Active Learning with Transfer Learning

Chunyong Luo, Yangsheng Ji, Xinyu Dai, Jiajun Chen
State Key Laboratory for Novel Software Technology,
Department of Computer Science and Technology,
Nanjing University, Nanjing, 210046, China
{luocy, jiys, daixy, chenjj}@nlp.nju.edu.cn

Abstract

In sentiment classification, unlabeled user reviews are often free to collect for new products, while sentiment labels are rare. In this case, active learning is often applied to build a high-quality classifier with as small amount of labeled instances as possible. However, when the labeled instances are insufficient, the performance of active learning is limited. In this paper, we aim at enhancing active learning by employing the labeled reviews from a different but related (source) domain. We propose a framework Active Vector Rotation (AVR), which adaptively utilizes the source domain data in the active learning procedure. Thus, AVR gets benefits from source domain when it is helpful, and avoids the negative affects when it is harmful. Extensive experiments on toy data and review texts show our success, compared with other state-of-the-art active learning approaches, as well as approaches with domain adaptation.

1 Introduction

To get a good generalization in traditional supervised learning, we need sufficient labeled instances in training, which are drawn from the same distribution as testing instances. When there are plenty of unlabeled instances but labels are insufficient and expensive to obtain, active learning (Settles, 2009) selects a small set of critical instances from target domain to be labeled, but costs are incurred for each label. On the other hand, transfer learning (Ji et al., 2011), also known as domain adaptation (Blitzer et al., 2006), aims at leveraging instances from other related source domains to construct high-quality models in the target domain. For example, we may employ labeled user reviews of similar products, to predict sentiment labels of new product reviews. When the distributions of source and target domain are similar, transfer learning would work well. But significant distribution divergence might cause negative transfer (Rosenstein et al., 2005).

To further reduce the labeling cost and avoid negative transfer, we propose a framework, namely Active Vector Rotation (AVR), which takes advantage of both active learning and transfer learning techniques. Basically, AVR makes model’s parameter vector $\mathbf{w}$ actively rotate towards its optimal direction with as few labeled instances in target domain as possible. Specifically, AVR first applies certain unsupervised learning techniques to make source and target domain’s distributions more ‘similar’, and then leverages source domain information to query the most informative instances of target domain. Most importantly, it carefully reweights instances to mitigate the risk of negative transfer. AVR is general enough to incorporate various active learning and transfer learning modules, as well as varied basic learners such as LR and SVM.

2 Related Work

Shi et al. (2008) proposed an approach AcTraK, using labeled source and target domain instances to build a so-called ‘transfer classifier’ to help label actively selected target domain instances. AcTraK initially requires labeled target domain instances,
and relies too much on the transfer classifier. Thus it might be degenerated by negative transfer.

An ALDA framework was proposed in (Saha et al., 2011). ALDA employs source domain classifier $w_{src}$ to help label actively selected target domain instances. When conditional distributions $P(y|x)$ are a bit different (Chattopadhyay et al., 2011) or marginal distributions $P(x)$ are significantly different between source and target domain, ALDA would perform poorly. ALDA doesn’t discuss the negative transfer problem and gets hurts when it happens, while AVR actively avoids it by its projection and reweighting strategy.

Liao et al. (2005) proposed a method M-Logit, utilizing auxiliary data to help train LR. They also proposed actively sampling target domain instances using Fisher Information Matrix (Fedorov, 1972; Mackay, 1992). Besides, instance weighting was used to mitigate distribution difference between source and target domain in (Huang et al., 2006; Jiang and Zhai, 2007; Sugiyama et al., 2008). These can work as a module in our framework.

3 AVR: Active Vector Rotation

Without loss of generalization, we will constrain the discussion of AVR to binary classification tasks. But in fact, AVR can also be applied to multi-class classification and regression.

Given training set $D_{tr} = \{(x_i, y_i)\mid i = 1, \ldots, m\}$, $x_i \in \mathbb{R}^n$, $y_i \in \{-1, +1\}$, traditional supervised learning tries to optimize (Fan et al., 2008; Lin et al., 2008):

$$\min_w \|w\| + \frac{C}{m} \sum_{i=1}^{m} \varepsilon(w; x_i, y_i),$$

(1)

where the penalty parameter $C > 0$, controls the importance ratio between loss function $\varepsilon(w; x_i, y_i)$ and regularization parameter $\|w\|$. Loss function’s definition is diverse for different basic learners, e.g. LR uses $\log(1 + e^{-y_i w^T x_i})$, while L2-SVM uses $\max(1 - y_i w^T x_i, 0)^2$.

In the paper, we have the following assumptions:

1) Target domain $D_{tgt} = \{(x_i^{tgt}, y_i^{tgt})\mid i = 1, \ldots, N_{tgt}\}$, $x_i^{tgt} \in \mathbb{R}^{n_{tgt}}$, $y_i^{tgt} \in \{-1, +1\}$, $N_{tgt}$ is the size of $D_{tgt}$;

2) Source domain $D_{src} = \{(x_i^{src}, y_i^{src})\mid i = 1, \ldots, N_{src}\}$, $x_i^{src} \in \mathbb{R}^{n_{src}}$, $y_i^{src} \in \{-1, +1\}$, $N_{src}$ is the size of $D_{src}$;

3) $p(x^s) \neq p(x^t)$;

4) $N_{src}$ and $N_{tgt}$ are large enough;

5) Testing set $D_{test}$ and $D_{tgt}$ are i.i.d..

Under maximum labeling budget $N_b$, our goal is to employ source and target domain instances to maximize model accuracy:

$$\max_w a_{D_{test}}(w) = \sum_{(x_i, y_i) \in D_{test}} \frac{1 + y_i h_w(x_i)}{2y_i^2},$$

(2)

where the hypothesis is:

$$h_w(x) = \begin{cases} -1, & w^T x < 0 \\ +1, & w^T x \geq 0 \end{cases}.$$

(3)

So, we design the machine learning framework, Active Vector Rotation, to optimize $w$:

$$\min_w \|w\| + \sum_{i=1}^{m} c_i \varepsilon(w; x_i, y_i),$$

(4)

where the weight variables $c_i > 0$, control the importance of each instance in training. Larger $c_i$ means more necessity of $w$ to fit $(x_i, y_i)$).

Intuitively, $w$ of $D_{tr}$ should try harder to fit the instances from $D_{tgt}$ than the instances from $D_{src}$, so that the corresponding $c_i$ of instances from $D_{tgt}$ should be larger. The algorithm of AVR is described in Table 1, which is discussed in detail in the following subsections.

| Input: $D_{src}$, $D_{tgt}$, $D_{test}$, $N_b$; Output: $w$, $a_{D_{test}}(w)$ |
|---------------------------------------------------------------|
| 1. Project $x^s$ and $x^t$ to a common latent semantic space, where $x^s$, $x^t \in \mathbb{R}^n$. |
| 2. Actively select the least source domain instances, which can characterize source domain classifier $w_{src}$, into training set $D_{tr} = \{(x_i^{src}, y_i^{src})\mid i = 1, \ldots, N_{src}\}$. |
| 3. Initialize $w$ using $D_{tr}$. |
| 4. For $i = N_{src} + 1 : N_{src} + N_b$ |
| \hspace{1cm} 1) Actively select the most informative instance $(x_i^{src}, y_i^{src})$ from $D_{tgt}$. |
| \hspace{1cm} 2) Insert the new labeled instance into training set, $D_{tr} = D_{tr} \cup (x_i^{src}, y_i^{src})$. |
| \hspace{1cm} 3) Update $c_i$ for $j = 1 : i$. |
| \hspace{1cm} 4) Retrain $w$ using $D_{tr}$ and (4). |
| end |
| 5. Compute $a_{D_{test}}(w)$. |

Table 1: AVR algorithm

3.1 Projection of Source and Target Domain

$x^s$ and $x^t$ might be in different vector spaces. To employ $D_{src}$ in the training of $D_{tgt}$’s optimal $w$, we’d better project $x^s$ and $x^t$ into a common n-dimensional latent semantic space, where the distributions of the projected $x^s'$, $x^t' \in \mathbb{R}^n$ would be more similar. Varied projection approaches could be employed in different tasks. For example, Hardoon et al. (2004) used CCA to project text and
image to a latent semantic space, where image could be retrieved by text. Blitzer et al. (2007) and Ji et al. (2011) utilized SCL and VMVPCA respectively in sentiment classification. Huang et al. (2006) applied RKHS and KMM in breast cancer prediction.

Regarding the case where \( x^s \) and \( x^t \) are in the same vector space but certain approach is applied to make their distributions more similar, we also consider it as a kind of projection of \( D_{src} \) and \( D_{tgt} \).

### 3.2 Initialization of Training set

To reduce training cost and risk of negative transfer, AVR actively selects a relatively small set of instances from \( D_{src} \) into \( D_{tr} \). Transfer learning mainly leverages \( D_{src} \)'s separating hyperplane information, i.e. \( w_{src} \), while only a small set of critical instances from \( D_{src} \) can characterize the statistics of \( w_{src} \). AVR initializes \( D_{tr} \) by these critical instances. Different tasks may employ different selection strategy. E.g. in our experiments, the text classification task employs uncertainty sampling (Settles, 2009), while sentiment classification task selects the least \( N'_{src} \) instances which can accurately characterize \( w_{src} \), such that:

\[
\min_{1 \leq j_i \leq N_{src}} \sum_{i=1}^{N'_{src}} w_{src}^T x_{j_i}^{s'}.
\]

### 3.3 Query Strategy in Target Domain

After initialization of \( D_{tr} \), AVR uses certain basic learner, such as LR and SVM, to get \( w = w_{init} \). As the labeling budget \( N_b \) is limited, we need iteratively query the most informative instance and add the new labeled instance into \( D_{tr} \) to retrain \( w \).

AVR revises the query strategy of traditional active learning. After a few new labeled instances added to \( D_{tr} \), the retrained \( w \) would be different from \( w_{init} \) and closer to the optimum. Traditional active learning queries the instance in \( D_{tgt} \) w.r.t. \( w \), e.g. uncertainty sampling queries the instance closest to separating hyperplane, such that:

\[
\min_{x_i^t \in D_{tgt}} \left| w^T x_i^t \right|.
\]

However, AVR queries the most informative instance from which are identically classified by \( w \) and \( w_{init} \), e.g. for uncertainty sampling, AVR queries the instance such that:

\[
\min_{x_i^t \in D_{tgt}, \ w^T x_i^t \ w_{init}^T x_i^t > 0} \left| w^T x_i^t \right|.
\]

The instance queried by AVR makes \( w \) more quickly approach to its optimum, as to some extent, part of the statistics of the instances which are differently classified by \( w \) and \( w_{init} \), can be characterized by the new queried instances. But when \( w \) is very close to the optimum, AVR will query by traditional active learning strategy.

### 3.4 Reweighting \( c_i \)

Appropriate reweighting can help accelerate \( w \) rotating to the optimum and avoid negative transfer. Intuitively, the instances from \( D_{tgt} \) and the instances which have similar distribution with \( D_{tgt} \) should be given higher weight. Varied reweighting strategy, e.g. TrAdaBoost (Dai et al., 2007), could be applied in AVR framework. In our experiments, AVR employs a simple but efficient reweighting strategy, without iteration:

\[
c_i = \begin{cases} 
1, & i \leq N_{src}, w^T x_i^t w_{init}^T x_i^t > 0 \\
0, & i \leq N'_{src}, w^T x_i^t w_{init}^T x_i^t \leq 0 \\
b, & \text{otherwise.}
\end{cases}
\]

### 4 Experiments

We perform AVR on a set of toy data and two real world datasets, 20 Newsgroups Dataset 1 and Multi-Domain Sentiment Dataset 2, comparing it with several baseline methods. In this paper, we use model accuracy as the evaluation. We used LR and L2-SVM as basic learner respectively, but due to space limit, we only report the results of LR.

#### 4.1 Toy Data

We generate four bivariate Gaussian distributions as the positive and negative instances of \( D_{src} \) and \( D_{tgt} \) respectively as illustrated in Figure 1.
As shown in Figure 1, $D_{src}$ and $D_{tgt}$ randomly sample 1000 instances respectively, then $D_{test}$ randomly samples 200 instances from $D_{tgt}$. Circle and diamond, big plus and cross, small plus and cross, represent positive and negative instances of $D_{src}$, $D_{tgt}$ and $D_{test}$ respectively.

To this toy data, AVR’s configuration is:

1) $x^{s'} = x^s$, $x^{t'} = x^t$.
2) AVR uses uncertainty sampling to select the least 5 instances which can characterize $w_{src}$, to initialize $D_{tr}$ and $w_{init}$. In Figure 1, the 5 instances are marked by big filled circles or diamonds, the dash line draws the separating hyperplane $w_{init}^T x = 0$.

3) Then AVR queries instances as described in Section 3.3, the first 10 queried instances are marked by large numerals, with the first 3 are queried w.r.t. (7). The small numerals mark the first 3 instances which would be queried w.r.t. (6).

4) AVR reweights $c_i$ by (8), where $b = 4$. The black filled circles mark the instances whose corresponding $c_i = 0$. The solid line draws the current hyperplane $w^T x = 0$.

Baseline methods are briefly described in Table 2. Details about AcTraK and ALDA can be found in (Shi et al., 2008) and (Saha et al., 2011) respectively.

| Method | Target Domain Labeling Budget $N_p$ |
|--------|-------------------------------------|
|        | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Random | 50.05 | 59.35 | 79.88 | 86.04 | 90.26 | 93.01 | 94.41 | 95.30 | 96.03 | 96.41 |
| Active | 49.50 | 57.85 | 80.41 | 95.50 | 96.30 | 97.23 | 97.41 | 97.59 | 97.64 | 97.72 |
| AcTraK | 93.15 | 95.23 | 96.10 | 96.69 | 97.03 | 97.30 | 97.53 | 97.68 | 97.78 | 97.82 |
| O-ALDA | 77 | 77 | 77 | 77 | 77 | 77 | 77 | 77 | 77 | 77 |
| B-ALDA | 77 | 77 | 77 | 77 | 77 | 77 | 77 | 77 | 77 | 77 |
| Source-A | 80.50 | 95 | 85 | 96 | 98.50 | 96 | 98 | 97 | 97 | 96.50 |
| AVR-U | 80.50 | 94 | 94.50 | 97 | 98.50 | 97 | 98.50 | 97.50 | 98.50 | 98.50 |
| AVR-W | 80.50 | 94 | 94.50 | 97 | 98.50 | 97 | 98.50 | 97.50 | 98.50 | 98.50 |

Table 2: Performance of different methods on toy data, where AcTraK unfairly uses two more labels.

### 4.2 20 Newsgroups Dataset

20 Newsgroups Dataset is commonly used in machine learning and NLP tasks. It contains about 20000 newsgroup documents which are categorized into 6 top categories and 20 subcategories. We split it into 6 pair of $D_{src}$ and $D_{tgt}$, with each pair includes only two top categories documents, such as “comp” and “rec”, but $D_{src}$ and $D_{tgt}$ are drawn from different subcategories, e.g. $D_{src}$ has “comp.graphics” and “comp.graphics”, but $D_{tgt}$ has “comp.windows.x” and “sci.autos”. The task is to leverage $D_{src}$ to distinguish the top categories of documents in $D_{tgt}$.

Our settings of 20 Newsgroups Dataset is identical with that on toy data, with $N_{src}$ varies from 500 to 800 on different pairs.

Due to space limit, we only report results on the pair of “comp vs. rec” in Figure 2, with all methods are averaged over 30 runs. The results on other pairs are similar. Since AVR-U and AVR-W are variants of AVR, with similar performance, we only report the results of AVR.

![Figure 2: AVR outperforms others on the “comp vs. rec” pair.](image-url)
4.3 Multi-Domain Sentiment Dataset

The sentiment dataset consists of user reviews about several products (Book, DVD, Electronic, Kitchen) from Amazon.com, the task is to classify a review’s sentiment label as positive or negative. We have 12 pairs with each pair has two products as $D_{src}$ and $D_{tgt}$ respectively. On this dataset, AVR employs VMVPCA (Ji et al., 2011) to project $D_{src}$ and $D_{tgt}$, and initializes $D_{tr}$ with $N'_{src} = 1000$ instances from $D_{src}$ w.r.t. (5), while the other configuration is the same as that described in Section 4.1. To be comparable, the baseline methods which leverage $D_{src}$ are preprocessed by VMVPCA. We also add another baseline method Source-A’ here, which is identical with Source-A, except that it is not projected by VMVPCA. Given space limit, we only report the results on the pair “DVD $\rightarrow$ Kitchen”, with other pairs have similar performance.

Figure 3: AVR does better than previous work on the “DVD $\rightarrow$ Kitchen” dataset for all budget sizes.

4.4 Discussion

From inspection of experimental results, we get the following remarks.

Why to combine active learning and transfer learning?

- Active learning such as uncertainty sampling can significantly reduce the labeling cost. But when $w$ is far from the optimum, uncertainty sampling may oversample instances near a direction. For example, in Figure 2, Active method is worse than Random method when $N_b < 50$.

- $D_{src}$ could help $D_{tgt}$ in learning accurate $w$, e.g. in Figure 2, when $N_b < 200$, Source-A method with the help of $D_{src}$ outperforms Random and Active methods which never use $D_{src}$. But inappropriate use of $D_{src}$ may cause negative transfer, e.g. in Figure 2, when $N_b > 200$, Source-A, ALDA and AcTraK methods, which overuse $D_{src}$, underperform Active method.

- Thus, we realize that appropriate combination of transfer learning and active learning could advance and complement each other. Especially when $D_{tgt}$ has scarce labels, $D_{src}$ could help avoid oversampling instances near a direction. But with the increase of labels in $D_{tgt}$, $D_{src}$ should decrease its weight in training to avoid negative transfer.

Does each component of AVR work?

- Appropriate Projection of $D_{src}$ and $D_{tgt}$ could mitigate distribution divergence, e.g. in our sentiment classification task, Source-A and AVR which applied VMVPCA significantly and consistently outperforms Source-A’.

- Initialize $D_{tr}$ by a small set of critical instances from $D_{src}$ can significantly reduce training cost without loss of accuracy. E.g. in our experiments, when $N_b = 1$, AVR has better or comparable performance w.r.t. Source-A which initializes $D_{tr}$ by whole $D_{src}$. More importantly, AVR trims initial $D_{tr}$ size from 1000 to 5 in toy data, from 4000 to 500 in Newsgrups dataset, and from 2000 to 1000 in Sentiment dataset.

- The query strategy of AVR described in Section 3.3 advances traditional active learning, which is supported by the performance of AVR over AVR-U.

- Appropriately reweighting instances from $D_{src}$ and $D_{tgt}$ could result in accurate $w$ and avoid negative transfer meanwhile. For example, in our experiments, the reweighting strategy of (8) makes AVR outperform all baseline methods, while some of which suffer from negative transfer.

How about AcTraK’s performance?

- AcTraK works well on our toy data, just because it unfairly uses too much more labels of $D_{tgt}$, even though, it underperforms AVR when $N_b > 3$. Besides, AcTraK performs poorly on high dimensional data like text in our experiments.
5 Conclusion and Future Work

Our proposed machine learning framework AVR actively and carefully leverages information of source domain to query the most informative instances in target domain, as well as to train the best possible model of target domain. The four essential components of AVR, which establish its efficacy and help it avoid negative transfer, are validated in experiments.

In the future, we are planning to apply AVR in more tasks with appropriate specification of projection, query and reweighting strategy. Especially for sentiment classification, we will combine prior domain knowledge, such as domain sentiment lexicon, with AVR framework to further reduce labeling cost.

Acknowledgements

This work is supported by the National Fundamental Research Program of China (2010CB327903) and the Doctoral Fund of Ministry of Education of China (20110091110003). We also thank Shujian Huang, Ning Xi, Yinggong Zhao, and anonymous reviewers for their greatly helpful comments.

References

John Biltzer, Ryan Mcdonald, Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In Proc. EMNLP, pp.120-128.

John Biltzer, Mark Dredze, Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: domain adaptation for sentiment classification. In Proc. ACL, pp.432-439.

Rita Chattopadhyay, Jieping Ye, Sethuraman Panchanathan, Wei Fan, Ian Davidson. 2011. Multisource domain adaptation and its application to early detection of fatigue. In Proc. KDD, pp.717-725.

Wenyuan Dai, Qiang Yang, Gui-Rong Xue, Yong Yu. 2007. Boosting for transfer learning. In Proc. ICML, pp.93-200.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, Chih-Jen Lin. 2008. Liblinear: a library for large linear classification. JMLR, 9:1871-1874.

Valerii Vadimovich Fedorov. 1972. Theory of optimal experiments. Academic Press.

David R. Hardoon, Sandor Szedmak, John Shaw-Taylor. 2004. Canonical correlation analysis: An overview with application to learning methods. Neural Computation, 16(12): 2639-2664.

Jayyuan Huang, Alexander J. Smola, Arthur Gretton, Karsten M. Borgwardt, Bernhard Schölkopf. 2006. Correcting sample selection bias by unlabeled data. In Proc. NIPS, pp.601-608.

Yangsheng Ji, Jiajun Chen, Gang Niu, Lin Shang, Xinyu Dai. 2011. Transfer learning via multi-view principal component analysis. JCSS, 26(1):81-98.

Jing Jiang, ChengXiang Zhai. 2007. Instance weighting for domain adaptation in NLP. In proc. ACL, pp.264-271.

Xuejun Liao, Ya Xue, Lawrence Cain. 2005. Logistic regression with an auxiliary data source. In Proc. ICML, pp.505-512.

Chih-Jen Lin, Ruby C. Weng, S. Sathiya Keerthi. 2008. Trust region newton method for large-scale logistic regression. JMLR, 9:627-650.

David J. C. Mackay. 1992. Information-based objective functions for active data selection. Neural Computation, 5:590-604.

Michael T. Rosenstein, Zvika Marx, Leslie Pack Kaelbling, Thomas G. Dietterich. 2005. To transfer or not to transfer. In Proc. NIPS, December 9-10.

Avishek Saha, Piyush Rai, Hal Daumé III, Suresh Venkatasubramanian, Scott L. DuVall. 2011. Active supervised domain adaptation. In Proc. ECML-PKDD, pp.97-112.

Burr Settles. 2009. Active learning Literature Survey. In Computer Sciences Technology Report 1648, University of Wisconsin-Madison.

Xiaoxiao Shi, Wei Fan, Jiangtao Ren. 2008. Actively transfer domain knowledge. In Proc. ECML-PKDD pp.342-357.

Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul von Bünau, Motoaki Kawanabe. 2008. Direct importance estimation with model selection and its application to covariate shift adaptation. NIPS, pp.1433-1440.