Large Margin Gaussian Mixture Models with Differential Privacy

Manas A. Pathak and Bhiksha Raj

Abstract—As increasing amounts of sensitive personal information is aggregated into data repositories, it has become important to develop mechanisms for processing the data without revealing information about individual data instances. The differential privacy model provides a framework for the development and theoretical analysis of such mechanisms. In this paper, we propose an algorithm for learning a discriminatively trained multiclass Gaussian mixture model-based classifier that preserves differential privacy using a large margin loss function with a perturbed regularization term. We present a theoretical upper bound on the excess risk of the classifier introduced by the perturbation.

Index Terms—Differential privacy, machine learning.

1 Introduction

In recent years, vast amounts of personal data is being aggregated in the form of medical, financial records, social networks, and government census data. As these often contain sensitive information, a database curator interested in releasing a function such as a statistic evaluated over the data is faced with the prospect that it may lead to a breach of privacy of the individuals who contributed to the database. It is therefore important to develop techniques for retrieving desired information from a data set without revealing any information about individual data instances. *Differential privacy* [2] is a theoretical model proposed to address this issue. A query mechanism evaluated over a data set is said to satisfy differential privacy if it is likely to produce the same output on a data set differing by at most one element. This implies that an adversary having complete knowledge of all data instances but one along with a priori information about the remaining instance, is not likely to be able to infer any more information about the remaining instance by observing the output of the mechanism.

One of the most common applications for such large data sets such as the ones mentioned above is for training classifiers that can be used to categorize new data. If the training data contains private data instances, an adversary should not be able to learn anything about the individual training data set instances by analyzing the output of the classifier. Recently, mechanisms for learning differentially private classifiers have been proposed for logistic regression [3]. In this method, the objective function which is minimized by the classification algorithm is modified by adding a linear perturbation term. Compared to the original classifier, there is an additional error introduced by the perturbation term in the differentially private classifier. It is important to have an upper bound on this error as a cost of preserving privacy.

The work mentioned above is largely restricted to binary classification, while multiclass classifiers are more useful in many practical situations. In this paper, we propose an algorithm for learning multiclass Gaussian mixture model classifiers which satisfies differential privacy. Gaussian classifiers that model the distributions of individual classes as being generated from Gaussian distribution or a mixture of Gaussian distributions [4] are commonly used as multiclass classifiers. We use a large margin discriminative algorithm for training the classifier introduced by Sha and Saul [5]. To ensure that the learned multiclass classifier preserves differential privacy, we modify the objective function by introducing a perturbed regularization term.

2 Differential Privacy

In recent years, the differential privacy model proposed by Dwork [2] has emerged as a robust standard for data privacy. It originated from the statistical database model, where the data set D is a collection of elements and a randomized *query mechanism* M produces a response when performed on a given data set. Two data sets D and D' differing by at most one element are said to be *adjacent*. There are two proposed definitions for adjacent data sets the stronger one based on deletion: D' containing of one entry less than D, and the weaker one based on substitution: one entry of D' differs in value from D. We use the deletion based definition of adjacency previously where a single entry of the data set $D = \{x_1, \ldots, x_{n-1}, x_n\}$ is deleted to obtain an adjacent data set $D' = \{x_1, \ldots, x_{n-1}\}$.

The query mechanism M is said to satisfy differential privacy if the probability of M resulting in a solution S when performed on a data set D is very close to the probability of M resulting in the same solution S when executed on an adjacent data set D'. Assuming the query mechanism to be a function $M : D \mapsto \mathrm{range}(M)$ with a

- *The authors are with the Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213. E-mail: {manasp, bhiksha}@cs.cmu.edu.*

Manuscript received 30 Nov. 2010; revised 15 Apr. 2011; accepted 17 Feb. 2012; published online 2 Mar. 2012.
For information on obtaining reprints of this article, please send e-mail to: tdsc@computer.org, and reference IEEECS Log Number TDSCSI-2010-11-0224.
Digital Object Identifier no. 10.1109/TDSC.2012.27.

probability density P defined over the space of M, differential privacy is formally defined as follows:

Definition. *A randomized function M satisfies ϵ-differential privacy if for all adjacent data sets D and D' and for any $S \in \text{range}(M)$*

$$\left|\log \frac{P(M(D) = S)}{P(M(D') = S)}\right| \leq \epsilon.$$

The value of the ϵ parameter, which is referred to as *leakage*, determines the degree of privacy. As there is always a tradeoff between privacy and utility, the choice of ϵ is motivated by the requirements of the application.

In a machine learning setting, the query mechanism can be thought of as an algorithm learning the classification, regression or density estimation rule which is evaluated over the training data set. The output of an algorithm satisfying differential privacy is likely to be same when the value of any single data set instance is modified, and therefore, no additional information can be obtained about any individual training data instances with certainty by observing the output of the learning algorithm, beyond what is already known to an adversary. Differential privacy is a strong definition of privacy—it provides an *ad omnia* guarantee as opposed to most other models that provide ad hoc guarantees against specific set of attacks and adversarial behaviors.

2.1 Related Work

The earlier work on differential privacy was related to simple data mining tasks and data release mechanisms [6], [7], [8], [9]. Although many of these works have connection to machine learning problems, more recently the design and analysis of machine learning algorithms satisfying differential privacy has been actively studied. Kasiviswanathan et al. [10] present a framework for converting a general agnostic PAC learning algorithm to an algorithm that satisfies privacy constraints. Chaudhuri and Monteleoni [3] propose the sensitivity method to create a differentially private logistic regression classifier by adding Laplace noise to the estimated parameters. They propose another differentially private formulation which involves modifying the objective function of the logistic regression classifier by adding a linear term scaled by Laplace noise. The second formulation is advantageous because it is independent of the classifier sensitivity which difficult to compute in general and it can be shown that using a perturbed objective function introduces a lower error as compared to the exponential mechanism.

However, the above-mentioned differentially private classification algorithms only address the problem of binary classification. Although it is possible to extend binary classification algorithms to multiclass using techniques like one-versus-all, it is much more expensive to do so as compared to a naturally multiclass classification algorithm. Jagannathan et al. [11] present a differentially private random decision tree learning algorithm which can be applied to multiclass classification. Their approach involves perturbing leaf nodes using the sensitivity method, and they do not provide theoretical analysis of excess risk of the perturbed classifier.

In this paper, we extend the objective perturbation framework proposed by Chaudhuri and Monteleoni [3] to create differentially private classifiers based on large margin Gaussian mixture models [5], [12]. We show that the bound on the excess risk of the differentially private classifier is linear in the number of classes and inversely proportional to the square of the privacy parameter ϵ.

3 Large Margin Gaussian Classifiers

We investigate the large margin multiclass classification algorithm introduced by Sha and Saul [5]. The training data set $(\vec{x}, \vec{y})$ contains n d-dimensional iid training data instances $\vec{x}_i \in \mathbb{R}^d$ each with labels $y_i \in \{1, \ldots, C\}$.

3.1 Modeling Single Gaussian per Class

We first consider the setting where each class is modeled as a single Gaussian ellipsoid. Each class ellipsoid is parameterized by the centroid $\vec{\mu}_c \in \mathbb{R}^d$, the inverse covariance matrix $\mathbf{\Psi}_c \in \mathbb{R}^{d \times d}$, and a scalar offset $\theta_c \geq 0$. The decision rule is to assign an instance $\vec{x}_i$ to the class having smallest Mahalanobis distance [13] with the scalar offset from $\vec{x}_i$ to the centroid of that class

$$y_i = \arg\min_c \, (\vec{x}_i - \vec{\mu}_c)^T \mathbf{\Psi}_c (\vec{x}_i - \vec{\mu}_c) + \theta_c. \tag{1}$$

To simplify the notation, we expand $(\vec{x}_i - \vec{\mu}_c)^T \mathbf{\Psi}_c (\vec{x}_i - \vec{\mu}_c)$ and collect the parameters for each class as the following $(d+1) \times (d+1)$ *positive semidefinite* matrix:

$$\mathbf{\Phi}_c = \begin{bmatrix} \mathbf{\Psi}_c & -\mathbf{\Psi}_c \vec{\mu}_c \\ -\vec{\mu}_c^T \mathbf{\Psi}_c & \vec{\mu}_c^T \mathbf{\Psi}_c \vec{\mu}_c + \theta_c \end{bmatrix}, \tag{2}$$

and also append a unit element to each d-dimensional vector $\vec{x}_i$. The decision rule for a data instance $\vec{x}_i$ simplifies to

$$y_i = \operatorname*{argmin}_c \; \vec{x}_i^T \mathbf{\Phi}_c \vec{x}_i. \tag{3}$$

The discriminative training procedure involves estimating a set of positive semidefinite matrices $\{\mathbf{\Phi}_1, \ldots, \mathbf{\Phi}_C\}$ from the training data $\{(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)\}$ which optimize the performance on the decision rule mentioned above. We apply the large margin intuition about the classifier maximizing the distance of training data instances from the decision boundaries having a lower error. This leads to the classification algorithm being robust to outliers with provably strong generalization guarantees. Formally, we require that for each training data instance $\vec{x}_i$ with label y_i, the distance from $\vec{x}_i$ to the centroid of class y_i is at least less than its distance from centroids of all other classes by one

$$\forall c \neq y_i : \vec{x}_i^T \mathbf{\Phi}_c \vec{x}_i \geq 1 + \vec{x}_i^T \mathbf{\Phi}_{y_i} \vec{x}_i. \tag{4}$$

Analogous to support vector machines, the training algorithm is an optimization problem minimizing the *hinge loss* denoted by $[f]_+ = \max(0, f)$, with a linear penalty for incorrect classification. We use the sum of traces of inverse covariance matrices for each classes as a *regularization* term. The regularization requires that if we can learn a classifier which labels every training data instance correctly, we choose the one with the lowest inverse covariance or highest covariance for each class ellipsoid as this prevents the classifier from overfitting. The parameter γ controls the tradeoff between the loss function and the regularization

$$J(\mathbf{\Phi}, \vec{x}, \vec{y}) = \frac{1}{n}\sum_{i=1}^{n}\sum_{c\neq y_i}\left[1 + \vec{x}_i^T(\mathbf{\Phi}_{y_i} - \mathbf{\Phi}_c)\vec{x}_i\right]_+ + \gamma\sum_c \mathrm{trace}(\mathbf{\Psi}_c). \tag{5}$$

The inverse covariance matrix $\mathbf{\Psi}_c$ is contained in the upper left size $d \times d$ block of the matrix $\mathbf{\Phi}_c$. We replace it with $\mathbf{I_\Phi \Phi}_c\mathbf{I_\Phi}$, where $\mathbf{I_\Phi}$ is the truncated size $(d+1)\times(d+1)$ identity matrix with the last diagonal element $I_{\Phi_{d+1,d+1}}$ set to zero. The optimization problem becomes

$$J(\mathbf{\Phi}, \vec{x}, \vec{y}) = \frac{1}{n}\sum_{i=1}^{n}\sum_{c\neq y_i}\left[1 + \vec{x}_i^T(\mathbf{\Phi}_{y_i} - \mathbf{\Phi}_c)\vec{x}_i\right]_+ + \gamma\sum_c \mathrm{trace}(\mathbf{I_\Phi\Phi}_c\mathbf{I_\Phi}). \tag{6}$$

The objective function is convex function of positive semidefinite matrices $\mathbf{\Phi}_c$. The optimization can be formulated as a semidefinite programming problem [14] and be solved efficiently using interior point methods.

3.2 Generalizing to Mixtures of Gaussians

We extend the above classification framework to modeling each class as a mixture of K Gaussians ellipsoids. A simple extension is to consider each data instance $\vec{x}_i$ as having a mixture component m_i along with the label y_i. The mixture labels are not available a priori, these can be generated by training a generative GMM using the data instances in each class and selecting the mixture component with the highest posterior probability. Similar to the criterion in (4), we require that for each training data instance $\vec{x}_i$ with label y_i and mixture component m_i, the distance from $\vec{x}_i$ to the centroid of the Gaussian ellipsoid for the mixture component m_i of label y_i is at least one greater than the minimum distance from $\vec{x}_i$ to the centroid of any mixture component of any other class. If $\mathbf{\Phi}_{y_i,m_i}$ corresponds to the parameter matrix of the mixture component m_i of the class y_i, and $\mathbf{\Phi}_{cm}$ corresponds to the parameter matrix of the mixture component m of the class c

$$\forall c \neq y_i : \min_m \vec{x}_i^T\mathbf{\Phi}_{cm}\vec{x}_i \geq 1 + \vec{x}_i^T\mathbf{\Phi}_{y_i,m_i}\vec{x}_i.$$

In order to maintain the convexity of the objective function, we use the property $\min_m a_m \geq -\log\sum_m e^{-a_m}$ to rewrite the above constraint as

$$\forall c \neq y_i : -\log\sum_m e^{-\vec{x}_i^T\mathbf{\Phi}_{cm}\vec{x}_i} \geq 1 + \vec{x}_i^T\mathbf{\Phi}_{y_i,m_i}\vec{x}_i. \tag{7}$$

As before, we minimize the hinge loss of misclassification along with the regularization term. The objective function becomes

$$J(\mathbf{\Phi}, \vec{x}, \vec{y}) = \frac{1}{n}\sum_{i=1}^{n}\sum_{c\neq y_i}\left[1 + \vec{x}_i^T\mathbf{\Phi}_{y_i,m_i}\vec{x}_i + \log\sum_m e^{-\vec{x}_i^T\mathbf{\Phi}_{cm}\vec{x}_i}\right]_+ + \gamma\sum_{cm}\mathrm{trace}(\mathbf{I_\Phi\Phi}_{cm}\mathbf{I_\Phi}). \tag{8}$$

After this modification, the underlying optimization problem remains a convex semidefinite program and is tractable to solve. As compared to the single Gaussian case, however, the space of the problem increases linearly as the product of the number of classes and mixture components CK.

3.3 Making the Objective Function Differentiable and Strongly Convex

The hinge loss being nondifferentiable is not convenient for our analysis; we replace it with a surrogate loss function called Huber loss l_h. For small values of the parameter h, Huber loss has similar characteristics as hinge loss and provides the same accuracy [15]. Let us denote $\vec{x}_i^T\mathbf{\Phi}_{y_i}\vec{x}_i + \log\sum_m e^{-\vec{x}_i^T\mathbf{\Phi}_c\vec{x}_i}$ by $M(x_i, \mathbf{\Phi}_c)$ for conciseness. The Huber loss ℓ_h computed over data instances $(\vec{x}_i, y_i)$ becomes

$$\ell_h(\mathbf{\Phi}_c, \vec{x}_i, y_i) = \begin{cases} 0 \\ \quad \text{if } M(x_i, \mathbf{\Phi}_c) > h, \\ \frac{1}{4h}\left[h - \vec{x}_i^T\mathbf{\Phi}_{y_i}\vec{x}_i - \log\sum_m e^{-\vec{x}_i^T\mathbf{\Phi}_c\vec{x}_i}\right]^2 \\ \quad \text{if } |M(x_i, \mathbf{\Phi}_c)| \leq h \\ -\vec{x}_i^T\mathbf{\Phi}_{y_i}\vec{x}_i - \log\sum_m e^{-\vec{x}_i^T\mathbf{\Phi}_c\vec{x}_i} \\ \quad \text{if } M(x_i, \mathbf{\Phi}_c) < -h. \end{cases} \tag{9}$$

Finally, the regularized Huber loss computed over the training data set $(\mathbf{x}, \vec{y})$ is given by

$$\begin{aligned} J(\mathbf{\Phi}, \vec{x}, \vec{y}) &= \frac{1}{n}\sum_{i=1}^{n}\sum_{c\neq y_i}\ell_h\left[1 + \vec{x}_i^T\mathbf{\Phi}_{y_i}\vec{x}_i + \log\sum_m e^{-\vec{x}_i^T\mathbf{\Phi}_c\vec{x}_i}\right] \\ &\quad + \gamma\sum_{cm}\mathrm{trace}(\mathbf{I_\Phi\Phi}_{cm}\mathbf{I_\Phi}) \\ &= \frac{1}{n}\sum_{i=1}^{n} L(\mathbf{\Phi}, \vec{x}_i, y_i) + N(\mathbf{\Phi}) \\ &= L(\mathbf{\Phi}, \vec{x}, \vec{y}) + N(\mathbf{\Phi}), \end{aligned} \tag{10}$$

where $L(\mathbf{\Phi}, \vec{x}_i, y_i)$ is the contribution of a single data instance to the loss, $L(\mathbf{\Phi}, \vec{x}, \vec{y})$ is the overall loss function, and $N(\mathbf{\Phi})$ is the regularization term.

Our theoretical analysis requires that the regularized loss function minimized by the classifier is γ-strongly convex. The regularized loss function $J(\mathbf{\Phi}, \vec{x}, \vec{y})$ is convex as it is the sum of convex loss function $L(\mathbf{\Phi}, \vec{x}, \vec{y})$ and regularization term $N(\mathbf{\Phi})$, but it does not satisfy strong convexity. Toward this, we augment it with an additional ℓ_2 regularization term to get

$$\begin{aligned} J(\mathbf{\Phi}, \vec{x}, \vec{y}) &= \frac{1}{n}\sum_{i=1}^{n} L(\mathbf{\Phi}, \vec{x}_i, y_i) \\ &\quad + \gamma\sum_{cm}\mathrm{trace}(\mathbf{I_\Phi\Phi}_{cm}\mathbf{I_\Phi}) + \lambda\sum_{cm}\|\mathbf{\Phi}_{cm}\|^2 \\ &= L(\mathbf{\Phi}, \vec{x}, \vec{y}) + N(\mathbf{\Phi}), \end{aligned} \tag{11}$$

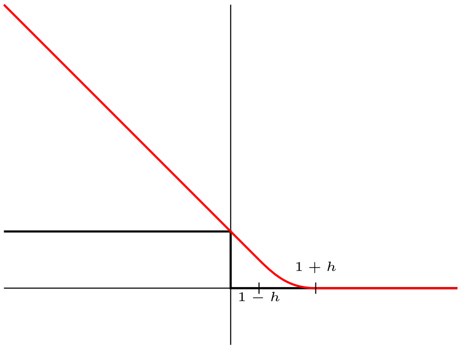

Fig. 1. Huber loss.

where $N(\mathbf{\Phi})$ now includes the extended regularization term. As the ℓ_2 regularization term satisfies 1-strong convexity, it can be easily shown that $J(\mathbf{\Phi}, \vec{x}, \vec{y})$ satisfies λ-strong convexity, i.e.,

$$J\left(\frac{\mathbf{\Phi}_1 + \mathbf{\Phi}_2}{2}, \vec{x}, \vec{y}\right) = \frac{J(\mathbf{\Phi}_1, \vec{x}, \vec{y}) + J(\mathbf{\Phi}_2, \vec{x}, \vec{y})}{2} - \frac{\lambda}{4}\sum_{cm} \|\mathbf{\Phi}_{1,cm} - \mathbf{\Phi}_{2,cm}\|^2. \tag{12}$$

4 Differentially Private Large Margin Gaussian Mixture Models

We modify the large margin Gaussian mixture model formulation to satisfy differential privacy by introducing a perturbation term in the objective function. As this classification method ultimately consists of minimizing a convex loss function, the large margin characteristics of the classifier by itself do not interfere with differential privacy.

We generate the size $(d+1) \times (d+1)$ perturbation matrix $\mathbf{b}$ with density

$$P(\mathbf{b}) \propto \exp(-\epsilon\|\mathbf{b}\|), \tag{13}$$

where $\|\cdot\|$ is the Frobenius norm (element-wise ℓ_2 norm) and ϵ is the privacy parameter. One method of generating such a $\mathbf{b}$ matrix is to sample $\|\mathbf{b}\|$ from $\Gamma((d+1)^2, \frac{1}{\epsilon})$ and the direction of $\mathbf{b}$ from the uniform distribution.

Our proposed learning algorithm minimizes the following objective function $J_p(\mathbf{\Phi}, \vec{x}, \vec{y})$, where the subscript p denotes privacy

$$J_p(\mathbf{\Phi}, \vec{x}, \vec{y}) = J(\mathbf{\Phi}, \vec{x}, \vec{y}) + \sum_c \sum_{ij} b_{ij}\Phi_{cij}. \tag{14}$$

As the dimensionality of the perturbation matrix $\mathbf{b}$ is same as that of the classifier parameters $\mathbf{\Phi}_c$, the parameter space of $\mathbf{\Phi}$ does not change after perturbation. In other words, given two data sets $(\vec{x}, \vec{y})$ and $(\vec{x}', \vec{y}')$, if $\mathbf{\Phi^p}$ minimizes $J_p(\mathbf{\Phi}, \vec{x}, \vec{y})$, it is always possible to have $\mathbf{\Phi^p}$ minimize $J_p(\mathbf{\Phi}, \vec{x}', \vec{y}')$. This is a necessary condition for the classifier $\mathbf{\Phi^p}$ satisfying differential privacy.

Furthermore, as the perturbation term is convex and positive semidefinite, the perturbed objective function $J_p(\mathbf{\Phi}, \vec{x}, \vec{y})$ has the same properties as the unperturbed objective function $J(\mathbf{\Phi}, \vec{x}, \vec{y})$. Also, the perturbation does not introduce any additional computational cost as compared to the original algorithm.

5 Theoretical Analysis

5.1 Proof of Differential Privacy

We prove that the classifier minimizing the perturbed optimization function $J_p(\mathbf{\Phi}, \vec{x}, \vec{y})$ satisfies ϵ-differential privacy in the following theorem. Given a data set $(\vec{x}, \vec{y}) = \{(\vec{x}_1, y_1), \ldots, (\vec{x}_{n-1}, y_{n-1}), (\vec{x}_n, y_n)\}$, the probability of learning the classifier $\mathbf{\Phi}^p$ is close to the probability of learning the same classifier $\mathbf{\Phi}^p$ given an adjacent data set $(\vec{x}', \vec{y}') = \{(\vec{x}_1, y_1), \ldots, (\vec{x}_{n-1}, y_{n-1})\}$ which *wlog* does not contain the nth instance. As we mentioned in the previous section, it is always possible to find such a classifier $\mathbf{\Phi}^p$ minimizing both $J_p(\mathbf{\Phi}, \vec{x}, \vec{y})$ and $J_p(\mathbf{\Phi}, \vec{x}', \vec{y}')$ due to the perturbation matrix being in the same space as the optimization parameters.

Our proof requires a strictly convex perturbed objective function resulting in a unique solution $\mathbf{\Phi}^p$ minimizing it. This in turn requires that the loss function $L(\mathbf{\Phi}, \vec{x}, y)$ is strictly convex and differentiable, and the regularization term $N(\mathbf{\Phi})$ is convex. These seemingly strong constraints are satisfied by many commonly used classification algorithms such as logistic regression, support vector machines, and our general perturbation technique can be extended to those algorithms. In our proposed algorithm, the Huber loss is by definition a differentiable function and the trace regularization term is convex and differentiable. Additionally, we require that the difference in the gradients of $L(\mathbf{\Phi}, \vec{x}, y)$ calculated over for two adjacent training data sets is bounded. We prove this property in Lemma A.1 given in the appendix.[1]

Theorem 5.1. *For any two adjacent training data sets $(\vec{x}, \vec{y})$ and $(\vec{x}', \vec{y}')$, the classifier $\mathbf{\Phi}^p$ minimizing the perturbed objective function $J_p(\mathbf{\Phi}, \vec{x}, \vec{y})$ satisfies differential privacy*

$$\left|\log \frac{P(\mathbf{\Phi}^p|\vec{x}, \vec{y})}{P(\mathbf{\Phi}^p|\vec{x}', \vec{y}')}\right| \leq \epsilon',$$

where $\epsilon' = \epsilon + k$ for a constant factor k.

Proof. As $J(\mathbf{\Phi}, \vec{x}, \vec{y})$ is strongly convex and differentiable, there is a unique solution $\mathbf{\Phi}^*$ that minimizes it. As the perturbation term $\sum_c \sum_{ij} b_{ij}\Phi_{cij}$ is also convex and differentiable, the perturbed objective function $J_p(\mathbf{\Phi}, \vec{x}, \vec{y})$ also has a unique solution $\mathbf{\Phi}^p$ that minimizes it. Differentiating $J_p(\mathbf{\Phi}, \vec{x}, \vec{y})$ w.r.t $\mathbf{\Phi}_{cm}$, we have

$$\frac{\partial}{\partial \mathbf{\Phi}_{cm}} J_p(\mathbf{\Phi}, \vec{x}, \vec{y}) = \frac{\partial}{\partial \mathbf{\Phi}_{cm}} L(\mathbf{\Phi}, \vec{x}, \vec{y}) + \gamma \mathbf{I_\Phi} + 2\lambda\mathbf{\Phi_{cm}} + \mathbf{b}. \tag{15}$$

Substituting the optimal $\mathbf{\Phi}^p_{cm}$ in the derivative gives us

$$\gamma \mathbf{I_\Phi} + \mathbf{b} + 2\lambda\mathbf{\Phi_{cm}} = -\frac{\partial}{\partial \mathbf{\Phi}_{cm}} L(\mathbf{\Phi}^p, \vec{x}, \vec{y}). \tag{16}$$

This relation shows that two different values of $\mathbf{b}$ cannot result in the same optimal $\mathbf{\Phi}^p$. As the perturbed objective function $J_p(\mathbf{\Phi}, \vec{x}, \vec{y})$ is also convex and differentiable, there is a bijective map between the perturbation $\mathbf{b}$ and the unique $\mathbf{\Phi}^p$ minimizing $J_p(\mathbf{\Phi}, \vec{x}, \vec{y})$.

[1] An appendix containing supplementary lemmas can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TDSC.2012.27.

Let $\mathbf{b_1}$ and $\mathbf{b_2}$ be the two perturbations applied when training with the adjacent data sets $(\vec{x}, \vec{y})$ and $(\vec{x}', \vec{y}')$, respectively. Assuming that we obtain the same optimal solution $\mathbf{\Phi}^p$ while minimizing both $J_p(\mathbf{\Phi}, \vec{x}, \vec{y})$ with perturbation $\mathbf{b}_1$ and $J_p(\mathbf{\Phi}, \vec{x}, \vec{y})$ with perturbation $\mathbf{b}_2$

$$\begin{aligned}
&\gamma \mathbf{I_\Phi} + 2\lambda \mathbf{\Phi_{cm}} + \mathbf{b}_1 = -\frac{\partial}{\partial \mathbf{\Phi}_{cm}} L(\mathbf{\Phi}^p, \vec{x}, \vec{y}),\\
&\gamma \mathbf{I_\Phi} + 2\lambda \mathbf{\Phi_{cm}} + \mathbf{b}_2 = -\frac{\partial}{\partial \mathbf{\Phi}_{cm}} L(\mathbf{\Phi}^p, \vec{x}', \vec{y}'),\\
&\mathbf{b}_1 - \mathbf{b}_2 = \frac{\partial}{\partial \mathbf{\Phi}_{cm}} L(\mathbf{\Phi}^p, \vec{x}', \vec{y}') - \frac{\partial}{\partial \mathbf{\Phi}_{cm}} L(\mathbf{\Phi}^p, \vec{x}, \vec{y}).
\end{aligned} \tag{17}$$

We take the Frobenius norm of both sides and apply the bound on the RHS as given by Lemma A.1. Assuming that $n > 1$, in order to ensure that $(\vec{x}', \vec{y}')$ is not an empty set

$$\begin{aligned}
\|\mathbf{b}_1 - \mathbf{b}_2\| &= \left\| \frac{\partial}{\partial \mathbf{\Phi}_{cm}} L(\mathbf{\Phi}^p, \vec{x}', \vec{y}') - \frac{\partial}{\partial \mathbf{\Phi}_{cm}} L(\mathbf{\Phi}^p, \vec{x}, \vec{y}) \right\|\\
&= \Bigg\| \frac{1}{n-1} \sum_{i=1}^{n-1} \frac{\partial}{\partial \mathbf{\Phi}_{cm}} L(\mathbf{\Phi}^p, \vec{x}_i, y_i)\\
&\quad - \frac{1}{n} \sum_{i=1}^{n-1} \frac{\partial}{\partial \mathbf{\Phi}_{cm}} L(\mathbf{\Phi}^p, \vec{x}_i, y_i) - \frac{1}{n} \frac{\partial}{\partial \mathbf{\Phi}_{cm}} L(\mathbf{\Phi}^p, \vec{x}_n, y_n) \Bigg\|\\
&= \frac{1}{n} \left\| \frac{1}{n-1} \sum_{i=1}^{n-1} \frac{\partial}{\partial \mathbf{\Phi}_{cm}} L(\mathbf{\Phi}^p, \vec{x}_i, y_i) - \frac{\partial}{\partial \mathbf{\Phi}_{cm}} L(\mathbf{\Phi}^p, \vec{x}_n, y_n) \right\|\\
&\leq \frac{2}{n} \leq 1.
\end{aligned}$$

Using this property, we can calculate the ratio of densities of drawing the perturbation matrices $\mathbf{b}_1$ and $\mathbf{b}_2$ as

$$\frac{P(\mathbf{b} = \mathbf{b}_1)}{P(\mathbf{b} = \mathbf{b}_2)} = \frac{\frac{1}{\mathrm{surf}(\|\mathbf{b}_1\|)} \|\mathbf{b}_1\|^d \exp[-\epsilon \|\mathbf{b}_1\|]}{\frac{1}{\mathrm{surf}(\|\mathbf{b}_2\|)} \|\mathbf{b}_2\|^d \exp[-\epsilon \|\mathbf{b}_2\|]},$$

where $\mathrm{surf}(\|\mathbf{b}\|)$ is the surface area of the $(d+1)$-dimensional hypersphere with radius $\|\mathbf{b}\|$. As $\mathrm{surf}(\|\mathbf{b}\|) = \mathrm{surf}(1)\|\mathbf{b}\|^d$, where $\mathrm{surf}(1)$ is the area of the unit $(d+1)$-dimensional hypersphere, the ratio of the densities becomes

$$\begin{aligned}
\frac{P(\mathbf{b} = \mathbf{b}_1)}{P(\mathbf{b} = \mathbf{b}_2)} &= \exp[\epsilon(\|\mathbf{b}_2\| - \|\mathbf{b}_1\|)]\\
&\leq \exp[\epsilon \|\mathbf{b}_2 - \mathbf{b}_1\|] \leq \exp(\epsilon).
\end{aligned} \tag{18}$$

The ratio of the densities of learning $\mathbf{\Phi}^p$ using the adjacent data sets $(\vec{x}, \vec{y})$ and $(\vec{x}', \vec{y}')$ is given by

$$\frac{P(\mathbf{\Phi}^p | \vec{x}, \vec{y})}{P(\mathbf{\Phi}^p | \vec{x}', \vec{y}')} = \frac{P(\mathbf{b} = \mathbf{b}_1)}{P(\mathbf{b} = \mathbf{b}_2)} \frac{|\det(\mathcal{J}(\mathbf{\Phi}^p \to \mathbf{b}_1 | \vec{x}, \vec{y}))|^{-1}}{|\det(\mathcal{J}(\mathbf{\Phi}^p \to \mathbf{b}_2 | \vec{x}', \vec{y}'))|^{-1}}, \tag{19}$$

where $\mathcal{J}(\mathbf{\Phi}^p \to \mathbf{b}_1 | \vec{x}, \vec{y})$ and $\mathcal{J}(\mathbf{\Phi}^p \to \mathbf{b}_2 | \vec{x}', \vec{y}')$ are the Jacobian matrices of the bijective mappings from $\mathbf{\Phi}^p$ to $\mathbf{b}_1$ and $\mathbf{b}_2$, respectively. In Lemma A.3 available in the online supplemental material, we show that the ratio of the Jacobian determinants is upper bounded by $\exp(k) = 1 + \frac{1}{n\lambda}$, which is constant in terms of the classifier $\mathbf{\Phi}^p$ and the data set $(\vec{x}, \vec{y})$. The proof of Lemma A.3 is similar to [16, Theorem 9].

By substituting this result into (19), the ratio of the densities of learning $\mathbf{\Phi}^p$ using the adjacent data sets becomes

$$\frac{P(\mathbf{\Phi}^p | \vec{x}, \vec{y})}{P(\mathbf{\Phi}^p | \vec{x}', \vec{y}')} \leq \exp(\epsilon + k) = \exp(\epsilon'). \tag{20}$$

Similarly, we can show that the probability ratio is lower bounded by $\exp(-\epsilon')$, which together with (20) satisfies the definition of differential privacy. □

5.2 Analysis of Excess Error

In this section, we bound the error on the differentially private classifier as compared to the original nonprivate classifier. We treat the case of a single Gaussian per class for simplicity, however this analysis can be naturally extended to the case of a mixture of Gaussians per class. In the remainder of this section, we denote the terms $J(\mathbf{\Phi}, \mathbf{x}, \mathbf{y})$ and $L(\mathbf{\Phi}, \mathbf{x}, \mathbf{y})$ by $J(\mathbf{\Phi})$ and $L(\mathbf{\Phi})$, respectively, for conciseness. The objective function $J(\mathbf{\Phi})$ contains the loss function $L(\mathbf{\Phi})$ computed over the training data $(\mathbf{x}, \mathbf{y})$ and the regularization term $N(\mathbf{\Phi})$—this is known as the regularized *empirical risk* of the classifier $\mathbf{\Phi}$. In the following theorem, we establish a bound on the regularized empirical excess risk of the differentially private classifier minimizing the perturbed objective function $J_p(\mathbf{\Phi})$ over the classifier minimizing the unperturbed objective function $J(\mathbf{\Phi})$.

Theorem 5.2. *With probability at least $1 - \delta$, the regularized empirical excess risk of the classifier $\mathbf{\Phi}^p$ minimizing the perturbed objective function $J_p(\mathbf{\Phi})$ over the classifier $\mathbf{\Phi}^*$ minimizing the unperturbed objective function $J(\mathbf{\Phi})$ is bounded as*

$$J(\mathbf{\Phi}^p) \leq J(\mathbf{\Phi}^*) + \frac{8(d+1)^4 C}{\epsilon^2 \lambda} \log^2\left(\frac{d}{\delta}\right).$$

Proof. We use the definition of

$$J_p(\mathbf{\Phi}) = J(\mathbf{\Phi}) + \sum_c \sum_{ij} b_{ij} \Phi_{cij}$$

and the optimality of $\mathbf{\Phi}^p$, i.e., $J_p(\mathbf{\Phi}^p) \leq J_p(\mathbf{\Phi}^*)$

$$\begin{aligned}
J(\mathbf{\Phi}^p) + \sum_c \sum_{ij} b_{ij} \Phi^p_{cij} &\leq J(\mathbf{\Phi}^*) + \sum_c \sum_{ij} b_{ij} \Phi^*_{cij},\\
J(\mathbf{\Phi}^p) &\leq J(\mathbf{\Phi}^*) + \sum_c \sum_{ij} b_{ij} \left(\Phi^*_{cij} - \Phi^p_{cij}\right).
\end{aligned} \tag{21}$$

Using the strong convexity of $J(\mathbf{\Phi})$ and the optimality of $J(\mathbf{\Phi}^*)$, we have

$$\begin{aligned}
J(\mathbf{\Phi}^*) &\leq J\left(\frac{\mathbf{\Phi}^p + \mathbf{\Phi}^*}{2}\right)\\
&\leq \frac{J(\mathbf{\Phi}^p) + J(\mathbf{\Phi}^*)}{2} - \frac{\lambda}{8} \sum_c \|\mathbf{\Phi}^*_c - \mathbf{\Phi}^p_c\|^2,\\
J(\mathbf{\Phi}^p) - J(\mathbf{\Phi}^*) &\geq \frac{\lambda}{4} \sum_c \|\mathbf{\Phi}^*_c - \mathbf{\Phi}^p_c\|^2.
\end{aligned} \tag{22}$$

Similarly, using the strong convexity of $J_p(\mathbf{\Phi})$ and the optimality of $J_p(\mathbf{\Phi}^p)$

$$\begin{aligned} J_p(\boldsymbol{\Phi}^p) &\le J_p\left(\frac{\boldsymbol{\Phi}^p+\boldsymbol{\Phi}^*}{2}\right) \\ &\le \frac{J_p(\boldsymbol{\Phi}^p)+J_p(\boldsymbol{\Phi}^*)}{2} - \frac{\lambda}{8}\sum_c \left\|\boldsymbol{\Phi}_c^p - \boldsymbol{\Phi}_c^*\right\|^2, \\ J_p(\boldsymbol{\Phi}^*) - J_p(\boldsymbol{\Phi}^p) &\ge \frac{\lambda}{4}\sum_c \left\|\boldsymbol{\Phi}_c^p - \boldsymbol{\Phi}_c^*\right\|^2. \end{aligned}$$

Substituting $J_p(\boldsymbol{\Phi}) = J(\boldsymbol{\Phi}) + \sum_c \sum_{ij} b_{ij}\Phi_{cij}$

$$\begin{aligned} J(\boldsymbol{\Phi}^*) + \sum_c\sum_{ij} b_{ij}\Phi_{cij}^* - J(\boldsymbol{\Phi}^p) - \sum_c\sum_{ij} b_{ij}\Phi_{cij}^p& \\ \ge \frac{\lambda}{4}\sum_c \left\|\boldsymbol{\Phi}_c^* - \boldsymbol{\Phi}_c^p\right\|^2&, \\ \sum_c\sum_{ij} b_{ij}(\Phi_{cij}^* - \Phi_{cij}^p) - (J(\boldsymbol{\Phi}^p) - J(\boldsymbol{\Phi}^*))& \\ \ge \frac{\lambda}{4}\sum_c \left\|\boldsymbol{\Phi}_c^* - \boldsymbol{\Phi}_c^p\right\|^2&. \end{aligned}$$

Substituting the lower bound on $J(\boldsymbol{\Phi}^p) - J(\boldsymbol{\Phi}^*)$ given by (22)

$$\begin{aligned} \sum_c\sum_{ij} b_{ij}\left(\Phi_{cij}^* - \Phi_{cij}^p\right) &\ge \frac{\lambda}{2}\sum_c \left\|\boldsymbol{\Phi}_c^* - \boldsymbol{\Phi}_c^p\right\|^2, \\ \left[\sum_c\sum_{ij} b_{ij}\left(\Phi_{cij}^* - \Phi_{cij}^p\right)\right]^2 &\ge \frac{\lambda^2}{4}\left[\sum_c \left\|\boldsymbol{\Phi}_c^* - \boldsymbol{\Phi}_c^p\right\|^2\right]^2. \end{aligned} \tag{23}$$

Using the Cauchy-Schwarz inequality, we have

$$\left[\sum_c\sum_{ij} b_{ij}\left(\Phi_{cij}^* - \Phi_{cij}^p\right)\right]^2 \le C\|\mathbf{b}\|^2 \sum_c \left\|\boldsymbol{\Phi}_c^* - \boldsymbol{\Phi}_c^p\right\|^2. \tag{24}$$

Combining this with (23) gives us

$$\begin{aligned} C\|\mathbf{b}\|^2 \sum_c \left\|\boldsymbol{\Phi}_c^* - \boldsymbol{\Phi}_c^p\right\|^2 &\ge \frac{\lambda^2}{4}\left[\sum_c \left\|\boldsymbol{\Phi}_c^* - \boldsymbol{\Phi}_c^p\right\|^2\right]^2, \\ \sum_c \left\|\boldsymbol{\Phi}_c^* - \boldsymbol{\Phi}_c^p\right\|^2 &\le \frac{4C}{\lambda^2}\|\mathbf{b}\|^2. \end{aligned} \tag{25}$$

Combining this with (24) gives us

$$\sum_c\sum_{ij} b_{ij}\left(\Phi_{cij}^* - \Phi_{cij}^p\right) \le \frac{2C}{\lambda}\|\mathbf{b}\|^2.$$

We bound $\|\mathbf{b}\|^2$ with probability at least $1-\delta$ as given by Lemma A.6 available in the online supplemental material.

$$\sum_c\sum_{ij} b_{ij}\left(\Phi_{cij}^* - \Phi_{cij}^p\right) \le \frac{8(d+1)^4 C}{\epsilon^2\lambda}\log^2\left(\frac{d}{\delta}\right). \tag{26}$$

Substituting this in (21) proves the theorem. □

The upper bound on the regularized empirical risk is in $O(\frac{C}{\epsilon^2})$. The bound increases for smaller values of ϵ which implies tighter privacy and therefore suggests a tradeoff between privacy and utility.

The regularized empirical risk of a classifier is calculated over a given training data set. In practice, we are more interested in how the classifier will perform on new test data which are assumed to be generated from the same source as the training data. The expected value of the loss function computed over the data is called the *true risk* $\tilde{J}(\boldsymbol{\Phi}) = \mathbb{E}[J(\boldsymbol{\Phi})]$ of the classifier $\boldsymbol{\Phi}$. In the following theorem, we establish a bound on the true excess risk of the differentially private classifier minimizing the perturbed objective function and the classifier minimizing the original objective function.

Theorem 5.3. *With probability at least $1-\delta$, the true excess risk of the classifier $\boldsymbol{\Phi}^p$ minimizing the perturbed objective function $J_p(\boldsymbol{\Phi})$ over the classifier $\boldsymbol{\Phi}^*$ minimizing the unperturbed objective function $J(\boldsymbol{\Phi})$ is bounded as*

$$\begin{aligned} \tilde{J}(\boldsymbol{\Phi}^p) \le \tilde{J}(\boldsymbol{\Phi}^*) &+ \frac{8(d+1)^4 C}{\epsilon^2\lambda}\log^2\left(\frac{d}{\delta}\right) \\ &+ \frac{16}{\lambda n}\left[32 + \log\left(\frac{1}{\delta}\right)\right]. \end{aligned}$$

Proof. Let $\boldsymbol{\Phi}^r$ be the classifier minimizing $\tilde{J}(\boldsymbol{\Phi})$, i.e., $\tilde{J}(\boldsymbol{\Phi}^r) \le \tilde{J}(\boldsymbol{\Phi}^*)$

Rearranging the terms, we have

$$\begin{aligned} \tilde{J}(\boldsymbol{\Phi}^p) &= \tilde{J}(\boldsymbol{\Phi}^*) + [\tilde{J}(\boldsymbol{\Phi}^p) - \tilde{J}(\boldsymbol{\Phi}^r)] + [\tilde{J}(\boldsymbol{\Phi}^r) - \tilde{J}(\boldsymbol{\Phi}^*)] \\ &\le \tilde{J}(\boldsymbol{\Phi}^*) + [\tilde{J}(\boldsymbol{\Phi}^p) - \tilde{J}(\boldsymbol{\Phi}^r)]. \end{aligned} \tag{27}$$

Sridharan et al. [17] present a bound on the true excess risk of any classifier as an expression of the bound on the regularized empirical excess risk for that classifier. With probability at least $1-\delta$

$$\tilde{J}(\boldsymbol{\Phi}^p) - \tilde{J}(\boldsymbol{\Phi}^r) \le 2[J(\boldsymbol{\Phi}^p) - J(\boldsymbol{\Phi}^*)] + \frac{16}{\lambda n}\left[32 + \log\left(\frac{1}{\delta}\right)\right].$$

Substituting the bound from Theorem 5.2

$$\tilde{J}(\boldsymbol{\Phi}^p) - \tilde{J}(\boldsymbol{\Phi}^r) \le \frac{8(d+1)^4 C}{\epsilon^2\lambda}\log^2\left(\frac{d}{\delta}\right) + \frac{16}{\lambda n}\left[32 + \log\left(\frac{1}{\delta}\right)\right]. \tag{28}$$

Substituting this result into (27) proves the theorem. □

Similar to the bound on the regularized empirical excess risk, the bound on the true excess risk is also inversely proportional to ϵ^2 reflecting the tradeoff between privacy and utility. The bound is linear in the number of classes C, which is a consequence of the multiclass classification. The classifier learned using a higher value of the regularization parameter λ will have a higher covariance for each class ellipsoid. This would also make the classifier less sensitive to the perturbation. This intuition is confirmed by the fact that the true excess risk bound is inversely proportional to λ.

5.3 Experiments

We analyze the differentially private large margin Gaussian mixture model classifier to empirically quantify the error introduced by the perturbation. We implemented the classifier using the `CVX` convex program solver [18]. We report the results on the experiments with the UCI Breast Cancer data set [19] consisting of binary labeled 683 data instances with 10 features. We split the data randomly into 583 instances for the training data set and 100 instances for the test data set.

We trained the classifier with the different random samples of the perturbation term b, each sampled with the

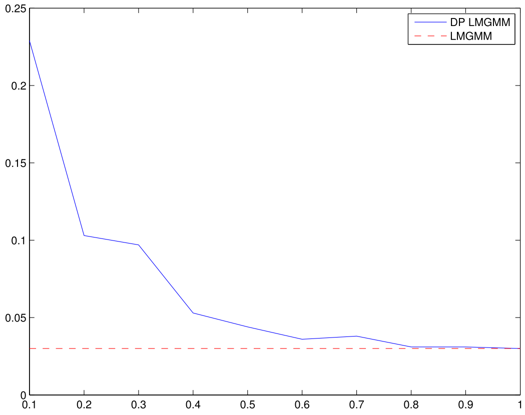

Fig. 2. Test error versus ϵ for the UCI breast cancer data set.

increasing values of ϵ, and the regularization parameter $\lambda = 0.31$ which is obtained via cross validation. The test error results averaged over 10 runs are shown in Fig. 2.

The dashed line represents the test error of the nonprivate classifier which remains constant with ϵ. We observe that for small value of ϵ implying tighter privacy constraints, we observe a higher error. By increasing ϵ, we see that the error steadily decreases and converges to the test error of the nonprivate classifier.

6 Conclusion

In this paper, we present a discriminatively trained Gaussian mixture model-based classification algorithm that satisfies differential privacy. Our proposed technique involves adding a perturbation term to the objective function. We prove that the proposed algorithm satisfies differential privacy and establish a bound on the excess risk of the classifier learned by the algorithm which is directly proportional to the number of classes and inversely proportional to the privacy parameter ϵ reflecting a tradeoff between privacy and utility.

In the future, we plan to extend this work along two main directions: extending our perturbation technique for a general class of learning algorithms and applying results from theory of large margin classifiers to arrive at tighter excess risk bounds for the differentially private large margin classifiers. Our intuition is that compared to other classification algorithms, a large margin classifier should be much more robust to perturbation. This would also give us insights into designing low error inducing mechanisms for differentially private classifiers.

Acknowledgments

A preliminary version of this paper appeared as [1].

References

[1] M. Pathak and B. Raj, "Large Margin Multiclass Gaussian Classification with Differential Privacy," *Proc. ECML/PKDD Workshop Privacy and Security Issues in Data Mining and Machine Learning,* 2010.

[2] C. Dwork, "Differential Privacy," *Proc. Int'l Colloquium Automata, Languages and Programming,* 2006.

[3] K. Chaudhuri and C. Monteleoni, "Privacy-Preserving Logistic Regression," *Proc. Neural Information Processing Systems (NIPS),* pp. 289-296, 2008.

[4] G. McLachlan and D. Peel, *Finite Mixture Models,* Wiley Series in Probability and Statistics. Wiley-Interscience, 2000.

[5] F. Sha and L.K. Saul, "Large Margin Gaussian Mixture Modeling for Phonetic Classification and Recognition," *Proc. IEEE Int'l Conf. Acoustics, Speech and Signal Processing (ICASSP),* pp. 265-268, 2006.

[6] I. Dinur and K. Nissim, "Revealing Information while Preserving Privacy," *Proc. Symp. Principles of Database Systems,* 2003.

[7] C. Dwork and K. Nissim, "Privacy-Preserving Datamining on Vertically Partitioned Databases," *Proc. 24th Ann. Int'l Cryptology Conf. (CRYPTO),* 2004.

[8] A. Blum, C. Dwork, F. McSherry, and K. Nissim, "Practical Privacy: The suLQ Framework," *Proc. Symp. Principles of Database Systems,* 2005.

[9] B. Barak, K. Chaudhuri, C. Dwork, S. Kale, F. McSherry, and K. Talwar, "Privacy, Accuracy, and Consistency Too: A Holistic Solution to Contingency Table Release," *Proc. Symp. Principles of Database Systems,* pp. 273-282, 2007.

[10] S.P. Kasiviswanathan, H.K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith, "What Can We Learn Privately?," *Proc. IEEE Symp. Foundations of Computer Science (FOCS),* pp. 531-540, 2008.

[11] G. Jagannathan, K. Pillaipakkamnatt, and R.N. Wright, "A Practical Differentially Private Random Decision Tree Classifier," *Proc. ICDM Workshop Privacy Aspects of Data Mining,* pp. 114-121, 2009.

[12] F. Sha and L.K. Saul, "Large Margin Hidden Markov Models for Automatic Speech Recognition," *Proc. Neural Information Processing Systems (NIPS),* pp. 1249-1256, 2007.

[13] P.C. Mahalanobis, "On the Generalised Distance in Statistics," *Proc. the Nat'l Inst. of Sciences of India,* vol. 2, pp. 49-55, 1936.

[14] L. Vandenberghe and S. Boyd, "Semidefinite Programming," *SIAM Rev.,* vol. 38, pp. 49-95, 1996.

[15] O. Chapelle, "Training a Support Vector Machine in the Primal," *Neural Computation,* vol. 19, no. 5, pp. 1155-1178, 2007.

[16] K. Chaudhuri, C. Monteleoni, and A.D. Sarwate, "Differentially Private Empirical Risk Minimization," *J. Machine Learning Research,* vol. 12, pp. 1069-1109, 2011.

[17] K. Sridharan, S. Shalev-Shwartz, and N. Srebro, "Fast Rates for Regularized Objectives," *Proc. Neural Information Processing Systems (NIPS),* pp. 1545-1552, 2008.

[18] M. Grant and S. Boyd, "CVX: Matlab Software for Disciplined Convex Programming, Version 1.21," http://cvxr.com/cvx, 2010.

[19] A. Frank and A. Asuncion, "UCI Machine Learning Repository," http://archive.ics.uci.edu/ml, 2010.

Manas A. Pathak received the BTech degree in computer science from Visvesvaraya National Institute of Technology, Nagpur, India, in 2006, the MS degree from Carnegie Mellon University (CMU) in 2009, and is currently working toward the PhD degree in the Language Technologies Institute at Carnegie Mellon University. He has done internships at PARC, MERL, IBM Research, and has nine papers published in various conferences and journals and two patents pending. His research interests include intersection of data privacy, machine learning, speech processing.

Bhiksha Raj received the PhD degree from CMU in 2000 and was at Mistubishi Electric Research Laboratories from 2001-2008. He is an associate professor and nontenured faculty chair at Carnegie Mellon University's Language Technologies Institute, and also holds the position of an associate professor by courtesy in the Electrical and Computer Engineering Department at CMU. His chief research interests include robust automatic speech recognition, machine learning and associated topics. Since 2005 he has also investigated topic models for signal processing, particularly in the context of modeling, enhancing, and modifying speech signals, and has published several papers on the topic.