Transfer Forest Based on Covariate Shift

Masamitsu Tsuchiya
SECURE, INC.
tsuchiya@secureinc.co.jp

Yuji Yamauchi, Takayoshi Yamashita, Hironobu Fujiyoshi
Chubu University
yuu@vision.cs.chubu.ac.jp, {yamashita, hf}@cs.chubu.ac.jp

Abstract

Random Forest, a multi-class classifier based on statistical learning, is widely used in applications because of its high generalization performance due to randomness. However, in applications such as object detection, disparities in the distributions of the training and test samples from the target scene are often inevitable, resulting in degraded performance. In this case, the training samples need to be reacquired for the target scene, typically at a very high human acquisition cost. To solve this problem, transfer learning has been proposed. In this paper, we present data-level transfer learning for a Random Forest using covariate shift. Experimental results show that the proposed method, called Transfer Forest, can adapt to the target domain by transferring training samples from an auxiliary domain.

1. Introduction

A Random Forest [3], a multi-class classifier based on statistical learning, is widely used in applications because of its high generalization performance due to randomness. In the field of computer vision, Random Forests are used in image classification [14], semantic segmentation [14], real-time keypoint recognition [10], and human detection and action recognition [20]. However, in certain applications such as object detection, disparities in the distributions of the training and test samples from the target scenes are often inevitable, resulting in degraded detection performance. In such a case, the training samples need to be reacquired for the target scenes, albeit at a very high human acquisition cost. For the target scene, sample acquisition must be correct without object position and scale variations of the images. Therefore, it is natural to re-use previously acquired data and expensive detectors.

Next, we address a problem. This study focuses on learning for two source samples, and learning is less for one source that supplementarily uses another. To overcome the problem, transfer learning has been proposed [18, 5, 11]. Transfer learning methods retrain the data efficiently by acquiring a small number of new samples from target scenes and the previously acquired training sample data to detect pedestrians in scenes with different camera-tilt and illumination. Therefore, we introduce a new Random Forest for transfer learning to the Random Forest family, which was organized into the following categories by Criminisi *et al.* [4]: classification forest, regression forest, density forest, manifold learning forest, and semi-supervised learning forest.

In this paper, we present transfer learning for Random Forests using covariate shift. Our experimental results show that our proposed method can adapt, with low cost, to target scenes by transferring training samples.

2. Transfer learning using covariate shift

Transfer learning is defined as the problem of retaining and applying the knowledge learned in one or more tasks to develop an effective hypothesis for a new task [12]. The related research can be roughly divided into three categories in accordance with the level of transfer knowledge. Model-level transfer learning estimates the hyper prior of a model's parameters from several tasks, and then transfers this hyper prior to similar tasks [9, 2, 19, 13]. Data-level transfer learning discovers useful samples from the auxiliary tasks, and then uses these together with the target samples for learning [5, 15]. Feature-level transfer learning searches for shared features with sufficient performance in two domains [1, 8, 16].

In data-level transfer learning, the training dataset is divided into target and auxiliary training samples, where all selected D_t and D_a, respectively, are found by the sampling sources. Although the auxiliary distribution of sample x, $p_a(x)$, generally differs from the target distribution $p_t(x)$, the conditional probability distribution can be considered equal: i.e., $p_a(y|x) = p_t(y|x)$, where y is the class label for two-class problems $y \in -1, 1$. The auxiliary sample must become the next target. Therefore, this auxiliary sample, which is not too close to the target, is assumed to be noise. Thus, data-level transfer learning with covariate shift is a weighting auxiliary sample using covariate loss that estimates both domains [11, 15]. Covariate loss λ is given by Eq. (1):

$$\lambda = \frac{p_t(y|x)}{p_a(y|x)}. \tag{1}$$

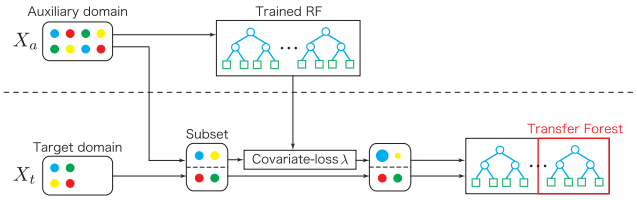


Figure 1. Overview of the Transfer Forest.

Pang *et al.* [11] reformulated the density ratio λ with the conditional probabilities for the boosting classifier as follows:

$$\lambda = \frac{1 + e^{-yH_a(x)}}{1 + e^{-yH_t(x)}}, \tag{2}$$

where H is a boosting classifier. H_a is trained on auxiliary data D_a, and H_t is adaptively trained with examples from both the auxiliary data D_a and target data D_t. This reformulation means that covariate loss can estimate both classifier outputs. We incorporate covariate loss λ into Random Forest training. Our proposed method in this paper is data-level transfer learning for Random Forests.

3. Transfer Forest

This section describes our proposed method called Transfer Forest. We introduce covariate shift loss into a framework of Random Forests as data-level transfer learning. Fig. 1 gives an overview of the Transfer Forest.

3.1. Auxiliary and target domains

A distribution disparity is often inevitable between the pedestrian training samples and test samples from a specific application scenario [17]. In this study, we define learning by training sample $X_a = \{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2, ..., \boldsymbol{\chi}_j\}$ as the auxiliary domain and training samples $X_t = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_i\}$ of specific scenes in an actual installation environment as the target domain. As the Random Forest in the auxiliary domain can be learned using offline processing, a large number of training samples are required.

For the target domain, we used specific scenes taken from different camera angles. In contrast to the auxiliary-domain data, target-domain data must be newly obtained, which means that having as few samples as possible is desirable. In this study, we address the problem of how to train an efficient Random Forest using a small number of training samples in the target domain on the basis of a previously trained Random Forest and training samples from the auxiliary domain X_a.

In our proposed method, we construct a Random Forest classifier using target and auxiliary samples. Both samples were selected with a degree of randomness for constructing a tree. During training, auxiliary samples are iteratively reweighted by evaluating "how far to the target" using the covariate loss between the target sample and auxiliary distributions. The covariate loss between the two domains is estimated from the classification output of the two forests already constructed: the auxiliary Random Forest and Transfer Forest in learning.

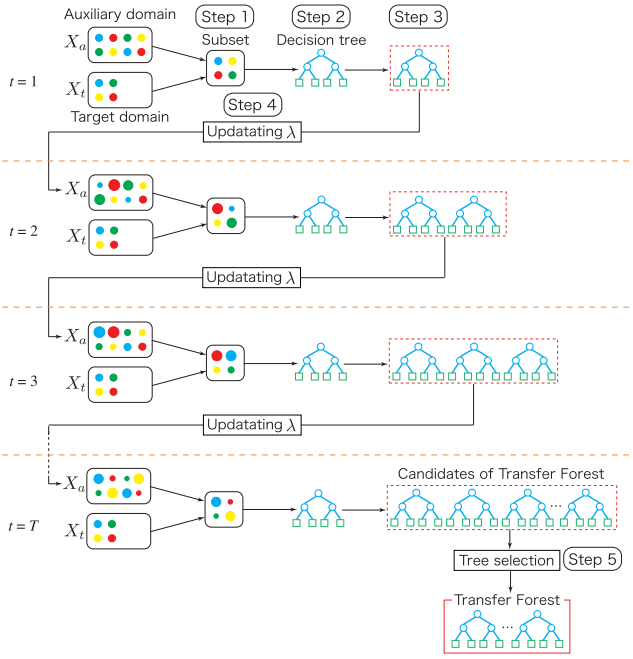


Figure 2. Training process for the Transfer Forest.

3.2. Training process of the Transfer Forest

Training of the Transfer Forest consists of the following five steps.

Step 1 First, we create a subset by randomly selecting samples from both domains X_t and X_a, where the number of training samples from each is equal.

Step 2 From the subset, we train a decision tree using covariate loss λ. Each training sample is weighted by the covariate loss λ.

Step 3 We include the decision tree trained in step 2 as a candidate for the Transfer Forest. The covariate loss λ is updated using the Transfer Forest and Random Forests previously trained using the auxiliary samples.

Step 4 These steps are repeated until a large number of candidates for the Transfer Forest is obtained.

Step 5 The latter half of the candidates were selected as the Transfer Forest.

The differences in the training processes for the original Random Forest and the proposed Transfer Forest are 1) the weighting and updating of the samples and 2) tree selection.

3.3. Training of decision tree using covariate loss

Here, we describe the details of the training algorithm for the Transfer Forest using covariate loss. First, subsets

X_t of the target training sample and X_a of the auxiliary one, $X_t = x_i, c_k; i \in [1, N], k \in [1, C]$ and $X_a = x_j, c_k; j \in [1, M], k \in [1, C]$, are created to train the decision trees. Then, both subsets consist of a randomly selected set of S sample images from each domain. Accordingly, subset λ_t of the covariate loss is partitioned for X_a, $\Lambda_t = \lambda_i; i \in [1, M]$. Λ is estimated as follows, which is incorporated in the Random Forest as Eq. (2):

$$\lambda_j = \frac{1 + e^{P_a(c_k|\boldsymbol{\chi}_j)}}{1 + e^{P_t(c_k|\boldsymbol{\chi}_j)}}, \tag{3}$$

where $P_a(c_k|\boldsymbol{\chi}_j)$ is the probability of a class label from the previously trained Random Forest using auxiliary domain samples, and $P_t(c_k|\boldsymbol{\chi}_j)$ is the probability of a class label from the Transfer Forest constructed using both domain samples. When the first tree is being constructed, $P_t(c_k|\boldsymbol{\chi}_j)$ is assumed to be 0, so λ_j is initialized to $1 + e^{P_a(c_k|\boldsymbol{\chi}_j)}$.

Nodes are constructed of a split function, a feature, and a threshold. The split function for the tree is the same as that for the original Random Forest, and the set I_n of training samples at node n is divided into child nodes I_l, I_r.

$$I_l = \{i \in I_n | f(\boldsymbol{x}_i) < t\}, \tag{4}$$
$$I_r = I_n \setminus I_l. \tag{5}$$

For prepared features $f_m; m \in [1, M]$ and thresholds $\theta m, k; k \in [1, K]$, the best combination is selected on the basis of information entropy calculated as

$$\Delta E = \frac{|I_l|}{|I|} E(I_l) \frac{|I_r|}{|I|} E(I_r). \tag{6}$$

Note that $E(I_l)$ and $E(I_r)$ denote Shannon entropy for the samples in each class when taking the left or right branch, respectively, for a given combination of features and thresholds. Shannon entropy is computed as

$$E(I) = \sum_{k=1}^{C} P(c_k) \log P(c_k), \tag{7}$$

while $P(c_k)$ is the probability distribution for class c_k at the node computed as

$$P(c_k) = \frac{|I^{t,c_k}| + |\Lambda^{a,c_k}|}{|I^t| + |\Lambda^a|}, \tag{8}$$
$$|\Lambda^{a,c_k}| = \sum_{j:y_j=c_k} \lambda_j. \tag{9}$$

$|I^{t,c_k}|$ denotes the number of target samples for class c_k, $|\Lambda^{a,c_k}|$ is the summation λ of auxiliary samples for class c_k, and $|I^t|$ is the number of target samples for all classes, and $|\Lambda^a|$ is summation λ of all auxiliary samples.

Subsets are partitioned using the features selected as described above. Feature values less than the threshold form the subset for the left child node, while values greater than the threshold form the subset for the right child node. This process is repeated at each child node using the newly formed subsets. Node generation terminates when the number of training samples is less than a pre-determined depth, when the training samples comprise only a single class, or when the nodes have reached a certain depth. Termination retains the stored probability distribution $P(c_k|l)$, which is computed as Eq. (8).

After the tree is constructed, λ is re-estimated using Eq. (3) with the updated Transfer Forest, and the next tree is constructed.

Fig. 3 shows an example of covariate loss λ after the training. Some training samples of the auxiliary domain are weighted with small values, which are not close to the target domain. These training samples are spoiled by the covariate loss λ for training the Transfer Forest by adapting to the target domain to prevent performance degrading.

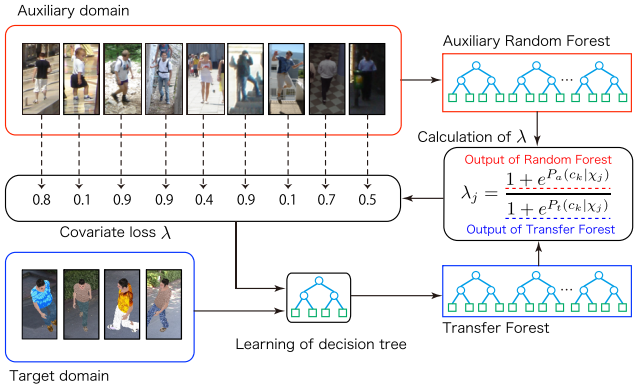


Figure 3. Example of covariate loss λ after the training.

3.4. Tree selection for Transfer Forest

Transferred parameters, such as covariate shift, are asymptotically approximated in sequential training. Hence, these values are not reliable when constructing the first tree, because the distribution estimates are insufficient. TrAdaBoost [5] was adopted after $T/2$ weak classifiers constituted a strong classifier without a forward constructed $T/2$. Based on the experimental results, this method using T outperforms other classifiers, although this is a weak boosting classifier that is not independent and chooses the best combinations of weak classifiers. If the best feature sets are chosen in the early stage, they contain inadequate numbers of strong classifiers. As the Random Forest is independent of each tree, we believe that using the same idea for the latter half of the forest is more reliable.

3.5. Classification process

Here we present the Transfer Forest classification algorithm, which is the same as the original random tree of the Transfer Forest. The algorithm begins by inputting an unknown sample x into each of the decision trees created by

the training algorithm. It then transverses each decision tree by branching left or right at each node using the splitting function and output class probability $P_t(c|x)$ saved at the leaf node previously arrived at. Then, as shown by Eq. 10, the algorithm calculates the average value of the outputs from the latter half of the decision tree in the Transfer Forest.

$$P(c|x) = \frac{2}{T} \sum_{t=T/2}^{T} P_t(c|x). \tag{10}$$

Final output $\hat{y}$ obtained by Eq. 11 determines the class with the highest probability.

$$\hat{y} = \arg\max_{c} P(c|x). \tag{11}$$

4. Experiments

The proposed method has been evaluated on both synthetic and real datasets. Using the real dataset, we performed two experiments on pedestrian detection to show the effectiveness of the proposed method in terms of binary and multi-class classification.

4.1. Synthetic data experiments

Fig. 4(a) illustrates the synthetic data experiment on two-dimensional spiral data. Solid lines in the spiral shape in Fig. 4(a) show 1,000 auxiliary samples, with a single color representing a separate class. Wavy lines in Fig. 4(a) show the training samples of the target domain, which were rotated 30 degrees from the auxiliary domain. We trained the Transfer Forest with the number of trees set to 50 and the maximum depth of a tree set to 5.

Fig. 4(b) and (c) shows the decision boundaries of the Transfer Forest with respect to a varying number of target samples and those of the original Random Forest using both the auxiliary and target training samples. When the number of target samples is reduced, the decision boundary of the Transfer Forest seems better than that of the original Random Forest. With a sufficient number of target samples, the decision boundaries of the Transfer Forest and original Random Forest are almost the same.

4.2. Real data experiments for binary classification

In this experiment, we used two different datasets of pedestrian detection data as a binary classification problem. To begin with, we pre-trained an Random Forest classifier on the basis of histograms of oriented gradient (HOG) features [6]. For the HOG features, the cell size was set to 8 and block size to 2 with a total of 3,780 dimensions. Our experiments used two datasets, namely, the INRIA person and DaimlerChrysler datasets, as the target and the auxiliary domains, respectively.

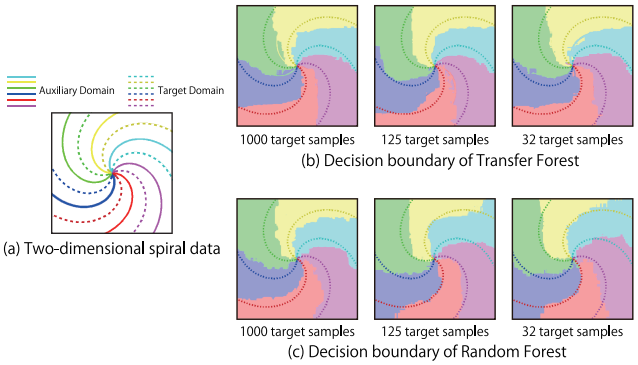


Figure 4. Experimental results using two-dimensional spiral data.

Auxiliary domain X_a: DaimlerChrysler Mono Pedestrian Detection Benchmark Dataset [7]

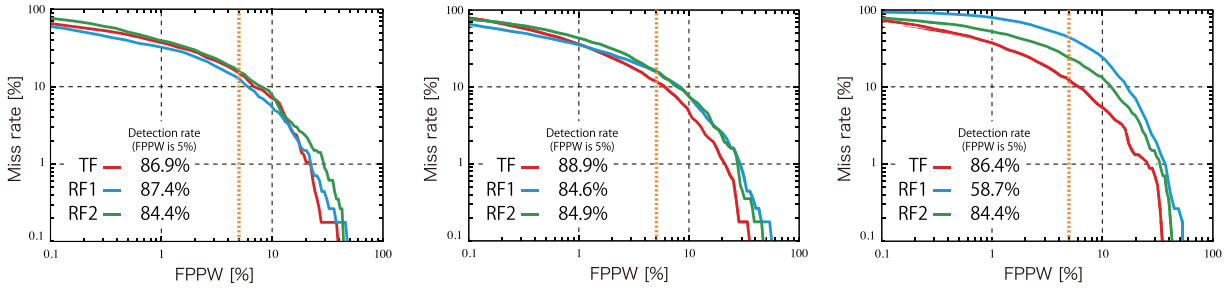
This dataset consists of 15,600 images of people for training, 6,700 background images for training, and 21,800 images, containing 56,500 pedestrians, for evaluation.

Target domain X_t: INRIA Person Dataset [6]

This dataset consists of 2,416 images of people for training, 1,218 background images for training, 1,135 images of pedestrians for evaluation, and 453 background images for evaluation.

We trained a Transfer Forest using both datasets described in above. To demonstrate the effectiveness of the proposed Transfer Forest, we compared it with a Random Forest (RF1) trained using only training samples of the target domain and a Random Forest (RF2) trained using both the auxiliary and target samples. We used the same parameters for both the Transfer Forest and Random Forest with the number of trees set to 50 and the maximum depth of a tree set to 5.

The detection-error-tradeoff curve while the number of target samples reduces from 2,414 (full) to 100 is shown in Fig. 5. From Fig. 5(a), the classification performances are almost the same for the Transfer Forest (TF), the Random Forest (RF1) with only the target domain, and the Random Forest (RF2) with both domains, with a target sample of 2416. When the number of target samples is reduced to 800, the classification performances deteriorated for the Random Forest (RF1) using only the target domain and the Random Forest (RF2) with both domains. This is because the number of training samples of the target domain had decreased. In contrast, Transfer Forest (TF) maintains its classification ability by incorporating a large number of auxiliary samples with the small number of target samples. When the number of target samples is reduced to 100, the classification performance is very low for the Random Forest (RF1) with only a target domain. This is because 100 images of training samples are not sufficient to represent the distribution of training samples in a target domain. The proposed method



(a) Number of target samples: 2416 (b) Number of target samples: 800 (c) Number of target samples: 100

Figure 5. Comparison of the performances of Transfer Forest and the conventional method.

performs 11% better than the Random Forest (RF2) under both domains. Thus, we say that the Transfer Forest can adopt auxiliary samples to the target domain using a small number of target samples by the covariate shift approach.

5. Conclusion

In this paper, we proposed a framework for Random Forests incorporating data-level transfer learning with covariate shift. The proposed Transfer Forest can deal with multi-domain samples using covariate shift in the Random Forest. Experimental results show that the proposed Transfer Forest has the same or better performance with a small number of target samples than re-training using all target samples.

Future work includes incorporating feature-level transfer learning into the Random Forest framework.

References

[1] R. Ando and T. Zhang. A Framework for Learning Predictive Structures from Multiple Tasks and Unlabeled Data. *Journal of Machine Learning Research*, 6:1817–1853, 2005. 1

[2] B. Bakker and T. Heskes. Task Clustering and Gating for Bayesian Multitask Learning. *Journal of Machine Learning Research*, 4:83–99, 2003. 1

[3] L. Breiman. Random Forests. In *Machine Learning*, volume 45, pages 5–32, 2001. 1

[4] A. Criminisi, J. Shotton, and E. Konukoglu. Decision Forests: A Unified Framework for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning. *Foundations and Trends in Computer Graphics and Vision*, 7(2-3):81–227, 2012. 1

[5] W. Dai, Q. Yang, G. Xue, and Y. Yu. Boosting for Transfer Learning. In *International Conference on Machine Learning*, pages 193–200, 2007. 1, 3

[6] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 886–893, 2005. 4

[7] M. Enzweiler and D. M. Gavrila. Monocular Pedestrian Detection: Survey and Experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2179–2195, 2009. 4

[8] A. Farhadi and M. Tabrizi. Learning to Recognize Activities from the Wrong View Point. In *European Conference on Computer Vision*, pages 154–166, 2008. 1

[9] L. Fei-Fei, R. Fergus, and P. Perona. One-Shot Learning of Object Categories. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 28:594–611, 2006. 1

[10] V. Lepetit, P. Lagger, and P. Fua. Randomized trees for real-time keypoint recognition. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 775–781, 2005. 1

[11] J. Pang, Q. Huang, S. Yan, S. Jiang, and L. Qin. Transferring Boosted Detectors Towards Viewpoint and Scene Adaptiveness. *IEEE Transactions on Image Processing*, 20(5):1388–1400, 2011. 1, 2

[12] M. Rosenstein, Z. Marx, T. Dietterich, and L. Kaelbling. Transfer Learning with an Ensemble of Background Tasks. In *NIPS Workshop on Inductive Transfer: 10 Years Later*, 2005. 1

[13] M. Rosenstein, Z. Marx, L. Kaelbling, and T. Dietterich. To transfer or not to transfer. In *NIPS Workshop on Inductive Transfer: 10 Years Later*, 2005. 1

[14] J. Shotton, M. Johnson, and R. Cipolla. Semantic Texton Forests for Image Categorization and Segmentation. In *IEEE Conf. on Computer Vision and Pattern Recognition*, June 2008. 1

[15] M. Sugiyama, M. Krauledat, and R. Müller. Covariate Shift Adaptation by Importance Weighted Cross Validation. *Journal of Machine Learning Research*, 8:985–1005, 2007. 1

[16] S. Thrun. Is Learning The n-th Thing Any Easier Than Learning The First? In *Advances in Neural Information Processing Systems*, pages 640–646, 1996. 1

[17] M. Tsuchiya, Y. Yuji, Y. Yamashita, and H. Fujiyoshi. Hybrid Transfer Learning for Efficient Learning in Object Detection. In *Asian Conference on Pattern Recognition*, pages 69–73, 2013. 2

[18] M. Wang, W. Li, and X. Wang. Transferring a Generic Pedestrian Detector Towards Specific Scenes. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2012. 1

[19] A. Wilson, A. Fern, S. Ray, and P. Tadepalli. Multi-task reinforcement learning: a hierarchical Bayesian approach. In *International Conference on Machine Learning*, volume 227, pages 1015–1022, 2007. 1

[20] T. Yamashita, Y. Yamauchi, and H. Fujiyoshi. Human Detection for Multiple Pose by Boosted Randomized Trees. In *Asian Conference on Pattern Recognition*, pages 229–233, 2011. 1