What do you do when you know that you don't know?

Abhijit Bendale*, Terrance E. Boult
University of Colorado at Colorado Springs
{abendale,tboult}@vast.uccs.edu *

Abstract

Real-world biometrics recognition problems often have two unknowns: the person be recognized, as well as a hidden unknown - missing data. If we choose to ignore data that is occasionally missing, we sacrifice accuracy. In this paper, we present a novel technique to address the problem of handling missing data in biometrics systems without having to make implicit assumptions on the distribution of the underlying data. We introduce the concept of "operational adaptation" for biometric systems and formalize the problem. We present a solution for handling missing data based on refactoring on Support Vector Machines for large scale face recognition tasks. We also develop a general approach to estimating SVM refactoring risk. We present experiments on large-scale face recognition based on describable visual attributes on LFW dataset. Our approach consistently outperforms state-of-the-art methods designed to handle missing data.

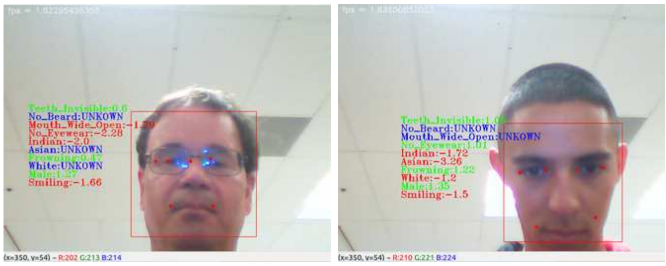

Figure 1. A system for describable visual attributes for faces based on [30] and extended for open-set recognition is shown above. In the image, green text is a positive attribute, red text is negative attributes and blue color signifies unknown/missing attribute. In the above images, the left image shows how bad lighting/feature-detection led to "UNKNOWN" labels for *Asian, While* attributes. The example on the right shows occlusion leading to the *no beard* attribute being labeled "UNKNOWN". Handling such unknowns at run-time, in a learning-based system, poses considerable operational challenges. This paper is about what the system should do when it knows that it does not know about some features.

1 Introduction

Biometrics systems have been widely adopted in various walks of life, thanks to significant progress in various subfields in the past decade. Cheap sensors, models learned with large amounts of data, an abundance of processing power have all led to development and deployment of biometrics systems beyond the narrow scope of research labs [15]. Biometric recognition in unconstrained settings imposes little restrictions on the data acquisition and processing procedure. Biometric recognition in the open world leads to multiple challenges. Failing assumptions, failing code, or missing inputs can then lead to missing data in higher-level feature descriptions. Matching models (especially learned-models) with missing data is challenging. How to build recognition system that operates in the face of these "unknowns" is the fundamental problem that we address in this paper.

In recent years, describable visual attributes have emerged as a powerful low-level feature representation for a wide range of face recognition applications [26, 18]. Kumar *et al*. [18] demonstrated a system to automatically train several attribute classifiers for faces, such as "brown hair", "mustache," "blonde", "pointy nose", "thin eyebrows", "wearing lipstick" etc. Attribute classifiers take an image as input and return a real-valued score representing the presence of the given attribute in the image. These real-valued scores can then be used in a full-fledged face recognition system for identification/retrieval [26]. While the system designed by Kumar *et al*. was primarily for a closed set face verification task, more recently, Wilber *et al*. [30] have proposed open set extensions for such systems. As noted by Scheirer *et al*. [28] "when a recognition system is trained and is operational, there are finite set of known objects in scenes with myriad unknown objects, combinations and configurations - labeling something new, novel or unknown should always be a valid outcome". In open-set systems, a specific face attribute is named unknown if the system is either unable to classify with sufficient confi-

*Research performed at Univ. of Colorado at Colorado Springs funded in part by NSF IIS-1320956. Abhijit Bendale is currently with Samsung Research America, Mountain View, CA

dence or is presented with an image/feature that it has not seen before. Such open set "unknown" labeling thus leads to known missing status for the respective attribute (see Fig 1). Systems designed to handle open set recognition have demonstrated excellent performance on many biometrics and computer vision systems in wild [30, 2, 6, 31].

This paper introduces and addresses a novel and practical problem, **Operational Adaptation**, where given only a previously trained operational system and a test instance $\boldsymbol{x}_t$ with some described difference from the normal instances, the system must adapt to the constraints of data to make predictions and to provide estimates of the risk of adaptation. While there is a growing body of work in the areas of domain adaptation and transfer learning that work towards adapting classifiers during the training phase, such approaches are not practical for a machine that may take days or weeks to train. In this work, we focus our attention on the more common and prevalent missing data problem, what [12, 19] calls the "nightmare at test time," where at test time the operational data is corrupted or missing. This is a nightmare because it cannot be avoided. We contend there are two important subproblems within the nightmare. The first is the obvious one, making decisions using partial data. The second, and generally overlookedproblem, is estimating how scared we should be using that partial data. Intuitively, many users would assume that losing 70% of features yields a nearly useless classifier, while losing only one feature is probably not bad. However, even one missing feature can lead to horrible performance if that is a critical feature, while the 70% missing may have little impact.

There are multiple contributions of this work. We formally define the problem of operational adaptation and present a novel solution for handling missing data with SVMs based on SVM re-factoring with bias re-optimization. Our solution offers superior results to many state-of-the-art approaches both in terms of accuracy and storage space. Further, we develop a general approach to estimating SVM Refactoring Risk. Our risk estimation process provides the associated risk when performing predictions with missing data. We show the proposed adaptation risk estimation is a better predictor of success/failure [25] than percent missing data. We use describable visual attribute representation on large scale face verification tasks for our experiments. The proposed approach consistently performs Labelled Faces in the Wild [13] and other machine learning datasets such as USPS [14] and MNIST [20]. Our new method is the first step toward addressing an important problem for operational use of machine learning for large scale biometrics recognition systems.

2 Related Work

Handling missing data in biometrics is an important problem and has been addressed by multiple researchers in the past. Ding *et al.* [9] performed a detailed study comparing multiple imputation methods for score fusion in biometrics. Poh *et al.* [23] proposed a framework for addressing kernel based multi-modal biometric fusion using neutral point substitution. Other notable works in the domain of handling missing data for biometric score fusion are by Fatukasi *et al.* [10] and Damer *et al.* [8]. Our work differs from these works in multiple aspects. Ding *et al.* showed promising results for score fusion with generative models with relatively lower feature dimensions. In our work, we focus mainly on large-scale discriminative models such as SVMs. Further, the problem of interest of this work is run-time [19] adaptation of learned models for verification/recognition systems, unlike the works of Fatukasi *et al.*, Poh *et al.* and Damer *et al.* where the focus is primarily on fusion rules for biometric score fusion.

Researchers in machine learning and statistics communities [21, 5, 24] have also addressed the problem of learning from missing data. Chechik *et al.* [5] proposed a max-margin learning framework that is based on geometric interpretation of the margin and aims to maximize the margin of each sample in its own relevant subspace. The work of [24] presents a comprehensive evaluation framework comparing imputation based methods and reduced feature models. Reduced feature models are constructed for each type of missing pattern separately making the problem computationally extremely expensive. Such methods are unsuitable for biometrics where feature dimensionality tends to be very high as it requires storing reduced model for every permutation of missing feature space[1].

3 Operational Adaptation

Given a training set $\{\mathcal{D}_S = x_i, y_i \in \mathcal{X} \text{ x } \mathcal{Y} : \{+1, -1\}\}$, where $\mathcal{X}$ is the input space and $\mathcal{Y}$ is a finite training set. The learning problem is to find a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ that obtains high predictive accuracy. In this work, we focus our attention on run-time adaptation of a pre-trained classifier to new operational domains, in the presence of limited data and model parameters. We assume existence of a Source Domain $\mathcal{D}_S$ and a family of Operational Domains $\mathcal{D}_{o_j} \neq \mathcal{D}_S$. We assume the training set $\mathcal{D}_S \in \mathbb{R}^n$ and each test sample belongs to an operational domain $\mathcal{D}_{o_j}$, where $\mathcal{D}_{o_j} \in \mathbb{R}^{k_j}$ where, in general, $k_j \leq n$. Let $\boldsymbol{M}_j$ be an operator such that $\boldsymbol{M}_j : \mathcal{D}_S \rightarrow \mathcal{D}_{o_j}$, i.e. it map items in the source domain to the operational domain. While the definition of operational adaptation can be more general, in this work, we focus on problems where the operational domain $\mathcal{D}_o$ has missing features compared to source domain $\mathcal{D}_S$, in which case, $\boldsymbol{M}$ projects away dimensions associated with missing features. For operational adaptation, we enforce

[1] e.g. for LBP like features where feature dimensionality is around 200 features, total number of reduced models that need to be stored would be 2^200

that, during operational time, we have access only to the operational data O_D, which includes the learned prediction function f and its associated parameters $(\theta_1..\theta_f)$. In terms of SVMs, one could view f, b, type of kernel, α and the support vectors as the operational data O_D.

Definition 1 Operational Adaptation: *Given a learned prediction function $f(\theta_1, ..\theta_f)$ over some source (training) domain $\mathcal{D}_S$ defined by operational data O_D, an operational domain $\mathcal{D}_o$, a transformation operator $\boldsymbol{M}$ relating $\mathcal{D}_S$ to $\mathcal{D}_o$, and test point $x \in \mathcal{D}_o$, the problem of Operational Adaptation is:*

1. *to obtain adapted prediction function $f_o()$ and an effective prediction function over the operational domain $\mathcal{D}_o$*
2. *obtain an associated operational adaptation risk measure $\mathcal{R}_o : (f_o(), x) \to [0, 1]$, which estimates the likelihood of failure of the prediction function f_o.*

In this paper, we focus on the difference between source domain $\mathcal{D}_S$ and operational domain $\mathcal{D}_o$ as difference in dimensionality, in particular in the remainder, we presume that $\boldsymbol{M}$ is linear projection. However, the idea of operational adaptation applies to any problem which satisfies the constraints mentioned in definition 1, e.g. the general definition includes operational domains that involve linear basis transformations or even non-linear remapping. In this particular definition, while we presume that $\boldsymbol{M}$ is given, a more general form may involve estimating $\boldsymbol{M}$.

3.1 SVM Refactoring and Run Time Bias Estimation

The primary intuition behind SVM classification is to map the data into a high dimensional space and find a max-margin separating hyperplane for efficient classification. In this section, we present a methodology to modify support vector machines and introduce the idea of operationally adapted instance specific bias. The estimation for bias in operational domain is based on modifying the independent variables in the dual of objective function of SVM to minimize the classification error over support vectors. Our SVM refactoring method is computationally efficient compared to reduced model methods, and more accurate than zero or mean imputations. The proposed method for bias estimation at prediction time is termed as Run Time Bias Estimation (RTBE). The intuition behind this method is explained in detail in Fig 2

We assume that we are given a set of training vectors $x_i \in \mathbb{R}^n, i = 1, ..m$ in two classes, and a vector of labels $\mathbf{y}$ such that $y_i \in \{1, -1\}$. The learning of such a classifier reduces to the constrained optimization problem as follows [3]:

$$\min_{\boldsymbol{w},b,\xi} \mathcal{P}(\boldsymbol{w}, b, \xi) = \frac{1}{2}\boldsymbol{w}^2 + C\sum_{i=1}^{m}\xi_i \tag{1}$$

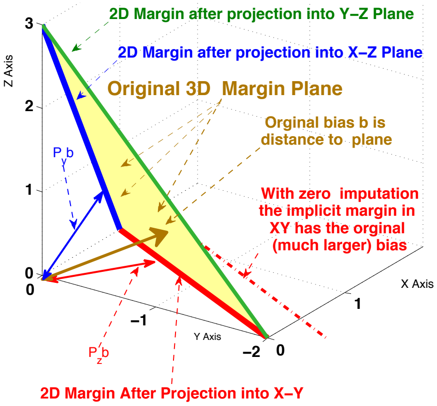
2D Margin after projection into Y–Z Plane
2D Margin after projection into X–Z Plane
Original 3D Margin Plane
Orginal bias b is distance to plane
With zero imputation the implicit margin in XY has the orginal (much larger) bias
2D Margin After Projection into X–Y
P_y b
P_z b
Z Axis
Y Axis
X Axis
3
2
1
0
0
–1
–2
0
1

Figure 2. With Bias re-factoring, the data and the bias vector are both projected and distances are computed in the lower-dimensional subspace. For this example, presume 3D features with the original margin plane in 3D with bias b. Classic imputation by zero, if Z is missing, computes distances in the X-Y plane but keeping the original bias sets a much higher threshold as shown in the dashed red line. When features are missing, the Projection of the Bias decomposed vector is like using a lower-dimensional margin for decision making. If the Z feature is missing, P_z project the data and the bias vector the X-Y plane, effectively using the Red margin, but if y is missing, P_y projects into the X-Z plane effectively using the blue margin. The approach of this paper, run time bias estimation (RTBE), adjust the bias from the margin to the original plane to appropriate the projection subspace.

$$\text{subject to} \left\{ \; y_i(\boldsymbol{w}^\intercal\boldsymbol{\phi}(x_i) + b) \geq 1 - \xi_i, \forall i \; \xi_i \geq 0, \forall i \right. \tag{2}$$

where training data is mapped to a higher dimensional space by the kernel function $\boldsymbol{\phi}(.)$, and C is a penalty parameter on the training error (trade-off between accuracy and model complexity), ξ_i are the slack variables used when training instances are not linearly separable and b is the classifier bias [22]. In the formulation of SVM in equation 1, the term w defines the orientation of hyperplane with respect to origin of the feature space $(\mathbb{R}^n)$ and the bias term b defines the distance of the hyperplane from the origin[3].

The dual of the problem in equation 1 is given as

$$\max \mathcal{F}(\alpha) = \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} y_i\alpha_i y_j\alpha_j\phi(x_i, x_j) \tag{3}$$

$$\text{subject to} \left\{ \; \forall i, 0 \leq \alpha_i \leq C \; \sum y_i\alpha_i = 0 \right. \tag{4}$$

where $K(x_i, x_j) = \phi(x_i)^\intercal\phi(x_j)$ is matrix of kernel values. Using positive Lagrange coefficients $\alpha + i \geq 0$, the Lagrangian of the dual problem is given as

$$\begin{aligned}\mathcal{L}(\boldsymbol{w}, b, \xi, \alpha) \quad &= \quad \frac{1}{2}\boldsymbol{w}^2 + C\sum_{i=1}^{m}\xi_i - \\ &\sum_{i=1}^{m}\alpha_i(y_i(\boldsymbol{w}^\intercal\phi(x_i) + b) - 1 + \xi_i)\end{aligned} \tag{5}$$

which leads to the formal dual objective function $\mathcal{F}(\boldsymbol{\alpha})$ as:

$$\mathcal{F}(\boldsymbol{\alpha}) = \min_{\boldsymbol{w},b,\xi} \mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}) \text{ subject to } \forall i, \xi_i \geq 0 \quad (6)$$

The optimization of the dual objective function directly produces $\boldsymbol{\alpha}*$, yielding w.

Let $\boldsymbol{\alpha}^* = (\alpha_1^* .. \alpha_m^*)$ be a solution of the dual problem of equation 6 where $\boldsymbol{\alpha}^*$ satisfies the dual constraints. The vector $\boldsymbol{\alpha}^*$ is generally sparse, with many zero elements. Let v be the number of non-zero elements α^*, let $\boldsymbol{s} = [s_i], i = 1 \dots v$, the support vectors, be a remaining of the training points x_j associated with the non-zero elements of α^*. The operational data O_D for the SVM is thus $\{\boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{s}, \boldsymbol{y}, b, K\}$ and $\boldsymbol{M}$. Given these, the optimal value of b can be obtained via 1-dimensional optimization over the decision function (i.e. the primal problem) using the training data [3].

Let us now derive our approach to SVM refactoring, an operational adaptation approach that provides both improved classification as well as risk estimation. Our challenge is to define a solution in the reduced dimensional space using only operational data. A plausible solution for operational adaptation would be to treat the support vectors as a training set, project them into the operational domain and train a new optimal SVM solution. While plausible, our results show that this approach does not provide acceptable rate of classification on test data, e.g. in Figure 3, it is evaluated on the USPS dataset where it is only slightly better than zero imputation. On some other examples, it did much worse than zero imputation.

Thus, we seek an approach that will reuse more of learned structure than just the knowledge of the support vectors, in particular, to adapt the optimal weights. If we revisit the dual of objective function in equation 6, we note that re-optimizing values of α, ξ or $\boldsymbol{w}$ for the operational domain would require projection of all the training data which would violate the definition of operational adaptation. Thus, the only reasonable perturbation/optimization that can be performed is re-optimization of the bias term b.

Letting $\ell(x, y)$ be the loss function for estimate x given label y, and let L_n be the empirical loss, over full support vectors $\boldsymbol{s}$ using original SVM f in $\mathbb{R}^n$. Then we first define our refactored projected error as:

$$\varepsilon(\hat{f}(\cdot; b)) = \frac{1}{v} \sum_{j=1}^{v} \ell(\sum_{i=1}^{v} \alpha_i y_i \phi(\boldsymbol{M} s_i, \boldsymbol{M} s_j) + b), y_j) \quad (7)$$

And using this we define our refactor risk as:

$$R_r(\hat{f}(\cdot; b), \boldsymbol{s}) = 1 - \frac{\min\left(1 - L_n,\ 1 - \varepsilon(\hat{f}(\cdot; b))\right)}{1 - L_n} \quad (8)$$

where we normalize by $1 - L_n$, so the refactor risk is relative risk compared to the original loss. We include the $\min()$ because noise or irrelevant variables may result in the projected support vectors having lower empirical risk than the original dimensional version. In the paper, we generally exclusively used 0-1 error, thus $1 - L_n$ is average accuracy. The changes to use absolute, square, or other loss functions should be minimal because we are only measuring the loss on the projected support vectors.

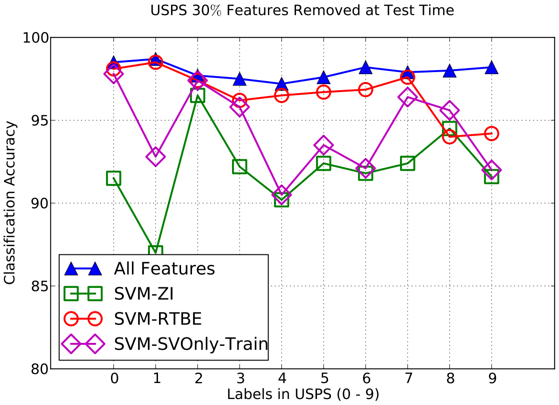

Figure 3. The above figure contains test classification accuracy for each digit in USPS dataset when 30% features were deleted from test samples. The results were obtained by training SVM with RBF kernel. The methods shown correspond to nature of test set. a) All Features Present : All the features were present during test time. b) Zero Imputation: 30% features were removed at test time and missing features were imputed with zero. c) Training with support vectors only: 30% features were removed from test samples. Corresponding features were removed from Support Vectors and a new model with these support vectors was trained. Results shown are classification results obtained with this new trained model d) RTBE: 30% features were removed at test time. SVM bias was re-optimized using our approach and classification results were obtained using optimized bias for operational domain $\mathcal{D}_o$.

3.2 Adaptation Risk Estimation

Since the classification accuracy depends on dimensions of the missing features and the technique used to reclassify data in the reduced dimensions, we refer to our re-factored risk model as the **Adaptation Risk Estimator** (eqn 8). It applies to *any* reduced dimensional model $\hat{f}$ for dealing with the partial data, not just re-factored SVM with factored bias. In particular, if the projection matrix $\boldsymbol{M}_0$ (transformation operator $\boldsymbol{M}$ between $\mathcal{D}_S$ and $\mathcal{D}_o$) is N x N and fills in "missing" data with zero, then the model is zero imputation and we can estimate the risk of using zero-imputation. It equally applies to mean-imputation.

Intuitively, our risk model is conservative as it uses difficult examples to estimate risk of failure. The actual performance can be much better if the example is far from the boundary, suggesting better risk estimators could be developed. In case of classifiers like decision trees or random forests [4], where only thresholds of tree splits are retained, operational adaptation could be achieved by retaining an "operational validation set" at run-time. Note the set of support vectors is an extremely biased and correlated set and hence it might violate the assumptions of statistical tests for

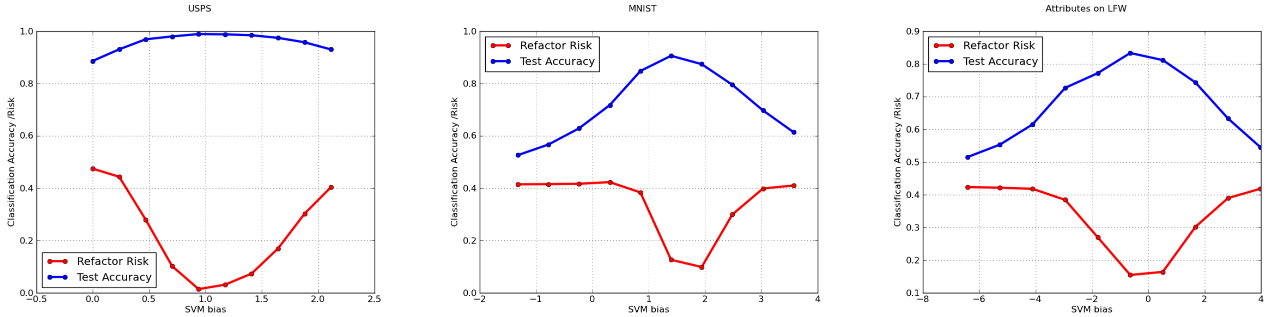

Figure 4. Effect of Varying SVM Bias b on classification accuracy on test set and associated refactor risk. It can be observed that when our refactor risk is minimum, maximum classification accuracy over test is obtained (Test Accuracy was scaled between 0 - 1 to fit this plot). Our objective is to obtain value of SVM bias b for which refactor risk is minimum. In the above experiment, 30% features were removed for each dataset during testing phase. The datasets shown are (from L-R) USPS [14], MNIST [20] and Attributes on LFW [13]

distributional shifting such as those considered in [7].

Returning to the re-optimization bias b_o for the refactored machine. Note that our refactor risk estimation applies to any machine f, and in particular we explicitly called out that it is a function of the underlying bias b. Using this, our refactored approach, we will seek a new operational bias b_o for equation 7, which is obtained from the 1-D optimization problem minimizing refactor risk:

$$b_o = \operatorname*{argmin}_{b} R_r(f_o(\cdot; b), \boldsymbol{s}) \qquad (9)$$

i.e., the re-optimization of the bias is performed based on minimization of risk over the support vectors projected into $\mathbb{R}^k$. If using 0-1 loss, as we have in the paper, the loss function is non-convex, however explicit 1-D optimization is still very efficient. We also note that, in practice, for operational adaptation with just missing variables, the computations of both the classification f_o and optimization of b_o can replace the matrix multiplication by $\boldsymbol{M}$, which is mostly zeros, with a selection operation that simply selects the relevant dimensions. With that, the cost to optimize b_o is dominated by estimating risk at the values of b associated with the v projected support vector locations. In our experiments, we found that minimizing refactor risk was in fact a good predictor of performance on test-set. To show this relationship, we plot results of varying bias for a particular operational domain and noting the associated refactor risk and test set classification accuracy 4. However, it should be noted that this experiment is done to show the relationship between refactor risk and test accuracy and during optimization process, we do not assume any knowledge about test set apart from the test-sample under consideration.

4 Experiments

In this section, we evaluate the proposed algorithm for SVM based re-factoring (i.e. Run Time Bias Estimation - RTBE) on USPS [14], MNIST [20] and Labelled Faces in the Wild [13] datasets (Fig 5). USPS and MNIST are leading handwriting recognition datasets and results proposed on those

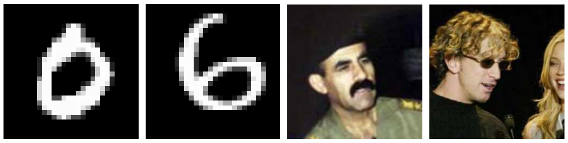

Figure 5. Left are example images from MNIST [20], and right are from and LFW [13].

can be compared against a wide variety of methods across different disciplines (e.g. biometrics, computer vision, machine learning, statistics etc.). The feature dimensionality considered for all datasets is high to show the suitability of the proposed methods on large-scale recognition tasks. USPS dataset [14] contains 9298 handwritten digits (0 - 9) (7291 for training and 2007 for testing) collected from mail envelopes in Buffalo. Each image is represented as a 256 dimensional feature vector. The MNIST database consists of 60,000 training samples and 10000 testing samples for digits between (0-9). The digits are size-normalized and centered in a fixed-size image. The size of each image is 28 x 28 leading to a feature vector of size 784. Scaled pixel values are provided for performing supervised classification task.

LFW face dataset [13] is designed for large scale face verification task and contains 13233 images of 5749 individuals. View 1 of LFW is used for building models, feature selection and finding optimal operating parameters. View 2 consists of 6000 pairs of face images on which performance is to be reported. The 6000 pairs are further divided in 10 splits to allow 10-fold cross-validation. Overall classification performance is reported on View 2 by using only the signs of the outputs and counting the number of errors in classification. We use describable visual attributes [18] on LFW dataset for face verification task. Attribute classifiers are created by using describable visual traits such as gender, race, hair color etc. These visual traits are used to construct classifiers C_K. These classifiers are then used to detect presence/absence of attribute in a given face image and a score is assigned to it. Each image in LFW dataset is thus represented as a vector $C(I_i) = \langle C_1(I_i), C_2(I_i)..C_K(I_i)\rangle$ (where K is total number of attributes/traits). To decide if the image belongs to the same person, these classifier scores are compared $\{C(I_i), C(I_j)\}$. Verification classifier for a pair of images is given as $v(I_i, I_j) = (|C_i - C_j|, (C_i.C_j), \frac{1}{2}(C_i + C_j))$. These classifier scores are used as input features for face verification task.

We systematically delete features from testing as percentages of total features present for each dataset. Each set of missing features leads to a new operational domain $\mathcal{D}_o$. We consider percentage of missing features in range of 10%, 20%, 30%, 40% and 50%. The process is kept similar for all three datasets. For each dataset, we trained SVM with linear and RBF kernel essentially training model in $\mathbb{R}^n$ (where n = { 256, 146 and 784} for USPS, Attributes and MNIST respectively). During test phase, for zero imputation method, all the missing features are substituted by zero and classification is carried out. With RTBE approach, we detect the missing features, project the Support Vectors in corresponding operational domain and obtain new optimal bias b_o by minimizing refactor risk over support vectors (operational data). As a baseline, we also obtain results on respective datasets without deleting any features. It is obvious that the performance of the system would be best when all the features are present. We observe that the proposed approach of RTBE consistently outperforms zero imputation across multiple datasets. We also note that rate of performance degradation for RTBE is lower compared to zero imputation. For USPS and MNIST dataset, the performance obtained with RTBE continues to remain stable even under extreme cases (e.g. 40% and 50% missing data).

4.1 Comparison with Other Methods for Handling Missing Data

The state of the art for handling missing data for USPS dataset is multi-class Gaussian Process [29] yielding 94.2% (error of 5.8 %) at 25% features missing (64 pixels out of 256). In the same work, authors noted Zero imputation resulted in classification accuracy of 94.15 (5.85% Error) and mean imputation yielded 93.92 (6.08% error). On the same dataset with similar train/test protocol, our method of RTBE achieves overall classification accuracy of 95.11 % (4.89 % error) using linear kernel and 98.26 % (error 1.76 %) using RBF kernel (a 69.65 % reduction of error over the state of the art). Chechik *et al.* [5] reported their results on MNIST dataset by considering classification on digits '5' and '6'. They remove a square patch randomly from the image and report a performance of 95% using their geometric margin method (similar performance is reported for their averaged norm method in the same study). Our approach on RTBE yields 96.6% on similar problem of classifying '5' and '6'. To the best of our knowledge, no study has been done on handling missing data on attributes on LFW [18]. Comparison with few other approaches obtained from [5] is shown in Table 1

Alg/Dataset	MNIST	Pima	USPS
Zero	$95 \pm .5$	66 ± 4	-
Mean	$94 \pm .7$	65 ± 4	93.92
EM [11]	$95 \pm .04$	65 ± 3	-
Avg W [5]	$95 \pm .5$	64 ± 5	-
Geom.Margin [5]	$95 \pm .5$	66 ± 5	-
Guass.Process [29]	-	-	94.2
RTBE (this paper)	$\mathbf{96.6 \pm .5}$	64 ± 4	$\mathbf{98.26}$

Table 1. Comparison with other methods [5], with best algorithm in bold if more than 1 std.dev. above others. For MNIST and PIMA the data randomly dropped 90% of the features and Geom.Margin were the state of the art algorithm. For USPS , 25% of data missing and Guassian processess were the state of the art. We include Pima as an example where all tested algorithms, including the new RTBE approach, are not statstically different.

4.2 Risk Estimation for Missing Data

To evaluate the effectiveness of the Adaptation Risk Estimation, we use the meta-recognition evaluation paradigm, MRET (Meta-Recognition Error Trade-off Curves), proposed in [27, 25], which considers how often the risk estimator correctly predicts the failure/success of the underlying classifier. We consider two risk estimation approaches: percentage of missing features (wrt to total features) and the risk estimator from Eq. 8. For risk estimation with percentage of missing features, we drop features in steps (e.g. 10%, 20% etc.) and at each step we predict success/failure. When using refactor risk $\mathcal{R}(f_o)$ as a risk estimator (from Eq. 8), the range of refactor risk is divided into steps and at each step success and failure is computed using formulae from 10. For each of the risk estimators we consider both zero-imputation and the SVM refactoring via bias factoring. One can threshold the risk estimator and predict any instance below threshold to be successfully classified and predict failure for those above it.

In particular, we define

C_1 = # instance when the risk estimator is below threshold yet the adapted SVM misclassifies

C_2 = # instance when the risk estimator is above threshold yet the adapted SVM correctly classifies

C_3 = # instance when the risk estimator is below threshold and the adapted SVM correctly classifies

C_4 = # instance when the risk estimator is above threshold yet the adapted SVM misclassifies

Finally, we calculate the Meta-Recognition False Accept Rate (MRFAR), the rate at which thresholded risk estima-

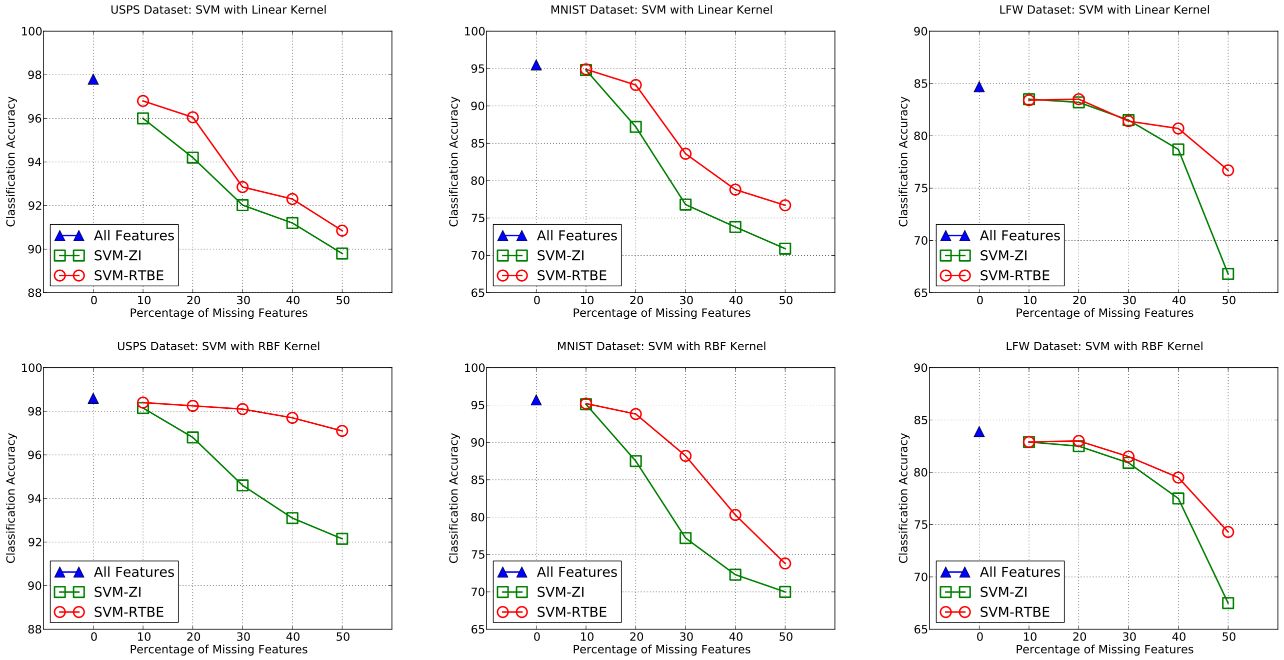

Figure 6. The above figure shows results on the three image datasets used for recognition: USPS, MNIST and LFW. The top row corresponds to results with SVM with linear kernel and bottom row corresponds to SVM with RBF kernel. Classification accuracy (Y - axis) is plotted as a function of percentage of missing features (X - Axis). It can be observed that RTBE consistently performs better than imputation with zero when features are missing. The difference is especially large when the percentage of missing features increases

tor incorrectly predicts success, and the Meta-Recognition Miss Detection Rate (MRMDR), the rate at which the thresholded risk estimator incorrectly predicts failure, as

$$MRFAR = \frac{|C_1|}{|C_1| + |C_4|}, \quad MRMDR = \frac{|C_2|}{|C_2| + |C_3|} \tag{10}$$

and then vary our threshold to compute the curves shown in Fig. 7. The resulting MRET curves show the proposed adaptation risk estimator is superior, and is more effective when combined with our novel SVM re-factoring. At operation time, just as one uses a traditional DET or ROC curve to set verification system parameters, the threshold on the risk parameter $\mathcal{R}(f_o)$ on MRET curve can be used to tune the rejection for an acceptable risk due to missing data. The results of this experiments[2] are shown in Fig 7.

5 Discussion and Conclusion

We noted that support vectors are an extremely biased and correlated set, and hence it might violate the assumptions of statistical tests for distributional shifting [7]. Detailed analysis of such correlation is an important future work. In streaming settings (incremental SVMs) for face recognition [1], the operational data available is always changing as support vectors are continuously updated. Handling missing data in such settings is another important aspect of operational adaptation. Some other problems such as adapt-

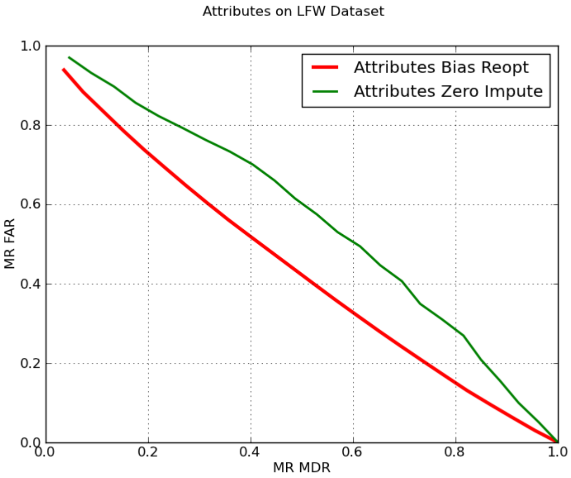

Figure 7. Meta-recognition comparison curve for evaluation adaptation risk estimators on partial data. Meta-recognition False accept rate (Y axis, Eq. 10) is the fraction of time the risk was low but the SVM classification failed, and the Meta-recognition miss detection rate (X axis) is the fraction of time the risk was high yet the SVM corrected classified the partial data. The ideal location is the lower-left. The plot shows the adaptation risk estimator for SVM refactoring (RTBE) is better than risk estimation for zero-imputation.

[2] We obtained similar results on USPS and MNIST dataset, but are not shown here

ing pre-trained classifiers for face-detection [17, 16] can be viewed as operational adaptation problems.

This paper provides theory and a novel solution for handling missing data in large-scale recognition problems. It adapts the solution at testing time, with virtually no loss in computational speed/efficiency, but significant improvements in accuracy compared to the state of the art. Further, it does not require a-priori knowledge of missing features. SVM refactoring with bias factoring performed consistently well on leading datasets compared to current de-facto methods, and when only modest data was missing, significantly outperformed the competition. Our method is suitable for large scale recognition tasks for many applications in computer vision like object recognition, feature tracking, action recognition, etc. that use supervised learning in the form of SVMs when features are missing. The second significant contribution of the is a technique for estimating the risk associated with classification with missing data, using only the data in the operational SVM. Our approach reclassifies the SVM in the reduced space and estimates the associated risk. Experiments show this risk measure is a better estimator of expected performance on the reduced dataset than just using the fraction of data missing. Finally, we show that the concept of operational adaptation is broader and applies to multiple areas beyond the domain of handling missing data.

References

[1] A. Bendale and T. Boult. Reliable posterior probability estimation for streaming face recognition. *CVPR Biometrics Workshop*, 2014.

[2] A. Bendale and T. Boult. Towards open world recognition. *CVPR*, 2015.

[3] L. Bottou and C.-J. Lin. Support vector machine solvers. *Large Scale Kernel Machines - MIT Press*, 2007.

[4] L. Breiman, J. Friedman, R. Olshen, and C. Stone. Classification and regression trees. *Wadsworth*, 1984.

[5] G. Chechik, G. Heitz, G. Elidan, P. Abbeel, and D. Koller. Max-margin classification of data with absent features. *J. of Machine Learning Research*, pages 1–21, 2008.

[6] G. Chiachia, A. Falcao, N. Pinto, A. Rocha, and D. Cox. Learning person-specific representations from faces in the wild. *IEEE TIFS*, 2014.

[7] D. A. Cieslak and N. V. Chawla. A framework for monitoring classifiers performance: when and why failure occurs? *Knowledge and Information Systems*, 18(1):83–108, 2009.

[8] N. Damer, B. Fuhrer, and A. Kuijper. Missing data estimation in multi-biometric identification and verification. *IEEE Biometric Measurements and Systems for Security and Medical Applications Workshop*, 2013.

[9] Y. Ding and A. Ross. A comparison of imputation methods for handling missing scores in biometric fusion. *Pattern Recognition*, pages 919–933, 2012.

[10] O. Fatukasi, J. Kittler, and N. Poh. Estimation of missing values in multimodal biometric fusion. *IEEE BTAS*, 2008.

[11] Z. Ghahramani and M. I. Jordan. Supervised learning from incomplete data via an em approach. In *Advances in neural information processing systems 6*. Citeseer, 1994.

[12] A. Globerson and S. Roweis. Nightmare at test time: robust learning by feature deletion. *ICML*, 2006.

[13] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. Technical Report 07-49, Univ. of Mass., Amherst, October 2007.

[14] J. J. Hull. A database for handwritten text recognition research. *IEEE TPAMI.*, 16(5):550–554, May 1994.

[15] A. Jain and A. Kumar. Biometrics of next generation: An overview. *Springer*, 2010.

[16] V. Jain and S. Farfade. Adapting classification cascades to new domains. *ICCV*, 2013.

[17] V. Jain and E. Learned-Miller. Online domain-adaptation of a pre-trained cascade of classifiers. *CVPR*, 2011.

[18] N. Kumar, A. Berg, P. Belhumeur, and S. Nayar. Describable visual attributes for face verification and image search. *IEEE TPAMI*, 2011.

[19] A. R. C. Lampert. Classifier adaptation at prediction time. *CVPR*, 2015.

[20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. of the IEEE*, 86(11):2278–2324, 1998.

[21] B. Marlin. Missing data problems in machine learning. *PhD Thesis, University of Toronto*, 2008.

[22] T. Ming-Huang and V. Kecman. Bias term b in svms again. *ESANN*, pages 441–448, 2004.

[23] N. Poh, D. Windrige, V. Mottl, A. Tatarchuk, and A. Eliseyev. Addressing missing values in kernel-based multimodal biometric fusion using neutral point substitution. *IEEE Trans. on Info Forensics and Security*, 2010.

[24] M. Saar-Tsechansky and F. Provost. Handling missing values when applying classification models. *J. of Machine Learning Research*, 2007.

[25] W. Scheirer, A. Bendale, and T. Boult. Predicting biometric facial recognition failure with similarity surfaces and support vector machines. *CVPR Biometrics Workshop*, 2008.

[26] W. Scheirer, N. Kumar, P. Belhumeur, and T. Boult. Multi-attribute spaces: Calibration for attribute fusion and similarity search. In *CVPR*, 2012.

[27] W. Scheirer, A. Rocha, R. Michaels, and T. E. Boult. Meta-Recognition: The Theory and Practice of Recognition Score Analysis. *IEEE TPAMI*, 33(8):1689–1695, Aug. 2011.

[28] W. Scheirer, A. Rocha, A. Sapkota, and T. Boult. Towards open set recognition. *IEEE TPAMI*, 2013.

[29] A. J. Smola, S. V. N. Vishwanathan, and T. Hofmann. Kernel methods for missing variables. In *Proc. Wksp on Articial Intelligence and Statistics*, 2005.

[30] M. Wilber, E. Rudd, B. Heflin, Y. Lui, and T. Boult. Exemplar codes for facial attributes and tattoo recognition. *WACV*, 2014.

[31] M. Wilber, W. Scheirer, P. Leitner, B. Heflin, J. Zott, D. Reinke, D. Delaney, and T. Boult. Animal recognition in the mojave desert: Vision tools for field biologists. *WACV*, 2013.