Rethink about whether a domain-knowledge can help improving the classification

Shengzhe Xu, Bo Shen, Jihoon Chung

1. Problem setting and importance

1.1 Research problem

A lot of general models are designed and trained by big companies, where only the model interfaces or model parameters are available while raw training dataset is not. Examples are GPS road navigation or Xbox Kinect gesture recognition. It's impossible when a researcher finds the general model is imperfect in his/her specific scenario and wants to do some corrections on the model (maybe a classifier) based on his/her own small dataset. In this project, we're going to implement several approaches to do a classifier corrector, based on combination of classifier with domain knowledge.

1.2 Why the problem is interesting/important and novel?

Currently, when researchers want to so a similar classifier modification or correction, they need all of the big data, expert advisors and enough machine learning knowledge. Therefore, with the help of the approach presented by this project, the accuracy of the classification can be enhanced, the needs of professional involvement can be reduced, the time of professional study is reduced, the cost of data collection is diminished. When we do experiment in our project, the big companies corresponds to data set of U.S. people. Small data set is corresponding to data set of other countries. We want to classify the label of incomes of data for other countries. If income $> 50k$, label 1; otherwise, label 0.

1.3 Problem setting

Assume that we have well trained classifier from big data set of U.S. people. Even though the classifier is well trained to U.S people, we can't guarantee this classifier is also best to data set of other countries. Thus, we motivated to use domain knowledge of other countries. Thus, we trained new classifier of other countries data with the new feature which indicate the domain knowledge. However, as the data size of data for other countries is limited, this classifier also have limitations. Therefore, we notice the combination of classifier of big U.S data set and classifier of limited data for other country data.

2. Related work

Actively Transfer Domain Knowledge

The problem setting is similar to the setting for active transfer learning in (Xiaoxiao Shi, 2008). In their setting, we cannot get the labeled examples, therefore, domain experts are required to label a small set of examples. However, the cost is incurred for each label. While transfer learning could borrow labeled examples from a different domain without incurring any labeling cost, there is no guarantee that the transferred examples will actually help improve the learning accuracy. To solve these two problems, for labeled examples from a different domain are examined on the basis of their likelihood to correctly label the examples of the current domain. If the likelihood is low, the domain experts are required to label the current domain example since we do not trust the label generated from the different domains examples.

Learning Bayesian network parameters under incomplete data with domain knowledge (LiaoWenhui, 2009)

The paper is dealing with the situation that when training data is incomplete how can we learn the parameter in Bayesian network. If training data is incomplete, parameters could easily get trapped among copious local maxima when learning process. To prevent that the paper proposed method using the domain knowledge to regularize so that prevent the problem. Thus paper using Bayesian concept to make incomplete training data into better training data. We also have motivation for the project in this paper. They use domain knowledge as prior information and use it to compensate the limitation of training data. The difference from our project is that to make incomplete training data into better data, our project uses other classifier with new domain knowledge.

Uncertain<T>: A First-Order Type for Uncertain Data (James Bornholt, (2014))

Some recent works already start introducing the conception of uncertain aspect in the programming language system. These two paper discussed the new programming language abstraction for uncertain data as well as uncertain element computation. However, the difference is the existing work discuss the convenience computing of general probabilistic in the language system, but our project can improve the classification without re-training as well as knowing it's training data. Furthermore, there is no existing work has worked on the one-click classifier correction programming language system, which our project can address.

3. Project Implementation

For classifier in our project, we use logistic classifier. We have two logistic classifier. One is learned from big amount of U.S data set and denote this as C1. The other is learned from small amount of other nation's data set with new feature and denote this as C2. Let classifier which is combination of C1 and C2 is C3.

Let β as coefficient of classifier C1 and as γ coefficient of classifier C2.

Thus, we can calculate probability from C1 and C2 as follows. Y indicates new feature.

C1: $\frac{1}{1+\exp(-\beta X)}$ C2: $\frac{1}{1+\exp(-\gamma(X,\mathrm{Y}))}$

To combine the both classifier we should select the optimal proportion of each classifier.
Denote the proportion as α.

3.1 Linear Combination

This is the optimization problem to find α.

$$\min_{\alpha} \sum_i \left\| Z_i - \alpha \frac{1}{1+\exp(-\beta X_i)} - (1-\alpha) \frac{1}{1+\exp(-\gamma(\mathrm{X_i},\mathrm{Y_i}))} \right\|_2$$

Subject to $0 \leq \alpha \leq 1$

Where Z_i's are the true labels, $(X_i, Y_i)'s$ are the instances

3.2 Integer Programming

Since we figure out that the objective function in Section 3.1 is not that efficient, therefore, we changed the objective function as below (using big M for binary variables in objective function):

$$\min_{\alpha} \sum_i \| Z_i - Z_i^{\alpha} \|_2$$

Subject to $0 \leq \alpha \leq 1$

$$\alpha \frac{1}{1+\exp(-\beta X_i)} + (1-\alpha) \frac{1}{1+\exp(-\gamma(\mathrm{X_i},\mathrm{Y_i}))} < M Z_i^{\alpha} + 0.5 \qquad i = 1, \dots, m$$

$$\alpha \frac{1}{1+\exp(-\beta X_i)} + (1-\alpha) \frac{1}{1+\exp(-\gamma(\mathrm{X_i},\mathrm{Y_i}))} \geq M(Z_i^{\alpha} - 1) + 0.5 \quad i = 1, \dots, m$$

Where Z_i's are the true labels, $(X_i, Y_i)'s$ are the instances and Z_i^{α} is the prediction label from C3.

Z_i^{α} =1 if $\alpha \frac{1}{1+\exp(-\beta X_i)} + (1-\alpha)\frac{1}{1+\exp(-\gamma(\mathrm{X_i},\mathrm{Y_i}))} \geq$ 0.5, otherwise Z_i^{α} =0.

We used the data, which is the same as Section 3.1. We used AMPL, which is an algebraic modeling language to describe and solve high-complexity problems for large-scale mathematical computing (i.e., large-scale optimization and scheduling-type problems), to solve the optimization problem above. The result is that in most case, $\alpha = 0$ or 1, which means that C3 is C2 or C1. The accuracy from C3 is almost bounded by the best accuracy from C1 and C2. It means that our proposed method including two classifiers cannot improve accuracy significantly. From this result, we determined to change our framework. We figured out that we need to focus on the data set which have danger to misclassify from either C1 or C2 if we want improvement of accuracy by combining both of classifier. The proposed method we are describing in below section is related with this concept.

3.3 SVM-Kmeans

3.3.1 Disadvantage of SVM and K-Means

1. Disadvantage of SVM

SVM has its own advantages in solving small amount of instances, high dimension and linear non-separable problem, so use SVM for this assignment is a good choice. Because actually we are going to find the heuristic judgment of which instance (outliers that is wrongly predicted) is going to be corrected. The flaw that a SVM classification algorithm can be attacked is, instances, that far away from the optimal classification hyper plane, can be correctly classified with a high confidence level, but the instances, that near the optimal classification hyper plane, are in danger that to be predicted wrongly.

2. Disadvantage of K-Means

K-Means is a kind of greedy algorithm. It usually only depends on the distance and a small number of other samples. Because of the instability of the data distribution, its correct rate is also very unstable. Also because of the K value and the judgment threshold is varying from different situation, its performance is not very good when face to large number of instances.

3.3.2 Motivated Feasibility

When using SVM method to do classification and analyze the instance that be given a wrong label, it can be found that SVM classifier is same as other classifier that most of the wrongly predicted instances are laying in the vicinity of the optimal classification hyperplane. This inspires us to make full use of the position information of correctly or wrongly classified instances, to correct the instances that wrongly predicted in some highly dangerous regions.

3.3.3 SVM-KMeans Semi-Supervised Learning

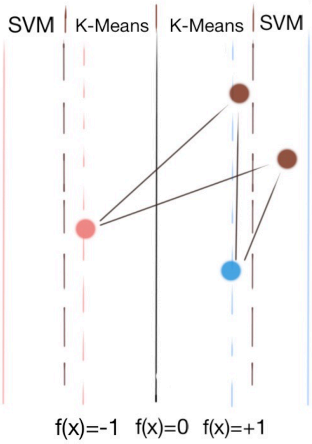
ALGORITHM 1: Corrective Semi-Supervised Learning Algorithm

Input: given existing SVM classifier c1, small test set d2 with new knowledge

Output: Corrected prediction result, the value is {0, 1}

Procedure:

```
// SVM prediction
y_hat_1: Get SVM prediction result for the d2, where SVM is based on RBF kernel in this project.
// K-means with SVM vote
k_cluster: First get K clusters from K-Means
k_cluster_label: K-means prediction with the help of SVM vote. Then use the SVM prediction result to do the vote for each clusters to decide what is the major label of this cluster.
y_hat_2: Use the major label of the cluster to re-label the instance in the whole cluster
// Heuristic correction
x_distance: For a new coming instance to classify, calculate the Euler distance between the instance and the classification hyper plane.
y_hat_3: if the distance is less than a threshold, we re-label it with the K-means algorithm prediction result, otherwise, label it with the SVM prediction result.
End
```



3.3.4 Result

Approach 1 (3.1 Linear Combination)

We did two comparative experiments based on how good the second classifier C2, the one based on small dataset, is trained. First is the situation that when C2 is trained good enough. We observed that alpha always tends to be 0, which means in the combination classifier, we always believe the C2. Second is the situation that when C2 is trained poor enough. We observed that C1 is relatively good in most of time. Thus, this approach always tends to choose a compromise between the two classifier.

Here is a representative accuracy rate during the experiment:

Classifier	Accuracy
C1-Logistic Regression	82.5%
C2-Logistic Regression	80%
C3-Combination of C1 and C2	82.4%

Finding 1: it's hard to generally adjust the classifier by only using parameters distribution approach like α*C1 + $(1-\alpha)$*C2.

Approach 2 (3.2 Integer Programming)

α=0, Finding : The accuracy from C3 is almost bounded by the best accuracy from C1 and C2.

Approach 3 (**3.3 SVM-Kmeans)**

Based on the 7-feature of the dataset, we trained the SVM/K-means and get the prediction accuracy of them. We enumerate the additional feature (new feature knowledge of the small dataset) and get the best helpful feature (the 8th one). The accuracy improvement is like the following table shows.

Classifier	Experiment Setting	Accuracy
C1-SVM	First 7 features	77.6%
C2-Kmeans with SVM vote	7 features + 8th feature, k=2	76.8%
C3-Corrective Learning	Distance threshold = 0.9	80%

4. Conclusion and future work

4.1 Conclusion

In our project, we try to solve the classification problem when there are limited training data set. By combining other well trained classifier and new feature, we solve the problem. At first, we try to use convex combination of logistic linear classifier and try to find optimal value of proportional of each classifier by integer programming. However, we figure out that accuracy is bounded by one of existing classifiers. Thus, we change our framework to just focusing the region of instances, which is in danger of misclassification from existing classifier and how can we improve classification accuracy by combining both existing classifier. Thus, we believe the prediction of SVM in the region that far away from the optimal classification hyper plane, while we adopt the result of the SVM-KMeans decision in the region that near the optimal classification hyper plane.

4.2 Future work

From our project, we figure out our proposed method have good performance and success for solving the problem what we described in section 1. We expect our method give a lot of advantages in certain situation.

For future work, we have not finished to figure out the properties of new feature. We want to figure out that in this problem setting, we want to detect which feature could improve the accuracy of classifier. That is, if we know the properties of new feature which could improve the classifier, we don’t have to do experiment to check. To fulfill this, at first we collect the new features which improved the accuracy. Then, we are thinking about using PCA or KL-divergence whether there are any relationship between existing feature and new feature that improve the accuracy.

Another future work is try to use Bayesian concepts. If we consider classifier with big data set as prior information and classifier result with limited small data set with new feature as likelihood function, we think it is also possible to apply Bayesian concepts in our problem. It will have advantage when dealing the feature which is continuous variables and easy to check the properties such as distribution and statistics of each feature.

References

Liao, W. a. (2009). Learning Bayesian network parameters under incomplete data with domain knowledge. *Pattern Recognition*, 3046--3056.

Xiaoxiao Shi, W. F. (2008). Actively Transfer Domain Knowledge. *Machine Learning and Knowledge Discovery in Databases* (pp. 342-353). Antwerp, Belgium: Springer.

James Bornholt, (2014). Uncertain<T>: a first-order type for uncertain data. Proceedings of the 19th international conference on Architectural support for programming languages and operating systems Pages 51-66