Privacy-Preserving SVM Classification using Non-metric MDS

Khaled Alotaibi and Beatriz de la Iglesia
School of Computing Sciences
University of East Anglia, Norwich, UK
{K.Alotaibi,B.Iglesia}@uea.ac.uk

***Abstract*—Privacy concerns are a critical issue in outsourcing data mining projects. Data owners are often unwilling to release their private data for analysis, as this may lead to data disclosure. One possible solution to address such concerns is to perturb the original data values so that they become hidden, thereby preserving privacy. This paper proposes a privacy-preserving technique using Non-metric Multidimensional Scaling, which not only preserves privacy but also maintains data utility for Support Vector Machine (SVM) classification. The perturbed data are subject to high uncertainty and have no information that can be exploited to disclose the original data. They also exhibit better class separation and compactness, which greatly eases the SVM task. The results show that the accuracy of the original and the perturbed data is similar, as the distances between the data objects both before and after the perturbation are well-preserved.**

***Keywords*—Privacy; Classification; Data Perturbation; SVM.**

I. INTRODUCTION

Multidimensional scaling (MDS) is a dimensionality reduction technique used to project data into a lower dimensional space, so that better data visualisation can be achieved [1]. The basic idea of the MDS technique is as follows: Given a matrix of proximities (similarities or dissimilarities) between data objects, find a configuration of data points (usually in a lower dimensional space) whose distances fit these proximities best. In non-metric MDS, the interpoint distances between the data points in the new space approximate a non-linear transformation derived from those proximities. The main motivation for using non-metric MDS as a data perturbation technique is that it has the ability to generate data in a new space thereby hiding the original data as the data points are located in their positions using only the rank-order distances so their original position cannot be inferred from the perturbed data. Furthermore, each data object is represented by completely different data values and each data variable has a different pdf.

The concept of "data perturbation" refers to transforming data, and thereby concealing any private details whilst preserving the underlying probabilistic properties, so that the inherent patterns can still be accurately extracted. However, in real cases, achieving these two objectives is a challenging task, as they are naturally in conflict. In this paper, we investigate whether our privacy model proposed in [2] preserves data utility for SVM classification whilst maintaining privacy. We distort the original data values using non-metric MDS transformation. Then, we use the perturbed data to carry out the classification analysis using SVM with three kernels–linear, polynomial and radial basis function, and show that the results are similar (if not better) to those obtained from the original data.

The structure of this paper is as follows. Section II presents some related literature. Section III introduces an overview of SVM. Section IV presents the proposed privacy-preserving data mining method, and discusses its privacy and utility preservation. Experimental results are introduced in Section V. Finally, Section VI presents our conclusion.

II. RELATED WORK

Most works on data perturbation are based on linear transformations using additive or multiplicative noise. Additive perturbation was designed for microdata protection where a random noise is added to the value of each attribute to produce new data values representing the perturbed data [3]. Multiplicative perturbation can provide, to some extent, a good data utility for data mining algorithms. Here, the basic idea is to multiply the original data matrix by either a rotation matrix or a projection matrix. Chen and Liu [4] proposed a rotation-based perturbation technique that generates the perturbed data by multiplying the original data with an orthogonal rotation matrix. They showed that the SVM classifier with the most popular kernels (polynomial, radial basis and neural network) is invariant to rotation transformation. Similarly, [5] suggests a geometric perturbation technique where extra components are added to the rotation model. These components are a random translation matrix and addition of noise so more data protection can be achieved while preserving the basic geometric properties of the data for SVM classification. In [6], [7], the original data are projected into a lower dimension using random projection matrix. This perturbation model was also performed using a PCA-based approach [8].

The drawback of additive perturbation is that the added noise will distort the distances between data points and therefore poor results will be obtained when applying data mining algorithms on the perturbed data. Furthermore, the additive noise can be filtered out and the privacy can then be compromised [9], [10]. Although multiplicative perturbation can provide a better solution to overcome the shortcomings of additive perturbation, the privacy model is not secure enough. The attacker can exploit some theoretical properties of the random matrices (they usually have a predictable structure) to disclose the original data values [11], [12].

Our work, in this paper, is categorised as a perturbation-based approach, where all data are distorted before they are released to a third party for analysis. However, unlike other methods, the proposed method preserves much of the statistical properties for the classification task using SVM and provides perfect data protection as the perturbed data are generated under a high level of uncertainty.

 ISBN: 978-1-61208-298-1

III. OVERVIEW OF SVM

The SVM is a distance-based learning approach that is widely used in data classification [13]. The basic idea is to find a hyperplane that separates the data into two classes with as great a margin as possible. The optimal hyperplane (decision boundary) is the one that separates these two classes and that maximizes the distance between the two closest points from either class (known as *support vectors*). Assume that the classes of data are separable. Consider a binary classification problem consisting of m pairs of training examples $(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)$, where $x_i \in \mathbb{R}^n$ and $y_i \in \{-1, 1\}$; the hyperplane is defined by

$$\mathbf{w} \cdot \mathbf{x} + b = 0, \tag{1}$$

where $\mathbf{w}$ is the weight vector and b is the bias. "." denotes the dot product in the feature space. Both parameters $\mathbf{w}$ and b must be chosen in such a way that the following two conditions are met:

$$\begin{aligned} \mathbf{w} \cdot \mathbf{x}_i + b \geq 1 \ \text{ when } \ y_i = 1, \\ \mathbf{w} \cdot \mathbf{x}_i + b \leq -1 \ \text{ when } \ y_i = -1. \end{aligned} \tag{2}$$

The classification rule of an unseen test object x' is defined by

$$g(x') = sign(\mathbf{w} \cdot \mathbf{x}' + b). \tag{3}$$

Maximizing the distance from a point $\mathbf{x}$ to the hyperplane in (1) determines the optimal hyperplane which creates the maximal margin between the negative and positive training examples. The distance from a hyperplane $H(\mathbf{w}, b)$ to a given data point $\mathbf{x}_i$ is simply

$$d(H(\mathbf{w}, b), \mathbf{x}_i) = \frac{\mathbf{w} \cdot \mathbf{x}_i + b}{||\mathbf{w}||} \geq \frac{1}{||\mathbf{w}||}. \tag{4}$$

That is, SVM finds the hyperplane that maximizes the margin by minimizing the squared norm of the hyperplane

$$\begin{aligned} \min_{\mathbf{w}} \ \frac{1}{2} ||\mathbf{w}||^2 \\ \text{subject to } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \, i = 1, 2, \ldots, m. \end{aligned} \tag{5}$$

For non-separable data, SVM can also deal with overlapping classes by maximizing the margin, allowing any misclassified data points to be penalised using a method known as the *soft margin* approach [14]. The misclassification bias can be defined by so-called *slack variables*, $\xi = \xi_1, \xi_2, \ldots, \xi_s$. Let $\xi_i \geq 0$; the constraints of the optimisation can be rewritten as

$$\begin{aligned} \mathbf{w} \cdot \mathbf{x}_i + b \geq 1 - \xi_i \ \text{ when } \ y_i = 1, \\ \mathbf{w} \cdot \mathbf{x}_i + b \leq -1 + \xi_i \ \text{ when } \ y_i = -1, \end{aligned} \tag{6}$$

and the learning task in SVM can be formalized as follows:

$$\begin{aligned} \min_{\mathbf{w}} \ \frac{1}{2} ||\mathbf{w}||^2 + C \sum_{i=1}^{l} \xi_i \\ \text{subject to } \begin{cases} y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i, \ i = 1, 2, \ldots, m, \\ \xi_i \geq 0, \ \Sigma \xi_i \leq C \end{cases} \end{aligned} \tag{7}$$

where the constant C is a regularisation parameter used to create a balance between a maximum margin and a small number of misclassified data points.

The SVM described so far finds linear boundaries in the input space. However, in many real problems, data may have non-linear decision boundaries, which would make finding a hyperplane that can successfully separate two overlapping classes a difficult task. One solution to this problem is to use the so-called *kernel trick*. The trick here is to transform the data X in d-dimensional input space into a higher D-dimensional feature space $\mathcal{F}$ (also known as *Hilbert* space), $\Phi : \mathbb{R}^d \rightarrow \mathbb{R}^D$ where $D \gg d$. This would make the overlapping classes separable in the new space $\mathcal{F}$. The transformation is performed via a kernel function K that satisfies Mercer's condition [15] so that better class separation can be achieved [16]. The function K can be defined by

$$K(\mathbf{u}, \mathbf{v}) = \Phi(\mathbf{u}) \cdot \Phi(\mathbf{v}), \tag{8}$$

where $\Phi : X \rightarrow \mathcal{F}$ and "." denotes the dot product in the feature space $\mathcal{F}$. By defining a proper K, we simply replace all occurrences of $\mathbf{x}_i$ in the SVM model with $\Phi(\mathbf{x}_i)$. That is, the feature space $\mathcal{F}$ is never explicitly dealt with, but rather we evaluate the dot product, $\Phi(\mathbf{x}_i) \ . \ \Phi(\mathbf{x}_j)$, directly using function K in the input space. Intuitively, computing only the dot product using K, in the feature space, is substantially cheaper than using the transformed attributes. For example, the Radial Basis Function (RBF) kernel unfolds into an infinite-dimension Hilbert space.

IV. DATA PERTURBATION

To disguise the original data values and provide unreal data values (synthetic data) that preserve as much as possible data properties for data mining task, we used the perturbation model proposed in [2], which is defined by some transformation T,

$$Y = T(X), \tag{9}$$

where $T : \mathbb{R}^n \rightarrow \mathbb{R}^p$ is a non-metric MDS transformation [17] such that

1) T preserves the rank ordering of the distances between objects in X and Y, i.e.

$$\begin{aligned} ||x_i - x_j|| < ||x_k - x_l|| \Longleftrightarrow \\ ||T(x_i) - T(x_j)|| < ||T(x_k) - T(x_l)||, \end{aligned} \tag{10}$$

and

2) T minimizes the sum of squared differences of the distances, i.e., it minimizes

$$\sum_{i,j} (||x_i - x_j|| - ||T(x_i) - T(x_j)||)^2. \tag{11}$$

For presentation convenience, we use different notation to distinguish between the dissimilarities in the original space, X, and the perturbed space, Y. The distances between points in Y are $||T(x_i) - T(x_j)|| = d_{ij}$. The above first condition (10) is satisfied through a monotonic function, f, that maintains a monotone relationship between the dissimilarities, δ_{ij}, in the original space, $\mathbb{R}^n$, and the distances, d_{ij}, in the lower space, $\mathbb{R}^p$, i.e., $d_{ij} = f(\delta_{ij})$. The estimates of point locations in the lower dimensional space should yield predicted distances, d_{ij}, between the points that "closely approximate" the observed dissimilarities, δ_{ij}. To quantify the discrepancy (the stress) and to find the best solution, the second condition (11) should be applied.

The monotone relationship is obtained by a non-linear approach (monotonic regression) which fits a non-linear function, $f : \delta_{ij} \mapsto d_{ij}$, and minimizes the stress, S. Let $M = m(m-1)/2$ be the number of possible dissimilarities, δ_{ij}, that can be calculated from the data matrix X. The stress is given by

$$S = \sqrt{\sum_{i,j}^{M} (\hat{d}_{ij} - d_{ij})^2 / \sum_{i,j}^{M} d_{ij}^2}, \qquad (12)$$

where $\hat{d}_{ij}$ (also known as *disparities*) are numbers representing a monotone least-square regression of d_{ij} on δ_{ij} . That is, the disparities are merely an admissible transformation of d_{ij}, chosen in optimal way, to minimize S over the data configuration matrix, Y.

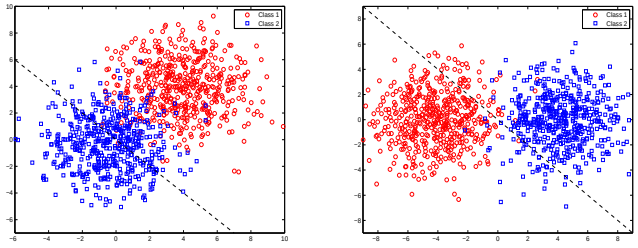
Non-metric MDS is quite similar to non-parametric procedures that are based on ranked data. The dissimilarities, δ_{ij}, are ranked by ordering them from lowest to highest and the disparities, $\hat{d}_{ij}$, should also follow the same monotonic ordering. This constraint implies the so-called *monotonicity* requirement

$$\text{if } \delta_{ij} < \delta_{kl} \text{ then } \hat{d}_{ij} \leq \hat{d}_{kl}. \qquad (13)$$

A. Data Utility Preservation

Non-metric MDS attempts to produce a new compact feature space with higher discriminative power [18]. In some senses, it may be considered similar to the kernel trick which generates a higher dimensional feature space to achieve better separation of the classes. It is expected that non-metric MDS can achieve good separation of negative examples from positive ones and as well as better class compactness (minimizing the intra-class distances while maximizing the inter-class separation) [19]. That is, we want to test whether the perturbed data, Y, can be as useful for SVM classification as the original data by measuring classification accuracy (or analogously the generalisation error) on both the original and the perturbed data.

The wider the margin between two groups of data, the better the SVM model will be at predicting the group for new instances. Figure 1 gives an insight into how non-metric MDS is able to discriminate two overlapping classes in a toy dataset, providing high data utility to the classification algorithm. In this example, the original dataset consists of 1,000 points in 4-dimensional space. We transform the data and generate a new dataset in 2-dimensional space. Then we train the SVM classifier on both datasets using a 50% training set, and test it on the remaining examples. As we expected, the perturbed data exhibit better class compactness and separation. It turns out that an optimal hyperplane that can successfully separate the two classes can be easily found in the lower dimensional space.



(a) Original data X (b) Perturbed data Y

Figure 1. Toy dataset, class separation in the original data X and the perturbed data Y using SVM classifier. For data X, the obtained hyperplane (dashed line) fails to separate the two classes whereas, for data Y, a better class separation has been achieved with a small relative error.

B. Privacy Preservation

For rigorous privacy analysis, we proposed a distance-based attack in order to disclose the location of a given point in the perturbed data using the technique of *Multilateration* [20], which uses knowledge about the location of $n+1$ data points. However, unlike [21], we do not require any prior knowledge beyond the known $n+1$ data points in the perturbed data. Then we show how this attack would fail to disclose the original data values because the perturbed data are subject to high uncertainty particularly in placing data points in the lower dimensional space.

1) Distance-Based Attack: Let $X \in \mathbb{R}^n$ be an $m \times n$ data matrix and x be unknown point for which we want to find the location. Given a set of $n+1$ known reference points, $R = \{r_1, r_2, \cdots, r_{n+1}\}$. Let d_{xr_i} be the Euclidean distance between the point x and each reference point r_i. The location of the point x is determined by minimizing

$$G(x) = \sum_{i=1}^{n+1} g_i(x)^2, \qquad (14)$$

where

$$g_i(x) = [(x_1 - x_{i1})^2 + (x_2 - x_{i2})^2 + \ldots + (x_n - x_{in})^2]^{1/2} - d_{xr_i}$$

is a non-linear function of n variables representing the coordinates of the point x. That is, we choose estimates $\hat{x_1}, \hat{x_2}, \ldots, \hat{x_n}$ that minimize $G(x)$. To solve this problem and find the minimum of the sum of squares, we use Gauss-Newton method which starts with a guess for x and iteratively move toward a better solution along the gradient of $G(x)$ until convergence.

 ISBN: 978-1-61208-298-1

Input: A set of $n+1$ known points, $r_1, r_2, \ldots, r_{n+1}$, an initial guess, x^0, a tolerance, $t > 0$, and a maximum number of iterations, $maxItr$.
Output: An estimation, $\hat{x}$, of the unknown point, x.

repeat
  Calculate $n+1$ distances, $d_{xr_1}, d_{xr_2}, \ldots, d_{xr_{n+1}}$, from the current x^k to each reference point, r;
  Evaluate $g_i(x^k)$ and $\nabla g_i(x^k)$ for $i = 1, 2, \ldots, n+1$;
  Move x^k a bit toward better location along the gradient, $x^{k+1} = ((A^k)^T A^k)^{-1}(A^k)^T b$;
  Calculate the error, err;
until *the error becomes less than the tolerance,* $err < t$, *or maximum number of iterations is exceeded,* $k > maxItr$;

Figure 2. Distance-Based Attack Algorithm.

If $g(x)$ is differentiable, then the refinement of the point x at iteration k can be achieved by the following linear approximation:

$$g(x) \approx g(x^k) + \nabla g(x^k)^T (x - x^k), \tag{15}$$

where $\nabla G(x^k)$ is the gradient (Jacobian) matrix that composes all first derivatives of x. To find x^{k+1} from x^k, we should minimize the sum of the squares of the linearized residuals, i.e.

$$\sum_{i=1}^{n+1} \Big(g_i(x^k) + \nabla g_i(x^k)^T (x - x^k) \Big)^2, \tag{16}$$

which is equivalent to solve the system $A^k x - b^k = 0$ which is defined by $(A^k)^T A^k x = (A^k)^T b^k$ and always consistent even when $A^k x = b^k$ is not consistent [22]. If A^k is non-singular, then there is a unique solution for x which represents the new position for the point x,

$$x^{k+1} = ((A^k)^T A^k)^{-1}(A^k)^T b. \tag{17}$$

To attack any given point, we develop a simple but effective search algorithm that can estimate the location of the unknown point while minimizing the sum of least-squares. The main steps are as follows: Start with an initial guess and move around in the direction where the relative error is minimized. The process is then repeated until convergence as described in Figure 2. The algorithm requires $O((n+1)^2 m)$ assuming that $m > n+1$.

To quantify the privacy for any given point x, we compute the ratio of the differences between x and its estimate $\hat{x}$ to the average distance from x to the $n+1$ known points $r_1, r_2, \cdots, r_{n+1}$, i.e.

$$\rho^* = \frac{||x - \hat{x}||}{\frac{1}{n+1} \sum_{j=1}^{n+1} ||x - r_j||}. \tag{18}$$

The overall privacy is then given by

$$\rho = \frac{1}{N} \sum_{i=1}^{N} \frac{||x_i - \hat{x}_i||}{\frac{1}{n+1} \sum_{j=1}^{n+1} ||x_i - r_j||}, \tag{19}$$

where N is the number of the remaining unknown points.

2) Uncertainty Measure: The process of placing points in our model is not straightforward, but rather it depends on preserving the order of dissimilarities. This implies that there is uncertainty about the exact location of any given point in the lower dimensional space, Y, and hence, a better protection against distance-based disclosure is achieved. To illustrate the basic idea, consider the following example. Let x be an unknown point and on distances d_{xr_1}, d_{xr_2} and d_{xr_3} from three other known points, r_1, r_2 and r_3, respectively. Assume that d_{xr_1}, d_{xr_2} and d_{xr_3} confirm the following order

$$d_{xr_1} < d_{xr_2} < d_{xr_3}.$$

To preserve the ordering (monotonicity), the point x should be placed somewhere within an area where the above order is satisfied. Assume that each point, here in this example, represents a single value, say salary. Assume also that $r_1 = 2K, r_2 = 5K$ and $r_3 = 7K$. If this information together with the distances from each point, r, to the point x are available to the attacker, she can guess that x is more likely to fall in an interval, say $[1K, 3K]$ with an assumption that the minimum salary is $1K$, as this would ensure the above ordering. Indeed, the probability that any attacked point locates within any given area is a measure of how well the original data are hidden. The probability P that the point x locates in area E, where $E \in \mathbb{R}$ is the domain of all possible outcomes, is

$$P(E) = \int_E f(x) dx. \tag{20}$$

This suggests that the probability of finding a given point x is inversely proportional to the area where the rank order is satisfied. The information available from the rank ordering would make the solution of non-metric MDS highly uncertain, as the points are not located to specific locations but rather to areas where the ordering is preserved. Therefore, the distance-based adversary attacks would fail to determine the exact location of the point.

V. EXPERIMENTS

For an empirical evaluation of training the SVM classifier on our privacy-preserving model, we use four numerical datasets collected from the UCI machine learning repository [23]. The datasets are Breast Cancer (699/9), Credit Approval (690/14), Pima Diabetes (768/8) and Hepatitis (155/19). All datasets have a binary class, i.e., positive and negative groups.

As we are not interested in the visual representation of data in the lower space, rather in achieving high data utility as far as possible, for some of the experiments the number of dimensions p was fixed to $n-1$, and the data projected into that lower dimensional space. Then, we used the generated data, Y, in p-dimensions, to carry out the SVM classification and to compare it with the results obtained from the original data, X.

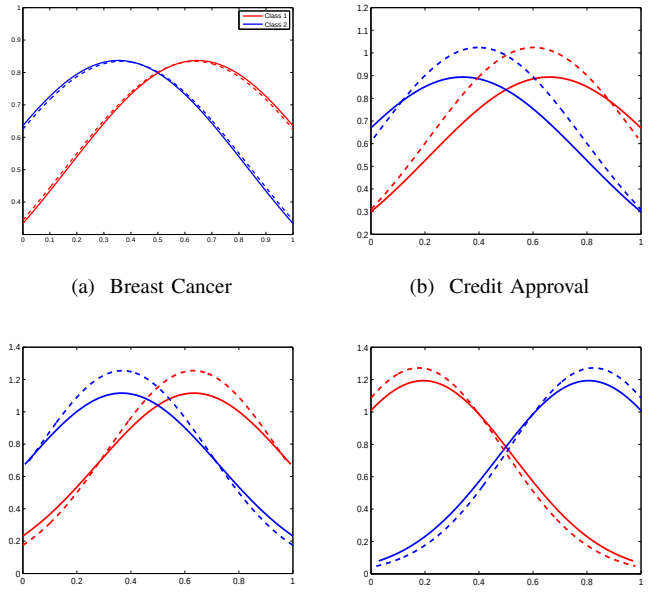
 ISBN: 978-1-61208-298-1

TABLE I. THE ACCURACIES OF LINEAR SVM ON THE ORIGINAL DATA, X, AND THE PERTURBED DATA, Y, AT REDUCED DIMENSIONS.

Dimension(n)	BreastCancer	CreditApproval	PimaDiabetes	Hepatitis
$X(n)$	0.9685	0.8623	0.7800	0.8099
$Y(n-1)$	0.9692	0.8638	0.7698	0.8314
$Y(n-2)$	0.9557	0.8584	0.7693	0.7930
$Y(n-3)$	0.9749	0.8725	0.7396	0.7721
$Y(n-4)$	0.9718	0.8649	0.7464	0.7721
$Y(n-5)$	0.9690	0.8557	0.7326	0.7876
$Y(n-6)$	0.9618	0.8630	0.7235	0.7876

To perturb data X and perform the classification, we used an implementation in Matlab. The dissimilarities, δ_{ij}, between the objects in X were first calculated. Then, we transformed the dissimilarities and generated Y. The initial configuration was chosen randomly. The stress S (12) was used as a data utility measure, as it determines the size of change in the interpoint distances in data Y as a result of the transformation.

We evaluated the classification accuracy on both the original and the perturbed data. 10-fold cross-validation was performed and the error rates of the testing set were evaluated for both data. The regularisation parameter, C, was set to 1 in all experiments because, in this set of experiments, our main concern is not to get an optimal SVM model but rather to compare the SVM models applied on the original and the perturbed data. Table I summarises the average accuracies of all the datasets at different dimensions using a linear SVM. For Pima Diabetes dataset, the accuracy on both the original and the perturbed data is lows because the data have imbalanced classes which inherently biased toward the majority concept. However, the difference in accuracy at the $n-1$ dimensional space was still plausible (0.01) exhibiting low distortion.

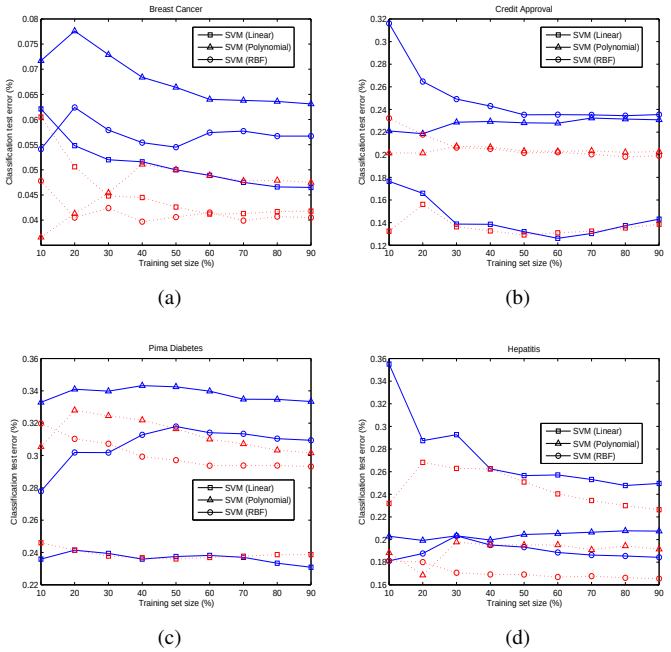


(a) Breast Cancer (b) Credit Approval

(c) Pima Diabetes (d) Hepatitis

Figure 3. An estimation of the classes' posterior probability in the original data (solid lines) and the perturbed data (dashed lines).

To show the usefulness of the perturbed data in terms of data utility, we assume independence of the variables and plot the distribution of the estimated posterior probability of assigning an object x to a class C_i. From naïve Bayes theorem with strong (naive) independence assumptions, the posterior probability is $P(C_i|x) = P(x|C_i)P(C_i)/P(x)$. This would help in estimating the quality of the classification on the perturbed data in comparison with the original data because it may give an insight of the classes overlapping in the original and in the perturbed space. As the outputs of the classifier are expected to approximate the corresponding posteriori class probabilities if it is reasonably well trained, $P(C_i|x)$ may also help to discover any potential decision boundaries that are typically expected to be close to Bayesian decision boundaries [24]. The results are shown in Figure 3. The distributions of classes before and after the perturbation almost coincide for all datasets. For the Hepatitis dataset, the separation of classes in both perturbed and original data appears to be better so an optimal hyperplane should be easily found. For instance, the accuracy on the perturbed data at the $n-1$ dimensional was 2% better than on the original data.



(a) (b)

(c) (d)

Figure 4. Classification error of SVM (with different kernels linear, polynomial and radial basis function) using training set with different sizes on the original data (solid lines) and the perturbed data (doted lines).

To evaluate the generalisation error when the SVM classifier is trained using a different training set size, we split each dataset into a training set containing $10\%, 20\%, \ldots, 90\%$ of the samples and a testing set containing the remaining corresponding samples. The results are depicted in Figure 4. The SVM error rates on the perturbed data were markedly lower, suggesting improved performance of the classifier on the perturbed data compared with the original data.

Finally, to assess the privacy of our model, we transformed the data into 6 different dimensions and attempted to estimate the original data values using distance-based attack. The number of known points was also incrementally varied to see its effect on the accuracy of locating unknown points. The average privacy of each dataset is shown in Figure 5. The results

 ISBN: 978-1-61208-298-1

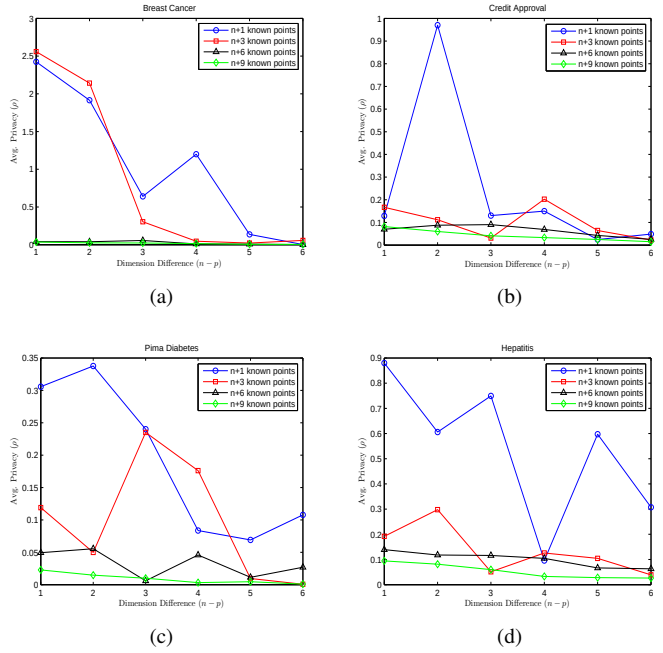


Figure 5. Average privacy (ρ) at different dimensions using different numbers of the known points.

indicate more resistance to the attack, especially at higher dimensions, and also confirm that the error of determining the location of an unknown point increases when the number of dimensions increases. That is, transforming the data into the few lower dimensions gives reasonable utility and privacy.

VI. Conclusion

The experiment results confirm that the perturbation method using non-metric MDS can provide high uncertainty for privacy preservation without affecting the accuracy of the SVM model and hence the utility. The perturbed data are still good enough to provide for reasonable discrimination between classes for SVM, and in some cases the data in the lower dimensionality provides improved classification performance. The main advantages of our method are that the perturbed data are independent from the original data and subject to a high degree of uncertainty; only the rank-order of dissimilarities are non-linearly mapped into distances in the perturbed data. In other words, the attacker knows nothing about the mapping function. Finally, most privacy-preserving data mining methods attempt to modify the learning algorithms in order to protect the original values from disclosure. This could decrease the efficiency of the algorithms, compromising the quality of the results. In contrast, the proposed method allows one to apply the algorithms directly to the perturbed data without any modification.

References

[1] M. Cox and T. Cox, "Multidimensional scaling," Handbook of data visualization, 2008, pp. 315–347.

[2] K. Alotaibi, V. J. Rayward-Smith, W. Wang, and B. de la Iglesia, "Non-linear dimensionality reduction for privacy-preserving data classification," in Proceedings of IEEE Fourth International Conference on Privacy, Security, Risk and Trust (PASSAT 2012). IEEE, 2012, pp. 694–701.

[3] R. Agrawal and R. Srikant, "Privacy-preserving data mining," ACM Sigmod Record, 2000, vol. 29, no. 2, pp. 439–450.

[4] K. Chen and L. Liu, "Privacy preserving data classification with rotation perturbation," in Proceedings of the Fifth IEEE International Conference on Data Mining, 2005, pp. 589–592.

[5] K. Chen, G. Sun, and L. Liu, "Towards attack-resilient geometric data perturbation," in Proceedings of the 2007 SIAM Data Mining Conference. SDM'07, 2007, pp. 78–89.

[6] K. Liu, H. Kargupta, and J. Ryan, "Random projection-based multiplicative data perturbation for privacy preserving distributed data mining," IEEE Transactions on Knowledge and Data Engineering, 2006, vol. 18, no. 1, pp. 92–106.

[7] S. Oliveira and O. Zaïane, "Privacy-preserving clustering to uphold business collaboration: A dimensionality reduction based transformation approach," International Journal of Information Security and Privacy, 2007, vol. 1, no. 2, pp. 13–36.

[8] R. Banu and N. Nagaveni, "Preservation of data privacy using pca based transformation," in Proceedings of the International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom'09). IEEE, 2009, pp. 439–443.

[9] D. Agrawal and C. Aggarwal, "On the design and quantification of privacy preserving data mining algorithms," in Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. ACM, 2001, pp. 247–255.

[10] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar, "Random-data perturbation techniques and privacy-preserving data mining," Knowledge and Information Systems, 2005, vol. 7, no. 4, pp. 387–414.

[11] S. Guo and X. Wu, "Deriving private information from arbitrarily projected data," Advances in Knowledge Discovery and Data Mining, 2007, pp. 84–95.

[12] K. Liu, C. Giannella, and H. Kargupta, "An attackers view of distance preserving maps for privacy preserving data mining," Knowledge Discovery in Databases, 2006, pp. 297–308.

[13] H. Drucker, D. Wu, and V. Vapnik, "Support vector machines for spam categorization," IEEE Transactions on Neural Networks, 1999, vol. 10, no. 5, pp. 1048–1054.

[14] K. Veropoulos, C. Campbell, and N. Cristianini, "Controlling the sensitivity of support vector machines," in Proceedings of the international joint conference on artificial intelligence, 1999, pp. 55–60.

[15] V. Vapnik, "The support vector method of function estimation," Nonlinear Modeling: Advanced Black-Box Techniques, 1998, vol. 55, pp. 55–85.

[16] T. Cover, "Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition," IEEE Transactions on Electronic Computers, 1965, no. 3, pp. 326–334.

[17] J. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," Psychometrika, 1964, vol. 29, no. 1, pp. 1–27.

[18] T. Cox and G. Ferry, "Discriminant analysis using non-metric multidimensional scaling," Pattern Recognition, 1993, vol. 26, no. 1, pp. 145–153.

[19] A. Gorban and A. Zinovyev, "Principal manifolds and graphs in practice: From molecular biology to dynamical systems," International Journal of Neural Systems, 2010, vol. 20, no. 3, pp. 219–232.

[20] W. Navidi, W. Murphy, and W. Hereman, "Statistical methods in surveying by trilateration," Computational statistics & data analysis, 1998, vol. 27, no. 2, pp. 209–227.

[21] E. Turgay, T. Pedersen, Y. Saygın, E. Savaş, and A. Levi, "Disclosure risks of distance preserving data transformations," in Scientific and Statistical Database Management. Springer, 2008, pp. 79–94.

[22] C. Meyer, Matrix analysis and applied linear algebra. Society for Industrial Mathematics, 2000, no. 71.

[23] A. Frank and A. Asuncion, "UCI machine learning repository," 2010, university of California, Irvine, School of Information and Computer Sciences. [Online]. Available: http://archive.ics.uci.edu/ml

[24] K. Tumer and J. Ghosh, "Analysis of decision boundaries in linearly combined neural classifiers," Pattern Recognition, 1996, vol. 29, no. 2, pp. 341–348.

 ISBN: 978-1-61208-298-1