Restricted Transfer Learning for Text Categorization

Rajhans Samdani, Gideon Mann
Google Research, {rajhans, gmann}@google.com

Abstract

In practice, machine learning systems deal with multiple datasets over time. When the feature spaces between these datasets overlap, it is possible to transfer information from one task to another. Typically in transfer learning, all labeled data from a *source* task is saved to be applied to a new *target* task thereby raising concerns of privacy, memory and scaling. To ameliorate such concerns, we present a semi-supervised algorithm for text categorization that transfers information across tasks without storing the data of the souce task. In particular, our technique learns a sparse low-dimensional projection from unlabeled and the source task data. In particular, our technique learns low-dimensional sparse word clusters-based features from the source task data and a massive amount of additional unlabeled data. Our algorithm is efficient, highly parallelizable, and outperforms competitive baselines by up to 9% on several difficult benchmark text categorization tasks.

1 Introduction

Centralized machine learning systems observe multiple labeled classification problems over time. Researchers [9] have wondered if after observing one task (called the *source* task), it is possible for a system to get better accuracy on the next task (called the *target* task.) A large body of work on transfer learning [9, 2, 10, 8] tries to address this question.

In this paper, we consider a restricted setting for transfer learning, which we refer to as *Restricted Incremental Transfer* (RIT)[1]. In the RIT setting, we cannot store the labeled source task data as such for a variety possible of reasons including privacy, memory, and scalability[2]. Thus a transfer learning algorithm for RIT must embed the information from the source task in a compact intermediate layer without the knowledge of the target task. To clarify, we use transfer learning to refer to a setting where the source and target tasks likely involve prediction over different label spaces.

In particular, we focus on text categorization and present a semi-supervised algorithm for tranferring information from source to target via a sparse low-dimensional projection of words. We call our algorithm *Projection-learning for Restricted Incremental Transfer* (PRIT.) PRIT uses word clusters constructed from unlabeled data and adapts them using labeled source data to create an intermediate word clustering, which is subsequently used for the target domain. Using information gathered from a massive amount of unlabeled data helps us scale to a large vocabulary of unseen words. We present experimental results on benchmark datasets on newsgroup categorization [7] and Wikipedia document categorization [1]. Our experiments show that PRIT achieves significant improvements over baseline algorithms by transferring information between different, yet related, tasks.

2 Preliminaries and Notation

The task of text categorization involves mapping a document to a given category or label. Formally, let a document be represented by the vector $\mathbf{x}$ where x_j is the count of word j in the document, and let y be the desired output label for that document. The goal then is to learn a function s.t.

[1] Related to what [10] refer to as representational transfer.

[2] A direct application of PRIT is in pay-for-use machine learning services which deal with confidential data from multiple clients over time.

$\max_{\hat{y}}\ \mathrm{f}(\mathbf{x}, \hat{y}) = y$, given a set of training data tuples $(\mathbf{x}, y)$. Here we consider the case where the system is presented with two unrelated text categorization training sets, $\mathbb{S}$ and $\mathbb{T}$, with distinct output label sets $\mathbb{Y}^s$ (the source) and $\mathbb{Y}^t$ (the target). In transfer learning, the goal is to improve the accuracy of learning the target function $\mathrm{f}^t(\mathbf{x}, \hat{y})$ given the source data $\mathbb{S}$ in addition to $\mathbb{T}$.

Cluster Projection Based Features: Tasks in text categorization and NLP suffer from word sparsity: a large fraction of words seen during testing may not be seen during training. To alleviate this problem, several researchers (e.g. [6]) project the words on to an n-dimensional "cluster space" (or *topic* space) with $n \ll d$, where each dimension can be thought of as a cluster or a topic.

Let $\mathbb{C}$ be a $n \times d$ cluster projection matrix such that $\mathbb{C}[i, j]$ is the weight of word j over the i^{th} cluster. Techniques like K-Means [5], LDA [3], or Brown clustering [4] can be used to learn $\mathbb{C}$. When the underlying clustering is a hard clustering (e.g. K-means), each word belongs to a few clusters with equal affinity. In the hard clustering case, we will interchangeably represent the cluster projection as a set (or a *clustering*) of hard word clusters, $\mathbb{C} = \{C_1, \ldots, C_n\}$. In the matrix form, $\mathbb{C}$ will be a sparse binary matrix with $\mathbb{C}[i, j] = 1$ *iff* word j is in C_i. Given this matrix representation, the product $\mathbb{C}\mathbf{x}$ yields the projection of the word counts onto the cluster space.

In this paper, we focus on conventional log-linear models of the form: $\mathrm{f}(\mathbf{x}, \hat{y}) = \Pr[\hat{y}|\mathbf{x}; \Theta] \propto \exp(\theta_{\hat{y}}^T \phi(\mathbf{x}))$. In order to integrate cluster features into a learned model, we augment the feature transformation $\phi(\mathbf{x})$ with cluster projection features: $\phi(\mathbf{x}; \mathbb{C}) = [\mathbf{x}^T \ \ (\mathbb{C}\mathbf{x})^T]^T$.

3 Projection learning for Restricted Incremental Transfer (PRIT)

In restricted incremental transfer, our goal is to improve the predictions of f^t by transferring information from $\mathbb{S}$ without retaining $\mathbb{S}$ as available initially. To do so, we create an intermediate representation using only $\mathbb{S}$ that is subsequently combined with $\mathbb{T}$ to construct the final model.

We present an algorithm for RIT for text categorization which we call *Projection-learning for Restricted Incremental Transfer* (PRIT.) PRIT proceeds in three main steps: (1) Using unsupervised data $\mathbb{U} = \{\mathbf{x}\}$, we construct word similarities and initial word clusters $\mathbb{C}$; (2) we split these word clusters $\mathbb{C}$ into smaller "label" clusters using source training data $\mathbb{S}$ creating an intermediate representation; and (3) we learn the final sparse cluster-projection matrix (along with the classifier parameters) on the target training data $\mathbb{T}$.

A high level overview of PRIT is given in Alg. 1. We now describe each step of PRIT.

1) Unsupervised information (line 1) We use a large amount of publicly available unsupervised data, the Google N-gram corpus [5], and represent each word as a vector based on its neighboring words in the corpus. Using this representation, we compute two kinds of information: 1) An initial coarse clustering $\mathbb{C} = \{\mathbb{C}_1, \ldots, \mathbb{C}_n\}$ using K-means and 2) the pairwise word similarities $\mathrm{sim}[u, v]$ between words using Jaccard similarity between their representations. Both of this tasks are highly parallelizable, which is necessary to deal with the enormous Google N-gram corpus.

2) Clustering based on the source task (lines 2-4) Given the initial clustering $\mathbb{C}$ and the word-similarity measure $\mathrm{sim}[u, v]$, each cluster $\mathbb{C}_i$ is split into *sub-clusters* based on the association of words in $\mathbb{C}_i$ with labels in the source data $\mathbb{S}$. This step is performed independently and in parallel for all clusters to produce a new clustering projection matrix $\mathbb{C}^s$.

Let $G_i(\sigma)$ be a graph with a node for each word $u \in \mathbb{C}_i$ and edges $E_i(\sigma) = \{(u, v) : u, v \in \mathbb{C}_i, \mathrm{sim}[u, v] \geq \sigma\}$, such that only words with similarity at least σ are connected. The edge (u, v) in E_i is weighted $\mathrm{sim}[u, v]$. Now, we sequentially perform the following three steps.

a) Initialize label distribution (line 2): Using the source training data $\mathbb{S}$, we compute the conditional label distribution, $q_{uy} = \Pr_{\mathbb{S}}[y|w]$, $\forall w \in C_i, \forall y \in \mathbb{Y}^s$, by counting as in naïve Bayes with Laplace smoothing. Let $\mathbb{C}_i(\rho) = \{w : \max_y q_{wy} \geq \rho\}$ be the set of words associated with probability greater than a constant ρ with at least one of the labels in $\mathbb{Y}^s$. These strongly associated words alone would be good candidates for cluster splits, but we leverage this information further.

b) Propagate label distribution (line 3): We spread the label distribution from words in the set $\mathbb{C}_i(\rho)$ to $\mathbb{C}_i \setminus \mathbb{C}_i(\rho)$, the words in $\mathbb{C}_i$ that are not strongly associated with a particular label. We achieve this by encouraging neighboring words in the similarity graph $G_i(\sigma)$ to have similar label distributions. Let $\mathbf{U}$ be the uniform distribution over labels $\mathbb{Y}^s$, and κ be a fixed regularization con-

Algorithm 1 An overview of PRIT algorithm.

Input: Unsupervised data $\mathbb{U}$, Training data for the source and target tasks: $\mathbb{S}$ and $\mathbb{T}$
1: Obtain from $\mathbb{U}$: initial word clusters: $\mathbb{C} = \{\mathbb{C}_1, \ldots, \mathbb{C}_n\}$ and a word similarity metric: sim
for $i = 1$ to n **do**
  2: Compute Label-Distribution $\mathbf{q}_w, \forall w \in \mathbb{C}_i$
  3: Perform Label-Propagation($\mathbb{C}_i$, sim)
  4: Split cluster $\mathbb{C}_i$ based on label distribution
  (Reclustering using source data)
end for
5: Combine all clusters to create clustering $\mathbb{C}^s$
6: Learn($\mathbb{C}^t, \Theta|\mathbb{T}, \mathbb{C}^s$), while regularizing $\mathbb{C}^t - \mathbb{C}^s$ (Learning over target)

stant. We obtain a label distribution $\mathbf{q}$ over all words by minimizing the following convex function:

$$\sum_{u \in C_i \setminus C_i(\rho)} \left(\kappa \|\mathbf{q}_u - \mathbf{U}\|^2 + \sum_{v \in C_i, (u,v) \in E_i} \text{sim}[u,v] \|\mathbf{q}_u - \mathbf{q}_v\|^2 \right) \quad (1)$$

$$\text{s.t.} \quad \forall v \in \mathbb{C}_i, \forall y \in \mathbb{Y}^s, \forall w \in \mathbb{C}_i(\rho) : \sum\nolimits_{y'} q_{vy'} = 1 \text{ and } q_{vy} \geq 0 \text{ and } q_{wy} = \Pr_{\mathbb{S}}[y|w]$$

The term $\kappa\|\mathbf{q}_u - \mathbf{U}\|^2$ regularizes the distributions to be close to uniform so that a word is not associated with any label without significant label information. We minimize (1) efficiently via a graph-based *label propagation* algorithm [11].

c) Split clusters (line 4): Using the final label distribution $\mathbf{q}$, we split the cluster $\mathbb{C}_i$ into smaller clusters each containing words associated with different labels: $\mathbb{C}_{iy} = \{w : q_{wy} \geq \rho\}$, $\forall y \in \mathbb{Y}^s, w \in \mathbb{C}_i$, and a cluster containing the remaining words $\overline{\mathbb{C}_i} = \mathbb{C}_i \setminus (\cup_y \mathbb{C}_{iy})$. Finally, we output a clustering containing all resulting clusters: $\mathbb{C}^s = \cup_{i,y} \mathbb{C}_{iy} \cup_i \overline{\mathbb{C}_i}$ (line 5.)

3) Sparse learning over the target task (line 6): When considering the target task f^t, we have access to the cluster projection matrix $\mathbb{C}^s$ which we further adapt to the target task. Given labeled data $\mathbb{T}$, we learn the final projection matrix $\mathbb{C}^t$ along with the parameters Θ_t using a novel sparse projection learning step. Let λ_1 and λ_2 be two positive regularization parameters. We learn as:

$$\min_{\Theta_t, \mathbb{C}^t \geq 0} \frac{\lambda_1}{2}\|\Theta_t\|^2 + \frac{\lambda_2}{2}\|\mathbb{C}^t - \mathbb{C}^s\|_{1,1} - \frac{1}{|\mathbb{T}|} \sum_{(x_t, y_t) \in D_t} \left(\theta_{y_t}^T \phi(x_t; \mathbb{C}^t) - \log\Big(\sum_{y \in \mathbb{Y}^t} e^{\theta_y^T \phi(x_t; \mathbb{C}^t)} \Big) \right), \quad (2)$$

where $\|\mathbb{C}^t - \mathbb{C}^s\|_{1,1}$ is the $\ell_{1,1}$ norm of $\mathbb{C}^t - \mathbb{C}^s$ ($\|A\|_{1,1} = \sum_{i,j} |A_{ij}|$). We choose $\ell_{1,1}$ norm as it encourages $\mathbb{C}^t$ to be sparse as $\mathbb{C}^s$ itself is a sparse matrix. Since the objective function in Eq. (2) is non-convex w.r.t Θ_t and $\mathbb{C}^t$, but is convex w.r.t. any one of the two individually, we follow an alternate optimization procedure, which iteratively optimizes Θ_t and $\mathbb{C}^t$, for 10 rounds.

4 Experiments and Conclusion

We present experiments on two datasets: 20 Newsgroup [7] and the ECML/PKDD 2012 Pascal Wikipedia document categorization challenge [1]. From the 20 Newsgroup dataset, we select four related newsgroups (based on their hardness of categorization as very easy to separate categories are not interesting) from the *comp* category: *comp.graphics (Graphics)*, *comp.windows.x (X)*, *comp.sys.ibm.pc.hardware (Hardware)*, and *comp.os.ms-windows.misc (Misc)*. We define two tasks: task1 is separating *Graphics* from *X* and task2 is separating *Misc* from *Hardware*. The Pascal dataset contains a large collection of Wikipedia documents each belonging to certain categories. The provided category labels form a hierarchy. We select three different sets of *primitive* categories (i.e. categories with no subcategories), each set having a common parent category in the hierarchy — thus we know that the categories within each of these sets are somehow related. We create binary categorization tasks within each of the sets which are as follows. 1) **American entertainment people by occupation:** Task1: *American directors* (*Directors*) vs *American producers* (*Producers*) and Task2: *American music video directors* (*Music Video Directors*) vs *American choreographers/dancers* (*Dancers*). 2) **American actors by state:** From the eight given categories classifying American actors based on their state, we randomly form four task pairs for binary classification. 3) **Ice Hockey players:** Again, we randomly define four different tasks from eight provided categories

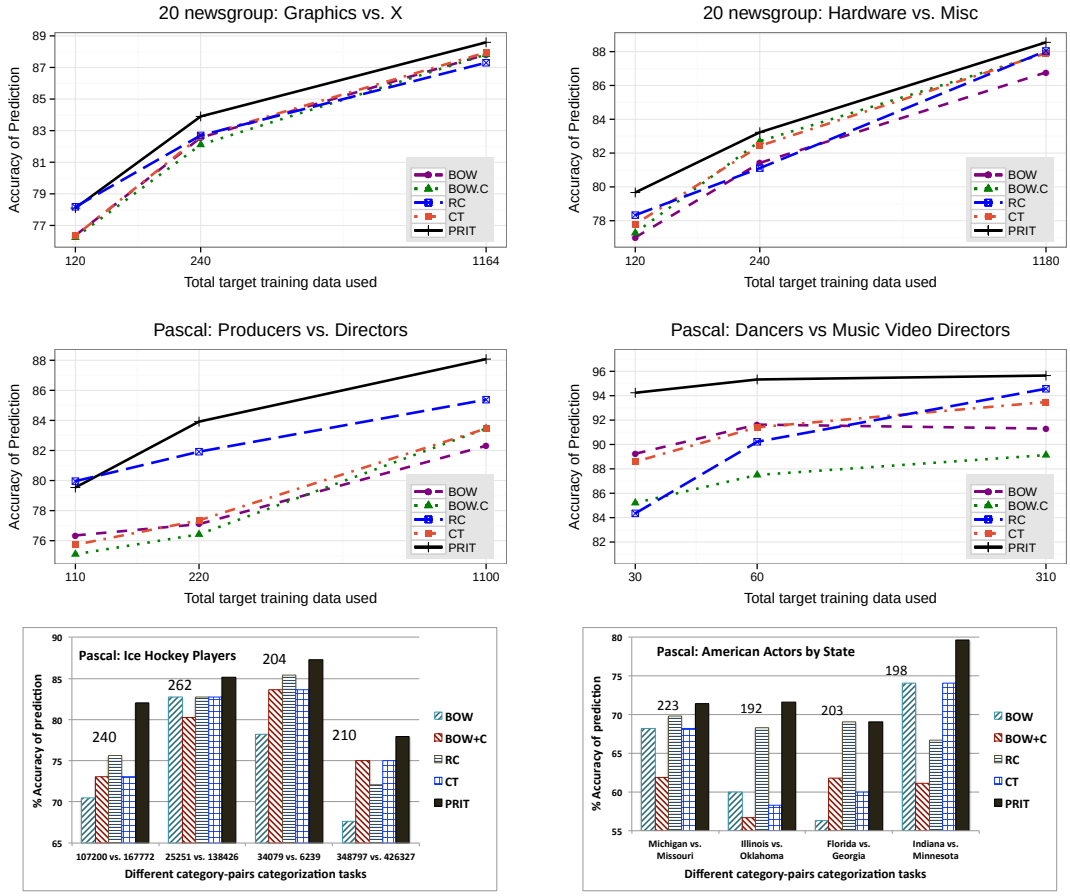

Figure 1: Comparing % accuracy of BOW, BOW+C, RC, CT, and our algorithm, PRIT. The top row corresponds to the 20 newsgroup dataset, the middle row corresponds to Pascal data for American entertainment people, and the bottom row contains Pascal Data for Ice Hockey Players (left) and American Actors by State (right). We use 100% of source training data for all experiments. For the top two rows, we vary the size of target training data; in the bottom row experiments (with 4 related tasks) we only report results with 100% of target data (exact training size is reported for each task).

(identified by their numerical ids from the dataset.) For a given set of related tasks, we experiment with each task as target and the remaining tasks as source tasks. If there are more than one possible source tasks, we simply use held-out target training data to first pick the best source task clustering.

Baselines and results: As baselines in our experiments, we use the following three styles of algorithms which obey the RIT restrictions (most algorithms for transfer learning cannot be used for RIT as they need access to the source labeled data.) **Simple baselines:** Includes the simple bag-of-words (**BOW**) baseline and another baseline which adds unsupervised cluster features (**BOW+C**). **Feature Learning Baseline:** This baseline performs the reclustering step (lines 4-6 of Alg. 1) as well as the learning step in Eq. 2 using only the target data. We call this baseline the Re-Clustering (**RC**) algorithm. We compare with RC to show that task-transfer is indeed essential to improve the performance with PRIT. **Classifier Transfer (CT):** CT uses the label probabilities output by a classifier trained on the source data as features in the target classification task (thus the source classifier forms the intermediate representation in this case.) The results are shown in Figure 1. PRIT outperforms the competing baselines by 1-9% in 18 out of 20 comparisons.

Conclusion: We considered a restricted transfer learning scenario motivated by practical considerations of privacy, memory, and scalability. Our proposed algorithm for this scenario significantly improves over competitive baselines in our experiments. Future work includes developing an approach that iteratively refines the learned features through a series of tasks, which is philosophically similar to the idea of *life long learning* [9].

References

[1] Joint ECML/PKDD PASCAL Workshop on Large-Scale Hierarchical Classification, 2012.

[2] J. Baxter. A bayesian/information theoretic model of learning to learn viamultiple task sampling. *Maching Learning*, 1997.

[3] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *JMLR*, 2003.

[4] P. Brown, V. D. Pietra, P. deSouza, J. Lai, and R. Mercer. Class-based n-gram models of natural language. *CL*, 1992.

[5] D. Lin, K. Church, H. Ji, S. Sekine, D. Yarowsky, S. Bergsma, K. Patil, E. Pitler, R. Lathbury, V. Rao, K. Dalwani, and S. Narsale. New tools for web-scale n-grams. In *LREC*, 2010.

[6] S. Miller, J. Guinness, and A. Zamanian. Name tagging with word clusters and discriminative training. In *NAACL*, 2004.

[7] T. M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., 1997.

[8] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Trans. on Knowl. and Data Eng.*, 2010.

[9] S. Thrun. Is learning the n-th thing any easier than learning the first? In *NIPS*, 1996.

[10] S. Thrun and L. Pratt, editors. *Learning to learn*. Kluwer Academic Publishers, 1998.

[11] X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical report, CMU, 2002.