Report of COMP90051 Statistical Machine Learning Project 2

Hualong Deng
Student No: 1103512
The University of Melbourne
hualongd@student.unimelb.edu

1 Introduction

The essential part of machine learning is feeding model some data, and the model could recognize the data. The fundamental basis is that the data should be from the same distribution. However, in many practical tasks, training data and test data are from different distributions. What's more, if the target data set is not big enough to train the model and we should use the data from different distribution to help training, it is significant to find some methods make sure the model still has excellent performance. This process is also called transfer learning. In this report, we analyze the approaches provided by Daumé III (2007), and compare their results on the data supplied from the project to find the best method of them. For extension, We also evaluate the transfer learning method Two-stage TrAdaBoost.R2 and use it to improve the performance of the approaches.

2 Data set

The data provided by project is from Evgeniou,et al. (2005). After preprocessing, there are three data sets: Male set, Female set and Mixed set, which corresponds to three different types of school: single-sex male, single-sex female, and mixed gender, respectively. There is an exam score in data features, which is what model should predict. Obviously, three data sets represent three different distributions, or we could call them three domains. How to use the data from one domain to help model predicting the score from another domains well is the target of this report.

3 Transfer learning methods

3.1 Baseline methods

In transfer learning field, many researchers have done great works and show many solutions. There are six baseline approaches provided by Daumé III (2007), to solve this domain adaption problem:

- SRCONLY: ignore the target data and train the model only with the source data.
- TGTONLY: ignore the source data and train the model only with the target data.
- ALL: train the model with the source data and the target data.
- WEIGHTED: When the target data is small compared to the source data, we often see weights getting tuned to the source domain. To avoid this, we can replicate the target data to size it up to the source data.
- PRED: use the SRCONLY model to make predictions on target data and use them as an additional feature to train on the target domain.
- LININT: train SRCONLY and TGTONLY models separately and linearly interpolate their predictions.

Conference on Neural Information Processing Systems (NeurIPS), .

3.2 Feature augmentation

There is also an approach called Feature Augmentation provided by Daumé III (2007). This approach supposes each feature could be three versions: general version, source version and target version. The feature from the target domain will be target version and the feature from the source domain will be source version. Both of them will be general version. The feature in the data will be extended to three features to represent its version situation. Suppose function f transforms feature to three features. If feature x is from target domain, f(x) = (x,x,0). If feature x is from source domain, f(x) = (x,0,x). Three elements in tuple represent feature's general version, source version and target version respectively. These expansion is actually redundant and these functions could be optimized to f(x) = (x,x) and f(x) = (x,0). And we could learn the weight of source and general components and the target and general components in the development set, to find the best hyper-parameter setting.

3.3 Two-stage TrAdaBoost.R2

AdaBoost, short for Adaptive Boosting, is a machine learning meta-algorithm and could be used in many other learning algorithms to improve performance. It groups multiple classifiers with each one progressively learning from the others' wrongly classified objects and to make the whole model stronger. AdaBoost is most usually used to boost the performance of decision trees on binary classification problems, which does not fit our task. However, AdaBoost.R2 (Drucker, 1997) is an improved version of AdaBoost to solve the linear regression task. TrAdaBoost is another improved version of AdaBoost, but is still for classification task. We could combine it with the principle of AdaBoost.R2 and get a new method TrAdaBoost.R2, but the new method is highly susceptible to over-fitting. To overcome this weakness, there is a improved method called Two-stage TrAdaBoost.R2 (David, 2010), which adjusts and updates the weight in two stages. In stage one, the weights of source instances are adjusted downwards gradually until reaching a certain point (determined through cross validation). In stage two, the weights of all source instances are frozen while the weights of target instances are updated as normal.

4 Experiment

4.1 Model

To evaluate all approaches, we consider use Lasso (least absolute shrinkage and selection operator) Regression and a 1-hidden-layer neural network. For the hyper-parameters of Lasso Regression model, we set its alpha (constant that multiplies the L1 term) is 1. For the hyper-parameters of neural network model, we set its number of neuron in hidden layer is 20, optimizer is Adam and learning rate is 10^{-2}. The performance of approaches will be with their best hyper-parameter settings.

4.2 Data and Evaluation metrics

There are three domains in the data, so we use a form of 3-fold cross-validation, where the folds are the domains. One domain will be the target data and another two domains will be source data, which means that there will be three possible combinations. In data-impoverished domain adaptation scenario, the training data of target domain will be reduced, so we set the volume of training instance and development instance are 100. What's more, four features in data is categorical features: Year, VR Band of Student, Ethnic group of student and School denomination. We use one-hot encoding to preprocess these features and there are 23 features in data finally.

We use Mean Square Error to evaluate (MSE) the performance of approaches. The performance of an approach will be the mean of its MSEs in three domain combinations.

4.3 Result and analysis

4.3.1 Approach comparisons

Comparing six baseline approaches in Table 1, we could find that WEIGHT with Lasso has the best performance among them. AUGMENT with Lasso is better than WEIGHT but there is just a tiny improvement. This shows that AUGMENT is actually the best domain adaption approach on data

Table 1: All approaches' MSE performance.

Target	SRCONLY		TGTONLY		ALL		WEIGHTED		PRED	
Domain	L	NN	L	NN	L	NN	L	NN	L	NN
Female	160.3	215.1	161.3	204.2	160.1	214.6	157.6	207.0	185.8	213.9
Male	134.7	170.0	135.1	170.0	134.6	170.2	133.0	170.0	163.0	171.1
Mixed	118.7	158.9	123.2	153.1	118.7	158.4	119.8	155.5	143.9	154.9
Mean	137.9	180.3	139.9	175.7	137.8	181.1	**136.8**	176.8	164.3	180.0

Target	LININT		AUGMENT		TWO-STAGE-WEI		TWO-STAGE-AUG	
Domain	L	NN	L	NN	L	DT	L	DT
Female	182.9	179.7	157.7	340.2	157.1	142.5	156.9	143.1
Male	188.2	165.8	133.4	221.5	137.8	123.4	134.4	139.5
Mixed	172.7	156.6	118.8	213.7	121.7	112.5	116.4	120.7
Mean	181.3	167.4	**136.6**	258.4	138.9	**126.1**	135.9	134.4

processing in this report, but WEIGHT is also a excellent approach and faster than AUGMENT in practice. Apply these two approaches on Two-stage TrAdaBoost.R2 method with Lasso, and we could find Two-stage improve the performance of AUGMENT with 0.7, but it makes WEIGHT'MSE higher. This situation shows that Two-stage does not guarantee a improvement and it may works better if the volume of features is bigger. What's more, we also test two approaches on Two-stage TrAdaBoost.R2 method with decision tree which could be improved great with AdaBoost. From the result, we coudl find that decision tree does so well and makes a big improvement compared to Lasso, especially in TWO-STAGE-WEI.

4.3.2 The effect of the volume of training data from the target domain

We also try to find that whether the volume of training data in the target domain will affect the performance or not. We set the volume of training data in the target domain be 100, 500, 1000 and test it on approach TGTONLY and ALL with Lasso. From Table 2, we could find that more the data will produce better performance, but the improvement decreased during the data volume increasing.

5 Conclusion

This report compares six domain adaption baseline approaches and AUGMENT approach, and the result shows that AUGMENT has the best performance but a tiny improvement compare to WEIGHT approach. We also use Two-stage TrAdaBoost.R2 method on these two approaches to improve their performance, and it really does well. The Two-stage TrAdaBoost.R2 with decision tree classifier gets the best performance in this report. What's more, this report also shows that more training data from the target domain will improve the performance of target-domain-related approach.

Table 2: Approaches' MSE performance with different volume of training data in the target domain.

Target	TGT			ALL		
Domain	100	500	1000	100	500	1000
Female	161.3	137.2	138.4	160.1	157.4	155.4
Male	135.1	134.0	132.9	134.6	134.1	133.7
Mixed	123.2	115.7	115.3	118.7	118.1	117.7
Mean	139.9	129.0	128.9	137.8	136.6	135.6

References

[1] David, P. & Peter, S. (2010) Boosting for Regression Transfer. In *Proceedings of the Twenty-Seventh International Conference on Machine Learning*, pp. 863–870.

[2] Wenyuan, D., Qiang, Y. & Yong, Y. (2007) Boosting for Transfer Learning. In *Proceedings of the 24th international conference on Machine learning*, pp. 193–200.

[3] Drucker, H. (1997) Improving regressors using boosting techniques. In *Proceedings of the Fourteenth International Conference on Machine Learning*, pp. 193–200.

[4] Daumé III, H. (2007) Frustratingly easy domain adaptation. In *Proceedings of 45th Annual Meeting of the Association for Computational Linguistics*.

[5] Evgeniou, T., Charles, M. & Massimiliano, P. (2005) Learning multiple tasks with kernel methods. *Journal of machine learning research 6.Apr (2005)*, PP. 615-637.