Vol.2013-NL-212 No.4
2013/7/18

Domain Adaptation for Word Sense Disambiguation under the Problem of Covariate Shift

Hironori Kikuchi[1,a)] Hiroyuki Shinnou[1,b)]

Abstract: Word sense disambiguation(WSD) is the task of identifying the meaning of an ambiguous word in a sentence. It can be solved by supervised learning. The problem of domain adaptation for WSD is considered that the distribution of source domain and target domain are different. However given the nature of WSD, The equivalence of the distribution between source domain and target domain is formed and the problem caused by the difference of the distribution between source domain's sense ratio and target domain's one. This problem of domain adaptation for WSD can be regarded as a problem of covariate shift. In this paper, we solve the problem by parameter learning method to weight the probability density ratio, which is the solution of covariate shift. In comparison with technique of Daumé, which is a standardized approach for adaptive region, the effect of solving the problem in covariate shift was shown.

Keywords: word sense disambiguation, domain adaptation, covariate shift, maximum entropy method, BCCWJ corpus

1. Introduction

In this paper, we indicate that the problem of domain adaptation, whose task is word sense disambiguation (WSD), is the problem of covariate shift. We solve this problem by the parameter learning method, which is weighted with the probability density ratio.

In many natural language processing tasks, the inductive learning method has been used. In this method, we create training data corresponding to the task from corpus A and learn a classifier from the training data. The classifier solves the problem of domain adaptation for WSD. However, the data to be applied to the classifier often belong to corpus B, whose domain is different from corpus A. In this case, the classifier learned from corpus A (source domain) cannot analyze accurately the data of corpus B (target domain). This is the problem of domain adaptation[*1], and we conduct domain adaptation in the task of WSD.

WSD is a task that identifies the meaning $c \in C$ of an ambiguous word w in statement $\boldsymbol{x}$. It is the question to solve $\arg\max_{c \in C} P(c|\boldsymbol{x})$ in posterior probability maximization. It is normally supposed to be solved by supervised learning. But as mentioned above, there is the problem of domain adaptation for WSD. In domain adaptation, $P_s(c|\boldsymbol{x})$ can be derived from source domain S, so we try to estimate $P_t(c|\boldsymbol{x})$ on target domain T by using $P_s(c|\boldsymbol{x})$ and other data. In this time, the meaning of the word in the same sentence is not considered to have changed if it appears on the corpus of any domain. That is, $P(c|\boldsymbol{x})$ does not depend on the domain, and hence $P_s(c|\boldsymbol{x}) = P_t(c|\boldsymbol{x})$. It need not necessarily estimate $P_t(c|\boldsymbol{x})$ because $P_s(c|\boldsymbol{x})$ can be derived. However, the identification accuracy is low if we use $P_s(c|\boldsymbol{x})$, which is estimated only from the source domain. This is caused by $P_s(\boldsymbol{x}) \neq P_t(\boldsymbol{x})$. In this paper, we consider that the problem of domain adaptation for WSD is the problem of covariate shift, and solve the problem by using the solution of covariate shift.

The training data is denoted by $D = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$. In covariate shift, we set the probability model $P(y|\boldsymbol{x}; \theta)$ to $P_t(y|\boldsymbol{x})$ and build $P_t(y|\boldsymbol{x})$ by finding θ, which maximizes the following log-likelihood, which is weighted with the probability density ratio $r_i = p_t(\boldsymbol{x}_i)/p_s(\boldsymbol{x}_i)$.

$$\sum_{i=1}^{N} r_i \log p(y_i|\boldsymbol{x}_i; \theta)$$

In our experiment, we select two domains, PB(BOOK) and OC(Yahoo! Chie-Bukuro), from BCCWJ corpus[7] and 17 words, whose frequency is more than 50, from both domains. We use a model of maximum entropy method[*2] as the probabilistic model and estimate the probability density ratio from $P_t(\boldsymbol{x}_i)$ and $P_s(\boldsymbol{x}_i)$. We show the effect of solving the problem in the covariate shift and compare it with the method of Daumé.

2. Related Work

Domain adaptation for natural language processing is the subject arising in all tasks that use the inductive learning method. The utilizing method can be divided into two types roughly. One uses labeled data from target domain (supervised domain adaptation) and the other does not use labeled data from target domain (unsupervised domain adaptation). In this essay, our method is classified into supervised case, so we introduce traditional researches

[1] 4-12-1 Nakanarusawa, Hitachi, Ibaraki 316–8511, Japan
[a)] 13nm705g@hcs.ibaraki.ac.jp
[b)] shinnou@mx.ibaraki.ac.jp

[*1] Domain adaptation is regarded as a kind of transfer learning[3] in part of machine learning.

[*2] As the tool, we used the **classias** presented at the following site. http:www.chokkan.orgsoftwareclassias

about the method of supervised case in this section.

The point of supervised domain adaptation is the application of data in source domain. In the case of the distance between source domain and target domain is remote, accuracy of the classifier gets worse if we use too much data from source domain. To prevent this problem, it is necessary to use the distance to control the application of data in source domain.

Asch[11] indicated that it is possible to estimate how much accuracy is reduced by using similarity between domains in conducting domain adaptation for part-of-speech task. Harimoto[13] investigated a factor that decline the accuracy, which task is syntactic analysis, when target domain is changed and proposed new measure to calculate similarity between the domains. Plank[8] selected the appropriate source domain to analyze target domain in syntactic analysis by calculating the similarity between domains. Ponomareva[9] and Remus[10] used the similarity between domains as parameter within learning in sentiment classification task. In these case, the similarity is measured to each task. The similarity in WSD is depend on target word. Komiya[5][4][6] changed the learning method to apply to each target word by natures including the distance of domains[*3].

Komiya's studies are a kind of ensemble learning. In those learning method, only the weight that is applied to data in source and target domain is different. That is, the approach to adjust the weight and apply it to learning method is effective. Jiang excluded the data which has large different of $P_s(y|\boldsymbol{x})$ and $P_t(y|\boldsymbol{x})$ as "misleading" from training data[2]. This can be regarded as weighting method because "misleading" is weighted with 0. Our method is also regarded as above.

The method of Daumé[1] is also regarded as weighting technique. In the method, vector $\boldsymbol{x}_s$ of training data in source domain are fixed to the three times length vector $(\boldsymbol{x}_s, \boldsymbol{x}_s, 0)$ and vector $\boldsymbol{x}_t$ of training data in target domain are fixed to $(0, \boldsymbol{x}_t, \boldsymbol{x}_t)$. Classification problem is solved by using the vector. This approach is very easy and highly effective. The effect for domain adaptation is obtained by applying the weight of the common features between source domain and target domain.

The studies to work out the solution for domain adaptation under covariate shift are Jiang's experiment[2] and Saeki's one[12]. Jiang controlled density ratio manually and used the logistic regression as a model. Saeki modeled P(x) by the unigram and used the maximum entropy method as a model. However, both tasks are not WSD.

3. Covariate Shift in Expected Loss Minimization

The set of the target words w is denoted by C, and the expected loss function is denoted by $r(\boldsymbol{x}, c, \boldsymbol{\theta})$ when the sense of w is identified c in the statement $\boldsymbol{x}$. In this formula, $\boldsymbol{\theta}$ represents a model. If $P_t(\boldsymbol{x}, c)$ is the distribution on the target domain, the expected loss in the problem of domain adaptation is described as follows.

$$R = \sum_{\boldsymbol{x},C} r(\boldsymbol{x}, c, \theta) P_t(\boldsymbol{x}, c)$$

[*3] The similarity between domains are including those all properties.

And if $P_s(\boldsymbol{x}, c)$ is the distribution on the target domain, following is established.

$$R = \sum_{\boldsymbol{x},C} r(\boldsymbol{x}, c, \boldsymbol{\theta}) \frac{P_t(\boldsymbol{x}, c)}{P_s(\boldsymbol{x}, c)} P_s(\boldsymbol{x}, c)$$

When the training data is denoted by $D = \{(\boldsymbol{x}_i, c_i)\}_{i=1}^N$ and $\hat{P}_s(\boldsymbol{x}, c)$ is the empirical distribution of $P_s(\boldsymbol{x}, c)$, following is established.

$$\begin{aligned} R &\approx \sum_{\boldsymbol{x},C} r(\boldsymbol{x}, c, \boldsymbol{\theta}) \frac{P_t(\boldsymbol{x}, c)}{P_s(\boldsymbol{x}, c)} \hat{P}_s(\boldsymbol{x}, c) \\ &= \frac{1}{N} \sum_{i=1}^{N} r(\boldsymbol{x}_i, c_i, \theta) \frac{P_t(\boldsymbol{x}_i, c_i)}{P_s(\boldsymbol{x}_i, c_i)} \end{aligned}$$

The assumption of covariate shift establishes the following.

$$\frac{P_t(\boldsymbol{x}_i, c_i)}{P_s(\boldsymbol{x}_i, c_i)} = \frac{P_t(\boldsymbol{x}_i) P_t(c_i|\boldsymbol{x}_i)}{P_s(\boldsymbol{x}_i) P_s(c_i|\boldsymbol{x}_i)} = \frac{P_t(\boldsymbol{x}_i)}{P_s(\boldsymbol{x}_i)}$$

Namely, if we think from the point of the expected loss minimization, we can solve the problem of covariate shift by finding $\boldsymbol{\theta}$, which minimizes the following formula L.

$$L = \sum_{i=1}^{N} \frac{P_t(\boldsymbol{x}_i)}{P_s(\boldsymbol{x}_i)} r(\boldsymbol{x}_i, c_i, \boldsymbol{\theta}) \tag{1}$$

L takes the form of minimizing the loss function $r(\boldsymbol{x}_i, c_i, \boldsymbol{\theta})$, whose weight is the probability density ratio $r_i = \frac{P_t(\boldsymbol{x}_i)}{P_s(\boldsymbol{x}_i)}$.

4. Maximization of the Weighted Log-Likelihood

In the case of expected loss minimization for the classification problem, 0/1-loss is chosen as the loss function usually. If we use this loss function, the decision function $d(\boldsymbol{x})$ that determines the output of input $\boldsymbol{x}$ is denoted by following formula. This describes the discrimination based on the maximizing posterior.

$$d(\boldsymbol{x}) = c_k \quad s.t. \quad k = \arg\max_j P(c_j|\boldsymbol{x})$$

It is difficult to construct $P(c|\boldsymbol{x})$ to minimize the Eq (1) strictly. However, there are many situations to use the solution of maximum likelihood as the expected loss minimization experientially. In fact, it is possible to find the model which can maximizes the log-likelihood in target domain in covariate shift by finding the parameter, which maximizes the likelihood weighted with r_i.

In this paper, we use the maximum entropy method.

$$P(c|\boldsymbol{x}, \boldsymbol{\lambda}) = \frac{1}{Z(\boldsymbol{x}, \boldsymbol{\lambda})} \exp\left(\sum_{j=1}^{M} \lambda_j f_j(\boldsymbol{x}, c) \right)$$

The input is denoted by $\boldsymbol{x} = (x_1, x_2, \cdots, x_M)$. The class is denoted by c. $f_j(\boldsymbol{x}, c)$ is the feature function. $Z(\boldsymbol{x}, \boldsymbol{\lambda})$ is a term to normalize $P(\cdot)$.

$$Z(\boldsymbol{x}, \boldsymbol{\lambda}) = \sum_{c \in C} \exp\left(\sum_{j=1}^{M} \lambda_j f_j(\boldsymbol{x}, c) \right)$$

$\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \cdots, \lambda_M)$ is the weight parameter corresponding to the features. In the maximum entropy method, we find $\boldsymbol{\lambda}$ by the maximum likelihood method using the training data $\{(\boldsymbol{x}_i, c_i)\}_{i=1}^N$.

Namely, we find the $\boldsymbol{\lambda}$ which maximizes the following formula.

$$L(\boldsymbol{\lambda}) = \prod_{i=1}^{N} \log P(c_i, \boldsymbol{x}_i)$$

The following formula is formed by differentiating $L(\boldsymbol{\lambda})$ in each λ_j and applying the extremeproblem.

$$\begin{aligned}\frac{\partial L(\boldsymbol{\lambda})}{\partial \lambda_j} &= \sum_{i=1}^{N} f_j(\boldsymbol{x}_i, c_i) \\ &\quad - \sum_{i=1}^{N} \sum_{c \in C} P(c|\boldsymbol{x}_i, \boldsymbol{\lambda}) f_j(\boldsymbol{x}_i, c) \\ &= 0\end{aligned}$$

And we can find $\boldsymbol{\lambda}$ by gradient method.

If the probability density ratio is denoted by $r(\boldsymbol{x})) = P_t(\boldsymbol{x})/P_s(\boldsymbol{x})$, we find $\boldsymbol{\lambda}$, which maximizes the following formula.

$$L(\boldsymbol{\lambda}) = \prod_{i=1}^{N} r(\boldsymbol{x}_i) \log P(c_i, \boldsymbol{x}_i)$$

And we can obtain the following formula by the same procedure as above.

$$\begin{aligned}\frac{\partial L(\boldsymbol{\lambda})}{\partial \lambda_j} &= \sum_{i=1}^{N} r(\boldsymbol{x}_i) f_j(\boldsymbol{x}_i, c_i) \\ &\quad - \sum_{i=1}^{N} \sum_{c \in C} P(c|\boldsymbol{x}_i, \boldsymbol{\lambda}) r(\boldsymbol{x}_i) f_j(\boldsymbol{x}_i, c) \\ &= 0\end{aligned}$$

We define a new feature function g_j.

$$g_j(\boldsymbol{x}, c) = r(\boldsymbol{x}) f_j(\boldsymbol{x}, c)$$

Then the above formula can be deformed to the following formula and solved by maximum entropy method.

$$\begin{aligned}\frac{\partial L(\boldsymbol{\lambda})}{\partial \lambda_j} &= \sum_{i=1}^{N} g_j(\boldsymbol{x}_i, c_i) \\ &\quad - \sum_{i=1}^{N} \sum_{c \in C} P(c|\boldsymbol{x}_i, \boldsymbol{\lambda}) g_j(\boldsymbol{x}_i, c) \\ &= 0\end{aligned}$$

Specifically, we multiply the each elements of training data $\boldsymbol{x}_i$ by $r(\boldsymbol{x}_i)$ and use normal maximum entropy method.

5. Calculation of the Density Ratio

In the learning under covariate shift, it needs to calculate the probability density ratio. It is possible to estimate the probability density directly but we find the probability density simply. The feature of examples $\boldsymbol{x}$ of the target word w is denoted by $\{f_1, f_2, \cdots, f_n\}$. And we calculate the probability density ratio $P_R(\boldsymbol{x})$ on domain $R \in \{S, T\}$. At first, we assume the following formula.

$$P_R(\boldsymbol{x}) = \prod_{i=1}^{n} P_R(f_i)$$

We create a feature list for all examples of w in the corpus of the domain R and denote the frequency of the features f by $n(R, f)$. And also we denote the total frequency of the features by $N(R)$. Namely, $N(R) = \sum_{f \in R} n(R, f)$. Then, we denote the number of types of the features related to w in domain S and T by M. $P_R(f)$ is defined as follows.

$$P_R(f) = \frac{n(R, f) + 1}{N(R) + M}$$

6. Experiments

In this experiment, we select two domains, PB(BOOK) and OC(Yahoo! Chie-Bukuro), from Balanced Corpus of Contemporary Written Japanese (BCCWJ[7]) corpus [7] and 17 words, whose frequency is more than 50, from both domains.

Table 1 shows those words, the number of sense in a dictionary, frequency and the number of sense in each corpus[*4].

The patterns of domain adaptation are from PB(source) to OC(target) and from OC(source) to PB(target).

Table 1 target words

word	# of senses in dic.	freq. in PB	# of senses in PB	freq. in OC	# of senses in OC
言う (iu)	3	1114	2	666	2
入れる (ireru)	3	56	3	73	2
書く (kaku)	2	62	2	99	2
聞く (kiku)	3	123	2	124	2
来る (kuru)	2	104	2	189	2
子供 (kodomo)	2	93	2	77	2
時間 (jikan)	4	74	2	53	2
自分 (jibun)	2	308	2	128	2
出る (deru)	3	152	3	131	3
取る (toru)	8	81	7	61	7
場合 (baai)	2	137	2	126	2
入る (hairu)	3	118	4	68	4
前 (mae)	3	160	2	105	3
見る (miru)	6	273	6	262	5
持つ (matsu)	4	153	3	62	4
やる (yaru)	5	156	4	117	3
ゆく (yuku)	2	133	2	219	2
average	3.35	193.9	2.94	150.6	2.88

In each word and each domain adaptation, we extract 15 labeled data from target domain at random and define the rest as test data.

The training data is constituted by labeled data in source domain and 15 labeled data in target domain. We estimate the accuracy of discrimination of sense. We conduct this experiment for five times and find the average of the accuracy.

Table 2 and Table 3 show the result of the domain adaption from PB to OC and from OC to PB. The value of Table 2 and Table 3 show the average of accuracy rate. The file of "S+T" is the result whose training data is constituted by labeled data from source domain and target domain. The file of "T-only" is the result whose training data is constituted by only target domain. The file of "D3" is the result which uses the method of Daumé. The file of "PM" is the result of the proposed method under covariate shift in this paper. All method use the maximum entropy method. The result of the proposed method is more better than the one of the other methods.

[*4] The semantics refers to the Iwanami dictionary.We targeted the middle classification within there. "入る (hairu)" has three senses in the dictionary but there are four senses in PB and OC because BCCWJ corpus has tags of new sense.

Table 2 result(PB → OC)

word	S+T	T-only	D3	PM
言う (iu)	0.8305	0.7582	0.8391	0.8188
入れる (ireru)	0.7172	0.7310	0.7172	0.7310
書く (kaku)	0.7330	0.7330	0.7330	0.7330
聞く (kiku)	0.7172	0.6451	0.7006	0.6803
来る (kuru)	0.7991	0.7863	0.7991	0.7933
子供 (kodomo)	0.5843	0.8734	0.6917	0.8701
時間 (jikan)	0.8316	0.8316	0.8316	0.8368
自分 (jibun)	0.8732	0.8732	0.8732	0.8732
出る (deru)	0.6591	0.6807	0.6643	0.6609
取る (toru)	0.4565	0.4800	0.4609	0.5600
場合 (baai)	0.9365	0.9817	0.9365	0.9728
入る (hairu)	0.6302	0.6629	0.6717	0.6896
前 (mae)	0.9103	0.8745	0.9036	0.8722
見る (miru)	0.7036	0.8287	0.8302	0.8359
持つ (motsu)	0.8836	0.7441	0.9179	0.7494
やる (yaru)	0.9421	0.9516	0.9421	0.9421
ゆく (yuku)	0.7165	0.7067	0.7382	0.6998
average	0.7603	0.7731	0.7795	0.7835

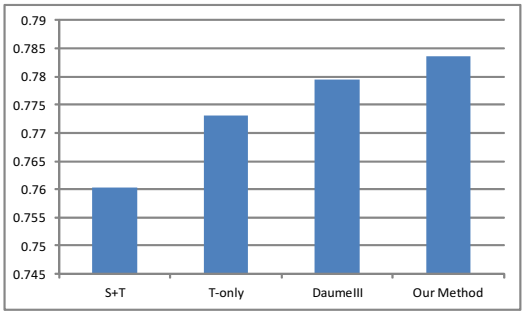

Fig. 1 Precision (PB → OC)

Table 3 result(OC → PB)

word	S+T	T-only	D3	PM
言う (iu)	0.8219	0.7079	0.8055	0.8048
入れる (ireru)	0.7304	0.6473	0.7009	0.6678
書く (kaku)	0.8085	0.9059	0.8340	0.8979
聞く (kiku)	0.7225	0.7560	0.7375	0.7095
来る (kuru)	0.9723	0.9791	0.9746	0.9723
子供 (kodomo)	0.5013	0.7597	0.5737	0.7441
時間 (jikan)	0.8814	0.8980	0.8780	0.8949
自分 (jibun)	0.9610	0.9760	0.9644	0.9630
出る (deru)	0.5754	0.5446	0.5885	0.5827
取る (toru)	0.3456	0.4357	0.3918	0.4027
場合 (baai)	0.8503	0.8487	0.8503	0.8503
入る (hairu)	0.5296	0.5436	0.5652	0.5596
前 (mae)	0.7986	0.7153	0.8472	0.7889
見る (miru)	0.8464	0.8393	0.8449	0.8401
持つ (motsu)	0.7918	0.7099	0.7801	0.7055
やる (yaru)	0.9341	0.9368	0.9341	0.9341
ゆく (yuku)	0.8813	0.8657	0.8830	0.8640
average	0.7619	0.7688	0.7737	0.7754

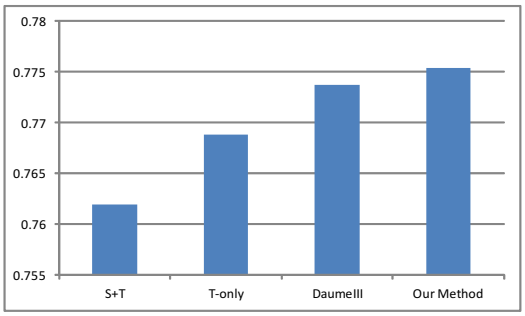

Fig. 2 Precision (OC → PB)

7. Discussions

7.1 Variation of the Average Accuracy Rate

In the learning under covariate shift, the difference of variation in the each accuracy is large. In the result of the previous section, the variation might affect the result because the difference between the method of Daumé and proposed method is slight.

In this section, we confirm the variation problem. We conducted the experiment that we selected 15 labeled data from target domain at random and included those data in training data for five times in the previous section. Finding the average accuracy of target word can get the accuracy of the five experiments. The result of the previous section is the average of the five experiments. We investigate the maximum value and the minimum value of the experiments and the difference between two values. At the same time, we conduct the experiment with the weight $r^{0.5}$. Table4 and 5 show the result.

We can see the variation of average accuracy rate in the proposed method is the smallest from those tables. If the weight is $r^{0.5}$, the result from OC to PB is better than the proposed method slightly, but the variation is larger than the proposed method. Considering this result, the proposed method tends to produce good results.

Table 4 Variation in the evaluation value(PB → OC)

method	average	maximum	minimum	variation
S+T	0.7603	0.7716	0.7444	0.02711
T-Only	0.7731	0.7870	0.7567	0.03034
D3	0.7795	0.7964	0.7687	0.02779
PM(r)	0.7835	0.7879	0.7780	0.00988
PM($r^{0.5}$)	0.7825	0.7899	0.7714	0.01854

Table 5 Variation in the evaluation value(OC → PB)

method	average	maximum	minimum	variation
S+T	0.7619	0.7704	0.7554	0.01499
T-Only	0.7688	0.7867	0.7555	0.03122
D3	0.7737	0.7786	0.7704	0.00818
PM(r)	0.7754	0.7796	0.7718	0.00779
PM($r^{0.5}$)	0.7777	0.7821	0.7720	0.01015

7.2 Use of the Discriminative Model

In this paper, we use the maximum entropy method. The learning under covariate shift estimates the parameter by using weighted log-likelihood. So learning techniques that we can use are limited to the kind of the generated model. But it is considered that the same effect can be obtained if we conduct learning to give a weight of the probability density ratio to data. Namely, better result can be got if there is a model, which can use the learning to give the weight to data.

Support Vector Machine (SVM) is difficult to conduct learning to give the weight to data. However, discriminative model is available in the method of Daumé because the method can use any learning method. In this section, we conduct the experiment that uses SVM for the method of Daumé. The results are show in Table 6 and Table 7. The file of “S+T” is the average accuracy rate when using SVM that uses all data from source and target domain. The file of “T-only” is the average accuracy rate when using SVM that uses data from target domain only. The file of “D3-SVM” is the average accuracy rate when using SVM under the method of Daumé. The file of “PM” is the average accuracy

rate of the proposed method in this paper. The accuracy of the proposed method is better than the one of SVM.

But, if we can conduct the learning to give weight to data in discriminative model, there is a possibility that the accuracy can be improved.

The file of “d-SVM” in Table 6 and Table 7 are shown for reference, it is the result of conducting SVM that multiplies the feature value by the weight of the probability density ratio. From this result, it is difficult to conduct the learning to give weight to data in SVM.

Table 6 SVM(PB → OC)

word	S+T	T-Only	D3-SVM	d-SVM	PM
言う (iu)	0.8118	0.7644	0.6671	0.8241	0.8188
入れる (ireru)	0.7414	0.7069	0.7483	0.6759	0.7310
書く (kaku)	0.7354	0.7257	0.7306	0.7281	0.7330
聞く (kiku)	0.6562	0.6746	0.6358	0.6340	0.6803
来る (kuru)	0.7656	0.7563	0.7794	0.7713	0.7933
子供 (kodomo)	0.6267	0.8800	0.7177	0.8052	0.8701
時間 (jikan)	0.8421	0.8316	0.8368	0.8421	0.8368
自分 (jibun)	0.8750	0.8536	0.8679	0.8732	0.8732
出る (deru)	0.6836	0.6627	0.6766	0.6470	0.6609
取る (toru)	0.4826	0.5000	0.4522	0.4783	0.5600
場合 (baai)	0.8929	0.9710	0.9201	0.8384	0.9728
入る (hairu)	0.6528	0.6642	0.6792	0.7094	0.6896
前 (mae)	0.9044	0.8689	0.9067	0.8711	0.8722
見る (miru)	0.8627	0.8546	0.8627	0.8586	0.8359
持つ (motsu)	0.8596	0.7277	0.8638	0.7702	0.7494
やる (yaru)	0.9421	0.9421	0.9421	0.9421	0.9421
ゆく (yuku)	0.7530	0.7765	0.7647	0.7402	0.6998
average	0.7699	0.7742	0.7678	0.7652	0.7835

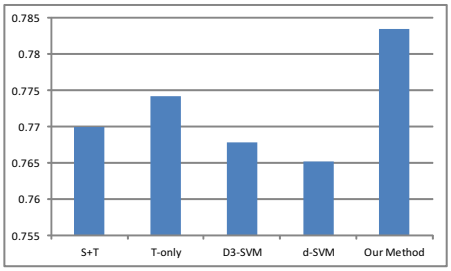

Fig. 3 SVM (PB → OC)

Table 7 SVM(OC → PB)

word	S+T	T-Only	D3-SVM	d-SVM	PM
言う (iu)	0.8237	0.7245	0.8084	0.7674	0.8048
入れる (ireru)	0.7171	0.6585	0.6683	0.6634	0.6678
書く (kaku)	0.7574	0.9064	0.7915	0.8340	0.8979
聞く (kiku)	0.7076	0.7486	0.7170	0.6388	0.7095
来る (kuru)	0.9746	0.9746	0.9746	0.9746	0.9723
子供 (kodomo)	0.5686	0.7674	0.6384	0.7287	0.7441
時間 (jikan)	0.8542	0.8949	0.8610	0.8949	0.8949
自分 (jibun)	0.9179	0.9740	0.9391	0.9295	0.9630
出る (deru)	0.6018	0.5548	0.6105	0.6486	0.5827
取る (toru)	0.3948	0.4353	0.4164	0.3826	0.4027
場合 (baai)	0.8553	0.8635	0.8602	0.8339	0.8503
入る (hairu)	0.5573	0.5256	0.5711	0.5889	0.5596
前 (mae)	0.8222	0.7681	0.8667	0.7764	0.7889
見る (miru)	0.8407	0.8298	0.8415	0.8368	0.8401
持つ (motsu)	0.7846	0.6928	0.7875	0.6885	0.7055
やる (yaru)	0.9273	0.9315	0.9273	0.9315	0.9341
ゆく (yuku)	0.8899	0.8486	0.8899	0.8675	0.8640
average	0.7644	0.7705	0.7747	0.7639	0.7754

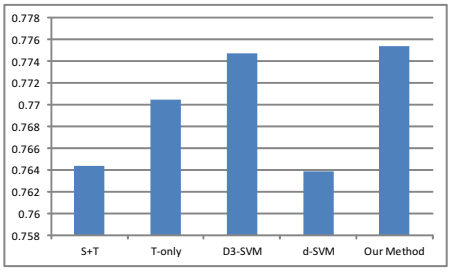

Fig. 4 SVM (OC → PB)

7.3 Application to Unsupervised Leaning

It is not necessary that the training data contain data of target domain. If data of target domain are not included, the method is considered unsupervised learning under covariate shift. We carry out a experiment to confirm it. Table 8 and Table 9 show the result. The file of “S-Only” is the result of learning with the training data of source domain. Namely, the weight of each data is 1. “Weight of density ratio” is the file that the weight is probability density ratio. It is corresponding to learning under covariate shift. The effect in the domain adaptation from OC to PB is observed slightly, but the accuracy is down in the domain adaptation from PB to OC.

The reason is considered to be a lack of data with weights. For example, in case of the weight of the data x_1 in c_1 is 0.01 and the weight of the data x_2 in c_2 is 0.02, if we don't consider the weight, $P(c_1) = P(c_2) = 0.5$. If we consider the weight, $P(c_1) = 1/3$ and $P(c_2) = 2/3$. Naturally, the low weights means that there is no reliable data, so it is reasonable that the $P(c_1) = P(c_2) = 0.5$ is estimated empirically. If large amount of data whose weight is low are contained, it is not possible to estimate properly. The data that weight to some extent is necessary. For this reason, it is difficult to use unsupervised learning simply. This is the problem for the future.

Table 8 unsupervised method(PB → OC)

word	S-only	weight of density ratio
言う (iu)	0.8305	0.8145
入れる (ireru)	0.7000	0.7483
書く (kaku)	0.7330	0.7330
聞く (kiku)	0.7006	0.6230
来る (kuru)	0.7991	0.7991
子供 (kodomo)	0.1557	0.1298
時間 (jikan)	0.8263	0.8263
自分 (jibun)	0.8732	0.8732
出る (deru)	0.6556	0.6189
取る (toru)	0.2696	0.3644
場合 (baai)	0.9201	0.9165
入る (hairu)	0.6189	0.5018
前 (mae)	0.9193	0.8722
見る (miru)	0.5646	0.5646
持つ (motsu)	0.8276	0.7458
やる (yaru)	0.9421	0.9421
ゆく (yuku)	0.6762	0.6811
average	0.7066	0.6914

Table 9 unsupervised method(OC → PB)

word	S-only	weight of density ratio
言う (iu)	0.8204	0.8046
入れる (ireru)	0.7450	0.5350
書く (kaku)	0.7830	0.9021
聞く (kiku)	0.6685	0.5680
来る (kuru)	0.9723	0.9723
子供 (kodomo)	0.2687	0.5994
時間 (jikan)	0.8814	0.8949
自分 (jibun)	0.9569	0.9569
出る (deru)	0.5549	0.5666
取る (toru)	0.2345	0.2757
場合 (baai)	0.8503	0.8503
入る (hairu)	0.4584	0.4441
前 (mae)	0.7514	0.7444
見る (miru)	0.8440	0.8393
持つ (motsu)	0.7904	0.7067
やる (yaru)	0.9368	0.9368
ゆく (yuku)	0.8812	0.8640
average	0.7293	0.7330

8. Conclusions

In this paper, we indicated that the problem of domain adaptation, whose task is word sense disambiguation (WSD), is the problem of covariate shift. We solved this problem by using parameter learning method, which is weighted with the probability density ratio.

In the learning under covariate shift, training data $(\boldsymbol{x}, y)$ is weighted with probability density ratio $\frac{P_t(\boldsymbol{x})}{P_s(\boldsymbol{x})}$. We calculated density ratio by estimating $P_t(\boldsymbol{x})$ and $P_s(\boldsymbol{x})$ directly at this time. And we used the maximum entropy method for leaning. In this experiment, we selected two domains, PB(BOOKS) and OC(Yahoo! Chie-Bukuro), from BCCWJ corpus and 17 words, whose frequency is more than 50, from both domains. We show the effect of our approach and compare it with the method of Daumé.

The future issues are finding more appropriate method to calculate density ratio, using the learning method of the discriminative model that is weighted with probability density ratio and application of the proposed method to unsupervised learning.

References

[1] Daumé III, Hal: Frustratingly Easy Domain Adaptation, *ACL-2007*, pp. 256–263 (2007).
[2] Jiang, J. and Zhai, C.: Instance weighting for domain adaptation in NLP, *ACL-2007*, pp. 264–271 (2007).
[3] Kamishima, T.: Transfer Learning, *The Japanese Society for Artificial Intelligence*, Vol. 25, No. 4, pp. 572–580 (2010).
[4] Komiya, K. and Okumura, M.: Automatic Determination of a Domain Adaptation Method for Word Sense Disambiguation using Decision Tree Learning, *IJCNLP-2011*, pp. 1107–1115 (2011).
[5] Komiya, K. and Okumura, M.: Automatic Domain Adaptation for Word Sense Disambiguation Based on Comparison of Multiple Classifiers, *PACLIC-2012*, pp. 75–85 (2012).
[6] Komiya, K. and Okumura, M.: Automatic selection of domain adaptation method for WSD using decision tree learning (In Japanese), *Journal of NLP*, Vol. 19, No. 3, pp. 143–166 (2012).
[7] Maekawa, K.: Design of a Balanced Corpus of Contemporary Written Japanese, *Symposium on Large-Scale Knowledge Resources (LKR2007)*, pp. 55–58 (2007).
[8] Plank, B. and van Noord, G.: Effective measures of domain similarity for parsing, *ACL-2011*, pp. 1566–1576 (2011).
[9] Ponomareva, N. and Thelwall, M.: Which resource is best for cross-domain sentiment analysis?, *CICLing-2012* (2012).
[10] Remus, R.: Domain Adaptation Using Domain Similarity- and Domain Complexity-based Instance Selection for Cross-domain Sentiment Analysis, *Proceedings of the 2012 IEEE 12th International Conference on Data Mining Workshops (ICDMW 2012) Workshop on Sentiment Elicitation from Natural Text for Information Retrieval and Extraction (SENTIRE)*, pp. 717–723 (2012).
[11] Van Asch, V. and Daelemans, W.: Using domain similarity for performance estimation, *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, pp. 31–36 (2010).
[12] Yosuke Saiki, H. T. and Okumura, M.: Domain Adaptation in Sentiment Classification by Instance Weighting, *IPSJ SIG Technical Report. SIG-NL Report*, Vol. 2008, No. 33, pp. 61–67 (2008).
[13] Yusuke Miyao, K. H. and Tsujii, J.: Kobunkaiseki no bunyatekiou ni okeru seido teika youin no bunseki oyobi bunyakan kyori no sokutei syuhou (In Japanese), *The 16th Annual Meeting on Journal of Natural Language Processing*, pp. 27–30 (2010).