Industrial Engineering
& Management Systems
Vol 17, No 2, June 2018, pp.334-340
ISSN 1598-7248 | EISSN 2234-6473 |
https://doi.org/10.7232/iems.2018.17.2.334

Instance Weighting Domain Adaptation Using Distance Kernel

Woojin Lee, Jaewook Lee, Saerom Park*
Department of Industrial Engineering, Seoul National University, Seoul, Republic of Korea

(Received: October 20, 2017 / Revised: December 15, 2017 / Accepted: December 25, 2017)

ABSTRACT

Domain adaptation methods aims to improve the accuracy of the target predictive classifier while using the patterns from a related source domain that has large number of labeled data. In this paper, we introduce new kernel weight domain adaptation method based on smoothness assumption of classifier. We propose new simple and intuitive method that can improve the learning of target data by adding distance kernel based cross entropy term in loss function. Distance kernel refers to a matrix which denotes distance of each instances in source and target domain. We efficiently reduced the computational cost by using the stochastic gradient descent method. We evaluated the proposed method by using synthetic data and cross domain sentiment analysis tasks of Amazon reviews in four domains. Our empirical results showed improvements in all 12 domain adaptation experiments.

Keywords: Distance Kernel, Domain Adaptation, Sentimental Analysis

* Corresponding Author, E-mail: psr6275@snu.ac.kr

1. INTRODUCTION

Recently, supervised learning algorithms has shown great performance due to deep learning algorithms and large scale dataset. Unfortunately collecting labeled data for machine learning model is expensive and time-consuming process. Obtaining labeled data is hard, but it may be still possible to transfer the knowledge from well-known training data to unlabeled data set. However, this approach suffers from difference in the data distribution because traditional machine learning procedure assumes that training data and test data come from an equal distribution.

Domain adaptation has received recent attention in order to build a robust classifier that could be applied to different domains. Domain adaptation considers an environment that training data and test data are from different distributions (Glorot *et al.*, 2011). It considers training data from source domain and test data from target domain. The purpose of domain adaptation is to build a predictive classifier for target domain by utilizing the knowledge in source domain.

There are two approaches to solve the domain adaptation problem. The first approach is feature based domain adaptation. It aims to find common feature structure that can link two different domains for domain adaptation (Pan and Yang, 2010). Glorot *et al.* (2011) conducted a study which used stacked marginalized auto-encoder to extract common feature between different domains. Ganin *et al.* (2016) used the idea of adversarial training to extract the common features that cannot discriminate between the source and the target domains. Bousmalis *et al.* (2016) suggested domain separation networks that could learn the representation which is unique to each domain.

The second one is instance-based approach. This approach focuses on revising the training of the classifier by adding various terms in loss function. There has been researches using importance weight to reweight the labeled instances from source domain (Shimodaira, 2000). Also there were researches that estimated importance weights that are used in instance based domain adaptation (Bickel *et al.*, 2009; Sugiyama *et al.*, 2007; Gretton *et al.*, 2009).

Although latest deep learning based approach extracts the transferrable representation that can be applied to source data as well as target data, we will focus on instance based learning approach that could improve the performance after the representation leaning phase. Therefore, in this research we will assume that common feature that is relevant on source and target data is already extracted, so we focus on instance based domain adaptation algorithm.

In this research, we propose new instance-based domain adaptation approach by using distance kernel based loss function. By adding distance-kernel based cross entropy in loss function, we could train a model that fits well in target domain by using the source input data, source label data, and target input data.

2. LITERATURE REVIEW

Instance based Domain Adaptation approach is most based on importance weighting algorithm. Let's assume that we have N_s source data $\mathrm{D}_\mathrm{s}^l = \{(x_1^s, y_1^s), (x_2^s, y_2^s), (x_{N_s}^s, y_{N_s}^s)\}$ and N_t unlabeled target data $\mathrm{D}_\mathrm{t}^u = \{x_1^t, x_2^t, \ldots, x_{N_t}^t\}$. We construct predictive function $\mathrm{f}_\theta(x)$ for conditional probability $p(y|x;\theta)$. The loss function $l(y_i, f(x_i))$ below, denotes cross entropy loss between predicted $f(x_i)$ and the label y_i. The empirical risk minimization using labeled source data can be written as follows.

$$\hat{\theta}_{\mathrm{ERM}} = \mathrm{argmin}\left[\frac{1}{N_s}\sum_{i=1}^{N_s} l\left(y_i^s \;\; f_\theta\left(x_i^s\right)\right)\right]$$

Since Domain Adaptation doesn't assume train set and test set belong to equal distributions, empirical risk minimization isn't consistent (Shimodaira, 2000). In Domain Adaptation setting we want to learn a model that can minimize the following equation.

$$\theta^* = \mathrm{argmin}\left[\frac{1}{N_t}\sum_{i=1}^{N_t} l\left(y_i^t, f_\theta\left(x_i^t\right)\right)\right]$$

Since target data's labeled data doesn't exist in this settings, we need to use the knowledge of source data instead. Let p_s and p_t be the probability density functions related to source and target data respectively. We can change it to following equation.

$$\begin{aligned}\theta^* &= \mathrm{argmin}\left[\frac{1}{N_t}\sum_{i=1}^{N_t} l\left(y_i^t, f_\theta\left(x_i^t\right)\right)\right] \\ &\approx \mathrm{argmin}\left[\frac{1}{N}\sum_{i=1}^{N}\frac{p_t}{p_s} l\left(y_i^s, f_\theta\left(x_i^s\right)\right)\right]\end{aligned}$$

Therefore, if we estimate the ratio between source data and target data (importance weights) in all data spaces, we can solve the domain adaptation problems (Shimodaira, 2000).

Importance weight based domain adaptation approach requires distribution in source and target data to calculate importance weight in all samples. However, in many real data sets, distribution of target and the source data in unknown and therefore it is not easy to estimate the importance weights.

There were several approaches to estimate importance weight in source and target data. There was approach by Sugiyama *et al.* (2007), which used kernel density estimation between source and target data. Then researchers focused on directly estimating the importance weights without estimating source and target data distribution. Bickel *et al.* (2009) suggested using kernel logistic regression method and an exponential model classifier for covariate shift. Sugiyama *et al.* (2008) estimated importance weight by finding the weight that minimizes Kullback-Leibler divergence from the source density and the estimated density. Gretton *et al.* (2009). They suggested Kernel Mean Matching (KMM) method that matches covariate distributions between source and target sets in a high dimensional feature space. They solved this problem by changing it to quadratic problem.

However, kernel density estimation has to estimate the density of the data and then estimate again the importance weights, but it suffered from curse of dimensionality. Rather than estimating the distribution or the importance weight, we thought we could directly use the label of the source data by using the kernel based similarity.

3. PROPOSED METHOD

3.1 Unsupervised Domain Adaptation

In unsupervised domain adaptation settings, we have only labeled data in source datasets. Therefore, following the notation in section 2, the empirical loss for source data will be given as:

$$L^l\left(D_s^l;\theta\right) = \frac{1}{N_t}\sum_{i=1}^{N_t} l\left(y_i^s \;\; f_\theta\left(x_i^s\right)\right) \quad (1)$$

However, the above empirical loss cannot reflect the difference between training and test distribution. In most covariate shift methods, although this problem was addressed by introducing importance weight, they needed to assume that the support of source distribution contains the support of target distribution.

In this paper, we introduce new kernel weight domain adaptation method based on smoothness assumption of classifier, which means that similar inputs have similar class distributions (LeCun *et al.*, 2015).

The kernel weighted loss is defined as follows:

$$L^u\left(D_s^l, D_t^u;\theta\right)=\frac{1}{N_s}\frac{1}{N_t}\sum_{i=1}^{N_s}\sum_{j=1}^{N_t}k\left(x_i^s\ x_j^t\right)l\left(y_i^s\ f_\theta\left(x_j^t\right)\right) \quad (2)$$

where $k(x, x')$ is kernel function of the inputs x and x'. Since kernel function is a kind of similarity function, the more similar the inputs are, the larger the kernel function value of them becomes. We used popular kernel function rbf kernel $k(x, x') = \exp(-qx - x'\|^2)$ in our metric. This loss makes prediction of a target input that resembles true labels of source data depending on the input similarities.

In unsupervised domain adaptation, we jointly minimize (1) and (2), and prediction function is learned not only to fit the source data but also to utilize source labels for target prediction based on kernel weights of source and target inputs. Because the kernel weighted loss harnesses smoothness assumption, it is necessary to train the model with equation (1) when using equation (2).

From the perspective of kernel density estimation, we can see that using kernel weights is related to estimating target class with source distribution. Let $\hat{p}(x)$ be an unnormalized estimated distribution from data $\{x_1, \ldots, x_N\}$, then its kernel density estimation is:

$$\hat{p}\left(x\right)\propto\frac{1}{N}\sum_{i=1}^{N}k(x-x_i) \quad (3)$$

Using equation (3), when the binary cross entropy loss is used, $\hat{p}(\mathrm{y}=+|\mathrm{x})=\mathrm{f}(\mathrm{x})$ and $\hat{p}(\mathrm{y}=-|\mathrm{x})=1-\mathrm{f}(\mathrm{x})$, we can represent (2) as follows:

$$\frac{1}{N_t}\sum_{j=1}^{N_t}\begin{bmatrix}\hat{p}_s\left(x_j^t,+\right)\log f_\theta\left(x_j^t\right)\\+\hat{p}_s\left(x_j^t,-\right)\log\left(1-f_\theta\left(x_j^t\right)\right)\end{bmatrix} \quad (4)$$

where
$\hat{p}_s\left(x,+\right)=\frac{N_{s_+}}{N_s}\hat{p}_s(x|+),\ \hat{p}_s\left(x,-\right)=\frac{N_{s_-}}{N_s}\hat{p}_s(x|-)N_{s+}$ is the number of source data with positive label, and $\mathrm{N_{s\_}}$ is the number of source data with negative label. The detailed derivation of equation (4) is given in Appendix. The equation (4) demonstrates the effect of equation (2) that we calculate the cross entropy loss with the estimated target label from source distribution instead of true target label. This equation can be easily extended to multi-class classification by using cross entropy loss and class probabilities.

When training the classification model $f(x)$, we combine the two loss functions (1) and (2) with γ.

$$L^T(D_s^l, D_t^u;\theta,\gamma)=L^l\left(D_s^l;\theta\right)+\gamma L^u\left(D_s^l,\mathrm{D_t^u};\theta\right) \quad (5)$$

Where γ is non-negative hyper-parameter adjusting the weight between equation (1) and (2).

While kernel matrix from source and target input, $\mathrm{K}\in\mathbb{R}^{\mathrm{N_s}\times\mathrm{N_t}}, K_{i,j}=k(x_i,x_j)$, is computed once, the cost of computing Loss2 is still high due to evaluating the loss function on every example in entire dataset. We disentangle this problem through stochastic gradient descent method. Instead of updating parameters with entire dataset, the parameters are updated with randomly sampled mini batch dataset. It can reduce the computational cost and converge much faster (LeCun *et al.*, 2015).

3.2 Semi-Supervised Domain Adaptation

In our domain adaptation method, we can easily expand our model to semi-supervised model when $\mathrm{N_t^l}$ partial labels of target data are given and there are also few unlabeled data in source domain. We denote the labeled target dataset as $\mathrm{D_t^l}=\{(\mathrm{x_1^t},\mathrm{y_1^t}),\ldots,(\mathrm{x_{N^l}^t},\mathrm{y_{N^l}^t})\}$ where $\mathrm{N_t^l}\le N_t$ and the unlabeled source dataset as $\mathrm{D_s^u}=\{x_1^s, x_2^s, \ldots, x_{N_s}^s\}$. Using the labeled target dataset $\mathrm{D_t^l}$, we can add two additional loss $\mathrm{L^l}(D_t^l)$ and $\mathrm{L^u}(D_t^l, D_s^u)$ to equation (5):

$$\begin{aligned}\mathrm{L_{semi}^T}\left(D_s^l, D_s^u, D_t^l, D_t^u;\theta,\gamma\right)=&\\ L^l\left(D_s^l;\theta\right)+\gamma_1 L^u\left(D_s^l, D_t^u;\theta\right)&\\ +\gamma_2 L^l\left(D_t^l;\theta\right)+\gamma_3 L^u\left(D_t^l, D_s^u;\theta\right)&\end{aligned} \quad (6)$$

where $\boldsymbol{\gamma}=\{\gamma_1,\gamma_2,\gamma_3\}$.

The combined loss learns the shared prediction function $\mathrm{f_\theta}(x)$ which predicts similar class label even if given similar inputs across domains.

4. EXPERIMENT

4.1 Toy Data

As a first experiment, we examined the proposed distance kernel based domain adaptation algorithm on the toy data. We wanted to find out similarity based approach works on intuitive toy examples. We used isotropic Gaussian blobs data which is illustrated in Figure 1. The black data represents source data, and the markers O and X represents positive and negative labels respectively. The grey scale data represents target data. In this example, the source data and target data distribution similar but different. And the distance between samples with same label is closer than the other. The results are in Table 1.

As we can see in the Table 1. This problem is not easy to transfer the knowledge to source domain to target domain, because the baseline algorithm which is based on equation (1) is significantly lower than the model trained on target dataset. This model will be notated as goal of domain adaptation problem, and it can be notated as equation (7)

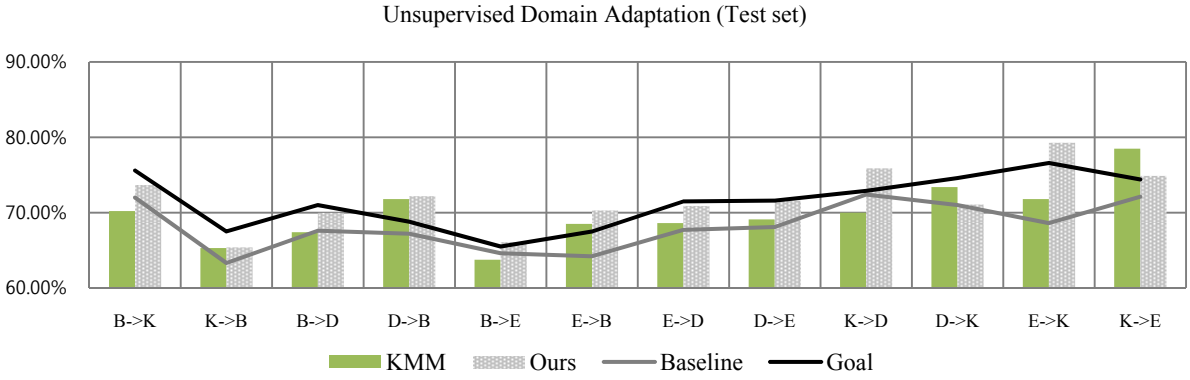

Figure 1. Mean classification accuracy (%) on the target test set, which was not used to calculate the kernel distance between source data and target data set.

$$L^l\left(D_t^l;\theta\right)=\frac{1}{N_t}\sum_{i=1}^{N_t} l\left(y_i^t \;\; f_\theta\left(x_i^t\right)\right) \tag{7}$$

Our model showed comparable achievement to the goal in this toy data. We can conclude that if the distance between source and target data with the same label is close, our model works well.

4.2 Real Data

In this paper, we used amazon review data regarding four different domains: Books, DVDs, Electronics, and Kitchen. We regarded reviews with rating larger than 3 as positive reviews and rating smaller than 3 as negative ones. Our review data contained large difference in number of reviews in each domain, and also the number of positive and negative reviews was imbalanced. Therefore, balanced 2500 reviews in every domain to balance the number of the data used in our experiment. We then used 500 reviews as test set.

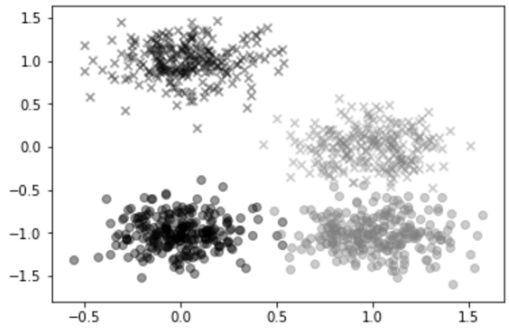

Figure 2. Isotropic Gaussian blobs data. Source positive data, source negative data, target positive data, and target negative data has their centers in (0, -1), (0, 1), (1, -1), and (1, 1) respectively.

Table 1. Mean classification accuracy (%) for unsupervised domain adaptation in toy data. Baseline refers to simple logistic regression trained only on source data, and Goal refers to logistic regression trained on target data

	Ours	Baseline	Goal
Train (Source)	100.00%	100.00%	98.00%
Train (Target)	95.20%	68.60%	100.00%
Test (Source)	100.00%	99.00%	99.00%
Test (Target)	91.00%	67.00%	100.00%

4.3 Unsupervised Domain Adaptation

We conducted 12 cross domain experiments with four domains. Each review test was preprocessed as bag-of-words and transformed in to binary vectors by using unigram model. We used 6786 sentiment words suggested by Baccianella *et al*. (2010) as dictionary in our bag-of-words representation. Since bag-of-words representation had high dimension (6274) we used linear dimension reduction method (SVD) with source and target domain together to lower the dimension to 100.

We compared four methodologies in our experiment. The baseline is using logistic regression model with cross entropy equation (1) that is trained on source data, and directly applied to target data. The goal baseline is model based on equation (7). We used Kernel Mean Matching as a comparison to out proposed method.

The results of unsupervised domain adaptation are illustrated in Figure 2. Figure 2 shows mean classification accuracy (%) on the Amazon review data set. It illustrated performance in the test set of target data set, our proposed method was significantly better than baseline logistic regression method in all twelve experiments. Also ten pair of experiments were better than KMM. Moreover, ours is better accuracy than the goal algorithm in five areas, which used target data as training.

4.4 Semi-Supervised Domain Adaptation

As we have shown in section 3.2, we can easily expand our method to semi-supervised model. In this experimental setting, we used 500 data in target data as labeled

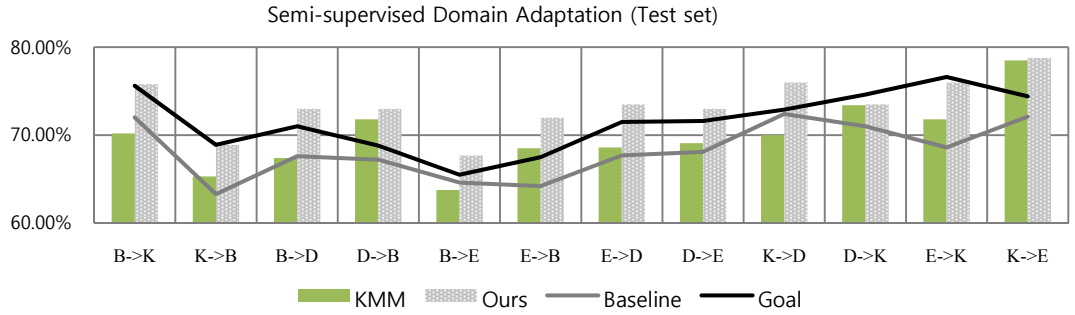

Figure 3. Mean classification accuracy (%) on the target test set, which was not used to calculate the kernel distance between source data and target data set.

data. We then trained the model by using 2000 labeled source data and 2000 unlabeled target data with 500 labeled target data by using the loss function equation (6).

Figure 3 illustrates the performance of our model in semi-supervised setting. It shows higher accuracy compared to KMM significantly in all experimental pairs. Moreover, it showed average 2%p higher accuracy than the goal which used 2500 samples of labeled target data set.

5. CONCLUSION

We present in this work a distance kernel based domain adaptation model that improves existing importance weight based domain adaptation technique. The model does so by assuming similar inputs have similar class distributions and making prediction of a target input that resembles true labels of source data depending on the input similarities. Our results on Amazon dataset outperformed baseline method and existing method. We expect our paper will contribute to domain adaptation in sentimental analysis when labeled data in target data does not exist. The proposed method is intuitive and simple approach that can be applied in various domain adaptation problem.

In our further research, we plan to consider combining our proposed loss function with previous models in domain adaptation. Then the average accuracy of the experiments will be better than this research. Moreover, the classifier we used in this model can be simply expanded to deeper neural network based models which can yield better performance. Finally, since we expect we can combine this model to recent deep learning based representation domain adaptation algorithms, to construct a transferrable classifier as well as fine representation.

ACKNOWLEDGEMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MEST) (No. 2016R1A2B30140 30). Also it was supported by the National Research Foundation of Korea (NRF) Grant funded by the Korean Government (MSIP) (No.2017R1A5A10156 26).

REFERENCES

Baccianella, S., Esuli, A., and Sebastiani, F. (2010), Senti Word Net 3.0: An enhanced lexical resource for sentiment analysis and opinion mining, *LREC*, 2200-2204.

LeCun, Y., Bengio, Y., and Hinton, G. (2015), Deep learning, *Nature*, **521**, 436-444.

Bickel, S., Brückner, M., and Scheffer, T. (2009), Discriminative learning under covariate shift, *Journal of Machine Learning Research*, **10**(Sep), 2137-2155.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F. Marchand, M., and Lempitsky, V. (2016), Domain-adversarial training of neural networks, *Journal of Machine Learning Research,* **17**, 1-35.

Glorot, X., Bordes, A., and Bengio, Y. (2011), Domain adaptation for large-scale sentiment classification: A deep learning approach, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 513-520.

Gretton, A., Smola, A. J., Huang, J., Schmittfull, M., Borgwardt, K. M., and Schölkopf, B. (2009), *Covariate shift by kernel mean matching.* In: Quiñonero-Candela, J., Sugiyama, M., Schwaighofer, A., and Lawrence, N. D. (Eds.), MIT Press, Cambridge, MA, USA, 131-160.

Bousmalis, K., Trigeorgis, G., Silberman, N., Krishnan, D., and Erhan, D. (2016), Domain separation networks, In *Advances in Neural Information Processing Systems*, 343-351.

Pan, S. J. and Yang, Q. (2010), A survey on transfer learning, *IEEE Transactions on Knowledge & Data Engineering*, **22**(10), 1345-1359.

Shimodaira, H. (2000), Improving predictive inference under covariate shift by weighting the log-likelihood function, *Journal of Statistical Planning and Infe-*

rence, **90**(2), 227-244.

Sugiyama, M., Krauledat, M., and MÃžller, K. R. (2007), Covariate shift adaptation by importance weighted cross validation, *Journal of Machine Learning Research*, **8**(May), 985-1005.

Sugiyama, M., Nakajima, S., Kashima, H., Buenau, P. V., and Kawanabe, M. (2008), Direct importance estimation with model selection and its application to covariate shift adaptation, *Proceedings of the 20th International Conference on Neural Information Processing Systems*, Vancouver, British Columbia, Canada, 1433-1440.

APPENDIX

In this section, we induce the equation (4) from equation (2) using equation (3).

$$\frac{1}{N_s}\frac{1}{N_t}\sum_{j=1}^{N_t}\sum_{i=1}^{N_s}k\left(x_i-x_j\right)l\left(y_i,f_\theta\left(x_j\right)\right)$$

$$\propto\frac{1}{N_s}\frac{1}{N_t}\sum_{j=1}^{N_t}\left\{\sum_{i\in S_+}k\left(x_i^s-x_j^t\right)\log f_\theta\left(x_j^t\right)+\sum_{i\in S_-}k\left(x_i^s-x_j^t\right)\log\left(1-f_\theta\left(x_j^t\right)\right)\right\}$$

$$=\frac{1}{N_t}\sum_{j=1}^{N^t}\left\{\frac{N_{s_+}}{N_s}\log f_\theta\left(x_j^t\right)\frac{1}{N_{s_+}}\sum_{i\in S_+}k\left(x_i^s-x_j^t\right)+\frac{N_{s_-}}{N_s}\log\left(1-f_\theta\left(x_j\right)\right)\frac{1}{N_{s_-}}\sum_{i\in S_-}k\left(x_i^s-x_j^t\right)\right\}$$

$$=\frac{1}{N_t}\sum_{\mathrm{j}=1}^{\mathrm{N_t}}\left\{p_s\left(+\right)\hat{p}_{s|+}\left(x_j^t|+\right)\log f_\theta\left(x_j^t\right)+p_s\left(-\right)\hat{p}_{s|-}\left(x_j^t|-\right)\log\left(1-f_\theta\left(x_j^t\right)\right)\right\}$$

$$=\frac{1}{N_t}\sum_{j=1}^{N_t}\left\{p_s\left(x_j^t,+\right)\log f_\theta\left(x_j^t\right)+p_s\left(x_j^t,-\right)\log\left(1-f_\theta\left(x_j^t\right)\right)\right\}$$