Dictionary based Sparse Representation for Domain Adaptation

Rishabh Mehrotra
BITS Pilani
Rajasthan, India
erishabh@gmail.com

Rushabh Agrawal
BITS Pilani
Rajasthan, India
rushabh0803@gmail.com

Syed Aqueel Haider
MIT Manipal
Karnataka, India
sahrizvi@ymail.com

ABSTRACT

Machine Learning algorithms are often as good as the data they can learn from. Enormous amount of unlabeled data is readily available and the ability to efficiently use such amount of unlabeled data holds a significant promise in terms of increasing the performance of various learning tasks. We consider the task of *supervised* Domain Adaptation and present a Self-Taught learning based framework which makes use of the K-SVD algorithm for learning sparse representation of data in an unsupervised manner. To the best of our knowledge this is the first work that integrates K-SVD algorithm into the self-taught learning framework. The K-SVD algorithm iteratively alternates between sparse coding of the instances based on the current dictionary and a process of updating/adapting the dictionary to better fit the data so as to achieve a sparse representation under strict sparsity constraints. Using the learnt dictionary, a rich feature representation of the few labeled instances is obtained which is fed to a classifier along with class labels to build the model. We evaluate our framework on the task of domain adaptation for sentiment classification. Both self-domain (requiring very few domain-specific training instances) and cross-domain classification (requiring 0 labeled instances of target domain and very few labeled instances of source domain) are performed. Empirical comparisons of self-domain and cross-domain results establish the efficacy of the proposed framework.

Categories and Subject Descriptors

H.3.3 [**Knowledge Management**]: Classification and Clustering

General Terms

Algorithms

Keywords

Transfer Learning, Sparse Representation, Domain Adaptation

1. INTRODUCTION

Machine Learning algorithms are often as good as the data they can learn from. The expense involved and the difficulty of obtaining labeled data poses a severe bottleneck in the applicability of several machine learning algorithms for classification tasks. Enormous amount of unlabeled data is readily available and thus the ability to use such amount of unlabeled data holds significant promise in terms of increasing the performance of various learning tasks.

Raina et al[3] proposed a new machine learning framework called Self-Taught Learning for using unlabeled data in supervised classification task. Self-Taught Learning requires that the structure learned from unlabeled data be “useful” for representing data from the classification task. Specifically considering the case of sentiment classification of reviews and domain adaptation, a large amount of labeled instances in each of the domains are required to achieve a reasonable classification accuracy. If automatic sentiment classification were to be used across a wide range of domains, the effort to annotate corpora for each domain may become cumbersome. Learning a different system for each of the domain would prevent us from exploiting the information shared across domains.

A better strategy could be to learn a single system from the set of domains for which labeled and unlabeled data are available and then apply it to any target domain. For this strategy to succeed the system should be able to discover intermediate abstractions that are shared and meaningful across domains. Other challenges faced while performing classification tasks include the problem of high dimensionality of observed data: an excessively large number of data features which hinder the model-building process and the inability to represent data (signals, images, etc.) in the most parsimonious terms. Models that propose that the signal of interest is sparse in some transform domain are preferred.

We feel that the intermediate abstractions learnt by the dictionary could yield a better transfer across domains. Some of the customer sentiments on product quality, product servicing, etc. make sense across a wide range of domains. Since the same words or tuples of words may be used across domains to indicate the presence of these higher-level concepts, it should be possible to discover them.

In this paper we present a framework which proposes solution to the above mentioned problems. We use the K-SVD algorithm (Aharon et al[2]) to learn a dictionary based sparse representation using unlabeled reviews spanning multiple domains. Given a set of unlabeled instances, we seek the dictionary that leads to the best representation for each member in this set, under strict sparsity constraints. We then learn the representation of few labeled instances based on the obtained dictionary and feed it to a classifier. The K-SVD algorithm makes use of unlabeled instances to learn a representation that models intermediate abstractions across the domains thereby fulfilling the objective of Self-Taught Learning while at the same time solving the problem of high dimensionality by keeping a check on the number of dictionary elements to train. We evaluate our framework on a standard Amazon review dataset created by Blitzer et al [1].

2. SPARSE REPRESENTATION

Sparse and redundant representation modeling of data assumes an ability to describe signals (y∈R^n) as linear combinations of a few atoms $\{d_j\}_{j=1}^{k}$ from a pre-specified dictionary. As such, the

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.
Copyright 2012 ACM 978-1-4503-1156-4/12/10...$15.00

choice of the dictionary that sparsifies the signals is crucial for the success of this model. Representing a signal involves the choice of a dictionary, which is the set of elementary signals or atoms used to decompose the signal. When the dictionary forms a basis, every signal is uniquely represented as the linear combination of the dictionary atoms.

2.1 Sparse Coding and Dictionary Training

Sparse coding is the process of computing the representation coefficients x based on the given signal y and the dictionary D. This process, commonly referred to as “atom decomposition,” requires solving:

$$\boldsymbol{min_x}\|\boldsymbol{x}\|_{\mathbf{0}} \text{ subject to } \mathbf{y = Dx} \qquad (1)$$

or

$$\boldsymbol{min_x}\|\boldsymbol{x}\|_{\mathbf{0}} \text{ subject to } \|\mathbf{y\text{-}Dx}\|_{\mathbf{2}} \leq \boldsymbol{\epsilon} \qquad (2)$$

where ‖.‖o is the 1o norm, counting the nonzero entries of a vector; and this is typically done by a “pursuit algorithm” that finds an approximate solution.

Sparse Coding is a necessary stage in the K-SVD algorithm we describe later. Exact determination of the sparsest representation is an NP hard problem and thus various approximate solutions have been proposed. The simplest ones are the matching pursuit (MP)[5] and the orthogonal matching pursuit (OMP) algorithms[6][7].

Dictionary training is a much more recent approach to dictionary design, and as such, has been strongly influenced by the latest advances in sparse representation theory and algorithms. The most recent training methods focus on L0 and L1 sparsity measures, which lead to simple formulations and enable the use of recently developed efficient sparse-coding techniques[8][9]. We next describe the KSVD algorithm (as proposed by [3]) used for learning adaptive dictionaries.

2.2 K-SVD Algorithm

Given a set of examples Y, the goal of the K-SVD is to find a dictionary D and a sparse matrix X which minimize the representation error,

$$\boldsymbol{argmin_{D,X}} \|\boldsymbol{Y - Dx}\|_F^2 \text{ subject to } \left\|\boldsymbol{\gamma_i}\right\|_{\mathbf{0}}^{\mathbf{0}} \leq \boldsymbol{T} \ \ \forall_{\boldsymbol{i}}$$

where $\boldsymbol{\gamma_i}$ represent the columns of X, and the L0 sparsity measure; $\|.\|_{\mathbf{0}}^{\mathbf{0}}$counts the number of non-zeros in the representation. The K-SVD algorithm alternates between sparse-coding and dictionary update steps.

Sparse coding is performed for each signal individually using any standard technique. The objective function is

$$\mathbf{min}\ _{\mathbf{D,X}}\|\boldsymbol{Y - DX}\|_F^2 \text{ subject to } \forall_{\boldsymbol{i}}\,, \|\boldsymbol{x_i}\|_{\mathbf{0}} \ \leq \boldsymbol{T_0}$$

The penalty term can be written as:

$$\|\boldsymbol{Y - DX}\|_F^2 \ = \ \sum_{\boldsymbol{i=1}}^{\boldsymbol{N}}\|\boldsymbol{Y - Dx_i}\|_{\mathbf{2}}^{\mathbf{2}} \qquad \text{-(3)}$$

Thus the objective function can further be decomposed to N distinct problems of the form:

$$\mathbf{min} \quad _{\boldsymbol{x_i}}\|\boldsymbol{y_i - Dx_i}\|_{\mathbf{2}}^{\mathbf{2}}$$

subject to

$$\|\boldsymbol{x_i}\|_{\mathbf{0}} \ \leq \boldsymbol{T_0} \ \ \boldsymbol{for\ i = 1, 2, \ldots., N}$$

This problem is adequately addressed by the pursuit algorithms discussed above. For the part of dictionary updating, only 1 column of the dictionary and the corresponding row in X is questioned, thereby re-writing the penalty term:

$$\begin{aligned}\|\boldsymbol{Y - Dx}\|_F^2 \ &= \ \left\|\boldsymbol{Y} - \textstyle\sum_{\boldsymbol{j=1}}^{\boldsymbol{K}} \boldsymbol{d_j X_T^j}\right\|_F^{\mathbf{2}} \\ &= \left\|\left(\boldsymbol{Y} - \sum_{\boldsymbol{j \neq k}} \boldsymbol{d_j X_T^j}\right) - \ \boldsymbol{d_k X_T^k}\right\|_F^{\mathbf{2}} \\ &= \|\boldsymbol{E_k - d_k X_T^k}\|_F^2 \qquad\qquad - (\mathbf{4})\end{aligned}$$

Defining ω_k as the group of indices pointing to the examples $\{\boldsymbol{y_i}\}$ that use the atom $\boldsymbol{d_k}$ i.e. those where x_T^k is zero. Ω_k is defined as the matrix of size $\boldsymbol{N} \times |\boldsymbol{\omega_k}|$ with ones on the $(\boldsymbol{\omega_k}(\boldsymbol{i}), \boldsymbol{i})\boldsymbol{th}$ position and zeroes elsewhere.

When multiplying $\boldsymbol{X_R^k = X_T^k \Omega_k}$, this shrinks the row vector by discarding of the zero entries, resulting with the row vector of length $|\boldsymbol{\omega_k}|$. Thus the equation (4) becomes:

$$\|\boldsymbol{E_k \Omega_k - d_k X_T^k \Omega_k}\|_F^2 = \|\boldsymbol{E_k^R - d_k X_R^k}\|_F^2$$

and SVD can be used to find the final solution. For details the reader is directed to [3].

We use the K-SVD algorithm to train a dictionary based on all the unlabeled instances of all the domains and get their corresponding co-efficient matrix using OMP. Henceforth we use the rows of the representation matrix X as our feature vectors.

3. SUPERVISED DOMAIN ADAPTATION

Domain adaptation considers the setting in which the training and testing data are sampled from different distributions. Assume we have two sets of data: a source domain S providing labeled training instances and a target domain T providing instances on which the classifier is meant to be deployed. We do not make the assumption that these are drawn from the same distribution, but rather that S is drawn from a distribution p_S and T from a distribution p_T. The learning problem consists in finding a function realizing a good transfer from S to T.

We consider the case of supervised Domain Adaptation, a setting where we have a large amount of labeled data from some source domain, a large amount of unlabeled data from a target domain, and additionally a small budget for acquiring labels in the target domain. In our setting we experimented on 3 different values of the fixed budget (5%, 10% & 20%) of the total unlabeled instances we had initially in the self-domain. The instances which constitute those percentages were randomly selected and the labels for those instances were acquired from the oracle**.**

Table 1 : Self-Domain Sentiment Classification

Domain	Books			DVD			Electronics			Kitchenware		
% dataset used for training / Learning Algorithm	5	10	20	5	10	20	5	10	20	5	10	20
Voted Perceptron	63.3	66.8	72.7	48.9	54.7	53.3	72.7	69.6	79.7	64	72.6	79.4
Logistic Regression	73.8	74.4	77.8	53.1	58.3	59.6	74.8	71.1	76.3	75.4	83.2	85.1
Bayesian Logistic	70.9	75.5	81.7	51.8	59.1	59.0	73.7	83.2	84.6	75.8	85.4	88.3

4. EXPERIMENTAL ANALYSIS

4.1 Dataset Description

Our data was based on the one used by [2]. They used the dataset consisting of product reviews for 4 items – books, DVDs, electronics and kitchenware. The raw dataset was taken from Amazon and consisted of product name, reviewer name, ratings (on a scale of 0-5), rating date and title followed by review text. The ratings were made binary (ratings>3 were written as 1 and rest were made 0) for practical purposes. The review text was in the form of frequencies of various words appearing in the text. The remaining attributes were discarded, as they were found to be too ambiguous. We took the same dataset for our practical experiments (the dataset had 1000 positive and 1000 negative reviews). The nature of our experiments required unsupervised training examples and hence, for each of the 4 items, 1000 labeled (500 positive and 500 negative) reviews were separated out for testing purposes and the rest (totaling 4000) with labels discarded, were used for setting up the dictionary. The same dictionary was used for modeling a sentiment predictor for each of the 4 domains (the domains here being the items) as well as for domain transfer between them (the model building aspect is explained at a later stage).

4.2 Experimental Setup

The basic preprocessing step was the one used by [2] in which the bag-of-words representation with frequency of each word was converted to a simple word presence/absence format. For computational reasons, only the most 5000 frequent unigrams and bigrams were shortlisted for further computations

4.3 Sentiment Classification & Domain Transfer (Self-Domain & Cross-Domain)

The experiments were conducted in the following steps :

Unsupervised Dictionary Learning

The first step was to come up with the dictionary representation using the 4000 unlabeled instances. By setting the appropriate parameters, the size of dictionary representation (the number of attributes in terms of which data instance can be represented) was kept at 50 (thus achieving dimensionality reduction from 5000 to 50). We came up with the dictionary by iteratively running the code till the error threshold was reached. To achieve sparse representation, the no, of coefficients to use in OMP coefficient calculation was empirically kept as 5.

Sparse Coding

The second step was to train the data for a particular domain. The remaining 4000 instances (1000 labeled ones from each of the domains, after removing their labels) were represented in terms of the dictionary achieved in the step 1 using OMP coefficient calculations.

Self-domain Sentiment Classification

The coefficient matrices obtained by the sparse coding stage were divided into training and testing sets of various sizes. In this paper, we show the results for training on 50,100 and 200 instances respectively (corresponding to 5%, 10% and 20% respectively in tables and figures) and testing on the remaining ones (from the 1000 labeled ones). The dictionary learned from unsupervised learning allowed training on so few samples to achieve reasonably satisfactory results. The algorithms used were Voted Perceptron, Logistic Regression and Bayesian Logistic Regression algorithms. The results are enumerated in Table 1.

Domain Transfer: Budgeted Domain Adaptation

The second objective of our method was to enable cross-domain training to allow a single model to be applied in other domains as well. This is accomplished by training on one of the domains and testing on the remaining 3 domains using the same 3 algorithms mentioned earlier. The results are enumerated in Table 2.

Table 2 : Domain Transfer Results

Test Domain / Train Domain	Books	DVDs	Electronics	Kitchenware
Books	-	63.4	87.9	89.8
DVDs	82.7	-	83.1	85.7
Electronics	82.7	69.5	-	90.92
Kitchenware	85.7	63.3	90.3	-

The next section provides a brief analysis of the results.

5. CONCLUSION

Below are a few points summarizing the results achieved and how they stack up against our initial aims.

1.As visible, the results of self-domain training in case of minimal labeled examples is comparable to that of cross-domain training, thus fulfilling the initial aims of the algorithm.

2. The results also show that as theorized, building representations from multiple sources tends to trap some higher level and more abstract patterns and structures in data as well. This accounts for the results achieved in both Table 1 and Table 2.

3. Like in the case of [18] , the quality of our features obtained allow such a cross-domain accuracy but unlike in their case, we achieved these results without using any kind of hierarchical or multi-layer k-SVD algorithm (they had used stacked Denoising auto-encoder). In addition we also achieved a dimension reduction from 5000 attributes to mere 50 attributes along with a sparse representation (the no, of coefficients to use in OMP coefficient calculation was kept as 5) . Also, they had used 80% ie 1600 samples for training purpose, much higher than our own.

4. The results for cross-domain were achieved without any prior knowledge about the source and target domains. Important observation in case of cross-domain results is that the domains are not necessarily significantly related to each other (like kitchenware and books, for example). But our dictionary manages to achieve commendable results in-spite of this.

5. Further improvements in results for practical purposes is further possible by implementing an hierarchical version of the K-SVD algorithm. Another aspect is that for building up the dictionary, unlabeled reviews from more varied domains could be taken for a better representation.

One important factor to note is that apart from the algorithmic difference from the previous approaches, the key difference in our framework is that in coherence to the Self-Taught Learning principle, we learn sparse representation using unlabeled instances from all four domains. We believe that such accuracies for domain transfer are achieved due to the rich representation learnt to represent reviews in our algorithm.

6. REFERENCES

[1] Blitzer, J., McDonald, R., and Pereira, F. (2006). Domain adaptation with structural correspondence learning. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'06).

[2] Blitzer, Dredze, Pereira F.(2007) Biographies, Bollywood, boom-boxes and blenders: Domain Adaptation for sentiment classification. In proceedings of ACl 2007.

[3] Daume III, Marcu D. (2006) Domain Adaptation for Statistical Classifiers. Journal of Artificial Intenlligence Research, 26, 101-126.

[4] M. Aharon, M. Elad, and A. M. Bruckstein, K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representation Technion—Israel Inst. of Technology, 2005, Tech. Ref

[5] S. Chen, S. A. Billings, and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification," Int. J. Contr., vol. 50, no. 5, pp. 1873–96, 1989

[6] J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," IEEE Trans. Inf. Theory, vol. 50, pp. 2231–2242, Oct. 2004.

[7] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in Conf. Rec. 27th Asilomar Conf. Signals, Syst. Comput., 1993, vol. 1.

[8] G. Davis, S. Mallat, and Z. Zhang, "Adaptive time-frequency decompositions," Opt. Eng., vol. 33, no. 7, pp. 2183–91, 1994.

[9] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," SIAM Rev., vol. 43, no. 1, pp. 129–159, 2001.

[10] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," Technical Report – Statistics, Stanford, 1995.

[11] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," SIAM Review, vol. 51, no. 1, pp. 34–81, 2009.

[12] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," 1993 Conference Record of The 27th Asilomar Conference on Signals, Systems and Computers, pp. 40–44, 1993.

[13] I. F. Gorodnitsky and B. D. Rao, "Sparse signal reconstruction from limited data using FOCUSS: a re-weighted minimum norm algorithm," IEEE Trans. Signal Process., vol. 45, no. 3, pp. 600–616, 1997.

[14] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," Appl. Comput. Harmon. Anal., vol. 26, no. 3, pp. 301–321, 2009.

[15] S. A. Haider, R. Mehrotra, "Corporate News Classification and Valence Prediction: A Supervised Approach", Proc. Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis, ACL-HLT 2011, pages 175–181, 24 June, 2011, Portland, Oregon, USA.

[16] S. Lesage, R. Gribonval, F. Bimbot, and L. Benaroya, "Learning unions of orthonormal bases with thresholded singular value decomposition," Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., vol. 5, pp. 293– 296, 2005.

[17] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis (GPCA)," IEEE Trans. Pattern Anal. Mach. Intell., vol. 27, no. 12, pp. 1945–1959, 2005.

[18] Glorot,Bordes, Bengio(2011) Domain Adaptation for Large-Scale Sentiment Classifiation: A Deep Learning Approach. Internation Conferene of Maine Learning 2011.