Empirical Risk Minimization and Stochastic Gradient Descent for Relational Data

Vicor Veitch[1] **Morgane Austern**[1] **Wenda Zhou**[1] **David M. Blei** **Peter Orbanz**
Columbia University

Abstract

Empirical risk minimization is the main tool for prediction problems, but its extension to relational data remains unsolved. We solve this problem using recent ideas from graph sampling theory to (i) define an empirical risk for relational data and (ii) obtain stochastic gradients for this empirical risk that are automatically unbiased. This is achieved by considering the method by which data is sampled from a graph as an explicit component of model design. By integrating fast implementations of graph sampling schemes with standard automatic differentiation tools, we provide an efficient turnkey solver for the risk minimization problem. We establish basic theoretical properties of the procedure. Finally, we demonstrate relational ERM with application to two non-standard problems: one-stage training for semi-supervised node classification, and learning embedding vectors for vertex attributes. Experiments confirm that the turnkey inference procedure is effective in practice, and that the sampling scheme used for model specification has a strong effect on model performance. Code is available at github.com/wooden-spoon/relational-ERM.

1 Introduction

Relational data is data that can be represented as a graph, possibly annotated with additional information. An example is the link graph of a social network, annotated by user profiles. We consider prediction problems for such data. For example, how to predict the preferences of a user of a social network using both the preferences and profiles of other users, and the network itself? In the classical case of i.i.d. sequence data—where the observed data does not include link structure—the data decomposes into individual examples. Prediction methods for such data typically rely on this decomposition, e.g., predicting a user's preferences from only the profile of the user, ignoring the network structure. Relational data, however, does not decompose; e.g., because of the link structure, a social network can not be decomposed into individual users. Accordingly, classical methods do not generally apply to relational data, and new methods cannot be developed with the same ease as for i.i.d. sequence data.

With i.i.d. sequence data, prediction problems are typically solved with models fit by empirical risk minimization (ERM) [24, 25, 22]. We give an (unusual) presentation of ERM that anticipates the relational case. The observed data is a set $\overline{\mathbb{S}}_n = \{\overline{X}_1, \dots, \overline{X}_n\}$ that decomposes into examples $\overline{X}_i = (X_i, Y_i)$. The task is to choose a predictor π that completes X by estimating missing information Y, e.g., a class label. An ERM model is defined by two parts: (i) a hypothesis class $\{\pi_\theta | \theta \in \mathcal{T}\}$ from which π is chosen, and (ii) a loss function L where $L(\bar{x}; \theta) \in \mathbb{R}_+$ measures the reconstruction error of predictor π_θ on example $\bar{x}$. The empirical risk is the expected loss on an example randomly selected from the dataset:

$$\hat{R}(\theta, \overline{\mathbb{S}}_n) := \mathbb{E}_{\overline{X} \sim \mathbb{F}(\overline{\mathbb{S}}_n)}[L(\overline{X}; \theta) | \overline{\mathbb{S}}_n], \tag{1}$$

where $\mathbb{F}(\overline{\mathbb{S}}_n)$ is the empirical distribution.[2] The ERM dogma is to select the predictor $\pi_{\hat{\theta}_n}$ given by $\hat{\theta}_n = \operatorname{argmin}_\theta \hat{R}(\theta, \overline{\mathbb{S}}_n)$. That is, the objective function that defines learning is the empirical risk.

ERM has two useful properties. (1) It provides a principled framework for defining new machine learning methods. In particular, when examples are generated i.i.d., model-agnostic results guarantee that ERM models cohere as more data is collected (e.g., in the sense

[1] Equal contribution

Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89.

[2] The empirical risk is more often equivalently written as $\hat{R}(\theta, \overline{\mathbb{S}}_n) = \frac{1}{n} \sum_{i \le n} L(\overline{X}_i; \theta)$.

of statistical convergence) [22]. (2) For differentiable models, mini-batch stochastic gradient descent (SGD) can efficiently solve the minimization problem (albeit, approximately). The ease of SGD comes from the definition of the empirical risk as the expectation over a randomly subsampled example: the gradient of the loss on a randomly subsampled example is an unbiased estimate of the gradient of the empirical risk. Combined with automatic differentiation, this provides a turnkey approach to fitting machine-learning models.

Returning to relational data, the observed data is now a graph $\overline{G}_n$ of size n (e.g., the number of vertices or edges). The graph is possibly annotated, e.g., by vertex labels. We further consider G_n as an incomplete version of $\overline{G}_n$. For example, G_n may censor labels of the vertices or some of the edges from $\overline{G}_n$. In relational learning, the task is to find a predictor π that completes G_n by estimating the missing information. Typically, π is chosen from a parameterized family $\{\pi_\theta | \theta \in \mathcal{T}\}$ to minimize an objective function $\mathcal{O}_n(\theta, \overline{G}_n)$. Unlike the empirical risk, the objective $\mathcal{O}_n$ is not built from a loss on individual examples; $\mathcal{O}_n$ must be specified for the entire observed graph.

In relational learning, there is not yet a framework that has properties (1) and (2) of ERM. The challenge is that relational data does not decompose into individual examples. Regarding (1), theory is elusive because the i.i.d. sequence assumption is meaningless for relational data. This makes it difficult to reason about what happens as more data is collected. Regarding (2), mini-batch SGD is not generally applicable even for differentiable models. SGD requires unbiased estimates of the full gradient. For a random subgraph G_k of G_n, the stochastic gradient $\nabla_\theta \mathcal{O}_k(\pi_\theta(G_k), \overline{G}_k)$ is not generally unbiased. In particular, the bias depends on the choice of random sampling scheme used to select the subgraph. Circumventing these two issues requires either careful design of the objective function used for learning [e.g., 19, 8, 3, 29, 9], or model-specific derivation and analysis. For example, graph convolutional networks [11, 12, 21, 23] use full batch gradients, and scaling training requires custom derivation of stochastic gradients [4].

This paper introduces relational ERM, a generalization of ERM to relational data. Relational ERM provides a recipe for machine learning with relational data that preserves the two important properties of ERM:

1. It provides a simple way to define (task-specific) relational learning methods, and

2. For differentiable models, relational ERM minimization can be efficiently solved in a turnkey fashion by mini-batch stochastic gradient descent. Relational ERM mitigates the need for model-specific analysis and fitting procedures.

Extending turnkey mini-batch SGD to relational data allows the easy use of autodiff-based machine-learning frameworks for relational learning. To facilitate this, we provide fast implementations of a number of graph subsampling algorithms, and integration with TensorFlow.[3]

In Section 2 we define relational ERM models and show how to automatically calculate unbiased mini-batch stochastic gradients. In Section 3 we explain connections to previous work on machine learning for graph data and we illustrate how to develop task-specific relational ERM models. In Section 4 we review several randomized algorithms for subsampling graphs. Relational ERM models require the specification of such algorithms. In Section 5 we establish theory for relational ERM models. The main insights are: (i) the i.i.d. assumption can be replaced by an assumption on how the data is collected [18, 27, 1, 5], and, (ii) the choice of randomized sampling algorithm is necessarily viewed as a model component. In Section 6, we study relational ERM empirically by implementing the models of Section 3. We observe that the turnkey mini-batch SGD procedure succeeds in efficiently fitting the models, and that the choice of graph subsampling algorithm has a large effect in practice.

2 Relational ERM and SGD

Our aim is to define relational ERM in analogy with classical ERM. The fundamental challenge is that relational data does not decompose into individual examples. Classical ERM uses the empirical distribution to define the objective function Eq. (1). There is no canonical analogue of the empirical distribution for relational data.

The first insight is that the empirical distribution may be viewed as a randomized algorithm for subsampling the dataset. The required analogue is then a randomized algorithm for subsampling a graph. In the i.i.d. setting, uniform subsampling is almost always used. However, there are many possible ways to sample from a graph. We review a number of possibilities in Section 4. For example, the sampling algorithm might draw a subgraph induced by sampling k vertices at random, or the subgraph induced by a random walk of length k. The challenge is that there is no a priori criterion for deciding which sampling algorithm is "best."

Our approach is to give up and declare victory: we *define* the required analogue as a *component of model*

[3] github.com/wooden-spoon/relational-ERM

design. We require the analyst to choose a randomized sampling algorithm Sample, where $\mathsf{Sample}(\overline{G}_n, k)$ is a random subgraph of size k. The choice of Sample defines a notion of "example." This allows us to complete the analogy to classical ERM.

A *relational ERM model* is defined by three ingredients:

1. A sampling routine Sample.
2. A predictor class $\{\pi_\theta | \theta \in \mathcal{T}\}$ with parameter θ.
3. A loss function L, where $L(\overline{G}_k; \theta)$ measures the reconstruction quality of π_θ on example G_k.

The objective function is defined in analogy with the empirical risk Eq. (1). The *relational empirical risk* is:

$$\hat{R}_k(\pi, \overline{G}_n) := \mathbb{E}_{\overline{G}_k = \mathsf{Sample}(\overline{G}_n, k)}[L(\overline{G}_k; \theta) \mid \overline{G}_n]. \quad (2)$$

Relational empirical risk minimization selects a predictor $\hat{\pi}$ that minimizes the relational empirical risk,

$$\hat{\pi} := \pi_{\hat{\theta}_n} \quad \text{where} \quad \hat{\theta}_n := \operatorname*{argmin}_\theta \hat{R}_k(\pi_\theta, \overline{G}_n) \,. \quad (3)$$

Stochastic gradient descent

A crucial property of relational ERM is that SGD can be applied to solve the minimization problem Eq. (3) without any model specific analysis. Define a stochastic gradient as $\nabla_\theta L(\mathsf{Sample}(G_n, k); \theta)$, the gradient of the loss computed on a sample of size k drawn with Sample. Observe that

$$\begin{aligned} \nabla_\theta \hat{R}_r(\theta, G_n) &= \nabla_\theta \mathbb{E}[L(\mathsf{Sample}(G_n, k); \theta) \mid \overline{G}_n] \\ &= \mathbb{E}[\nabla_\theta L(\mathsf{Sample}(G_n, k); \theta) \mid \overline{G}_n]. \end{aligned}$$

That is, the random gradient $\nabla_\theta L(\mathsf{Sample}(G_n, k); \theta)$ is an unbiased estimator of the gradient of the full relational empirical risk. If Sample is computationally efficient, then SGD with this stochastic estimator can solve the relational ERM.

To specify a relational ERM model in practice, the practitioner implements the three ingredients in code. Machine-learning frameworks provide tools to make it easy to specify a class of predictors and a per-example loss function, which are ingredients of classical ERM. Relational ERM additionally requires implementing Sample and integrating it with a machine-learning framework. In practice, Sample can be chosen from a standard library of sampling routines. To that end, we provide efficient implementations of a number of routines and integration with an automatic differentiation framework (TensorFlow).[4] This gives an effective "plug-and-play" approach for defining and fitting models.

[4] github.com/wooden-spoon/relational-ERM

3 Example Models

We consider several examples of relational ERM models. We split the parameter into a pair $\theta = (\gamma, \lambda)$: the global parameters γ are shared across the entire graph, and the embedding parameters λ provide low-dimensional embeddings λ_v for each vertex v. Informally, global parameters encode population properties—"people with different political affiliation are less likely to be friends"—and the embeddings encode per-vertex information—"Bob is a radical vegan."

Graph representation learning

Methods for learning embeddings of vertices are widely studied; see [10] for a review. Many such methods rely on decomposing the graph into neighborhoods determined by (random) walks of fixed size. One example is Node2Vec [8] (an extension of DeepWalk [19]). The basic approach is to draw a large collection of simple random walks, view each of these walks as a "sentence" where each vertex is a "word", and learn vertex embeddings by applying a standard word embedding method [16, 15]. To use mini-batch SGD, the objective function is restricted to a uniform sum over all walks. Unbiased stochastic gradients to be computed by uniformly sampling walks.

Relational ERM models include graph representation models of this kind. For example, Node2Vec [8] is equivalent to a relational ERM model that (i) predicts graph structure using a predictor parameterized only by embedding vectors, (ii) uses a cross-entropy loss on graph structure, and (iii) takes Sample as a random-walk of fixed length (augmented with randomly sampled negative examples).

A number of other relational learning methods also enable SGD by restricting the objective function to a uniform sum over fixed-size subgraphs [e.g., 8, 3, 29, 9]. Any such model is equivalent to a relational ERM model that takes Sample as the uniform distribution over fixed-size subgraphs. But, in general, relational ERM does not require restricting to sampling schemes of this kind. Note that "negative-sampling" algorithms—which are critical in practice—do not uniformly sample fixed size subgraphs.

The next examples illustrate relational ERM for problems that are difficult with existing approaches to graph representation learning.

Semi-supervised node classification

Consider a network G_n where each node i is labeled by binary features—for example, hyperlinked documents labeled by subjects, or interacting proteins labeled by function. The task is to predict the labels of a subset

of these nodes using the graph structure and the labels of the remaining nodes.

The model has the following form: Each vertex i is assigned a k-dimensional embedding vector $\lambda_i \in \mathbb{R}^k$. Labels are predicted using a parameterized function $f(\bullet;\gamma) : \mathbb{R}^k \to [0,1]^L$ that maps the node embeddings to the probability of each label. The presence or absence of edge i,j is predicted based on $\lambda_i^T \lambda_j$. This enables learning embeddings for unlabeled vertices. Let σ denote the sigmoid function; let label $l_{ij} \in \{0,1\}$ denote whether vertex i has label j; and let $q \in [0,1]$. The loss on subgraphs $G_k \subset G_n$ is:

$$L(G_k;\lambda,\gamma,l) = \tag{4}$$
$$q\Big(\sum_{i\in v(G_k)}\sum_{j=1}^{L} l_{ij}\log f(\lambda_i;\gamma)_j + (1-l_{ij})\log(1-f(\lambda_i;\gamma)_j)\Big)$$
$$+(1-q)\Big(-\sum_{i,j\in e(G_k)}\log\sigma(\lambda_j^T\lambda_i) - \sum_{i,j\in\bar{e}(G_k)}\log(1-\sigma(\lambda_j^T\lambda_i))\Big).$$

Here, v, e, and $\bar{e}$ denote the vertices, edges, and non-edges of the graph respectively. The loss on edge terms is cross-entropy, a standard choice in embedding models [10]. Intuitively, the predictor uses the embeddings to predict both the vertex labels and the subgraph structure.

The model is completed by choosing a sampling scheme Sample. Relational ERM then fits the parameters as

$$(\hat{\lambda}_n, \hat{\gamma}_n) \;=\; \operatorname*{argmin}_{\lambda,\gamma} \mathbb{E}[L\big(\lambda,\gamma;\; \mathsf{Sample}(G_n,k), l\big) \mid G_n].$$

We can vary the choice of Sample independent of optimization concerns; in Section 6 we observe that this leads to improved predictive performance.

Older embedding approaches use a two-stage procedure: node embeddings are first pre-trained using the graph structure, and then used as inputs to a logistic regression that predicts the labels [e.g., 19, 8]. Yang, Cohen, and Salakhudinov [29] adapt a random-walk based method to allow simultaneous training; their approach requires extensive development, including a custom (two-stage) variant of SGD. Relational ERM allows simultaneous learning with no need for model specific derivation.

Wikipedia category embeddings

We consider Wikipedia articles joined by hyperlinks. Each article is tagged as a member of one or more categories—for example, "`Muscles_of_the_head_and_neck`", "`Japanese_rock_music_groups`", or "`People_from_Worcester`." The task is to learn embeddings that encode semantic relationships between the categories.

Let G_n denote the hyperlink graph and let $\mathcal{C}(i)$ denote the categories of article i. Each category $c \in C$ is assigned an embedding γ_c, and the embedding of each article (vertex) is taken to be the sum of the embeddings of its categories, $\lambda_i \coloneqq \sum_{c\in\mathcal{C}(i)} \gamma_c$. The loss is

$$L(G_k, C; \lambda) = \tag{5}$$
$$-\sum_{i,j\in e(G_k)} \log\sigma(\lambda_j^T\lambda_i) \;-\; \sum_{i,j\in\bar{e}(G_k)} \log(1-\sigma(\lambda_j^T\lambda_i))\,,$$

where e and $\bar{e}$ denote, respectively, the presence and absence of hyperlinks between articles. Intuitively, the predictor uses the category embeddings to predict the hyperlink structure of subgraphs. Relational ERM chooses the embeddings as

$$\hat{\gamma}_n = \operatorname*{argmin}_{\gamma} \mathbb{E}[L\big(\lambda(\gamma);\; \mathsf{Sample}(G_n,k), C\big) \mid G_n]\,.$$

We write $\lambda(\gamma)$ to emphasize that the article embeddings are a function of the category embeddings. Category embeddings obtained with this model are illustrated in Fig. 1; see Section 6 for details on the experiment.

The point of this example is: relational ERM makes it easy to implement this non-standard relational learning model and fit it with mini-batch SGD. The use of mini-batch SGD is important because the data graph is large.

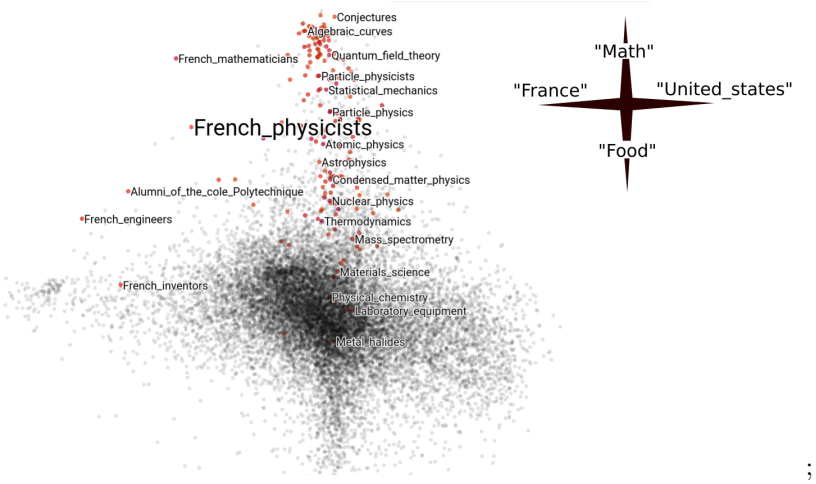
Statistical relational learning

Statistical relational learning takes the graph to encode the dependency structure between the units [17, 6, e.g.]. The idea is to infer a joint probability distribution over the entire dataset, respecting the dependency structure. The distribution can then be used to make graph-aware predictions. There is also work on adapting SGD to this setting [28]. Despite the similar goals, Relational ERM does not attempt to infer a distribution; the precise relationship with statistical relational learning is not clear.

4 Subsampling algorithms

In classical ERM, sampling uniformly (with or without replacement) is typically the only choice. In contrast, there are many ways to sample from a graph. Each such sampling algorithm Sample leads to a different notion of empirical risk in (2).

As described above, random walks underlie graph representation methods built in analogy with language models. A simple random walk of length k on a graph G_n selects vertices $v_1, \ldots, v_k$ by starting at a given vertex v_1, and drawing each vertex v_{i+1} uniformly from the neighbors of v_i. Typically, random-walk based methods



;

Figure 1: Trained Wikipedia category embeddings. Category embeddings are projected into a 2-dimensional space, with a projection chosen to maximally separate "`France`" and "`United_states`" horizontally, and "`Math`" and "`Food`" vertically. Highlighted categories are nearest neighbors of "`French_physicists`."

augment the sample by hallucinating additional edges using a strategy borrowed from the Skipgram model [16]:

Algorithm 1 (Random walk: Skipgram [19])**.**

(i) Sample a random walk $v_1, \dots, v_k$ starting at a uniformly selected vertex of $\overline{G}_n$.

(ii) Report $\overline{G}_k = \{(v_i, v_j) : d(v_i, v_j) < W\}$. The *window* W is a sampler parameter, and $d(v_i, v_j)$ is the number of steps between v_i and v_j.

Since relational ERM is indifferent to the connection with language models, a natural alternative augmentation strategy is:

Algorithm 2 (Random walk: Induced)**.**

(i) Sample a random walk $v_1, \dots, v_k$ starting at a uniformly selected vertex of $\overline{G}_n$.

(ii) Report $\overline{G}_k$ as the edge list of the vertex induced subgraph of the walk.

A simple choice is to sample k vertices uniformly at random and report $\overline{G}_k$ as the induced subgraph. Such an algorithm will not work well in practice since it is not suitable for sparse graphs. We are typically interested in the case $k \ll n$. If $\overline{G}_n$ is sparse then such a sample typically includes few or no edges, and thus carries little information about $\overline{G}_n$. The next algorithm modifies uniform vertex sampling to fix this pathology. The idea is to over-sample vertices and retain only those vertices that participate in at least one edge in the induced subgraph.

Algorithm 3 (p-sampling [27])**.**

(i) Select each vertex in $\overline{G}_n$ independently, with a fixed probability $p \in [0, 1]$.

(ii) Extract the induced subgraph $\overline{G}_k$ of $\overline{G}_n$ on the selected vertices.

(iii) Delete all isolated vertices from $\overline{G}_k$, and report the resulting graph.

Another natural sampling scheme is:

Algorithm 4 (Uniform edge sampling)**.**

(i) Select k edges in $\overline{G}_n$ uniformly and independently from the edge set.

(ii) Report the graph $\overline{G}_k$ consisting of these edges, and all vertices incident to these edges.

Many other sampling schemes are possible; see Leskovec and Faloutsos [14] for a discussion of possible options in a related context.

4.1 Negative sampling

For a pair of vertices in an input graph $\overline{G}_n$, a sampling algorithm can report three types of edge information: The edge may be observed as present, observed as absent (a *non-edge*), or may not be observed. The algorithms above do not treat edge and non-edge information equally: Algorithms 1, 2 and 4 cannot report non-edges, and the deletion step in Algorithm 3 biases it towards edges over non-edges. However, the locations of non-edges can carry significant information.

Negative sampling schemes are "add-on" algorithms

that are applied to the output of a graph sampling algorithm and augment it by non-edge information. Let $\overline{G}_k$ denote a sample generated by one of the algorithms above from an input graph $\overline{G}_n$.

Algorithm A (Negative sampling: Induced)**.**

(i) Report the subgraph induced by $\overline{G}_k$, in the input graph $\overline{G}_n$ from which $\overline{G}_k$ was drawn.

Another method, originating in language modeling [15, 7], is based on the unigram distribution: Define a probability distribution on the vertex set of $\overline{G}_k$ by $P_n(v) := \text{Prob}\{v \in \overline{H}_k\}$, the probability that v would occur in a separate, independent sample $\overline{H}_k$ generated from $\overline{G}_n$ by the same algorithm as $\overline{G}_k$. For $\tau > 0$, we define a distribution $P_n^\tau(v) := (P_n(v))^\tau / Z(\tau)$, where $Z(\tau)$ is the appropriate normalization.

Algorithm B (Negative sampling: Unigram)**.** For each vertex v in $\overline{G}_n$:

(i) Select k vertices $v_1, \ldots, v_k \overset{iid}{\sim} P_n^\tau$.

(ii) If (v, v_j) is a non-edge in $\overline{G}_n$, add it to $\overline{G}_n$.

The canonical choice in the embeddings literature is $\tau = \frac{3}{4}$ [15].

5 Theory

We now turn to formalizing and establishing theoretical properties of relational ERM. Particularly, (i) relational ERM satisfies basic theoretical desiderata, and (ii) Sample should be viewed as a model component. We first give the results, and then discuss their interpretation and significance.

When the data is unstructured (i.e., no link structure), theoretical analysis of ERM relies on the assumption that the data is generated i.i.d. The i.i.d. assumption is ill-defined for relational data. Any analysis requires some analogous assumption for how the data $\overline{G}_n$ is generated. Following recent work emphasizing the role of sampling theory in modeling graph data [18, 27, 1, 5], we model $\overline{G}_n$ as a random sample drawn from some large population network. Specifically, we consider a population graph $\mathcal{G}$ with $|\mathcal{G}|$ edges, and assume that the observed sample $\overline{G}_n$ of size n is generated by p-sampling from $\mathcal{G}$, with $p = n/\sqrt{|\mathcal{G}|}$. We assume the population graph is "very large," in the sense that $|\mathcal{G}| \to \infty$. The distribution of $\overline{G}_n$ in the "infinite population" case is well-defined [1].

The analogy with i.i.d. data generation is two-fold: Foundationally, the i.i.d. assumption is equivalent to assuming the data is collected by uniform sampling from some population [20], and p-sampling is a direct analogue [27, 1, 18]. Pragmatically, both assumptions strike a balance between being flexible enough to capture real-world data [2, 26] and simple enough to allow precise theoretical statements.

We establish results for several choices of $\mathsf{Sample}(G_n, k)$. Edges may be selected by either p-sampling with $p = k/\sqrt{n}$—note the size of $\mathsf{Sample}(G_n, k)$ is free of n—or by using a simple random walk of length k (Algorithm 1 or Algorithm 2). Negative examples may be chosen by Algorithm A or Algorithm B.

The main result guarantees that the limiting risk of the parameter we learn depends only on the population and the model, and not on idiosyncrasies of the training data.

Theorem 5.1. *Suppose that G_n is collected by p-sampling as described above, that $k \in \mathbb{N}$ is fixed, and that* Sample *is fixed to a sampling algorithm based on either p-sampling or random walk sampling as described above. Suppose further that the loss is bounded and parameter setting $\bar{\theta} = (\bar{\gamma}, \bar{\lambda})$ satisfies mild technical conditions given in the appendix. Then there is some constant $c_{\bar{\theta}}(\mathsf{Sample}, k) \in \mathbb{R}_+$ such that*

$$\hat{R}_k(\bar{\theta}; \overline{G}_n) \to c_{\bar{\theta}}(\mathsf{Sample}, k) \tag{6}$$

both in probability and in L_1 as $n \to \infty$. Moreover, there is some constant $c_*(\mathsf{Sample}, k) \in \mathbb{R}_+$ such that

$$\min_\theta \hat{R}_k(\theta; \overline{G}_n) \to c_*(\mathsf{Sample}, k) \tag{7}$$

both in probability and in L_1, as $n \to \infty$.

The limits depend on the choice of Sample *(and k), and usually do not agree between different sampling schemes.*

The result is proved for Sample based on p-sampling in supplement C and for random-walk based sampling in supplement D.

Classical ERM guarantees usually apply even to the parameter itself, not just its risk. In the relational setting, the possibly complicated interplay of the learned embeddings makes such results more difficult. The next two results build on Theorem 5.1 to establish (partial) guarantees for the parameter itself.

We establish a convergence result for the global parameters output by a two-stage procedure where the embedding vectors are learned first. Such a result is applicable, for example, when predicting vertex attributes from embedding vectors that are pre-trained to explain graph structure. The proof is given in supplement E.

Theorem 5.2. *Suppose the conditions of Theorem 5.1, and also that the loss function verifies a certain strong convexity property in γ, given explicitly in the appendix. Let $\tilde{\gamma}_n = \text{argmin}_\gamma \min_\lambda \hat{R}_k(\gamma, \lambda; \overline{G}_n)$. Then*

$\tilde{\gamma}_n \to \tilde{\gamma}_*(\mathsf{Sample}, k)$ *in probability for some constant* $\tilde{\gamma}_*(\mathsf{Sample}, k)$.

We next establish a stability result showing that collecting additional data does not dramatically change learned embeddings. The proof is given in supplement F.

Theorem 5.3. *Suppose the conditions of Theorem 5.1, and also that the loss function is twice differentiable and the Hessian of the empirical risk is bounded. Let* $\hat{\lambda}_{n+1}|_n$ *denote the restriction of the embeddings* $\hat{\lambda}_{n+1}$ *to the vertices present in* G_n. *Then* $\hat{\lambda}_n - \hat{\lambda}_{n+1}|_n \to 0$ *in probability, as* $n \to \infty$.

The examples of Section 3 do not satisfy the conditions of the theorem because the cross-entropy loss is unbounded. However, the models can be trivially modified to bound the output probabilities away from 0 and 1. In this case, the loss is bounded. Further, for the logistic regression model used in the experiments the convexity and Hessian conditions also hold, by direct computation.

Interpretation and Significance

The properties we establish are minimal desiderata that one might demand of any sensible learning procedure. Nevertheless, such results have not been previously established for relational learning methods. The obstruction is the need for a suitable analogue of the i.i.d. assumption. The demonstration that population sampling can fill this role is itself a main contribution of the paper. Indeed, the results we establish are weaker than the analogous guarantees for classical ERM, and main significance is perhaps the demonstration that such results can be established at all. This is important both as a foundational step towards a full theoretical analysis of relational learning, and because it strengthens the analogy with classical ERM.

A strength of our arguments is that they are largely agnostic to the particular choice of model, mitigating the need for model-specific analysis and justification. For example, our results include random-walk based graph representation methods as a special case, providing some post-hoc support for the use of such methods.

The limits in Theorems 5.1 and 5.2 depend on the choice of Sample. Accordingly, the limiting risk and learned parameters depend on Sample in the same sense they depend on the choice of predictor class and the loss function; i.e., Sample is a model component. This underscores the need to consider the choice in model design, either through heuristics—e.g., random-walk sampling upweights the importance of high degree vertices relative to p-sampling—or by trying several choices experimentally.

6 Experiments

The practical advantages of using relational ERM to define new, task-specific, models are: (i) Mini-batch SGD can be used in a plug-and-play fashion to solve the optimization problem. This allows inference to scale to large data. And, (ii) by varying Sample we may improve model quality. We have used relational ERM to define novel models in Section 3. The models are determined by (4) and (5) up to the choice of Sample. We now study these example models empirically.[5] The main observations are: (i) SGD succeeds in quickly fitting the models in all cases. And, (ii) the choice of Sample has a dramatic effect in practice. Additionally, we observe that the best model for the semi-supervised node classification task uses p-sampling. p-sampling has not previously been used in the embedding literature, and is very different from the random-walk based schemes that are commonly used.

Node classification problems

We begin with the semi-supervised node classification task described in Section 3, using the model Eq. (4) with different choices of Sample. We study the blog catalog and protein-protein interaction data reported in [8], summarized by the table to the right. We pre-process the data to remove self-edges, and restrict each network to the largest connected component. Each vertex in the graph is labeled, and 50% of the labels are censored at training time. The task is to predict these labels at test time.

	Blogs	Protein
Vertices	10,312	3,890
Edges	333,983	76,584
Label Dim.	39	50

Table 1: Average Macro-F1 for Two-Stage Training.

Choice of Sample	Alg. #	Blogs	Protein
rw/skipgram+ns	1+B	0.18	0.16
rw/induced+ind	2+A	0.08	0.08
rw/induced+ns	2+B	0.18	0.16
p-samp+ind.	3+A	0.17	0.14
p-samp+ns	3+B	0.22	0.16
unif. edge+ns	4+B	0.21	0.15

Two-stage training. We first train the model (4) using no label information to learn the embeddings (that is, with $q = 0$). We then fit a logistic regression to predict vertex features from the trained embeddings.

[5] Code at github.com/wooden-spoon/relational-ERM

Table 2: Average Macro-F1 for Simultaneous Training. Columns are labeled by the sampling scheme used to draw test vertices.

	Blog catalog			Protein-Protein		
Sample	Unif.	p-samp	rw	Unif.	p-samp	rw
rw/skipgram+ns (Alg. 1+B)	0.20	0.26	0.27	0.25	0.32	0.34
p-samp+ns (Alg. 3+B)	0.30	0.34	0.35	0.30	0.37	0.39
Node2Vec (reported)	0.26	-	-	0.18	-	-

This two stage approach is a standard testing procedure in the graph embedding literature, e.g. [19, 8]. We use the same scoring procedure as Node2Vec [8] (average macro F1 scores), and, where applicable, the same hyperparameters.

Table 1 shows the effect of varying the sampling scheme used to train the embeddings. As expected, we observe that the choice of sampling scheme affects the embeddings produced via the learning procedure, and thus also the outcome of the experiment. We further observe that sampling non-edges by unigram negative sampling gives better predictive performance relative to selecting non-edges from the vertex induced subgraph.

Simultaneous training. Next, we fit the model of Section 3 with $q = 0.001$—training the embeddings and global variables simultaneously. Recall that simultaneous training is enabled by the use of relational ERM. We choose label predictor π_γ as logistic regression, and adapt the label prediction loss to measure the loss only on vertices in the positive sample.

There is not a unique procedure for creating a test set for relational data. We report test scores for test-sets drawn according to several different sampling schemes. Results are summarized by Table 2. We observe:

- Simultaneous training improves performance.
- p-sampling outperforms the standard rw/skipgram procedure.
- This persists irrespective of how the test set is selected (i.e., it is not an artifact of the data splitting procedure).

Note that the average computed with uniform vertex sampling is the standard scoring procedure used in the previous table. The last observation is somewhat surprising: we might have expected a mismatch between the training and testing objectives to degrade performance. One possible explanation is that the random-walk based sampler excessively downweights low-connectivity vertices, and thus fails to fully exploit their label information.

Wikipedia Category Embeddings

We consider the task of discovering semantic relations between Wikipedia categories, as described in Section 3. This task is not standard; wholly new model is required.

We define a relational ERM model by choosing category embedding dimension $k = 128$, the loss function L in (5), and Sample as 1+B, the skipgram random walk sampler with unigram negative sampling. The data $\overline{G}_n$ is the Wikipedia hyperlink network from [13], consisting of Wikipedia articles from 2011-09-01 restricted to articles in categories containing at least 100 articles.

The challenge for this task is that the dataset is relatively large—about 1.8M nodes and 28M edges—and the model is unusual—embeddings are assigned to vertex attributes instead of the vertices themselves. SGD converges in about 90 minutes on a desktop computer equipped with a Nvidia Titan Xp GPU. Fig. 1 on page 5 visualizes example trained embeddings, which clearly succeed in capturing latent semantic structure.

7 Conclusion

Relational ERM is a generalization of ERM from i.i.d. data to relational data. The key ideas are introducing Sample as a component of model design, which defines an analogue of the empirical distribution, and using the assumption that the data is sampled from a population network as an analogue of the i.i.d. assumption. Relational ERM models can be fit automatically using SGD. Accordingly, relational ERM provides an easy method to specify and fit relational data models.

The results presented here suggest a number of directions for future inquiry. Foremost: what is the relational analogue of statistical learning theory? The theory derived in Section 5 establishes initial results. A more complete treatment may provide statistical guidelines for model development. Our results hinge critically on the assumption that the data is collected by p-sampling; it is natural to ask how other data-generating mechanisms can be accommodated. Similarly, it is natural to ask for guidelines for the choice of Sample.

Acknowledgments

VV and PO were supported in part by grant FA9550-15-1-0074 of AFOSR. DB is supported by ONR N00014-15-1-2209, ONR 133691-5102004, NIH 5100481-5500001084, NSF CCF-1740833, the Alfred P. Sloan Foundation, the John Simon Guggenheim Foundation, Facebook, Amazon, and IBM. The Titan Xp used for this research was donated by the NVIDIA Corporation.

References

[1] C. Borgs, J. T. Chayes, H. Cohn, and V. Veitch. *Sampling perspectives on sparse exchangeable graphs*. 2017. arXiv: 1708.03237.

[2] F. Caron and E. B. Fox. "Sparse graphs using exchangeable random measures". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 79.5 (2017), pp. 1295–1366. eprint: https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/rssb.12233.

[3] B. P. Chamberlain, J. Clough, and M. P. Deisenroth. "Neural Embeddings of Graphs in Hyperbolic Space". In: *ArXiv e-prints* (May 2017). arXiv: 1705.10359 [stat.ML].

[4] J. Chen, J. Zhu, and L. Song. "Stochastic Training of Graph Convolutional Networks with Variance Reduction". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by J. Dy and A. Krause. Vol. 80. Proceedings of Machine Learning Research. Stockholmsmässan, Stockholm Sweden: PMLR, 2018, pp. 942–950.

[5] H. Crane and W. Dempsey. *A framework for statistical network modeling*. 2015. arXiv: 1509.08185.

[6] L. Getoor and B. Taskar. *Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2007. ISBN: 0262072882.

[7] Y. Goldberg and O. Levy. *word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method*. 2014. arXiv: 1402.3722.

[8] A. Grover and J. Leskovec. "Node2Vec: Scalable Feature Learning for Networks". In: *Proc. 22nd Int. Conference on Knowledge Discovery and Data Mining (KDD '16)*. ACM, 2016, pp. 855–864.

[9] W. L. Hamilton, R. Ying, and J. Leskovec. *Inductive Representation Learning on Large Graphs*. June 2017. arXiv: 1706.02216.

[10] W. L. Hamilton, R. Ying, and J. Leskovec. *Representation Learning on Graphs: Methods and Applications*. 2017. arXiv: 1709.05584.

[11] T. N. Kipf and M. Welling. "Variational Graph Auto-Encoders". In: *ArXiv e-prints* (Nov. 2016). arXiv: 1611.07308 [stat.ML].

[12] T. N. Kipf and M. Welling. "Semi-Supervised Classification with Graph Convolutional Networks". In: *ICLR*. 2017.

[13] C. Klymko, D. Gleich, and T. G. Kolda. *Using Triangles to Improve Community Detection in Directed Networks*. 2014. arXiv: 1404.5874.

[14] J. Leskovec and C. Faloutsos. "Sampling from Large Graphs". In: *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '06. Philadelphia, PA, USA: ACM, 2006, pp. 631–636. ISBN: 1-59593-339-5.

[15] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. *Distributed Representations of Words and Phrases and their Compositionality*. 2013. arXiv: 1310.4546.

[16] T. Mikolov, K. Chen, G. Corrado, and J. Dean. *Efficient Estimation of Word Representations in Vector Space*. 2013. arXiv: 1301.3781.

[17] J. Neville and D. Jensen. "Relational Dependency Networks". In: *J. Mach. Learn. Res.* 8 (May 2007), pp. 653–692. ISSN: 1532-4435.

[18] P. Orbanz. *Subsampling large graphs and invariance in networks*. 2017. arXiv: 1710.04217.

[19] B. Perozzi, R. Al-Rfou, and S. Skiena. "DeepWalk: Online Learning of Social Representations". In: *Proc. 20th Int. Conference on Knowledge Discovery and Data Mining (KDD '14)*. ACM, 2014, pp. 701–710.

[20] D. N. Politis, J. P. Romano, and M. Wolf. *Subsampling*. Springer, 1999.

[21] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling. "Modeling Relational Data with Graph Convolutional Networks". In: *The Semantic Web*. Ed. by A. Gangemi, R. Navigli, M.-E. Vidal, P. Hitzler, R. Troncy, L. Hollink, A. Tordai, and M. Alam. Cham: Springer International Publishing, 2018, pp. 593–607. ISBN: 978-3-319-93417-4.

[22] S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[23] R. van den Berg, T. N. Kipf, and M. Welling. "Graph Convolutional Matrix Completion". In: *ArXiv e-prints* (June 2017). arXiv: 1706.02263 [stat.ML].

[24] V. Vapnik. "Principles of Risk Minimization for Learning Theory". In: *Advances in Neural Information Processing Systems 4*. 1992, pp. 831–838.

[25] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

[26] V. Veitch and D. M. Roy. *The Class of Random Graphs Arising from Exchangeable Random Measures*. Dec. 2015. arXiv: 1512.03099.

[27] V. Veitch and D. M. Roy. "Sampling and Estimation for (Sparse) Exchangeable Graphs". In: (2016). arXiv: 1611.00843.

[28] J. Yang, B. F. Ribeiro, and J. Neville. "Stochastic Gradient Descent for Relational Logistic Regression via Partial Network Crawls". In: *CoRR* abs/1707.07716 (2017).

[29] Z. Yang, W. Cohen, and R. Salakhudinov. "Revisiting Semi-Supervised Learning with Graph Embeddings". In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by M. F. Balcan and K. Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, 2016, pp. 40–48.