International Journal of Advances in Computer Science & Its Applications
Volume 6 : Issue 1 [ISSN 2250-3765]
Publication Date : 18 April, 2016

Transitive Associations for Domain Transfer Problem on Opinion Mining

Yilmaz Ar

***Abstract*—Classification algorithms need labeled examples to train their model. However there are not enough labeled examples in some domains. There is an approach that training a classifier in one domain and use it to classify examples on different domains. This method is not always successful and this is called domain transfer problem. Spectral Feature Alignment is proposed as a solution to this problem [3]. In this study I investigate this algorithm with a demonstrative example and I do classification experiments on randomly created datasets. Support vector machines are used as a classification tool and the aim of the experiments is to find the impact of the spectral feature alignment on the classification accuracy. Based on the results of these experiments, I will discuss the possible research opportunities on this area.**

***Keywords*—opinion mining, spectral feature alignment**

I. Introduction

The increasing use of new web technologies provides new opportunities to Internet users for expressing themselves on the virtual world. They write many reviews about their experiences on the services or on the products they have bought. The reviews are very valuable source of information for companies. They use reviews to evaluate their services or products. They can observe the shortcomings, the deficiencies or unsatisfactory parts of their products. Usually the bulk of reviews consist of only text, not any ratings. The companies need to classify these reviews therefore they build classifiers. A sufficient number of labeled examples (rated reviews) are necessary for training. Not always the enough number of rated reviews is available for all domains. Creating a classifier by using labeled examples in one domain and using this classifier to classify unlabeled examples in another domain would be very useful. The first domain is called as source domain and the second domain is called as target domain. The unlabeled examples on both domains would also be used in the process of creating a classifier. However creating a classifier in one domain and using it in another domain does not always give good classification results. This is called domain-transfer-problem and in recent years some approaches are developed for the solution. The main problem of transferring from one domain to other is that users usually prefer different domain specific words to show their ideas in different domains. There should be a connection between the domain specific words of different domains. One of the solutions is spectral feature alignment (SFA) method proposed in [3]. This method tries to establish relationships between the domain specific words in different domains. In the next section, SFA method will be discussed in detail. Section III presents a demonstrative example of SFA algorithm. Section IV provides the experiments performed on artificially created data. Later, in Section V the discussion of the results and the conclusion are given. The last section consists of the future work both on SFA and the transitive associations across domains.

Yilmaz Ar
Ankara University
Turkey

II. Spectral Feature Alignment

First the definitions and the motivation of spectral feature alignment concept [3] is given. Later the algorithm will be explained in detail.

A. *Definitions and Motivation*

The source domain is the domain that contains some amount of labeled data and unlabeled data. The main goal is creating a classifier using labeled data belonging to source domain. The target domain is the domain consisting of unlabeled data and in some cases very small amount of labeled data. The classifier trained on source domain is used to predict the reviews on the target domain. When the classifier is trained only using source domain data, generally the accuracy results on target domain is not good. Usually reviewers use different words in their feedback to evaluate a product or a service in different domains. They are called as domain specific words. For example “fast” and “readable” are domain specific words to laptop and book domains, respectively. Because the domain specific words are different, the classifier trained on the source domain is not successful on the target domain. On the other hand, not all the words are domain specific. Some words can be used in different domains. For example the word “good” may be used both in laptop and book domains. In addition, the word “awful” might be used in both domains. These words are called domain independent words, in other words; domain independent features. They are very important part of cross domain sentiment classification. Some algorithms only depend on these domain independent features. The SFA algorithm uses the domain independent features to create additional features. The key in this approach is the same domain independent features may have close connections to different domain specific words in different domains. These different domain specific features may form clusters and mainly these clusters are added as the new features. Let us assume that the following reviews are obtained from laptop and book domains:

Laptop Domain:
The laptop A is fast and it has built-in virus detection program. It is a good feature. (Positive)
I bought the laptop A two days ago, its performance is excellent. In addition it looks reliable. (Positive)
This computer has awful design and the battery life is very short. (Negative)

International Journal of Advances in Computer Science & Its Applications
Volume 6 : Issue 1 [ISSN 2250-3765]
Publication Date : 18 April, 2016

Book Domain:
The book B is very easy to read. The cover design is good. (Positive)
I read the book B and the story is impressive. The author wrote an excellent novel. (Positive)
The language use is awful and the story is boring. (Negative)

Domain Ind. Features	Domain Specific Features					
	fast	*readable*	*reliable*	*impressive*	*short − battery − life*	*boring*
good	1	1	0	0	0	0
excellent	0	0	1	1	0	0
awful	0	0	0	0	1	1

Figure 1. The co-occurrence relationship of domain independent and domain specific features for the motivating example

Fig 1. shows the co-occurrence relationship of domain independent and domain specific features for the example given above.

B. *Spectral Feature Alignment (SFA) Algorithm*

SFA algorithm was proposed in [3] and tries to find the feature alignment mapping function. The algorithm uses the idea of spectral clustering proposed in [2]. The feature alignment mapping function is used to create additional features for each example. Input: Labeled source domain data and unlabeled target domain data, the number of clusters k, the number of total features m, and the number of domain independent variables l. Output: Feature alignment mapping function.
The first step of SFA algorithm is determining domain independent features (l) among all features (m). Mutual information and frequency are used together on selecting domain independent features. Mutual information measures the relations between features and domains. A feature is assumed to be domain independent if it has low mutual information. Furthermore the domain independent features should appear frequently. Mutual information and frequency criteria are implemented as follows:

$$\mathbf{I}(\mathbf{X}^{\mathbf{i}};\mathbf{D}) = \sum_{d \in D} \sum_{x \in X^i, x \neq 0} p(x,d) \log_2 \left(\frac{p(x,d)}{p(x)p(d)} \right) \tag{1}$$

where $\mathbf{D}$ is a domain variable and $\mathbf{X}^{\mathbf{i}}$ is a specific feature. The features that has smaller $\mathbf{I}(\mathbf{X}^{\mathbf{i}};\mathbf{D})$ values are chosen as domain independent. If any feature does not appear in one of the domains, it is eliminated and not considered as domain independent. Both source and target domains are used to select domain independent features. The remaining m-l features are specified as the domain specific features. In the second step the co-occurrence matrix $\mathbf{M}$ in $\mathrm{R}^{(m-l)\times l}$ is computed by using domain independent and domain specific features. The rows represent the domain specific features and the columns represent the domain independent features. Using $\mathbf{M}$ the affinity matrix $\mathbf{A}$ is created as follows:

$$\mathbf{A} = \begin{bmatrix} 0 & M \\ M^T & 0 \end{bmatrix} \tag{2}$$

In this matrix, the first m-l rows and columns correspond to the m-l domain-specific features and the last l rows and columns correspond to the l domain-independent features. In the next step, the diagonal matrix $\mathbf{D}$ is constructed.

$$\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij} \tag{3}$$

and the matrix

$$\mathbf{L} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2} \tag{4}$$

is computed. The k largest eigenvectors of $\mathbf{L}$, $u1, u2, \ldots, uk$, are found and the matrix

$$\mathbf{U} = \begin{bmatrix} u_1 & u_2 & \cdots & u_k \end{bmatrix} \tag{5}$$

in R^{mxk} is formed. Later the feature alignment mapping function is created.

$$\vartheta(x) = x\mathbf{U}_{[1:m\text{-}l,:]} \tag{6}$$

$U_{[1:m\text{-}l,;]}$ shows the first m-l rows of U and x in $\mathrm{R}^{1x(m\text{-}l)}$. For any data example either in source domain or in target domain, to create a new representation first the domain specific features are extracted ($\Phi_{(DS)}(.)$). Then the feature alignment mapping function is applied on domain specific features to create new representations ($\vartheta(\Phi_{(DS)}(.))$.
Feature augmentation is the last step of the SFA method. Normally, the best approach should augment the domain independent features with the new features. However the algorithm may fail to perform feature alignment perfectly. Therefore all the features are augmented with the new learned features to create the new representation for the example. A trade-off parameter γ may be used to balance the effect of original features and new features. The new feature representation is defined as:

$$x_i = [x_i, \gamma\vartheta(\Phi_{(DS)}(.)] \tag{7}$$

where $x_i \in \mathrm{R}^{1xm}$, $x_i \in R^{1x(m+k)}$ and $0 \leq \gamma \leq 1$ [3].

III. A Demonstrative Example

Fig. 2 shows the manually created data including 12 features and 20 samples for domain A. a_{ij} indicates the existence of j^{th} feature on sample i. If $a_{ij}=1$, then the sample includes the feature.
Fig. 3 displays the same type of data for domain B. The first 10 examples in both figures correspond to reviews implying positive sentiment. The remaining corresponds to the reviews implying negative sentiment. When these two domains are considered together, the first 4 features are specified as domain independent features. Domain independent features are the features which occur in two domains. The next 8 features (features 5-12) are domain specific features which only and/or mostly occur in one domain. Domain specific features consist of the features which are specific to a domain. As shown in Fig. 2 and Fig. 3, features 5-8 are domain specific for the domain A, and the features 9-12 are domain specific to domain B.
Consider the following case: Let domain A and domain B are the Cell Phone and Laptop domains, respectively. Fig. 4 shows the feature numbers and their corresponding explanations. For example according to the matrix in Fig. 2 and Fig. 3, the reviewers used the first four words namely

"large display", "user interface", "storage" and "processing power" in reviews written for both domains.

Number of Examples	Number of Features											
	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	0	1	1	0	0	0	0	0	0
2	0	0	0	0	1	1	0	0	0	0	0	0
3	1	0	0	0	1	1	0	0	0	0	0	0
4	1	0	0	0	1	1	0	0	0	0	0	0
5	1	0	0	0	1	1	0	0	0	0	0	0
6	0	0	0	0	0	1	1	0	0	0	0	0
7	0	0	0	0	0	1	1	0	0	0	0	0
8	0	1	0	0	0	1	1	0	0	0	0	0
9	0	1	0	0	0	1	1	0	0	0	0	0
10	0	1	0	0	0	1	1	0	0	0	0	0
11	1	0	1	0	0	0	1	1	0	0	0	0
12	1	0	1	0	0	0	1	1	0	0	0	0
13	0	0	1	0	0	0	1	1	0	0	0	0
14	0	0	1	0	0	0	1	1	0	0	0	0
15	0	0	1	0	0	0	1	1	0	0	0	0
16	0	1	0	1	1	0	0	1	0	0	0	0
17	0	1	0	1	1	0	0	1	0	0	0	0
18	0	0	0	1	1	0	0	1	0	0	0	0
19	0	0	0	1	1	0	0	1	0	0	0	0
20	0	0	0	1	1	0	0	1	0	0	0	0

Figure 2. The artificially created data for domain A.

Number of Examples	Number of Features											
	1	2	3	4	5	6	7	8	9	10	11	12
1	1	0	1	0	0	0	0	0	1	1	0	0
2	1	0	1	0	0	0	0	0	1	1	0	0
3	1	0	0	0	0	0	0	0	1	1	0	0
4	1	0	0	0	0	0	0	0	1	1	0	0
5	1	0	0	0	0	0	0	0	1	1	0	0
6	0	1	0	1	0	0	0	0	0	1	1	0
7	0	1	0	1	0	0	0	0	0	1	1	0
8	0	1	0	0	0	0	0	0	0	1	1	0
9	0	1	0	0	0	0	0	0	0	1	1	0
10	0	1	0	0	0	0	0	0	0	1	1	0
11	0	0	0	0	0	0	0	0	0	0	1	1
12	0	0	0	0	0	0	0	0	0	0	1	1
13	0	0	1	0	0	0	0	0	0	0	1	1
14	0	0	1	0	0	0	0	0	0	0	1	1
15	0	0	1	0	0	0	0	0	0	0	1	1
16	0	0	0	0	0	0	0	0	1	0	0	1
17	0	0	0	0	0	0	0	0	1	0	0	1
18	0	0	0	1	0	0	0	0	1	0	0	1
19	0	0	0	1	0	0	0	0	1	0	0	1
20	0	0	0	1	0	0	0	0	1	0	0	1

Figure 3. The artificially created data for domain B.

The reviewer for example 1 in Fig. 2 used feature 5 and 6 that are "call quality" and "no ear problems" to indicate positive meaning in his review. On the other hand another reviewer 11 in Fig. 3 used features 11 and 12 that are "bag quality" and "slow DVD player" in the review written for laptop domain to imply negative meaning.

The first 10 examples on both datasets are positive examples and the remaining last 10 examples are negative examples. A SVM classifier was trained on domain A (model A) and another classifier was trained on domain B (model B). The data on domain A was tested by using model B and the accuracy result was 60%; and the data on domain B was tested by using model A and the result 60% was obtained.

Feature Number	Feature Name
1	Large Display
2	User Interface
3	Storage
4	Processing Power
5	Call Quality
6	No Ear Problems
7	Keypad
8	Cause Ear Problems
9	Keyboard
10	Good Dvd Player
11	Bag Quality
12	Slow DVD Player

Figure 4. The real feature names for the given feature numbers

Then SFA method was applied to learn new additional features. The number of additional features is between 1 and 12. With the additional features, the new views of each example consist of features between 13 and 25. The other input for SFA method is the number of domain independent variables; it is between 1 and 10. The SFA algorithm was run for 120 different combinations of domain independent variable and additional representation (feature). For each dataset, a SVM classifier was trained. Each dataset was tested by using the model created on the other domain.

Number of New Features	Number of Domain Independent Variables									
	1	2	3	4	5	6	7	8	9	10
1	60	60	60	60	60	60	60	60	60	60
2	60	60	60	60	60	60	60	60	60	50
3	60	60	70	80	60	60	50	60	60	50
4	60	60	80	100	80	60	50	60	60	50
5	60	50	80	100	80	65	50	60	60	50
6	60	50	80	100	90	50	50	60	60	50
7	60	50	80	100	90	50	80	60	60	50
8	60	50	80	100	100	50	60	60	60	50
9	60	50	80	60	90	80	60	60	60	50
10	60	40	65	30	80	60	60	60	60	50
11	60	55	65	50	60	60	60	60	60	50
12	60	55	60	50	60	60	60	60	60	50

Figure 5. The accuracy results (%) on Domain A using the SVM classifier trained on Domain B with SFA algorithm

The obtained accuracy results are given on Fig. 5 and Fig. 6 for domain A and on Fig. 7 and Fig. 8 for domain B. In these figures the light color areas show the high accuracy entries and the dark color parts represent the low accuracy entries. Each entry represents a different combination of the number of domain independent variable and the number of additional features. As shown, the accuracy is improved significantly especially when the number of domain independent variable is close to the number of real domain independent variables. In this manually created data, the 4 features were intentionally specified as domain independent features. They appear in both domains. The remaining 8 features were specified as domain specific features. The accuracy is increased up to 100% when the domain independent variable is chosen as 4. The accuracy is dropped sharply when the domain independent variable is chosen as 6 or more. Moreover when the number of additional features is increased to 9 or more, the accuracy is dropped below 50.00%.

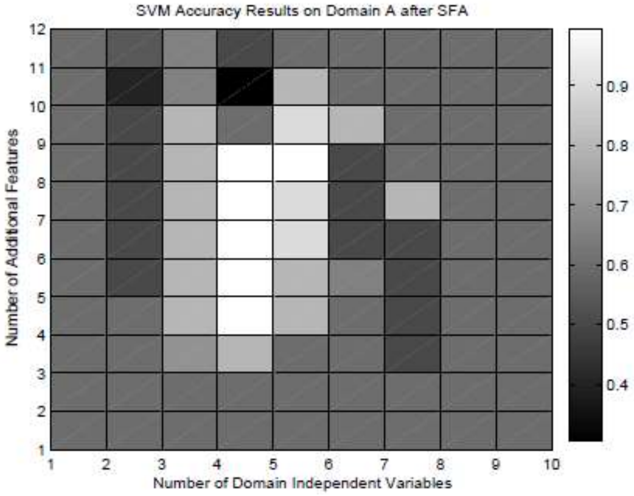

Figure 6. SVM Accuracy Results on Domain A after SFA

Actually these are very small datasets, and they do not resemble the actual datasets. Out of 12 features, 4 may not be proper for the number of domain independent variable. If we look at the study that introduced the SFA algorithm, 500 domain independent features were used in the experiments performed on the real datasets including more than 200 000 total features. They had experiments with the domain independent variables between 300 and 700 with step length 100. The other parameters (the number of additional features and the balancing factor for the effect of original and new features) were fixed. They obtained the better results when the domain independent features were between 400 and 700. In the experiments chapter, I will simulate the algorithm with the synthetically created datasets including 120 features.

Number of New Features	Number of Domain Independent Variables									
	1	2	3	4	5	6	7	8	9	10
1	60	60	60	60	60	60	60	60	60	60
2	60	60	60	60	60	60	30	60	60	60
3	60	60	60	80	80	45	60	60	60	60
4	60	60	80	100	50	55	80	60	60	60
5	60	60	70	100	65	60	80	60	60	60
6	60	50	70	100	80	80	80	60	60	60
7	60	55	70	100	80	80	50	60	60	60
8	60	55	70	100	80	80	50	60	60	60
9	60	40	60	60	80	50	50	60	60	60
10	70	40	70	60	80	50	50	60	60	60
11	70	45	55	60	60	50	50	60	60	60
12	70	45	55	60	60	50	50	60	60	60

Figure 7. The accuracy results (%) on Domain B using the SVM classifier trained on Domain A with SFA algorithm

IV. Experiments

First, I give information about my dataset. Then, random data generation with uniform distribution will be discussed. Later, the parameters and the classification tool are provided. Finally in this section the results of the experiments and the observations will be presented.

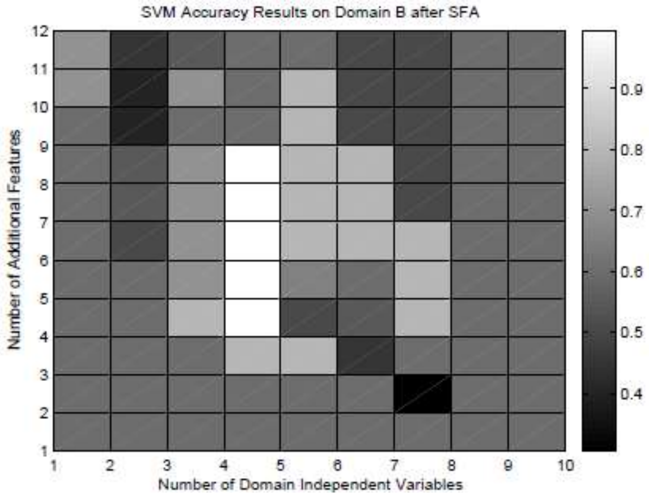

Figure 8. SVM Accuracy Results on Domain B after SFA

A. *Dataset*

In my experiments, I used artificially created dataset. There were 20 synthetic datasets in my study. Each dataset consists of two domains; domain A and domain B. In each domain, the dataset includes 1000 (500 positive and 500 negative) examples for training and 1000 (500 positive and 500 negative) examples for testing. In the SFA concept, the training means using both domains' data (labeled or unlabeled examples) to create the feature alignment mapping function. After this function is created, each example can be transformed to a new representation. In testing phase (testing the SFA approach), at first, each example is represented in new format (with additional features). The 1000 testing examples are transformed to new representation in each domain (domain A and domain B). Later 1000 examples in domain A were used to create model A and 1000 examples in domain B were used to create model B.

B. *Random Data Generation with Uniform Distribution*

In random data generation, the dataset is created according to uniform distribution between [0,1] for the specified areas. The zeros are placed on the remaining places. For example when a feature is considered as a domain specific feature for domain A, in domain B the entries of that feature in data matrix are assigned to 0. Additionally, if a feature is used only in positive samples, the entries in negative examples for that feature are filled with zeros. For the other entries, in which zeros are not placed, random numbers are generated in [0,1]. For the entries, that their numbers are below 0.5, I assign the value 1. Otherwise, I assign 0. For half of the dataset, I assign 0's and 1's based on the generated random number value to be able to simulate the sparseness. Basically, I assign 1's to the numbers below 0.25, and 0's to others. Therefore, the probability of inserting 1 to an entry in the randomly created dataset is dropped from 50% to 25% and this makes the dataset more sparse than the one that is created by uniform distribution between [0,1]. If we look at the study that introduced the SFA algorithm, an example includes about 100 features in the experiments performed on the real datasets including more than 400 000 total features.

C. *Parameters and Classification Tool*

The artificial data are randomly created. Some areas of the dataset are pre-specified, and in these areas the data creation is done with uniformly distributed manner. The data dimensions are the number of examples and the number of features. The randomly created datasets have 1000 examples (the first 500 of the examples are positive and the last 500 of the examples are negative) and include 120 features in each domain. There are two main groups of features; domain independent features and domain specific features. The domain independent features appear in both domains. In my classification, there are three different domain independent features. The first one is "general domain independent features" and they appear in both positive and negative examples in both domains. The second one is I call "negatively related features" and they appear in positive examples in one domain and in negative examples in the other domain. The last kind of domain independent features is "regular domain independent features" and they behave same in both domains. In other words if they occur in positive examples in a domain, they also occur in positive examples in other domain. The difference between "general independent features" and "regular independent features" is general ones may appear in both positive and negative examples but regular ones appear only positive examples or only negative examples. The domain specific features appear only in one domain. Fig. 9 displays the parameter settings of features on artificially created datasets. Experiments were conducted with 20 different randomly generated datasets with different parameter settings. For example, dataset 1 includes 10 general domain independent, 10 negatively related domain independent, 20 regular domain independent and 80 domain specific features.

The Simulation Parameters	Created Datasets																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
General Dom. Ind.	10	10	10	10	20	20	30	10	10	10	20	30	20	20	10	10	20	20	20	20
Negat. Rel. Dom. Ind.	10	10	20	20	10	10	20	20	30	30	20	30	10	30	30	50	20	20	30	30
Regular Dom. Ind.	20	30	10	10	10	10	30	30	30	30	20	30	30	30	40	40	40	40	30	30
Domain Specific	80	80	80	80	80	80	60	60	60	60	60	60	60	60	40	40	40	40	40	40
[illegible]	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Total Features	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120

Figure 9. The parameter settings of features on 20 artificially created datasets.

The SVM Light (Support Vector Machine) tool was used for the classification task [1]. To train the SVM classifier, I need labeled examples. The examples on my dataset are labeled as the first half is positive and the second half is negative. This tool was used to create models for both domains; model A for domain A and model B for domain B. As mentioned in previous parts of this work, after creating models in the learning phase, the other domain's model is used for testing.

D. *Results and Observations*

The SVM accuracy results for datasets 1 and 12 without using SFA method for domain A and domain B are given in Fig. 10. In the figure, domain A column shows the results of the model trained on domain B tested on domain A and domain B column shows the vice versa. For example for Dataset 1 (10 general domain independent features, 10 negatively related domain independent features, 20 regular domain independent features, and 80 domain specific features), the mean SVM accuracies without using SFA algorithm are 64.75% for domain A and 64.36% for domain B, respectively. The experiment repeated with 6 different random datasets.

	Domain A	Domain B
Dataset 1	64.75	64.36
Dataset 12	51.01	47.06

Figure 10. The mean SVM accuracy results (%) on two artificially created datasets. The experiment repeated with 6 different random datasets.

The feature alignment mapping function is constructed using the training data for this setting. After that the testing examples are transformed to new representations using this function. The SVM tool is used to train a model for domain A and another model for domain B. The model B is used to test the examples on Domain A and the model A is used to test the examples on Domain B.

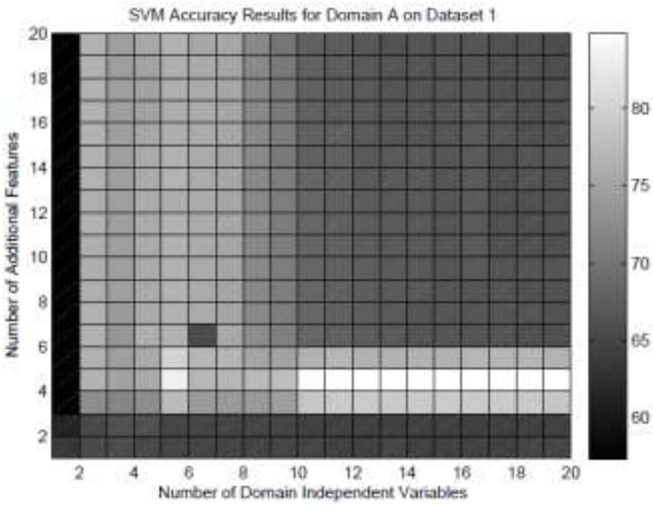

Figure 11. SVM Accuracy Results for Domain A on Dataset 1 after SFA

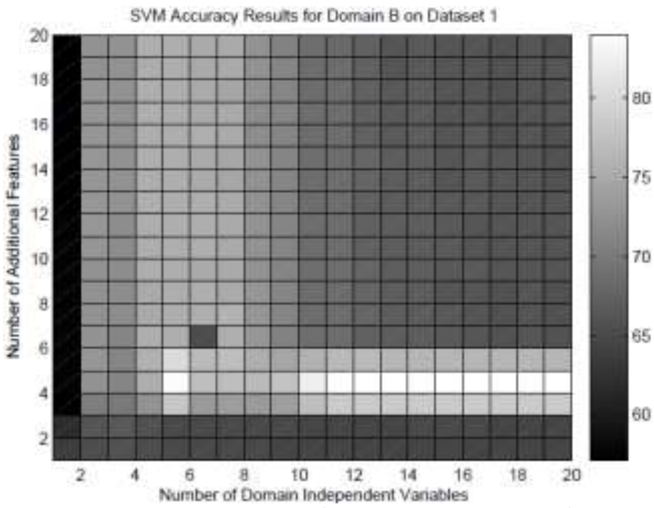

Figure 12. SVM Accuracy Results for Domain B on Dataset 1 after SFA

For the dataset 1, Fig. 11 represents mean SVM accuracy results on Domain A and Fig. 12 shows the results on Domain B. The experiments were carried out 6 times. The light color in the figure represents the high accuracy and dark color shows low accuracy. When the domain independent variable is between 2 and 8, for all the additional features, the SVM accuracy appears consistently above 70s% in Fig. 11. In addition, when the additional feature is 4 there is a consistent improvement. They are good improvements comparing the only-SVM results on domain A: 64.75%. On the other hand in some combination of domain independent variable and additional feature entries, there is not a significant improvement. The similar results are obtained on Domain B.

On the other example for Dataset 12 (20 general domain independent features, 20 negatively related domain independent features, 20 regular domain independent features, and 60 domain specific features), the SVM accuracy results without using SFA algorithm are 51.01% for Domain A and 47.06% for domain B, respectively. For the dataset 12, Fig. 13 represents mean SVM accuracy results on Domain A and Fig. 14 shows the results on Domain B. The experiments were performed 6 times.

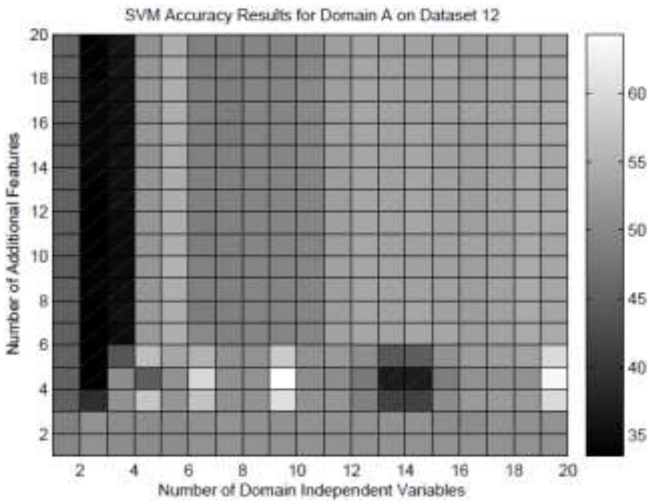

Figure 13. SVM Accuracy Results for Domain A on Dataset 12 after SFA

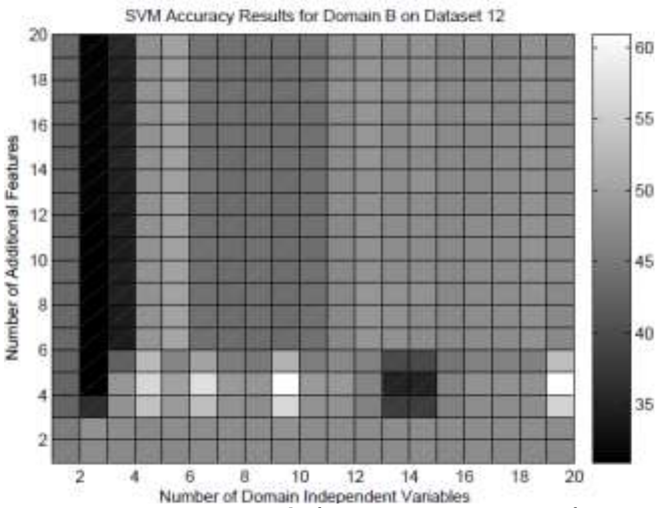

Figure 14. SVM Accuracy Results for Domain B on Dataset 12 after SFA

The results on this dataset do not show any consistent behavior both in domain A and domain B. For example in one of the runs, when the domain independent feature is 3, for all additional features the results are poorer. There is sharp increase and decrease in the results. For example for additional feature 4, there are some fluctuations observed. The reasons are discussed in Section V. The observations about the SFA algorithm impact on the SVM results can be listed as follows:

- When the number of general domain independent variable is higher than the number of other domain independent variables and data is created randomly using uniform distribution between [0,1], the SVM accuracy is increased significantly. If the data is sparser, the improvement is observed but not significant.
- When the number of negatively related domain independent variable is equal or higher than the number of other domain independent variables and data is created randomly using uniform distribution between [0,1], the SVM accuracy results have sharp increase and decrease for the different combination of domain independent variable and additional new features. If the data is sparser, there is no fluctuations and no significant improvement.
- Generally the increase in “general” and “regular” domain independent features causes the significant improvements and the increase in the “negatively related” domain independent features decreases the SVM accuracy.
- If any setting improves the SVM accuracy results, using more intense data in the same setting increases the improvement. On the other hand if any setting’s improvement is not good on the SVM accuracy results, using more intense data in the same setting makes the results fluctuated or worse.

Using all the single words in a review reduces the performance of SFA. Because some words such as “i”, “today”, and “bought” are not domain independent words, they may appear in any review, positive or negative. They may also appear in any domain. However, they may be selected as the domain independent features and there is no intuition to select “today” as domain independent. For example the following review segments consist of the given features:

. . . reliable . . . today (Positive in domain A)

. . . short-battery-life . . . today (Negative in domain B)

According to SFA algorithm, the feature “today” may be selected as domain independent feature and the algorithm expects that the features “reliable” and “short-battery-life” may form a cluster which will be used as an additional feature. However, there is no intuition to create a new feature that may connect two domain specific features with different polarity such as “reliable” and “short-battery-life”. Another problem is the process of independent feature selection. In some situations some noise values may reduce the quality of independent features.

It may happen as follows: A domain specific feature may appear many times in the examples of a domain. Usually that feature is not selected as the domain independent feature because if a feature does not appear in a domain it could not be domain independent. If that feature appears in the other domain as a noise, the feature is not eliminated. The high frequency of that feature in its own domain may cause that feature to be selected as domain independent.

International Journal of Advances in Computer Science & Its Applications
Volume 6 : Issue 1 [ISSN 2250-3765]
Publication Date : 18 April, 2016

V. Conclusion

The results show that the SFA algorithm-created additional features increase SVM accuracy in some of the settings in simulation. SFA improves the accuracy when the number of negatively related domain independent features is less than regular domain independent features. When the number of negatively related domain independent features is greater than the number of regular domain independent features, the results are not good, in other words the accuracy increase is not significant. Even in some cases the accuracy dropped below the baseline (without any additional features created by SFA). In addition to greater number of negatively related features, if the dataset is created randomly using uniform distribution between [0,1], the unexpected results are obtained. The unexpected results mean that with the change in the number of domain independent variables and the number of additional features, sharp increases or decreases occur in SVM accuracy results. For example; the dataset 12 (20 general domain independent features, 20 negatively related domain independent features, 20 regular domain independent features and 60 domain specific features, uniform distribution) has the results with that characteristic. When the number of additional features is 4, the change of the SVM accuracy results in one of the runs on domain A is given on Fig. 15.

The Number of Additional Features	The Number of Domain Independent Features							
	1	2	3	4	5	6	7	...
4	50.00	36.00	8.60	66.30	12.80	94.40	10.50	...

Figure 15. A part of SVM accuracy results on Dataset 12 in one of the Runs.

The sharp decreases and increases on accuracy results are seen on the results. The reason of these changes was investigated. When the number of additional feature is 4 and the number of domain independent feature is 5, the accuracy is 12.80%. On the other hand, the accuracy result is 94.40% when the number of additional feature is 4 and the number of domain independent feature is 6. This situation seems surprising because the difference between these two entries is not great. Both entries have 124 features (120 old features and 4 new features) and the difference is only the feature alignment mapping function. This function affects only additional new features and therefore between these two entries additional 4 features are different in for both training and testing examples. To analyze the situation the feature subset that best characterizes the statistical property of target classification variable is computed. For this calculation, the software package mRMR [4] is used. The 123rd feature is found as the first feature and 124th feature is third for the entry (4,5). Moreover the 123rd feature is detected as the third feature and 124th feature is first for the entry (4,6). These observations show that the created new additional features are significant and affect the accuracy results of SVM on domain transfer problem. Both the calculations and the accuracy results show that the additional features are valuable. Later because they are chosen among the most significant features, 123rd and 124th features are investigated. The investigation results show that there is a relation between these two features between testing examples and the training examples. For example in the first case (the entry (4,5)), the 123rd and 124th features have positive values in the positive examples and have negative values on the negative examples in testing phase. On the other hand the same features have negative values in positive examples and positive values in negative examples in training phase. I just changed the sign of these significant features values on the testing example; positive-valued features were made negative and negative-valued features were made positive. Then, I applied SVM to this new dataset; the accuracy result is increased from 12.80% to 97.10%.

VI. Future Work

I will apply the SFA algorithm on the real dataset to compare the results with ones obtained in the study [3]. I will study new domain independent feature selection methods to solve the problems discussed in Section IV. The fraction of the different types of the potential domain independent variables may be adjusted for a better learning for the mapping function. The polarity of the features will be included in the SFA algorithm. In SFA algorithm, a trade-off parameter is used to balance the effect of original features and new features. I will improve this parameter to balance each eigenvector in the feature alignment mapping function. I will define individual parameters for each eigenvector to create the mapping function. I believe that some of the relationships between the domain specific words of different domains are more powerful than the others and they should be weighted more in the mapping function.

References

[1] T. Joachims. SVMLight: Support Vector Machine. SVM-Light Support Vector Machine http://svmlight. joachims. org/, University of Dortmund, 1999.

[2] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In Advances in Neural Information Processing Systems 14: Proceeding of the 2001 Conference, pages 849–856, 2001.

[3] S.J. Pan, X. Ni, J.T. Sun, Q. Yang, and Z. Chen. Cross-domain Sentiment Classification via Spectral Feature Alignment. In Proceedings of the 19th international conference on World Wide Web, pages 751–760. ACM, 2010.

[4] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. IEEE Transactions on pattern analysis and machine intelligence, pages 1226–1238, 2005.