Deep Transfer Learning via Restricted Boltzmann Machine for Document Classification

Jian Zhang
CS Dept, Louisiana State University
Baton Rouge, LA 70803
Email: zhang@csc.lsu.edu

***Abstract*—Transfer learning aims to improve a targeted learning task using other related auxiliary learning tasks and data. Most current transfer-learning methods focus on scenarios where the auxiliary and the target learning tasks are very similar: either (some of) the auxiliary data can be directly used as training examples for the target task or the auxiliary and the target data share the same representation. However, in many cases the connection between the auxiliary and the target tasks can be remote. Only a few features derived from the auxiliary data may be helpful for the target learning. We call such scenario the deep transfer-learning scenario and we introduce a novel transfer-learning method for deep transfer. Our method uses restricted Boltzmann machine to discover a set of hierarchical features from the auxiliary data. We then select from these features a subset that are helpful for the target learning, using a selection criterion based on the concept of kernel-target alignment. Finally, the target data are augmented with the selected features before training. Our experiment results show that this transfer method is effective. It can improve classification accuracy by up to more than 10%, even when the connection between the auxiliary and the target tasks is not apparent.**

I. Introduction

When learning a complex task, we often need a large number of training examples to achieve a good performance. However, training examples are difficult to obtain. Human learning faces the same challenge but we use prior knowledge and experiences to alleviate the problem. Transfer learning [1], [2], [3], [4] is a learning paradigm that aims to follow a similar principle for developing better learning systems. In transfer learning, related learning tasks and data are explored to help learn a different target task. The former is often called the *auxiliary* tasks or data and the later the *target* task. Although there have been many research efforts in this direction, the extent to which knowledge transfer can be applied is quite limited. Many current transfer-learning approaches assume that the auxiliary and the target tasks are very similar. For example, in the instance-base transfer paradigm [5], [6], the instances in the auxiliary data are directly used as training examples for the target learning.

Human learning employs much deeper knowledge transfer. For example, when learning tennis, one's training and experiences on running may help. Although this training is not closely related to eye-hand coordination needed for playing tennis, it improves one's general motor skills and thus facilitates the learning. Similar scenarios may arise in machine learning. The auxiliary tasks may not match the target task closely but some (not all) features derived from the auxiliary data may still be applicable to the target task. We call transfer learning in this scenario the *deep* transfer learning.

We propose a feature-based transfer framework. Rather than reusing the model learned for the auxiliary task, we discover features from the auxiliary data in an unsupervised fashion. The features are then vetted using the target data. Those features that can help the target learning are then employed. We use restricted Boltzmann machine [7], [8] for feature discovery. The discovery process does not rely on the labels of the auxiliary data, because such labels may have little connection to the target labels. Our feature vetting (selection) is based on the concept of kernel alignment [9], [10]. Our experiment results show that this transfer mechanism is effective. The target learning can be improved significantly even when the auxiliary data are not closely connected to the target task. Our main contribution is the combination of the feature discovery mechanism based on restricted Boltzmann machine and the feature selection mechanism based on kernel-target alignment.

II. Deep Transfer Learning via Restricted Boltzmann Machine

We denote by $U = \{x^i\}_{i=1}^M$ the set of M auxiliary examples and $T = \{(x, y)^j\}_{j=1}^N$ the set of N target training examples. In both cases, x is the attribute vector of the example and $y \in \{+1, -1\}$ is the label of the example. We ignore the labels of the auxiliary data. For document classification, we consider the bag-of-words representation. x is the vector of word frequencies.

A. Feature Discovery via Restricted Boltzmann Machine

We assumes that there is a set of hidden features applicable to both the auxiliary and the target data. The data can be described (encoded) well using the features. The goal of feature discovery is to construct a family of feature functions (corresponding to these hidden features) from the auxiliary data.

978-0-7695-4607-0/11 $26.00 © 2011 IEEE
DOI 10.1109/ICMLA.2011.51

We distinguish the term "attribute" from the term "feature." We use *attribute* to refer to the raw data, e.g., the word frequencies used to represent a document. We use *feature* to denote a function that maps the raw data to a real value that describes a certain property of the data. Formally, we view the attribute vector as a vector of random variables $< x_1, x_2, \ldots, x_d >$ and the instances in the auxiliary dataset U as the samples of the random vector. We focus on binary features. Ideally, $f_j = 1$ if the feature f_j applies to an instance x and $f_j = 0$ otherwise. The features may be probabilistic. We treat them as random variables too and define the following probability:

$$P(f_j = 1|x) = \frac{e^{c_j + \sum w_{ij} x_i}}{1 + e^{c_j + \sum w_{ij} x_i}} = \text{sigm}(c_j + \sum w_{ij} x_i). \tag{1}$$

with parameters c_j and w_{ij}. We have two sets of random variables: one observable set $\{x_i\}$ and one hidden set $\{f_i\}$. Eq. 1 defines only the dependency of f on x. We further modify the dependency and make it mutual. This gives rise to a restricted Boltzmann machine [7], [8].

Following the probability definition of the restricted Boltzmann machine, the joint probability of the observable and the hidden random variables can be specified as:

$$P(f, x) \propto e^{\sum_j f_j c_j + \sum_i x_i b_i + \sum_{i,j} f_j x_i w_{i,j}}.$$

Our feature discovery process then naturally translates to learning this feature probability, i.e., the parameters c_j and w_{ij}, such that the probability of the data, i.e., $P(x)$ can be maximized.

We use the parameter-learning process introduced in [11], [12] to obtain the parameters. We give here a brief description of the process and refer the readers to [11], [12] for details. Let $G(x, f) = \sum_j f_j c_j + \sum_i x_i b_i + \sum_{i,j} f_j x_i w_{i,j}$ and $Z_0 = \sum_x \sum_f e^{G(x,f)}$. The marginal probability of the data is $\sum_f \frac{e^{G(x,f)}}{Z_0}$. Let $S(x) = \log \sum_f e^{G(x,f)}$ and $Z = \sum_x e^{S(x)}$. We have $P(x) = \frac{e^{S(x)}}{Z}$. Denote by θ the parameter vector. The average log-likelihood over all the data is then:

$$\mathbf{E}_{\hat{P}} \left[\frac{\partial \log P(x)}{\partial \theta} \right] = \mathbf{E}_{\hat{P}} \left[\frac{\partial S(x)}{\partial \theta} \right] - \mathbf{E}_{P} \left[\frac{\partial S(x)}{\partial \theta} \right].$$

where $\hat{P}$ is the empirical distribution of the data and P is the model distribution. Although the model distribution often cannot be calculated analytically, one can sample from P using Gibbs sampling and estimate the gradient using the samples. We follow the contrastive divergence approach developed in [11], [12] to approximate the gradient and train the restricted Boltzmann machine.

Once we obtain the parameters, we can calculate the probability that a particular feature applies to an instance. In fact, we use this probability to define a feature function:

$$\phi_i(x) = \mathbf{E}(f_i) = P(f_i = 1|x) \tag{2}$$

that transforms the raw data into a feature value. To obtain more complex functions, we apply recursively the restricted Boltzmann machine architecture on top of the features f_i. We may repeat this process to derive a multi-layer architecture and thus more and more complex features, i.e., $f_i^{(2)}, f_i^{(3)}, \ldots$. The architecture can be trained bottom up in a layer by layer fashion where $\{f_i^{(j-1)}\}$ are used as the observed variables and $\{f_i^{(j)}\}$ as the hidden variables. Once the parameters at all the layers are obtained, a feature function at layer n is defined recursively as:

$$\phi^n(x) = \text{sigm}(c^n + \sum w_j^n \phi_j^{n-1}(x)). \tag{3}$$

with $\phi_j^0(x) = x_j$. The feature functions are vetted and then used to augment the data for classification.

B. Feature Selection and Data Augmentation

Because in a deep transfer scenario, the auxiliary data are not closely related to the target task, not all the hidden features will be helpful. We eliminate irrelevant features using feature selection based on the concept of kernel-target alignment [9], [10]. Let $X =< x^1, x^2, \ldots, x^N >$ be the data matrix that contains the N target training examples and $y =< y^1, y^2, \ldots, y^N >^T$ the vector of the corresponding labels. The (linear) kernel matrix K can be calculated by $K = X^T X$. On the other hand, a kernel that is ideal for classification should take the form $K^{\text{ideal}} = yy^T$. If we expand K and K^{ideal} into two vectors by stacking the columns respectively, (we abuse notation and denote by K and K^{ideal} the two vectors as well.) the goodness of the kernel K in terms of classification can be measured by how well it aligns with K^{ideal} [9], i.e., by the quantity

$$\text{Align}(K) = \frac{K^T \cdot K^{\text{ideal}}}{\sqrt{K^T K} \sqrt{(K^{\text{ideal}})^T K^{\text{ideal}}}}.$$

We estimate the goodness of a feature $\phi(x)$ by the alignment of the kernel K_ϕ derived from that feature where $K_\phi =< \phi(x^1), \phi(x^2) \ldots, \phi(x^N) >^T < \phi(x^1), \phi(x^2) \ldots, \phi(x^N) >$.

After selecting a set of t features $\phi_1, \phi_2, \ldots, \phi_t$ using kernel alignment, we further measure the alignment of the kernel based on all the raw attributes and the alignment of the kernel based on all the selected features. Let Let b be the ratio of the later over the former, we augment the data in a weighted way such that the raw attribute vector x is transformed to be:

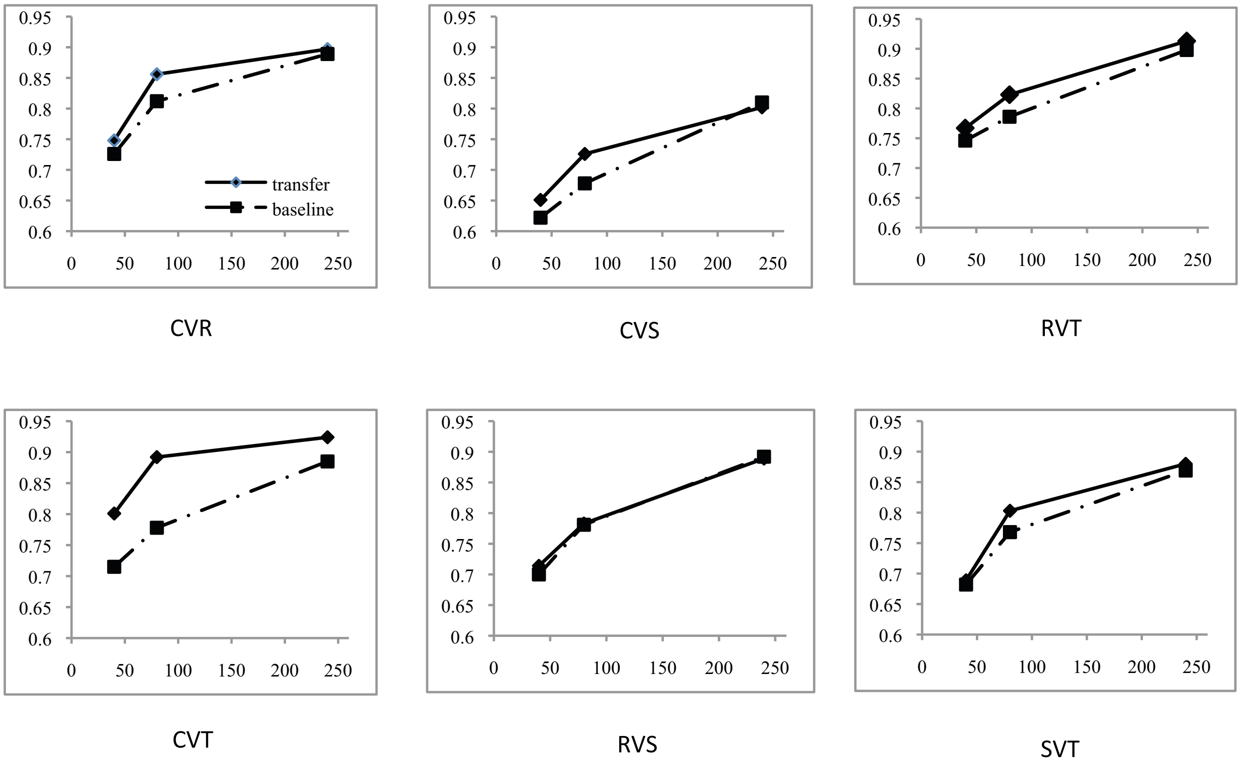
$$\hat{x} =< x_1, x_2, \ldots, x_d, b \cdot \phi_1(x), b \cdot \phi_2(x) \ldots, b \cdot \phi_t(x) >^T. \tag{4}$$

We then train a standard classifier on the augmented vector for classification.

III. Experiment Results

A. Datasets and Experiments

We use the 20-newsgroup dataset [13] in our experiments. Different from previous work in which the auxiliary and



CVR

CVS

RVT

CVT

RVS

SVT

Figure 1. Comparison of the Classification Accuracy between Transfer Learning and Baseline. x-axis: training sizes, 40, 80 and 240 are used. y-axis: classification accuracy. CVR: comp v.s. rec (sci and talk as aux); CVS: comp v.s. sci (rec and talk as aux); RVT: rec v.s. talk (comp and sci as aux); CVT: comp v.s. talk (rec and sci as aux); RVS: rec v.s. sci (comp and talk as aux); SVT: sci v.s. talk (comp and rec as aux).

the target data are from the same high level category (e.g., using "rec.autos" as auxiliary data for target task involving "rec.motocycles"), we use newsgroups from completely different top-level categories for auxiliary and target data. To evaluate the effectiveness of our transfer framework, we compared the classification accuracy with transfer to that of the baseline. The baseline was obtained by training and classification on the raw attribute vector. We used linear SVM (LibSVM [14]) for the classification. The articles in the 20 newsgroup dataset were pre-processed using the rainbow tools [15]. The pre-process removed the stop words and the header from each article.

B. Results

Fig. 1 plots the experiment results. For each classification task, we tested 3 different training sizes: 40, 80 and 240. For each experiment setting (classification pair + training size), we repeated the experiment 20 times with different random samples. In the subplot, the x-axis represents the training size and the y-axis represents the classification accuracy (the fraction of the test instances correctly classified). The accuracy value is the average of the 20 repetitions. In all except two cases, the discovered features lead to better classification. There is a case where by adding the discovered features, the classification accuracy is improved by more than 10%. (In comp v.s. talk, the accuracy is improved from 77.8% to 89.2% for the training size 80.)

We observed that the most improvement was obtained when training size was intermediate. When there are only a few training examples, the improvement may be small and so is the case when there are a lot of training examples. This is expected. With a lot of training examples, a good model can be learned and it may be impossible to make further improvement using not-so-closely-related auxiliary data. On the other hand, when the training set is small, the improvement can be small too. This is because the goodness of the features with respect to the target learning is tested using the target training set. With a small training set, feature selection may not be effective and thus the improvement is small if not negative.

IV. Related Work

Transfer learning is an area of intensive research. A comprehensive and detailed discussion on transfer learning can be found in the excellent survey given by Pan and Yang [16].

One transfer paradigm is the instance-based transfer learning. In such learning, the auxiliary data are combined with the target data to train a model for the target task [5], [6]. Clearly, to be effective, it requires that the auxiliary data and the target data be very close. Another transfer paradigm is the feature-presentation-based transfer learning. In such learning, a new common representation of the data is learned using the auxiliary (and the target) data. The common representation may improve learning accuracy for both the auxiliary and the target tasks [17], [18]. This approach differs from ours in that once a representation is learned, the data are completely presented in the format of this representation. In our case, the representation includes both the raw attributes and the selected newly-discovered features. Not all new features lead to a good representation since they are obtained from not-closely-related auxiliary data. Self-taught learning [18] can be viewed as a feature-presentation-based transfer where the new representation is learned in an unsupervised fashion.

V. Conclusion

In this paper, we introduced a novel framework for deep transfer learning where the auxiliary data are not closely related to the target learning task. In this scenario, one cannot directly include the auxiliary data into the target training and one cannot assume that there is a common representation that works for both the auxiliary and the target learning tasks. Our transfer framework is based on feature discovery and transfer. Using a hierarchical restricted Boltzmann machine architecture, we discover features that may be helpful for target learning from the auxiliary data. These features are further vetted using kernel-target alignment and only the ones beneficial to the target learning are employed to augment the target data. A standard learner can then be applied on the augmented target data to learn the target task. Our experiments show that this transfer framework is very effective. In some cases, transfer increases classification accuracy by more than 10%. This is significant because the auxiliary data are from categories very different from the ones involved in the target task.

Acknowledgment

This work was supported in part by NSF grant IIS-0905478 and Louisiana Board of Regents grant LEQSF(2009-11)-RD-A-09. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of National Science Foundation.

References

[1] S. P. Singh, "Transfer of learning by composing solutions of elemental sequential tasks," *Machine Learning*, vol. 8, pp. 323–339, 1992.

[2] S. Thrun, "Is learning the n-th thing any easier than learning the first?" in *NIPS*, 1995, pp. 640–646.

[3] J. Baxter, "A bayesian/information theoretic model of learning to learn via multiple task sampling," *Machine Learning*, vol. 28, no. 1, pp. 7–39, 1997.

[4] R. Caruana, "Multitask learning," *Machine Learning*, vol. 28, no. 1, pp. 41–75, 1997.

[5] W. Dai, Q. Yang, G. Xue, and Y. Yu, "Boosting for transfer learning," in *Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML)*, 2007, pp. 193–200.

[6] T. Kamishima, M. Hamasaki, and S. Akaho, "Trbagg: A simple transfer learning method and its application to personalization in collaborative tagging," in *The Ninth IEEE International Conference on Data Mining*, 2009, pp. 219–228.

[7] P. Smolensky, "Information processing in dynamical systems: Foundations of harmony theory," in *Parallel Distributed Processing*, D. E. Rumelhart and J. L. McClelland, Eds., 1986, vol. 1, pp. 194–281.

[8] Y. Freund and D. Haussler, "Unsupervised learning of distributions of binary vectors using 2-layer networks," in *NIPS*. Morgan Kaufmann, 1991, pp. 912–919.

[9] N. Cristianini, J. Kandola, A. Elisseeff, and J. Shawe-Taylor, "On kernel-target alignment," in *Advances in Neural Information Processing Systems 14*, 2002, pp. 367–373.

[10] G. R. G. Lanckriet, N. Cristianini, P. L. Bartlett, L. E. Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," in *Machine Learning, Proceedings of the Nineteenth International Conference (ICML)*, 2002, pp. 323–330.

[11] M. A. Carreira-Perpinan and G. E. Hinton, "On contrastive divergence learning," in *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics (AISTAT)*, 2005, p. 3340.

[12] Hinton and Salakhutdinov, "Reducing the dimensionality of data with neural networks," *SCIENCE: Science*, vol. 313, 2006.

[13] K. Lang, "NewsWeeder: learning to filter netnews," in *Proc. 12th International Conference on Machine Learning*, 1995, pp. 331–339.

[14] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.

[15] A. K. McCallum, "Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering," 1996, http://www.cs.cmu.edu/∼mccallum/bow.

[16] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng*, vol. 22, no. 10, pp. 1345–1359, 2010.

[17] A. Argyriou, T. Evgeniou, and M. Pontil, "Multi-task feature learning," in *NIPS*. MIT Press, 2006, pp. 41–48.

[18] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: transfer learning from unlabeled data," in *Proceedings of the Twenty-Fourth International Conference on Machine Learning(ICML)*, 2007, pp. 759–766.