Domain Adaptation under Label Shift with Active Learning Setting

Sha Sha UID: 2104167 Yu Wu UID: 2103504

1 Problem and Motivation

In supervised learning, our purpose is to make predictions on a target set without labels in the way of training a predictive model on a source set with labels and then use this model for prediction task. This works for many problems as we make the assumption that the source and target set are i.i.d. samples drawn from the same distribution. However, in many real-world cases, the source and target set may be drawn from different distributions. In this way, the model trained from source set may fail to achieve good performance on target set.

Consider that we need to train a model for the disease prediction task, the model is designed to make predictions on whether a patient has contracted a certain type of disease. Currently, we are only given diagnosis data, e.g., symptoms or medical reports, of city A, but we need to do the prediction at city B, where the disease is more prevalent. At this time, the distributions of contracted people in city A and city B are different, i.e., $P(Y=y) \neq Q(Y=y)$. The prediction task is much harder even if we can get the diagnosis data of the population of city B.

To deal with the problem, we firstly make a reasonable assumption that given whether the patient contracted the disease (label), the distributions of symptoms or medical reports are the same (training data), i.e., $P(X \mid Y) = Q(Y \mid X)$. The assumption is intuitively natural when viewing the data generating process as an anti-causal model [1] and simplifies the problem to make it tractable.

Moreover, the labels for target set may be achievable but expensive in real world, e.g., we can hire medical experts to help diagnose whether the patients have contracted the disease or not with the available data, but the cost of hiring experts can excess budget. In this scenario, we hope to select a limited number of the most “useful” samples to label which may help to obtain a high accuracy on the entire test set. Hence, the prediction performance may improve with a few target labels via active learning by actively querying labels, which is a specialized version of semi-supervised learning.

With the above problem and considerations concerned, we decide to deal with the domain adaptation problem and simplify it to a label shift problem by making the assumption that $P(Y=y) \neq Q(Y=y)$ and $P(X \mid Y) = Q(X \mid Y)$. Then, we apply the active learning method which may help further improve the performance on target set by requesting a few target sample labels iteratively to minimize cost.

2 Related Work

2.1 Domain Adaptation

Under the domain adaptation scenario, our task is to behave robust against the distributional shifts. To correct for the label shift, it mainly requires to estimate the importance weights $Q(Y=y)/P(Y=y)$ over the labels which typically live in a very low dimensional space, while the high dimensional space remains under-explored. There has been work on Bayesian approaches with a prior over the marginal label distribution assumed [2]. which requires to explicitly compute the posterior distribution of y. Recently, another method named Black Box Shift Estimation (BBSL) [3] has been proposed which is applicable to large scale data setting, but it lacks guarantees for excess risk. We refer to the recent work of Regularized Learning for Domain Adaptation under Label Shifts (RLLS) [4] which estimates importance weights with labeled source data and unlabeled target data, trains a classifier on the weighted source data and proposed a regularized estimator for small-sample regime.

2.1 Active Learning

The idea behind active learning is that a machine learning algorithm can perform better with less training if it is allowed choose the data from which it learns. An active learner may pose "queries", usually in the form of unlabeled data to be labeled manually. This approach is well-motivated in the scenario where unlabeled data may be abundant and easy to come by while training labels are difficult, time-consuming, or expensive to obtain. The article [5] provides an overview of empirical research in the field of active learning.

Two types of active learning are often discussed: stream-based selective sampling and pool-based sampling. Stream-based selective sampling obtains one instance at a time, sequentially from some streaming data source and makes each query decision individually. The procedure of this approach is shown in fig 1(a). On the other hand, Pool-based sampling evaluates and ranks the entire collection of unlabeled data before selecting the best query. The approach is illustrated in fig 1(b).

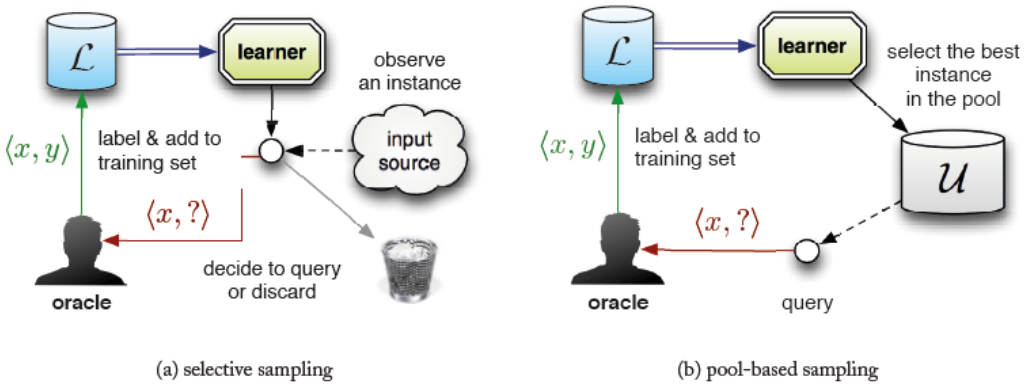

Fig 1 Selective sampling and pool-based sampling

Several query strategies are discussed in [5]. Uncertainty sampling queries the instance closest to the decision boundary. The basic premise of this strategy is that the learner can avoid querying the

instances it is already confident about, and focus its attention instead on the unlabeled instances it finds confusing. Searching through the hypothesis space is another method. Instead of deciding whether to query by single hypothesis, now more models are used to decide if a specific unlabeled data is informative. Query by disagreement is an example of this method. This approach essentially maintains the working version space V, and if a new data instance x comes along for which any two legal hypotheses disagree, then x's labeling cannot be inferred and its true label should be queried. If all the legal hypotheses do agree, then the label can be inferred and x can be safely ignored. Query by committee is a query strategy extended from query by disagreement. It relaxed two assumptions made in query by disagreement: 1) it is a measured among all hypotheses h belongs to V; 2) it is a binary measure. All one needs now are a method for obtaining hypotheses in the committee, and a heuristic for measuring disagreement among them. Since our goal of active learning is to achieve smaller classification error with less training, some query strategies try to optimize for that directly, which is to minimize expected error and variance. The idea is that one can identify all the possible outcomes, determine their values and probabilities, and compute a weighted sum to give an expected value for each action. The decision should be to choose the action that results in the best expected value, which would be the lowest expected future error.

3 Approach

Our approach is illustrated in fig 2, which can be mainly split into 3 steps.

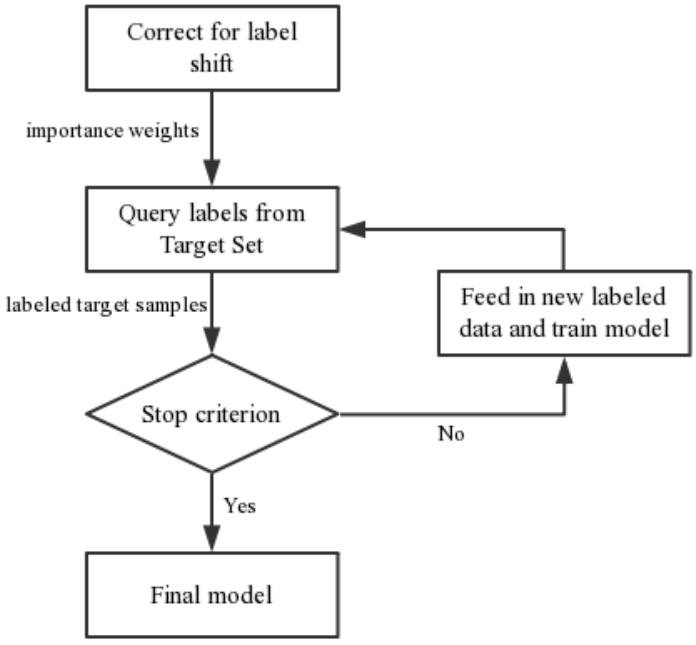

Fig 2 Flow chart of our algorithm

- Correct for the label shift

  As the source and target distributions are different ($P(Y=y) \neq Q(Y=y)$), we need to firstly correct for the label shift. The simple correlation between the label distributions p and q was noted in [3] as follows:

$$q_h(i) := \mathbb{Q}\left(h(X) = i\right) = \sum_{j=1}^{k} P\left(h(X) = i, Y = j\right) w_j$$

Here, we refer to the most recent advanced work of RLLS [4] mentioned in Section 2.1 to estimate the importance weights $w = Q(Y = y) / P(Y = y)$. Then, we are able to correct for the label shift with importance weights and train a model with training data which may also perform well in target set.

- Query data from target set
  After obtaining the weights used for domain adaptation, we now have the source domain and target domain with the same distribution. Then we want to query more data from target domain to be labeled as a supplement for the training data set. The algorithm we used to query labels is pool-based uncertainty sampling.
  With a trained model after label shift, data from target domain is predicted and the prediction is made according to the probability, which means that for each data point, the probability of each classification is calculated and the one with highest probability is selected as the final prediction for this data point. The probability is normalized such that the sum of the probability of all classifications is 1 and could be compared among all the data points. A higher probability means that the model is more confident about the prediction while a lower one implies less confidence. After evaluating all available data from target domain, the algorithm will query the top N data which the model is least confident of. These queried data will be labeled manually and then added to training dataset for next training.
- Re-train the model
  After querying a certain number of labels from the target set, we are now given extra information that helps us better estimate the label distribution of target set, which may further improve the prediction performance on target set. At this time, we have both training data, training labels, a certain number of target sample labels, we then feed all of them in and re-train the model.
- Stop criterion
  To simplified the problem, the total number of data from target domain to be labeled is given. Divided by the number of data to be labeled in each iteration, the number of iteration to query and re-train is decided and used for stop criterion.
  In further study, the stop criterion could be modified in some other ways, such as the cost of labelling data manually.

4 Experiments

In this section, we illustrate our theoretical analysis on artificially generated shift on the MNIST dataset.

- Data shift
  For this part, we refer to the shifts on MNIST dataset generated by the experiments of RLLS. In our work, we choose the tweak-one shift with $\rho = 0.2$.
- Compute importance weights
  For this part, we reproduce the part of importance weights estimation of RLLS, for which no prior

knowledge of $q(y)/p(y)$ is required.

- Train base model with training data
  We use the same two-layer fully connected neural network used in importance weight computing to the train the model with initial training data all from source domain applied with importance weights.
- Actively query labels of target samples
  With the model trained, we evaluate the uncertainty of each data point from target domain using the method illustrated in Section 3. After the top N data points which the model is least confident of are chosen, these data will be labeled and added to training data set.
- Re-train the model with new training data
  With queried labels of the target samples, we re-train our model with the new training data, which contains the original training data and all queried labeled data from target set by minimizing CrossEntropyLoss (weighted on training data and unweighted on queried data).
- Result
  Original Training Data Size = 20000, Original Test Data Size = 15000

Table 1 Result of the experiment

Iteration	Total Queried Samples	New Training Data Size	Accuracy
Initial State	0	20000	90.89%
1	50	20050	89.84%
2	100	20100	92.80%
3	150	20150	89.43%
4	200	20200	92.17%
5	250	20250	91.90%
6	300	20300	93.45%
7	350	20350	90.62%
8	400	20400	92.57%
9	450	20450	94.00%
10	500	20500	94.33%

In general, the test accuracy increases as more and more "useful" labeled samples selected from target domain are added to training data set, which meets our expectation. There are some small oscillations with the increase of queried data size. This is tolerable because there is some randomness during the training process, which may lead to the up and down of test accuracy.

Acknowledgement

Thanks Dr. Anqi Liu for providing the codes of Regularized Learning for Domain Adaptation Under Label Shift (RLLS) and all other help.

GitHub Link: https://github.com/wuyudd/CS165_Project

Reference

[1] Schölkopf, B., Janzing, D., Peters, J., Sgouritsa, E., Zhang, K., & Mooij, J. (2012). On causal and anticausal learning. *arXiv preprint arXiv:1206.6471*.

[2] Chan, Y. S., & Ng, H. T. (2005, July). Word Sense Disambiguation with Distribution Estimation. In *IJCAI* (Vol. 5, pp. 1010-5).

[3] Lipton, Z. C., Wang, Y. X., & Smola, A. (2018). Detecting and correcting for label shift with black box predictors. *arXiv preprint arXiv:1802.03916*.

[4] Azizzadenesheli, K., Liu, A., Yang, F., & Anandkumar, A. (2018). Regularized Learning for Domain Adaptation under Label Shifts.

[5] Settles, B. (2012). Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, *6*(1), 1-114.