Group-Sparse Matrix Factorization for Transfer Learning of Word Embeddings

Kan Xu[1] **Xuanyi Zhao**[1] **Hamsa Bastani**[1] **Osbert Bastani**[1]

Abstract

Sparse regression has recently been applied to enable transfer learning from very limited data. We study an extension of this approach to unsupervised learning—in particular, learning word embeddings from unstructured text corpora using low-rank matrix factorization. Intuitively, when transferring word embeddings to a new domain, we expect that the embeddings change for only a small number of words—e.g., the ones with novel meanings in that domain. We propose a novel group-sparse penalty that exploits this sparsity to perform transfer learning when there is very little text data available in the target domain—e.g., a single article of text. We prove generalization bounds for our algorithm. Furthermore, we empirically evaluate its effectiveness, both in terms of prediction accuracy in downstream tasks as well as the interpretability of the results.[1]

1. Introduction

While machine learning algorithms have proven to be tremendously effective at solving supervised and unsupervised problems, achieving good performance typically requires large training datasets. Yet, in many domains, there is very little data available for training. Thus, there has been a great deal of interest in transfer learning, where the goal is to leverage knowledge in a data-rich source domain to improve performance in a data-poor target domain.

A surprisingly effective transfer learning strategy is to simply fine-tune a model trained on data from the source domain (which we call the *proxy data*) on data from the target domain (which we call the *gold data*). For instance, this strategy has been used in transferring image classification models (Esteva et al., 2017), healthcare decision-making (Bastani, 2020), and word embeddings (Dingwall & Potts, 2018). Intuitively, stochastic gradient descent has regularization properties similar to ℓ_2 regularization (Ali et al., 2020), so this strategy can be interpreted as regularizing the parameters towards those of the proxy model—i.e., adopting a loss function of the form

$$\widetilde{\ell}(\theta; X_g, \widehat{\theta}_p) = \ell(\theta; X_g) + \|\theta - \widehat{\theta}_p\|_2^2,$$

where X_g is the gold training data, $\ell(\theta; X_g)$ is the unregularized loss, and $\widehat{\theta}_p$ are the parameters estimated using the proxy training data.

With this viewpoint, a natural strategy is to leverage alternative regularization strategies towards the source domain, instead of the ℓ_2 norm. Recent work has investigated using the ℓ_1 norm in the setting of (generalized) linear regression (Bastani, 2020)—i.e.,

$$\widetilde{\ell}(\theta; X_g, Y_g, \widehat{\theta}_p) = \|Y_g - X_g\theta\|_2^2 + \lambda \cdot \|\theta - \widehat{\theta}_p\|_1.$$

Intuitively, ℓ_1 regularization enables efficient learning in domains with very little data (Tibshirani, 1996; Chen et al., 1995; Candes & Tao, 2007; Bickel et al., 2009). The key assumption for this approach to work is that the values of the true parameters θ_g (for the target domain) must differ from θ_p (for the source domain) in only a few components—i.e., $\theta_g - \theta_p$ is sparse. If this assumption holds, then they prove that their strategy can learn from $\mathcal{O}(s \log d)$ samples instead of $\mathcal{O}(d)$ samples, where d is the dimension of θ_g and $s = \|\theta_g - \theta_p\|_0$ is the sparsity. A natural question is whether these techniques can be leveraged beyond the setting of generalized linear regression.

In this paper, we apply this approach to matrix factorization, which underlies one of the most basic unsupervised learning algorithms—namely, learning word embeddings from large-scale unlabeled text corpora such as Wikipedia (Pennington et al., 2014). While more sophisticated techniques have been developed (Devlin et al., 2018), approaches based on generalizations of matrix factorization remain competitive and widely used, and also tend to be more interpretable since we can visualize vector embeddings of individual words.

[1] University of Pennsylvania, Pennsylvania, USA. Correspondence to: Kan Xu <kanxu@sas.upenn.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

[1] We encourage readers to consult our extended paper at https://arxiv.org/abs/2104.08928, which includes proofs and additional results. Source code is available at https://github.com/kanxu526/GroupTLWordEmbedding.

The key question is identifying a notion of sparsity that we can leverage in this setting. Intuitively, we expect that only a small number of words in the target domain may change meaning compared to the source domain. For instance, in computer science, the term "object" (as in "object oriented") has a very different meaning than the usual English definition. Thus, we might expect very few word embeddings to change from the source domain to the target domain. More formally, let U_p^* denote the proxy word embedding matrix, whose i^{th} row U_p^{*i} is the true word embedding of word $i \in [d] = \{1, \cdots, d\}$ based on the proxy data; analogously, U_g^* denotes the gold word embedding matrix. Then, we expect that the word embeddings for most words are equal in both domains—i.e., $U_g^{*i} = U_p^{*i}$ for most $i \in [d]$.

Based on this intuition, we formulate an objective that encodes a *group-sparse* penalty (Friedman et al., 2010; Simon et al., 2013), where each row is a group. Intuitively, a group sparse penalty partitions the parameters into groups, and encodes that only a small number of groups contain non-zero parameters; it does so by encoding an ℓ_1 norm over the ℓ_2 norm of each group. We propose a two-stage estimator that uses this penalty to solve the transfer learning problem. The first stage estimates the proxy word embeddings using only proxy data. Then, the second stage estimates the word embeddings of the gold data using $\ell_{2,1}$ regularization to impose group sparsity compared to the proxy word embeddings:

$$\|U_g - \widehat{U}_p\|_{2,1} = \sum_{i=1}^{d} \|U_g^i - \widehat{U}_p^i\|_2,$$

where $\widehat{U}_p$ is the estimated proxy word embedding matrix.

We prove sample complexity bounds for our estimator, demonstrating how it can substantially improve the quality of the word embeddings for the gold data. In particular, assuming that most word vectors are preserved between the source and the target domains, we show that our estimator requires exponentially less gold data to achieve the same accuracy compared to using the gold data alone. Our proof relies on a tail inequality for the group lasso (Lounici et al., 2011), combined with an error bound for low-rank matrix problems (Ge et al., 2017).

While our main results are for word embeddings trained using matrix factorization, we show that our approach also applies to nonlinear extensions of matrix factorization. In particular, we show how our group sparse penalty term can be leveraged in conjunction with the GloVe word embedding objective (Pennington et al., 2014).

We evaluate our approach to learn word embeddings for Wikipedia articles in domains such as finance, math, computing etc. In particular, we demonstrate that our approach identifies words with novel meanings in this domain at a high rate. These results demonstrate the interpretability of word embeddings learned using our algorithm. Finally, we demonstrate the efficacy of our approach in a downstream task where the goal is to predict clinical trial eligibility based on unstructured clinical statements regarding inclusion or exclusion criteria.

Related work. Typically, transfer learning refers to learning with a large amount of data in the source domain, and a small amount of data in the target domain. Broadly speaking, the two domains must be connected in some way: they can either have the same covariate distribution $p(x)$ but different label distributions $p(y \mid x)$ and $q(y \mid x)$ (called *label shift*), or vice versa (called *covariate shift*). Approaches targeting the latter setting are typically referred to as *domain adaptation*.

Recently, Bastani (2020) applied sparse regression to handle label shift, when the shift is sparse—i.e., $y = x^T\beta_p + \epsilon$ vs. $y = x^T\beta_g + \epsilon$, where $\|\beta_p - \beta_g\|_0$ is much smaller than the dimension of β_p, β_g; here p refers to *proxy data* from the source domain, and g refers to *gold data* from the target domain. When the parameters β_g are sparse (i.e., $s = \|\beta_g\|_0$ is small), existing theory shows that the sample complexity of estimating β_g is $\mathcal{O}(s \log d)$ instead of $\mathcal{O}(d)$, where d is the dimension of β_g (Bühlmann & Van De Geer, 2011). Their key theoretical result is that the sample complexity of estimating β_g scales as s even though β_g itself may not be sparse. Instead, *relative sparsity* between β_g and β_p is sufficient to enable efficient transfer learning in high dimensions. A key limitation of their work is that they are limited to the supervised learning setting. Our motivation is to study the sample complexity of transfer learning in settings beyond supervised learning.

Given multiple proxy datasets as well as their "disparities" from the source domain, Crammer et al. (2008) study which proxy sources to use in supervised learning to minimize generalization error. Zhang et al. (2013) propose importance reweighting and sample transformation to correct the data distribution shift, and Ganin & Lempitsky (2015) add a domain classifier into their deep feed-forward neural network framework to fine-tune the source model. More relatedly, there has been work proving generalization bounds for unsupervised domain adaptation (Ben-David et al., 2007; 2010); unlike our work, they assume a large number of unlabeled examples from the target domain.

For word embedding models, a standard approach is to fine-tune the pre-trained word embeddings end-to-end. A closely related approach is to add an ℓ_2 penalty to the objective to regularize the word embeddings towards the existing ones Dingwall & Potts (2018); Yang et al. (2017). Other approach combines domain-specific word embeddings with pre-trained ones through Canonical Correlation Analysis (CCA) or the related kernelized version (KCCA) (Sarma et al., 2018). However, these approaches do not provide theoretical guarantees on their performance. In contrast, we prove theoretical bounds on the performance of our estimator under sparsity assumptions motivated by the domain distinction of word embeddings. We show that in the very low-data regime (e.g., a single article), using ℓ_1 regularization outperforms ℓ_2 regularization.

Another approach is to train *contextual embeddings* that capture different meanings of the same word based on their context (Devlin et al., 2018). Assuming the training corpus contains some data that covers the target domain, then one can automatically tailor word embeddings based on the given context. However, such techniques lack the interpretability of traditional word embeddings, since we cannot examine or visualize the embedding of a single word in isolation. Also, they may not work when the training corpus altogether omits content from the target domain.

We build on approaches to word embeddings based on low-rank matrix factorization. Given a few observations X_i about a matrix $\Theta \in \mathbb{R}^{d_1 \times d_2}$ with rank r, the goal is to compute a low-rank estimate $\widehat{\Theta}$. Recent work has provided an algorithm based on nuclear norm regularization, and proves a bound $\|\widehat{\Theta} - \Theta\|_F = \mathcal{O}(\sqrt{d/n})$ on the estimation error (Negahban & Wainwright, 2011). A more practical algorithm is the Burer-Monteiro approach (Burer & Monteiro, 2003), which replaces Θ explicitly with a low-rank representation UV^T, with $U \in \mathbb{R}^{d_1 \times r}$ and $V \in \mathbb{R}^{d_2 \times r}$, and minimizes the objective in terms of U and V. This approach is nonconvex but is simpler to implement and computationally efficient. Ge et al. (2017) show that the local minima of this nonconvex problem are also global minima under the restricted isometry property (RIP).

A simple way to construct word embeddings is to take Θ to encode the relationships between words (e.g., the co-occurrence matrix, in which Θ_{ij} counts how many times words i and j occur together in a fixed-length window), run low-rank matrix factorization to compute $UV^T \approx \Theta$, and then choose the i^{th} row of U to be the embedding of word i. Levy & Goldberg (2014) shows that skip-gram with negative sampling implicitly factorizes a word-context matrix, described by pointwise mutual information (PMI) matrix. GloVe (Pennington et al., 2014), which can be thought of as a nonlinear version of this approach, was a state-of-the-art technique until recently. We show how our approach can be extended to GloVe, although our theoretical guarantees only hold for the linear setting. Recently, contextual embeddings have been shown to outperform GloVe. However, they assign vectors to sequences of words, not to individual words, making them less widely applicable as well as less interpretable.

2. Problem Formulation

In this section, we formalize the problem of group sparse transfer learning for word embeddings. We begin by giving background on matrix factorization, and then formalize the transfer learning problem along with our assumptions on the group sparse structure of the word embeddings.

Notation. For any vector $v \in \mathbb{R}^d$, we use $\|v\|$ to denote its ℓ_2 norm. For a matrix $\Theta \in \mathbb{R}^{d_1 \times d_2}$ of rank r, we denote its singular values by $\sigma_1(\Theta) \geq \sigma_2(\Theta) \geq \cdots \geq \sigma_r(\Theta) > 0$, its Frobenius norm by $\|\Theta\|_F = \sqrt{\sum_{j=1}^r \sigma_j^2(\Theta)}$, its operator norm by $\|\Theta\| = \sigma_1(\Theta)$, its vector ℓ_∞ norm by $|\Theta|_\infty = \max_{i,j} |\Theta_{ij}|$, its vector ℓ_1 norm by $|\Theta|_1 = \sum_{i,j} |\Theta_{ij}|$, its j^{th} row by Θ^j, and

$$\|\Theta\|_{2,1} = \sum_{j=1}^{d_1} \|\Theta^j\|$$

to denote its matrix $\ell_{2,1}$ norm. Given $\Theta, \Theta' \in \mathbb{R}^{d_1 \times d_2}$, we denote the matrix dot product by $\langle \Theta, \Theta' \rangle = \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} \Theta_{ij} \Theta'_{ij}$. We let $[k] = \{1, 2, \cdots, k\}$.

Matrix sensing. In our formalism, we consider the general setting of matrix sensing—i.e., given noisy observations X_i of linear projections $\Theta^* = U^* U^{*T}$, recover the underlying matrix U^*. In the case of word embeddings, X_i are simply noisy observations of entries of Θ^*; we give details below.

Formally, consider an unknown matrix $U^* \in \mathbb{R}^{d \times r}$, and let $\Theta^* = U^* U^{*T}$; note that $\Theta^* \in \mathbb{R}^{d \times d}$ is symmetric and has rank r. The goal is to estimate U^* given observations $A_i \in \mathbb{R}^{d \times d}$ and $X_i \in \mathbb{R}$, for $i \in [n]$, where

$$X_i = \langle A_i, \Theta^* \rangle + \epsilon_i \tag{1}$$

and $\epsilon_1, \cdots, \epsilon_n$ are independent σ-subgaussian random variables. For instance, in the application to word embeddings, the $A_1, \cdots, A_{d^2}$ are the basis matrices—i.e., $A_{i+j \cdot d}$ equals 1 in position (i, j) and equals 0 elsewhere.

In this formulation, we can only compute U^* up to orthogonal change-of-basis since Θ^* is preserved under this transformation—i.e., if $\widetilde{U}^* = U^* R$ for an orthogonal matrix $R \in \mathbb{R}^{r \times r}$, then we have $\widetilde{U}^* \widetilde{U}^{*T} = U^* R R^T U^{*T} = U^* U^{*T} = \Theta^*$. Thus, the goal is to compute $\widehat{U}$ such that $\widehat{U} \approx U^* R$ for some orthogonal matrix R.

To simplify notation, we define the linear operator $\mathcal{A} : \mathbb{R}^{d \times d} \to \mathbb{R}^n$, where $\mathcal{A}(\Theta)_i = \langle A_i, \Theta \rangle$. Then, (1) becomes

$$X = \mathcal{A}(\Theta^*) + \epsilon,$$

where $X = \begin{bmatrix} X_1 & \cdots & X_n \end{bmatrix}^T$ and $\epsilon = \begin{bmatrix} \epsilon_1 & \cdots & \epsilon_n \end{bmatrix}^T$.

Now, given an estimator $\widehat{U}$, we measure the estimation error using the $\ell_{2,1}$ norm. We use this norm instead of the more typical Frobenius norm since it is more naturally compatible with the group-sparse structure. It is analogous to the fact that the ℓ_1 error of ℓ_1 regularized linear regression is more natural to bound; bounding the ℓ_2 norm requires additional regularity assumptions. We refer the reader to Chapter 6 of Bühlmann & Van De Geer (2011) for a discussion. In addition, since we can only identify U^* up to orthogonal change-of-basis, we consider the error in a specific direction as in Ge et al. (2017).

Definition 1. *Given $\widehat{U}, U^* \in \mathbb{R}^{d\times r}$, the error of $\widehat{U}$ is*

$$\ell(\widehat{U}, U^*) = \|\widehat{U} - U^* R_{(\widehat{U},U^*)}\|_{2,1},$$

where $R_{(\widehat{U},U^*)} = \arg\min_{R:R^TR=RR^T=I} \|\widehat{U} - U^*R\|_F$.

This definition of error is invariant under rotation.

Transfer learning. Consider unknown parameters $U_p^* \in \mathbb{R}^{d\times r}$ for the source domain, and unknown parameters $U_g^* \in \mathbb{R}^{d\times r}$ for the target domain. Our goal is to use data from the source domain to help estimate U_g^*. In particular, we assume given *proxy observations* $\mathcal{A}_p : \mathbb{R}^{d\times d} \to \mathbb{R}^{n_p}$ and $X_p \in \mathbb{R}^{n_p}$ from the source domain, along with *gold observations* $\mathcal{A}_g : \mathbb{R}^{d\times d} \to \mathbb{R}^{n_g}$ and $X_g \in \mathbb{R}^{n_g}$ from the target domain, where

$$X_p = \mathcal{A}_p(\Theta_p^*) + \epsilon_p$$
$$X_g = \mathcal{A}_g(\Theta_g^*) + \epsilon_g,$$

and $\epsilon_p \in \mathbb{R}^{n_p}$ and $\epsilon_g \in \mathbb{R}^{n_g}$ are vectors of independent σ_p and σ_g-subgaussian random variables, respectively.

We are interested in the setting $(n_g/\sigma_g^2) \ll (n_p/\sigma_p^2)$. Intuitively, this condition says that either we have many more proxy observations than gold observations (i.e., $n_g \ll n_p$), or that the proxy observations are much lower variance than the gold observations (i.e., $\sigma_p \ll \sigma_g$). The latter case sometimes arises in low-data settings due to the observation structure; for instance, as we describe below, this is the case for our application to word embeddings.

Group sparse structure. To leverage the proxy observations to help estimate the gold parameters U_g^*, we need to assume some relationship between the two. Letting

$$\Delta_U^* = U_g^* - U_p^*,$$

we assume that Δ_U^* has a row-sparse structure—i.e., most of its rows are 0. More precisely, letting

$$J = \{j \in [d] \mid \|\Delta_U^{*j}\| \neq 0\},$$

the *group sparsity* of Δ_U^* is $s = |J|$. Then, an accurate estimate of Δ_U^* can help to recover U_g^*, since estimating Δ_U^* requires less data to recover due to its sparse structure.

Importantly, note that the row-sparse structure of Δ_U^* is preserved under *simultaneous* orthogonal transformations of U_g^* and U_p^*—i.e., if $\widetilde{U}_p^* = U_p^*R$ and $\widetilde{U}_g^* = U_g^*R$ for an orthogonal matrix R, then $\widetilde{\Delta}_U^* = (U_g^* - U_p^*)R = \Delta_U^*R$ has the same row sparsity as Δ^*.

Application to word embeddings. To apply matrix factorization to compute word embeddings, we begin by constructing the *word co-occurrence* matrix $\widehat{\Theta} \in \mathbb{R}^{d\times d}$, where $\widehat{\Theta}_{ij}$ counts the number of times the two words indexed by i and j appear together (e.g., in some fixed-length window of text); here, d is the total number of the words. In addition, we normalize $\widehat{\Theta}$ (i.e., divide by the total count $\sum_{i,j} \widehat{\Theta}_{ij}$).

Intuitively, we think of $\widehat{\Theta}$ as an empirical estimate of Θ^*, and take the observations X_i to be the entries of $\widehat{\Theta}$. In particular, let $A_1, \cdots, A_{d^2} \in \mathbb{R}^{d\times d}$ such that $A_{i+j\cdot d} = \mathbb{1}(i=j)$. Then, we take

$$X_i = \langle A_i, \widehat{\Theta}\rangle,$$

in which case $\epsilon_i = \langle A_i, \widehat{\Theta}\rangle - \Theta^*$ is the error. This error is bounded (since $\widehat{\Theta}$ is normalized) and zero mean (by definition), so it is subgaussian. Thus, we can use matrix factorization on X_i's and A_i's to compute $\widehat{U}$ such that $\Theta^* \approx \widehat{U}\widehat{U}^T$. Finally, we take $\widehat{U}^i$ to be the word vector for word $i \in [d]$.

As discussed above, in this setting, the number of observations n scales the subgaussian parameter of ϵ_i rather than the number of observations, which is always d^2. In particular, as more observations become available, the variance of our estimate $\widehat{\Theta}$ of Θ^* becomes smaller.

3. Naïve Estimators

We begin by describing two naïve strategies for estimating U_g^*: one based on only using the gold data, and one based on only using the proxy data. Our proposed estimator (described in Section 4) builds on these ones.

3.1. Gold Estimator

First, we consider estimating U_g^* using only the gold data:

$$\widehat{U}_g = \arg\min_{U_g} \frac{1}{n_g}\|X_g - \mathcal{A}_g(U_gU_g^T)\|^2. \tag{2}$$

Now, we analyze the sample complexity of $\widehat{U}_g$ under standard regularity assumptions. In particular, we assume restricted well-conditionedness (RWC) (Li et al., 2019).

Definition 2. *A linear operator $\mathcal{A}$ satisfies the r-RWC(α,β) condition if*

$$\alpha\|Z\|_F^2 \leq \frac{1}{n}\|\mathcal{A}(Z)\|^2 \leq \beta\|Z\|_F^2,$$

with $3\alpha > 2\beta$ and for any $Z \in \mathbb{R}^{d\times d}$ with $\operatorname{rank}(Z) \leq r$.

This property is a generalization of the restricted isometry property (RIP), which is a common assumption in matrix sensing problems. Under the RIP condition, common low-rank matrix problems have no spurious local minima—i.e., all local minima are also global minima (Bhojanapalli et al., 2016; Ge et al., 2017). However, the RIP condition is very restrictive as it requires all the eigenvalues of the Hessian matrix to be within a small range of 1. The RWC condition applies to more general situations and also guarantees the statistical consistency for all local minima (Li et al., 2019).

Theorem 1. *Assume $\mathcal{A}_g$ satisfies $2r$-RWC. Then, we have*

$$\ell(\widehat{U}_g, U_g^*) = \mathcal{O}\left(\sqrt{\frac{\sigma_g^2(d^2 + d\log(\frac{1}{\delta}))}{n_g}}\right)$$

with probability at least $1 - \delta$.

The full statement and proof are given in our extended paper (Xu et al., 2021).

3.2. Proxy Estimator

Next, we consider a strategy that estimates U_g^* by estimating U_p^* and ignoring the bias term Δ_U^*:

$$\widehat{U}_p = \arg\min_{U_p} \frac{1}{n_p} \|X_p - \mathcal{A}_p(U_p U_p^T)\|^2. \tag{3}$$

We have the following result:

Theorem 2. *Assume $\mathcal{A}_p$ satisfies $2r$-RWC. Then, we have*

$$\ell(\widehat{U}_p, U_g^*) = \mathcal{O}\left(\|\Delta_U^*\|_{2,1} + \omega + \sqrt{\frac{\sigma_p^2(d^2 + d\log(\frac{1}{\delta}))}{n_p}}\right),$$

with probability at least $1 - \delta$, where

$$\omega = \|U_p^*(R_{(\widehat{U}_p, U_p^*)} - R_{(\widehat{U}_p, U_g^*)})\|_{2,1}. \tag{4}$$

Since U_p^* may not be aligned with U_g^*, the estimation error using $\widehat{U}_p$ as an estimator of U_g^* includes a term ω accounting for the difference between U_p^* and U_g^*. When $R_{(\widehat{U}_p, U_p^*)} = R_{(\widehat{U}_p, U_g^*)}$, we have $\omega = 0$. In this case, the error decomposes into the bias term $\|\Delta_U^*\|_{2,1}$ plus the error of $\widehat{U}_p$ compared to U_p^*. The full statement and proof are given in our extended paper (Xu et al., 2021).

4. Group Sparse Transfer Learning

In this section, we describe our proposed estimator that combines gold and proxy data. Then, we state the quadratic compatibility condition, which extends the standard compatibility condition from the sparse regression literature (Lounici et al., 2011) to the matrix factorization setting, and prove sample complexity bounds assuming this condition holds. Finally, we describe how our group-sparse penalty term can be leveraged in conjunction with the GloVe objective.

4.1. Estimation Procedure

We define our proposed joint estimator for gold task as through the following two steps:

$$\begin{aligned} \widehat{U}_p &= \arg\min_{U_p} \frac{1}{n_p} \|X_p - \mathcal{A}_p(U_p U_p^T)\|^2 \\ \widehat{U}_g &= \arg\min_{g(U_g) \le 2L} \frac{1}{n_g} \|X_g - \mathcal{A}_g(U_g U_g^T)\|^2 + \lambda \|U_g - \widehat{U}_p\|_{2,1} \end{aligned} \tag{5}$$

Since our problem is nonconvex, we follow Loh & Wainwright (2015) and define a compact search region for U_g: $g(U_g) = \|U_g - \widehat{U}_p\|_{2,1} \le 2L$. L is a tuning parameter that should be chosen carefully to make U_g feasible—specifically, $\|U_g^* - U_p^*\|_{2,1} \le L$.

4.2. Quadratic Compatibility Condition

We make the following key assumption, which generalizes the compatibility condition required in the traditional sparse regression setting:

Definition 3. *The quadratic compatibility condition is*

$$\frac{s}{n_g} \|\mathcal{A}_g(\Delta U_g^{*T} + U_g^* \Delta^T + \Delta\Delta^T)\|^2 \ge \kappa \left(\sum_{j \in J} \|\Delta^j\|\right)^2$$

for any $\Delta \in \mathbb{R}^{d \times r}$ that satisfies

$$\sum_{j \in J^c} \|\Delta^j\| \le 7 \sum_{j \in J} \|\Delta^j\|.$$

Compared to the standard compatibility condition in the group sparse setting (Lounici et al., 2011), we have an extra quadratic term $\Delta\Delta^T$ since we are considering the nonconvex matrix factorization setting. In our extended paper (Xu et al., 2021), we show that restricted strong convexity (commonly assumed in high-dimensional settings) implies the quadratic compatibility condition in our context.

4.3. Main Result

Our main result characterizes the estimation error of our joint estimator $\widehat{U}_g$.

Theorem 3. *Assume $\mathcal{A}_p$ satisfies $2r$-RWC, $\mathcal{A}_g$ satisfies the quadratic compatibility condition. Suppose $n_p = \Omega(d^2 + d\log(\frac{1}{\delta}))$, and*

$$\lambda = \mathcal{O}\left(\sqrt{\frac{\sigma_g^2 \log(\frac{d^2}{\delta})}{n_g}}\right).$$

Table 1. Error bound for naïve estimators and joint estimator. ω is defined in Equation (4).

Estimator	Joint	Gold	Proxy
Error Bound	$\mathcal{O}\left(\sqrt{\frac{s^2 \log d}{n_g}} + \sqrt{\frac{d^2}{n_p}}\right)$	$\mathcal{O}\left(\sqrt{\frac{d^2}{n_g}}\right)$	$\mathcal{O}\left(\|\Delta_U^*\|_{2,1} + \omega + \sqrt{\frac{d^2}{n_p}}\right)$

Then, we have

$$\ell(\widehat{U}_g, U_g^*) = \mathcal{O}\left(\sqrt{\frac{\sigma_g^2 s^2 \log(\frac{d^2}{\delta})}{n_g}} + \sqrt{\frac{\sigma_p^2(d^2 + d\log(\frac{1}{\delta}))}{n_p}}\right),$$

with probability at least $1 - \delta$.

The full statement and proof are given in our extended paper (Xu et al., 2021). We note that the required condition on n_p in Theorem 3 is easily satisfied in our "gold-scarce and proxy-rich" setting—i.e., $n_g \gg \log(d)$ and $n_p \gg d^2$.

We summarize the estimation error bounds we derived for the three different estimators in Table 1. In the regime of interest—we have access to lots of proxy data ($n_p \gg d^2$) but limited gold data ($n_g \ll d^2$)—the upper bound of our joint estimator is much smaller in contrast to the typical proxy and gold estimators. In particular, taking $n_p \to \infty$, our bound scales as $\sqrt{s^2 \log d / n_g}$ whereas the gold bound scales as $\sqrt{d^2/n_g}$, which is an improvement of s/d. Alternatively, the proxy bound scales as at least $\|\Delta_U^*\|_{2,1}$, which does not go to zero with n_g; in contrast, our bound does.

4.4. Application to GloVe

The original GloVe method solves the following optimization problem (Pennington et al., 2014):

$$\min_{U^i, V^j, b_i, c_j} \sum_{i,j \in [d]} f(X_{ij})(\log(X_{ij}) - (U^i V^{jT} + b_i + c_j))^2,$$

where X_{ij} is the total number of co-occurrences of word i and j, $\{U^i\}$ and $\{V^j\}$ are the two sets of word embeddings, and d is the vocabulary size; $f(X_{ij})$ is a weighting function that is non-decreasing in co-occurrence; $b_i, c_j \in \mathbb{R}$ are bias terms. In practice, GloVe takes the sum of the two sets of embeddings as the final embeddings, i.e., $U^i + V^i$ is the word vector for word i. To leverage our approach, we add a group lasso penalty to this objective:

$$\min_{U^i, V^j, b_i, c_j} \sum_{i,j \in [d]} f(X_{ij})(\log(X_{ij}) - (U^i V^{jT} + b_i + c_j))^2 + \lambda \sum_{i \in [d]} \|(U^i + V^i) - \widehat{U}_p^i\|, \tag{6}$$

where $\widehat{U}_p$ is the pre-trained GloVe embedding matrix.

5. Experiments

We evaluate our joint estimator on both synthetic and real data. On the synthetic data, we compare the error of our estimator with the ground truth parameters. Then, we consider two real datasets; in this case, we leverage our penalty in conjunction with the GloVe objective. First, we apply it to Wikipedia articles from specific domains (e.g., math), and evaluate whether it can identify words with novel meanings in that domain; this experiment demonstrates the interpretability of our approach. Second, we evaluate the downstream prediction accuracy of our word embeddings on a clinical trial eligibility data.

5.1. Experiments on Synthetic Data

Data. We focus on the low-data setting; in particular, we let $n_g = 50$, $n_p = 5{,}000$, and $d = 20$. We consider the exact low-rank case with $r = 5$. The observation matrices $A_{p,i}$'s (and $A_{g,i}$'s) are independent Gaussian random matrices whose entries are i.i.d. $N(0, 1)$. We generate Θ_p^* by choosing U_p^* with i.i.d. $N(0, 1)$ elements. To construct the gold data, we set the row sparsity of Δ_U^* to 10% ($s = 2$). Then, we randomly pick s rows and set the value of each entry to 1. We take both noise terms to be $\epsilon_{p,i}, \epsilon_{g,i} \sim N(0, 1)$.

Setup. We compute the gold, proxy, and joint estimators by solving optimization problems (2), (3), and (5), respectively. To construct the joint estimator, we also need to pick a proper value for the hyperparameter λ. We use 5-fold cross validation to tune λ and we keep 20% of the gold data as the cross validation set. As all the final estimates of U_g are invariant under an orthogonal change-of-basis, we instead report the Frobenius norm of the estimation error of Θ_g. We average this error over 100 random trials.

Results. Figure 1 shows the Frobenius error of the naïve estimators (i.e., the gold and proxy estimators from Section 3) and our joint estimator, with a 95% confidence interval. Our joint estimator significantly outperforms the other two estimators—in particular, the Frobenius error of our joint estimator is only around 4% of the proxy estimator and 2% of the gold estimator.

5.2. Experiments on Wikipedia

One advantage of our method is that it is more interpretable—in particular, we show that it can be used to identify the domain words (i.e., words that have a special meaning in

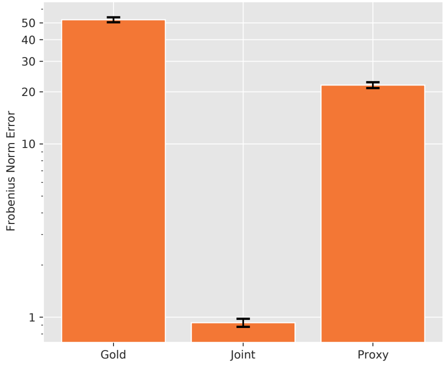

Figure 1. Frobenius norm estimation error of Θ_g over 100 trials, along with 95% confidence interval. The y-axis is in log scale.

a certain domain). We apply our method to GloVe and evaluate its performance on single domain Wikipedia articles in terms of the accuracy at identifying domain-specific words. We additionally compare our joint estimator with state-of-the-art fine-tuning heuristics Mittens (Dingwall & Potts, 2018) and CCA/KCCA (Sarma et al., 2018), as well as randomly selecting words.

Data. We manually curated 37 Wikipedia articles from the following four domains: finance, math, computer science, and politics. The articles selected all have a domain-specific word in their title—e.g., "put" in the article "put option" (in finance), "closed" in "closed set" (in math), "object" in computing, and "left" in "left wing politics" (in politics). All the Wikipedia text data were downloaded from English Wikipedia database dumps[2] in January 2020. We preprocess the text by splitting and tokenizing sentences, removing short sentences that contain less than 20 characters or 5 tokens, and removing stopping words.

We download the pre-trained word embeddings from GloVe's official website.[3] We take those trained using the 2014 Wikipedia dump and Gigaword 5, which contains around 6 billion tokens and 400K vocabulary words.

Setup. We solve the optimization problem (6) to construct our joint estimator for each single article. We take the pre-trained GloVe word embedding as described above. Similar to GloVe, we create the co-occurrence matrix using a symmetric context window of length 5. We choose the dimension of the word embedding to be 100 and use the default weighting function of GloVe. The Mittens word embeddings are obtained solving a similar problem as (6), but with the Frobenius norm penalty—i.e.,

$$\sum_{i\in[d]} \|(U^i + V^i) - \widehat{U}_p^i\|^2.$$

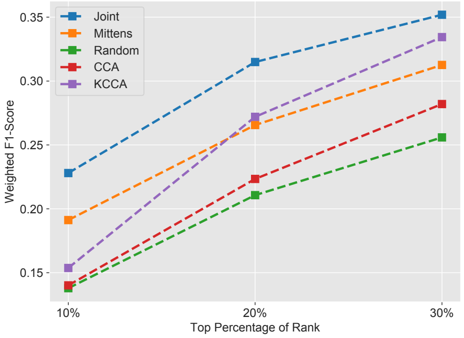

Figure 2. Weighted F1-score versus top percentage of the rank set for the threshold in the finance domain.

We fix $\lambda = 0.05$ for both approaches; we found our results to be robust to this choice. To construct the CCA estimator, we take a simple average of the aligned domain-specific word embeddings and the pre-trained word embedding. We set the standard deviation of the Gaussian kernel to be 1 to construct the KCCA estimator. Then, to identify domain-specific words, we score each word i by the ℓ_2 distance between its new embedding (e.g., our joint estimator or Mittens) and its pre-trained embedding; a higher score indicates a higher likelihood of being a domain-specific word.

To evaluate the accuracy of domain-specific word identification, we select and compare the top 10% of words selected by this score for each estimator—i.e., we treat all words in the top 10% as positives. We define a word to be a domain word if any of its definitions on Wiktionary is labeled with key words from that domain—i.e., "finance" or "business" for finance, "math", "geometry", "algebra", or "group theory" for math, "computing" or "programming" for computer science, and "politics" for politics. We compute the F_1-score of the selected domain words across articles in each domain. For the F_1-score of random selection, we compute the precision and recall in closed form (the precision is the fraction of domain words among all vocabulary words, and the recall is the top fraction we set, i.e., 10%).

Results. Table 2 shows the F_1-score of each domain weighted by article length for each estimator. Our approach consistently outperforms the baselines across all domains. While we observe that other approaches also identify domain-specific words, our approach does so more effectively, most likely since our sparsity assumption is satisfied by these datasets.

Next, we evaluate how our result varies with the value of selection threshold; in particular, we consider 10%, 20%, and 30%. Figure 2 shows the weighted F_1-score versus the top percentage set for the threshold in the finance domain. Our approach consistently outperforms the baselines. Finally,

[2] https://dumps.wikimedia.org/enwiki/latest/

[3] https://nlp.stanford.edu/projects/glove/

Table 2. Weighted F1-score of domain word identification across four domains, where we select the top 10% of words.

Domain	Joint	Mittens	CCA	KCCA	Random
Finance	**0.2280**	0.1912	0.1382	0.1560	0.1379
Math	**0.2546**	0.2171	0.2381	0.1605	0.1544
Computing	**0.2613**	0.1952	0.2224	0.2260	0.1436
Politics	**0.1852**	0.1543	0.0649	0.1139	0.0634

Table 3. Top 10 words in the rank sorted by absolute change of word embedding from source to target domain. We pick one article from each domain, with the domain words labeled in bold. The threshold is set to top 10% of the rank.

Short		Prime Number		Cloud Computing		Conservatism	
Joint	Mittens	Joint	Mittens	Joint	Mittens	Joint	Mittens
short	**short**	**prime**	**prime**	**cloud**	**cloud**	**party**	**party**
shares	percent	**formula**	still	**data**	**private**	**conservative**	**conservative**
price	due	**numbers**	**formula**	**computing**	large	social	second
stock	public	**number**	de	**service**	information	conservatism	social
security	customers	**primes**	**numbers**	**services**	devices	government	research
selling	prices	**theorem**	**number**	**applications**	**applications**	**liberal**	svp
securities	high	**natural**	great	**private**	security	**conservatives**	government
position	hard	integers	side	users	work	political	de
may	**shares**	**theory**	way	use	**engine**	**right**	also
margin	price	**product**	algorithm	**software**	allows	economic	church

Table 3 shows the top 10 words ranked by our approach and by Mittens for one article in each domain; our approach is more effective at identifying domain words (in bold).

5.3. Clinical Trial Eligibility

Another measure of the quality of the word embeddings is the prediction accuracy in downstream tasks. To this end, we consider a clinical trial eligibility prediction task, where the words are from the medical domain. We apply our approach in conjunction with GloVe, and compare it with two fine-tuning heuristics, Mittens, and CCA/KCCA. In addition, we compare our method with Mittens over different pre-trained word embeddings, i.e., Word2Vec (Mikolov et al., 2013) and Dict2Vec (Tissier et al. (2017), extends Word2Vec by adding dictionary definition of words) instead of GloVe.

Data. The inclusion criteria for cancer clinical trials are restrictive, and these protocols are typically described in text. Bustos & Pertusa (2018) use deep neural networks to classify short clinical statements into inclusion or exclusion criteria, which aims to help determine the eligibility of patients for cancer clinical trials. In this experiment, we analyze a 1-million subsample of the original 6-million clinical trial eligibility data used in Bustos & Pertusa (2018), which has been made publicly available by the authors on Kaggle[4]. The data provides a label (eligible or not eligible), and a corresponding short free-text statement that describes the eligibility criterion and the study intervention and condition.

Setup. We study the low-data setting by restricting to only 50 observations; we use a balanced sample as in Bustos & Pertusa (2018). As before, we solve (6) for our joint estimator, and solve the same objective with the Frobenius norm penalty for Mittens. We use CCA and KCCA as described in Section 5.2. In addition to using the pre-trained word embeddings of GloVe, we also use those of Word2Vec and Dict2Vec. For Word2Vec, we use the pre-trained word embeddings trained on the English CoNLL17 corpus, available at the Nordic Language Processing Laboratory (NLPL).[5] The pre-trained Dict2Vec embeddings are publicly available on Github.[6] As in many semi-supervised studies, we train word embeddings using all text from both the training and test sets. In this task, we set the word embedding dimension to 100. We tune our regularization parameter λ separately for each type of pre-trained embeddings (i.e., GloVe, Word2Vec, and Dict2Vec).

To predict eligibility, we use logistic regression with an ℓ_2 penalty. We split the 50 observations into 20% for testing and 80% for training and cross-validation. We use 5-fold cross-validation to tune the hyperparameters in regularized logistic regression. Since it is computationally expensive to feed all embeddings into the model, we instead take the average of embeddings of all words in an observation as the features for the logistic regression model.

Results. We compare the F_1-score of our joint estimator with the other baselines. Table 4 shows the average F_1-score

[4] https://www.kaggle.com/auriml/eligibilityforcancerclinicaltrials

[5] http://vectors.nlpl.eu/repository

[6] https://github.com/tca19/dict2vec

Table 4. F1-score of prediction over 10K trials. The 95% confidence interval is shown for each result.

Pre-trained	Estimator	Average F1-score
GloVe	Joint	**0.612554** ± 0.003405
	Mittens	0.604338 ± 0.003413
	CCA	0.598330 ± 0.003615
	Gold	0.583299 ± 0.003516
	KCCA	0.579160 ± 0.003351
	Proxy	0.568205 ± 0.003465
Word2Vec	Joint	**0.624428** ± 0.003466
	Mittens	0.604981 ± 0.003514
	Proxy	0.580473 ± 0.003503
Dict2Vec	Joint	**0.638923** ± 0.003154
	Mittens	0.627772 ± 0.003335
	Proxy	0.623337 ± 0.003491

over 10K trials and the 95% confidence interval. As can be seen, our joint estimator outperforms all baselines on the downstream prediction task. Furthermore, the performance of our joint estimator is robust across different pre-trained word embeddings.

6. Conclusion

We have proposed a novel estimator for transferring word embeddings to new domains. We cast the problem as a low-rank matrix factorization problem with a group-sparse penalty, regularizing the learned embeddings towards existing domain-agnostic embeddings such as GloVe. Under a sparsity assumption and standard regularity conditions, our estimator provably requires exponentially less data to achieve the same error compared to the gold and proxy estimators. Our experiments demonstrate the effectiveness of our approach in the low-data regime on synthetic data, a domain word identification task on single Wikipedia articles, and a downstream clinical trial eligibility prediction task.

References

Ali, A., Dobriban, E., and Tibshirani, R. The implicit regularization of stochastic gradient flow for least squares. In *International Conference on Machine Learning*, pp. 233–244. PMLR, 2020.

Bastani, H. Predicting with proxies: Transfer learning in high dimension. *Management Science*, 2020.

Ben-David, S., Blitzer, J., Crammer, K., Pereira, F., et al. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19:137, 2007.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

Bhojanapalli, S., Neyshabur, B., and Srebro, N. Global optimality of local search for low rank matrix recovery. *arXiv preprint arXiv:1605.07221*, 2016.

Bickel, P., Ritov, Y., and Tsybakov, A. Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics*, pp. 1705–1732, 2009.

Bühlmann, P. and Van De Geer, S. *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media, 2011.

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

Bustos, A. and Pertusa, A. Learning eligibility in cancer clinical trials using deep neural networks. *Applied Sciences*, 8(7):1206, 2018.

Candes, E. and Tao, T. The dantzig selector: statistical estimation when p is much larger than n. *The Annals of Statistics*, pp. 2313–2351, 2007.

Chen, S. S., Donoho, D. L., and Saunders, M. A. Atomic decomposition by basis pursuit. 1995.

Crammer, K., Kearns, M., and Wortman, J. Learning from multiple sources. *Journal of Machine Learning Research*, 9(8), 2008.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Dingwall, N. and Potts, C. Mittens: an extension of glove for learning domain-specialized representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 212–217, 2018.

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., and Thrun, S. Dermatologist-level classification of skin cancer with deep neural networks. *nature*, 542(7639):115–118, 2017.

Friedman, J., Hastie, T., and Tibshirani, R. A note on the group lasso and a sparse group lasso. *arXiv preprint arXiv:1001.0736*, 2010.

Ganin, Y. and Lempitsky, V. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pp. 1180–1189. PMLR, 2015.

Ge, R., Jin, C., and Zheng, Y. No spurious local minima in nonconvex low rank problems: A unified geometric analysis. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1233–1242. JMLR. org, 2017.

Levy, O. and Goldberg, Y. Neural word embedding as implicit matrix factorization. *Advances in neural information processing systems*, 27:2177–2185, 2014.

Li, Q., Zhu, Z., and Tang, G. The non-convex geometry of low-rank matrix optimization. *Information and Inference: A Journal of the IMA*, 8(1):51–96, 2019.

Loh, P.-L. and Wainwright, M. J. Regularized m-estimators with nonconvexity: Statistical and algorithmic theory for local optima. *The Journal of Machine Learning Research*, 16(1):559–616, 2015.

Lounici, K., Pontil, M., Van De Geer, S., Tsybakov, A. B., et al. Oracle inequalities and optimal inference under group sparsity. *The Annals of Statistics*, 39(4):2164–2204, 2011.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pp. 3111–3119, 2013.

Negahban, S. and Wainwright, M. J. Estimation of (near) low-rank matrices with noise and high-dimensional scaling. *The Annals of Statistics*, pp. 1069–1097, 2011.

Pennington, J., Socher, R., and Manning, C. D. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

Sarma, P. K., Liang, Y., and Sethares, W. A. Domain adapted word embeddings for improved sentiment classification. *arXiv preprint arXiv:1805.04576*, 2018.

Simon, N., Friedman, J., Hastie, T., and Tibshirani, R. A sparse-group lasso. *Journal of computational and graphical statistics*, 22(2):231–245, 2013.

Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

Tissier, J., Gravier, C., and Habrard, A. Dict2vec: Learning word embeddings using lexical dictionaries. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 254–263, 2017.

Xu, K., Zhao, X., Bastani, H., and Bastani, O. Group-sparse matrix factorization for transfer learning of word embeddings. *arXiv preprint arXiv:2104.08928*, 2021.

Yang, W., Lu, W., and Zheng, V. A simple regularization-based algorithm for learning cross-domain word embeddings. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2898–2904, 2017.

Zhang, K., Schölkopf, B., Muandet, K., and Wang, Z. Domain adaptation under target and conditional shift. In *International Conference on Machine Learning*, pp. 819–827. PMLR, 2013.