Adaptive Techniques for Intra-User Variability in Keystroke Dynamics

Hayreddin Çeker and Shambhu Upadhyaya
Department of Computer Science and Engineering
University at Buffalo
Buffalo NY 14260, USA
hayreddi@buffalo.edu, shambhu@buffalo.edu

Abstract

Conventional machine learning algorithms based on keystroke dynamics build a classifier from labeled data in one or more sessions but assume that the dataset at the time of verification exhibits the same distribution. A user's typing characteristics may gradually change over time and space. Therefore, a traditional classifier may perform poorly on another dataset that is acquired under different environmental conditions. In this paper, we investigate the applicability of transfer learning to update a classifier according to the changing environmental conditions with minimum amount of re-training. We show that by using adaptive techniques, it is possible to identify an individual at a different time by acquiring only a few samples from another session, and at the same time obtain up to 13% higher accuracy. We make a comparative analysis among the proposed algorithms and conclude that adaptive classifiers exhibit a higher start by a good approximation and perform better than the classifier trained from start-over.

1. Introduction

Biometrics has become ubiquitous and spurred common use in many authentication mechanisms. Despite its reliable identification and secure authentication, since different impressions of a user's acquired biometrics are not exactly the same, the verification system outputs a score that measures the degree of similarity between an existing profile and the samples acquired at the time of acquisition. For *behavioral biometrics* such as keystroke dynamics, gait and handwriting recognition, the verification is even more challenging [3, 6]. A user might be in different acquisition conditions and state of mind than at the time of enrollment. This may cause perturbations of behaviors and yet, reduce the similarity score.

Currently, a user is authenticated by being evaluated with a classifier that is trained on a single dataset even after a long time period has passed. However, the data is sensitive to several factors such as emotion, time of the day, keyboard layout, etc., for which we refer to as *environmental conditions* throughout the paper. Alternatively, the user may enroll in another training session at a different time period to generalize a system's recognition. Then, the inputs that have temporal variations are processed by a new classifier to acquire the changes incurred by the new environment. However, both options have their own limitations. While the former one deteriorates the accuracy of a system by time, the latter one requires a completely new classifier trained from scratch without taking advantage of a previously acquired profile.

In this paper, we propose a new adaptive classification mechanism called *transfer learning* to transfer acquired knowledge to a different domain in the context of keystroke dynamics. Our objective is to authenticate an enrolled user under different environmental conditions with the least amount of re-training. If the user moves to different conditions and types at a different time, the knowledge acquired from previous sessions is transferred via parameters that contain classifier information. Hence, the system can learn the updated profile faster by integrating the parameters, and recognize the latest typing pattern efficiently.

The contributions of this paper include the application of transfer learning on keystroke dynamics for the first time in the literature. We believe that transfer learning is a suitable tool to improve the recognition of behavioral biometrics. We employ 3 different adaptive SVM techniques described in [2] to enable knowledge transfer from one setting to another. The source profile is introduced as the regularizer of the target profile in the SVM cost function so that the classifier can learn from substantially less number of samples at a different time. Also, we make a comparative analysis among the adaptive schemes and the classifier trained from start.

The paper is organized as follows. The background information about transfer learning is provided in Section 2 followed by related work on the use of transfer learning on different domains and adaptive methods in keystroke dy-

namics in Section 3. Then, the details on data collection and feature extraction are given, and the experiments conducted are described in Section 4. Section 5 presents the results of using adaptive SVM techniques and their effectiveness in transferring knowledge to a new environment. Finally, Section 6 summarizes our findings and gives an insight into how the methods could be improved further.

2. Background

Ideally, the keystroke data collected at a session is expected to be an invariant representation of an individual's behavioral biometrics. In real applications, however, the data is sensitive to several factors such as emotion, time of the day and keyboard layout [18]. Therefore, an efficient adaptation mechanism is required to utilize an existing template and integrate new samples that may have a slightly different distribution.

2.1. Transfer Learning

Transfer learning allows an existing classifier to adapt to environmental conditions that may cause perturbations on the distribution of the original dataset. The existing classifier or the template that has been generated for a particular user is called the *source task*. The updated classifier that is adapted to the new environmental conditions is referred to *target task* in the transfer learning literature. Basically, the purpose in transfer learning is to create a target task by utilizing the knowledge that is acquired during the source task generation. In other words, transfer learning benefits from knowledge acquired from one or more tasks to recognize a related task in a faster and more efficient way. This way, prior knowledge is leveraged to generate a better and more representative model.

Another advantage of transfer learning is that the availability of the source data is not required. The parameters that shape up a classifier are sufficient to supervise the target task in the generation of an adapted profile. This way, the target task does not process the source data directly, and runs very fast to create the new profile. Furthermore, if a user's typing profile is to be transferred to another terminal, it is very efficient to send only a set of parameters instead of the entire dataset, especially when we consider large datasets.

3. Related Work

The concept of classifier adaptation can be considered as one of the important issues in machine learning. This problem is sometimes referred to concept drift in data mining [17] and sometimes as incremental learning [14] or cross-domain learning [15] in the literature.

Concept drift is different in the sense that the adapted profile is generated by using both the source and target datasets [18]. Whereas, transfer learning can directly manipulate the source task parameters without processing previous information. This way, the adaptation is efficient and more applicable where old data is inaccessible. *Incremental learning* is another related technique that shares a common background with transfer learning except it focuses on scalability problems in which the data is processed in part to consume less power [14]. As for *cross-domain learning*, it has been mostly used inter-changeably with transfer learning in the literature. It indicates a large variance among the domains that the knowledge transfer happens.

In biometric systems, the data acquired from an individual is susceptible to perturbations due to environmental conditions and sensor-based variations [16]. Biometric measurements tend to have a large *intra-class variability*, hence, it is very likely for an existing profile to be relatively different from a person's future profile. Solutions to intra-class variability problem include a template selection and periodic template update processes [10]. The main purpose of these operations is to involve the most representative features using various methods. This way, misclassification errors can be reduced by keeping the most similar[11], the most recently or the most frequently used records [12].

Specifically in keystroke dynamics, Giot et al. [4] propose a semi-supervised approach by involving the highly genuine samples in the update process. Similarly, Monrose et al. [9] use distinguishing features in keystroke dynamics updating with recent consistent inputs, thus, the reference profile acquires the modifications in a user's typing pattern. In [1, 7], the newest samples are appended to or replaced with the existing ones. In [5], the authors suggest a leave-one-out cross validation method using samples of the same pass-phrase typed by a particular user, and determine a threshold accordingly. Profile generation and threshold calculation are repeated each time a user is tested. This way, the threshold value becomes adaptive and can reflect the changes of the user over time.

However, the template update process discards existing templates and replaces them with the most current ones or consolidate two templates to reflect the most updated profile. In both cases, the recognition system runs from the beginning and may need previous set of features or raw data. Re-training of a system is very costly and computationally intensive, thus, a more efficient template update mechanism is required. In the next section, we present an adaptive mechanism that benefits from the existing profile (not the dataset!) and updates the parameters with only a few samples from the new environment.

4. Methodology

4.1. Dataset

In this paper, we consider the problem of recognizing individuals using their previously learned profiles. For this

goal, we use the public keystroke dataset published online by Killourhy and Maxion from CMU labs [8]. The data is collected in a controlled environment with 51 users at 8 different sessions. The sessions have at least one day interval in between, to reflect the variation incurred by time. All users are required to type the same password ".tie5Roanl", 50 times at each session for consistency. The key events and the corresponding time stamps are recorded, and feature vector (V) is created by the timing information of the password. It consists of 31 features for each password including the hold time (H_*: the time that each key in the password is kept pressed) and the flight times (DD_*, UD_*: elapsed time between press down and press up events of successive keys). A part of the feature vector is as follows:

$$V = \{..., H_5, DD_{5R}, UD_{5R}, H_R, DD_{Ro}...\}, \quad (1)$$

where it shows the key events related to '5' and 'R' keys in the password. Since all users type exactly the same input, the values in the feature vector are aligned based on the position in the password, and does not require post-processing. We believe that this dataset is suitable to demonstrate the applicability of transfer learning on keystroke dynamics since it has a time dimension that may incur variations among the sessions. When the variation is observed, the knowledge transfer can be realized by enabling a fast convergence and robust adaptation in the target task.

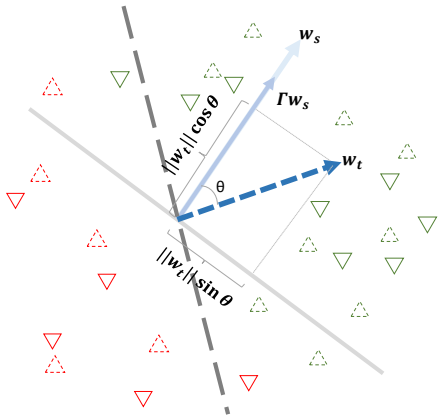

Figure 1: SVM Projection [2]

4.2. Proposed Techniques

It is assumed that the hyperplane that separates the data collected at another session, if not the same, is geometrically related to the hyperplane of the source dataset that is drawn by its constraints. Adaptive SVMs take advantage of this fact and adapt existing SVM to the new environment based on some projection techniques. Fig. 1 shows the projection of the target model parameter w_t onto the separating hyperplane of the source model.

- **Adaptive SVM (A-SVM) [18]:** Regularization of the distance between the model parameter of the source (w_s) and the target (w_t) is the basic part of this technique. A-SVM tries to shift and rotate the separating hyperplane by updating model parameter with the help of some samples from the target dataset. In this method, a new parameter Γ emerges to control the amount of transfer based on how the target samples are similar to the source profile. Accordingly, the objective function turns into Eq. 2:

$$\min \left(\frac{1}{2} \left\| w_t - \Gamma w_s \right\|^2 + C \sum_{i=1}^{l} \xi_i \right) \quad (2)$$

  where Γ corresponds to the amount of regularization. A-SVM aims at minimizing the distance between w_t and Γw_s, hence, keeping the angle θ at minimum ($||w_t|| cos\theta$ at maximum) while inducing transfer by fitting the target samples. Aytar and Zisserman provide a detailed analysis of A-SVM by describing the parameter Γ as a spring between w_t and Γw_s, and how it manages the tradeoff between margin maximization and knowledge transfer in [2].

- **Deformable Adaptive SVM (DA-SVM) [2]:** Knowledge transfer by adapting an hyperplane can also be performed by implementing local deformations on w_s to acquire the target samples and fit with a high accuracy. In this method, not only the hyperplane is adapted but also each element of the source model vector, w_s is transformed by a function of f into another element in target model vector, w_t. This way, small deformations in the adapted SVM can carry on a more flexible regularization. The objective function becomes:

$$\min \left(\frac{1}{2} \left\| w_t - \Gamma f(w_s) \right\|^2 + \lambda \Delta + C \sum_{i=1}^{l} \xi_i \right) \quad (3)$$

  where Δ refers to the amount of deformation, namely, the overall distance between the source and target elements of the model parameters. The additional parameter λ corresponds to the weight of deformations such that high values yield a similar solution to that of Eq. 2 while small values allow more deformations with less regularization.

- **Projective Model Transfer SVM (PMT-SVM) [2]:** An alternative way of reducing θ can be achieved by minimizing the projection of w_t onto the plane orthogonal to w_s. That is, $||w_t|| sin\theta$ in Fig. 1 is minimized with a regularization factor Γ while providing margin maximization. The objection function, in this case, is:

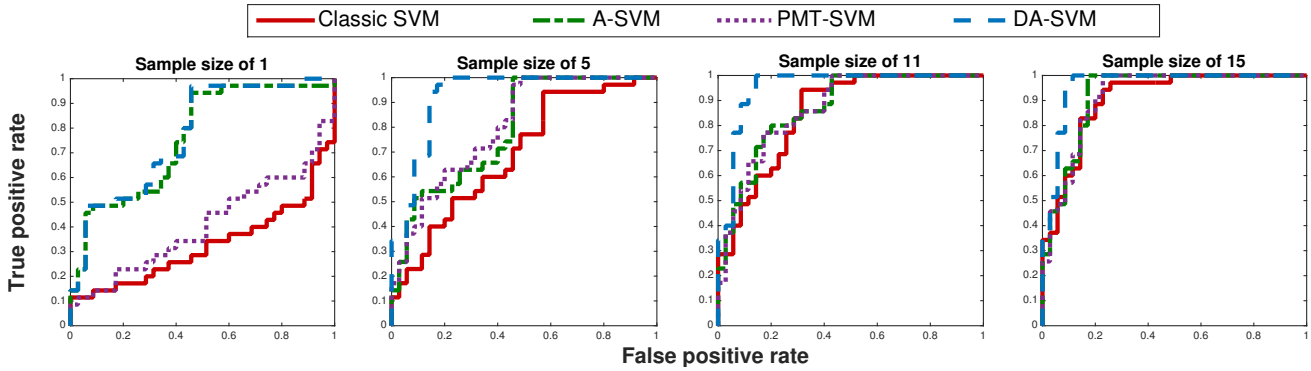

Figure 2: ROC with various step sizes

Sample Size:	1	5	11	15
Classic SVM	**71.28** $\pm$ 11.39	**80.01** $\pm$ 9.01	**93.71** $\pm$ 4.55	**96.20** $\pm$ 3.42
A-SVM	**80.11** $\pm$ 2.61	**85.54** $\pm$ 3.49	**93.33** $\pm$ 2.33	**94.86** $\pm$ 1.84
PMT-SVM	**82.64** $\pm$ 9.00	**87.12** $\pm$ 6.95	**95.86** $\pm$ 2.74	**95.76** $\pm$ 2.75
DA-SVM	**84.39** $\pm$ 4.03	**92.53** $\pm$ 3.42	**97.18** $\pm$ 1.10	**97.37** $\pm$ 1.01

AUC values are multiplied with 100 for higher precision

Table 1: Area Under Curve (AUC) by sample size

$$\min \left(\frac{1}{2} \|w_t\|^2 + \Gamma \|Pw_t\|^2 + C \sum_{i=1}^{l} \xi_i \right) \quad (4)$$

where $P = I - \frac{w_s w_s^T}{w_s^T w_s}$ corresponds to projection matrix.

4.3. Training

We pick two users from the dataset described in Section 4.1, and label one of the users as positive and the other as negative class to train the source (SVM_s) and target (SVM_t) classifiers. Out of 8 sessions, we randomly pick a source session for the users and train SVM_s using source dataset, D_s. Then, we pick a target session to be used for the target task, and partition the dataset, D_t into transferring (D_t^l: labeled) and testing (D_t^u: unlabeled) dataset. Our goal is to generate the target classifier, SVM_t using D_t^l to help classify D_t^u building upon SVM_s.

When SVM_s is trained with a complete session of the users, the model parameter, w_s is created. SVM_t builds w_t upon w_s by using D_t^l to learn the typing pattern faster. SVM classifiers are trained with a linear kernel and the cost variable is set $C = 0.002$ for all experiments similar to the experiments conducted in [2]. Other parameters including the regularization factor in the equations are set different based on the results of the experiments. The value of the parameters that we use in our experiments are as follows: $\Gamma = 0.01$ in Eq. 2; $\Gamma = 0.1$, $\lambda = 0.0001$ and $\Delta = 4$ in Eq. 3; $\Gamma = 5$ in Eq. 4.

5. Results

Our results are based on the the dataset that is publicly available from CMU labs [8]. The experiments are conducted by using the adaptive algorithms described earlier and an SVM classifier that is trained from start without any knowledge transfer (It is denoted by 'Classic SVM' in the tables and figures). We test the accuracy of the system step by step by increasing the sample size from the transferring dataset. *Sample* in the figures denotes to a user's attempt of typing the password at a time. At each step, the receiver operating characteristic (ROC) curve is plotted with respect to true positive rate (TPR) and false positive rate (FPR). ROC is a heavily used statistics in machine learning to illustrate the performance of a classifier as its discrimination threshold changes [13]. Also, the area under curve (AUC) is calculated by testing against the unlabeled testing dataset.

Fig. 2 plots the ROC curve of 4 different settings, each showing the results of different sample sizes for adaptive algorithms and the classic SVM classifier that is trained from start. The sample size of 1 (single-shot learning) clearly exhibits the advantage of using transfer learning over classic SVM since TPRs for adaptive algorithms are higher for almost all FPRs. As we increase the sample size, the ROC curves approach to the corner for all algorithms as expected. Nevertheless, the adaptive algorithms perform better and learn faster for almost all sample sizes since they start the learning process by initializing the target model parameter, w_t with a good approximation of w_s.

Accordingly, we calculate the AUC for scalar compar-

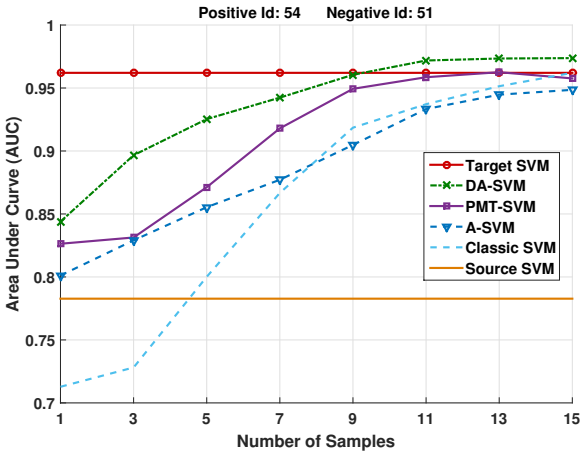

Figure 3: Comparison of SVM Algorithms

ison of the algorithms. We run the experiments 50 times to avoid bias and get consistent results, each time shuffling the target dataset. We take the mean and standard deviation for various sample sizes. Table 1 shows the corresponding AUC values and the standard deviation of the algorithms multiplied by 100 for higher precision. We can see that the adaptive SVMs outperform the classic SVM model, especially for a small number of samples. Also, it is important to note that the standard deviations for adaptive SVMs are less than the classic SVM, which indicates the robustness of the adaptive algorithms regardless of random good or bad sample selection during training.

To make a better comparison analysis, we plot the AUC results of the SVM algorithms including the source and the target SVMs. The source SVM is trained with the first (source) session, while the target SVM is trained with the transferring dataset (15 samples in this example) of the target task. For all SVMs plotted in Fig. 3, the AUC results are calculated by testing against the testing dataset (35 samples) that is left after partitioning. It is expected that the target SVM performs better than the source SVM since a classifier yields better results when testing against a dataset of the same session. Note that both source and target SVM are trained and tested once during this experiment to give a better sense of comparison, and this is why the trendlines are constant through the changing sample size. The AUC line of the target SVM can be seen as the upper limit of classic SVM since both of them are supposed to have the same configurations at the level of 15 samples.

Fig. 3 exhibits higher start and fast convergence of adaptive SVMs. In particular, DA-SVM has a relatively better learning rate among all others. It reaches the upper limit of classic SVM and even increases further by incorporating more samples from the target task. We believe that the use of a flexible regularization inside the square term has enabled us to exert a direct effect on the convergence speed of the algorithm. Also, by allowing local deformations on the source model parameter, w_s in Eq. 3, DA-SVM algorithm can switch to w_t in a more robust way. This fast adaptation of DA-SVM has also been proven by Aytar et al. [2] in an image classification experiment.

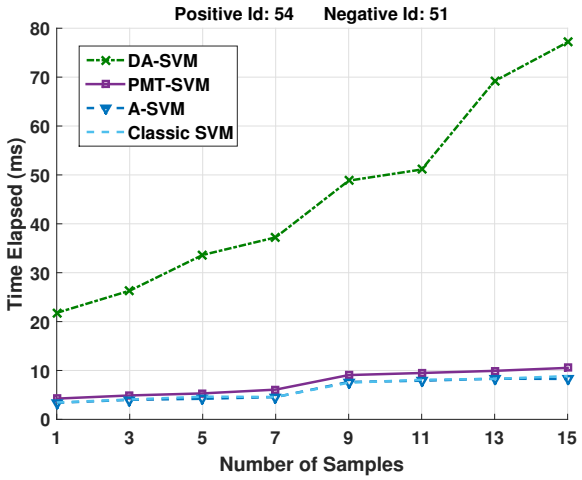

Figure 4: Performance of Adaptive-SVM Algorithms

Nevertheless, the boosted performance of DA-SVM algorithm requires intensive computation. To show the trade-off between the convergence rate and computational cost, we draw Fig. 4 that displays the elapsed time during the training process of each SVM by using Matlab's built-in *timeit* function. It runs the training method multiple times, and reports the median of the measurements. As expected, the flexibility provided by DA-SVM requires relatively more computation compared to other SVMs.

6. Conclusion

Keystroke dynamics is a behavioral biometrics that can be used as a means of authentication. Due to the nature of behavioral biometrics, the characteristics of an individual may change gradually based on environmental conditions. A user's profile can show a degradation of performance over time. Therefore, a classifier with an existing profile may perform poorly on another dataset that is acquired under psychologically, temporally or spatially different conditions. In this paper, we propose using 3 different adaptive techniques: *Adaptive SVM, Deformable Adaptive SVM and Projective Model Transfer SVM*. They are used to update a classifier and transfer the knowledge acquired from a learned profile to an adapted target profile. The source model parameter is introduced as the regularizer in target profile generation so that the classifier can learn from substantially less number of samples at a different time. By utilizing a publicly available keystroke dataset, we are able to demonstrate the effect of transfer learning in verifying users

at a different session. Our results show that, adapted SVMs exhibit a higher start by a good approximation and perform up to 13% better than the classifier trained from scratch. In addition, we make a comparative analysis between the adaptive SVMs with respect to accuracy results and computational costs. Although the size of the dataset in a session is not large enough to run extensive experiments, we believe this study can be considered as a proof-of-concept about the applicability of transfer learning on keystroke dynamics or behavioral biometrics, in general. We plan to move this research further and collect large scale data to validate the results and conduct thorough experiments in the future.

Acknowledgments

This research is supported in part by National Science Foundation Grant No. CNS: 1314803. Usual disclaimers apply.

References

[1] L. C. F. Araújo, L. H. R. Sucupira, M. G. Lizárraga, L. L. Ling, and J. B. T. Yabu-Uti. User authentication through typing biometrics features. *IEEE Transactions on Signal Processing*, 53(2 II):851–855, 2005.

[2] Y. Aytar and A. Zisserman. Tabula rasa: Model transfer for object category detection. *Proceedings of the IEEE International Conference on Computer Vision*, pages 2252–2259, 2011.

[3] H. Çeker and S. Upadhyaya. Enhanced Recognition of Keystroke Dynamics using Gaussian Mixture Models. In *Military Communications Conference, MILCOM 2015-2015 IEEE*, number 2015-02, pages 1305–1310, 2015.

[4] R. Giot, B. Dorizzi, and C. Rosenberger. Analysis of template update strategies for keystroke dynamics. *IEEE SSCI 2011 - Symposium Series on Computational Intelligence - CIBIM 2011: 2011 IEEE Workshop on Computational Intelligence in Biometrics and Identity Management*, 1:21–28, 2011.

[5] D. Hosseinzadeh and S. Krishnan. Gaussian mixture modeling of keystroke patterns for biometric applications. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 38(6):816–826, 2008.

[6] A. K. Jain, A. Ross, and S. Prabhakar. An introduction to biometric recognition. *Circuits and Systems for Video Technology, IEEE Transactions on*, 14(1):4–20, 2004.

[7] P. Kang, S.-s. Hwang, and S. Cho. Continual retraining of keystroke dynamics based authenticator. In *International Conference on Biometrics*, pages 1203–1211. Springer, 2007.

[8] K. S. Killourhy and R. A. Maxion. Comparing anomaly-detection algorithms for keystroke dynamics. In *Dependable Systems & Networks, 2009. DSN'09. IEEE/IFIP International Conference on*, pages 125–134. IEEE, 2009.

[9] F. Monrose, M. K. Reiter, and S. Wetzel. Password hardening based on keystroke dynamics. *International Journal of Information Security*, 1:69–83, 2002.

[10] A. Rattani, B. Freni, G. L. Marcialis, and F. Roli. Template Update Methods in Adaptive Biometric Systems : A Critical Review. *Advances in Biometrics. Springer Berlin Heidelberg*, pages 847–856, 2009.

[11] F. Roli, L. Didaci, and G. L. Marcialis. Adaptive biometric systems that can improve with use. *Advances in Biometrics: Sensors, Algorithms and Systems*, pages 447–471, 2008.

[12] T. Scheidat, A. Makrushin, and C. Vielhauer. Automatic Template Update Strategies for Biometrics. *Science*, (May):2–6, 2007.

[13] B. Song, G. Zhang, W. Zhu, and Z. Liang. ROC operating point selection for classification of imbalanced data with application to computer-aided polyp detection in CT colonography. *International journal of computer assisted radiology and surgery*, 9(1):79–89, jan 2014.

[14] N. Syed, S. Huan, L. Kah, and K. Sung. Incremental learning with support vector machines. *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1–6, 1999.

[15] M. E. Taylor and P. Stone. Transfer Learning for Reinforcement Learning Domains : A Survey. *Journal of Machine Learning Research*, 10:1633–1685, 2009.

[16] U. Uludag, A. Ross, and A. Jain. Biometric template selection and update: A case study in fingerprints. *Pattern Recognition*, 37(7):1533–1542, 2004.

[17] H. Wang, W. Fan, P. Yu, and J. Han. Mining concept-drifting data streams using ensemble classifiers. *Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2(1):226—-235, 2003.

[18] J. Yang, R. Yan, and A. G. Hauptmann. Adapting SVM Classifiers to Data with Shifted Distributions. *Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007)*, pages 69–76, 2007.