63, 2007. 3.1.1, 3.1.4

[101] Yoshihiro Yamanishi, Michihiro Araki, Alex Gutteridge, Wataru Honda, and Minoru Kanehisa. Prediction of drug–target interaction networks from the integration of chemical and genomic spaces. *Bioinformatics*, 24(13):i232–i240, 2008. 1.1.5

[102] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. *CoRR*, abs/1412.6575, 2014. URL http://arxiv.org/abs/1412.6575. 1.2.1, 1.2.2, 1.2.2, 1.2.4, 1.2.4, 1.2.5, 1.3, 1.4

[103] Yiming Yang and Xin Liu. A re-examination of text categorization methods. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 42–49. ACM, 1999. 1.2.5

[104] Kai Yu and Wei Chu. Gaussian process models for link analysis and transfer learning. In *Advances in Neural Information Processing Systems*, pages 1657–1664, 2008. 1.1.1

[105] Yu Zhang and Dit-Yan Yeung. A regularization approach to learning task relationships in multitask learning. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 8(3): 12, 2014. 3.1.1

[106] Xiaojin Zhu and Zoubin Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical report, Citeseer, 2002. 2.1.1

[107] Xiaojin Zhu, Zoubin Ghahramani, John Lafferty, et al. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, volume 3, pages 912–919, 2003. 1.1.2, 2.1.1

[108] Xiaojin Zhu, Jaz Kandola, Zoubin Ghahramani, and John D Lafferty. Nonparametric transforms of graph kernels for semi-supervised learning. In *Advances in neural information processing systems*, pages 1641–1648, 2004. 2.1.1, 2.1.2, 2.1.2, d, 1