Collaborative PCA/DCA Learning Methods for Compressive Privacy

S.Y. Kung, Thee Chanyaswad, J. Morris Chang, and Peiyuan Wu

***Abstract*—In the internet era, the data being collected on consumers like us are growing exponentially and attacks on our privacy are becoming a real threat. To better assure our privacy, it is safer to let data owner control the data to be uploaded to the network, as opposed to taking chance with the data servers or the third parties. To this end, we propose a privacy-preserving technique, named Compressive Privacy (CP), to enable the data creator to compress data via collaborative learning, so that the compressed data uploaded onto the internet will be useful only for the intended utility and will not be easily diverted to malicious applications.**

For data in a high-dimensional feature vector space, a common approach to data compression is dimension reduction or, equivalently, subspace projection. The most prominent tool is Principal Component Analysis (PCA). For unsupervised learning, PCA can best recover the original data given a specific reduced dimensionality. However, for supervised learning environment, it is more effective to adopt a supervised PCA, known as the Discriminant Component Analysis (DCA), in order to maximize the discriminant capability.

The DCA subspace analysis embraces two different subspaces. The signal subspace components of DCA are associated with the discriminant distance/power (related to the classification effectiveness), while the noise subspace components of DCA are tightly coupled with the recoverability and/or privacy protection. This paper will present three DCA-related data compression methods useful for privacy-preserving applications.

- **Utility-driven DCA: Because the rank of the signal subspace is limited by the number of classes, DCA can effectively support classification using a relatively small dimensionality (i.e. high compression).**
- **Desensitized PCA: By incorporating a signal-subspace ridge into DCA, it leads to a variant especially effective for extracting privacy-preserving components. In this case, the eigenvalues of the noise-space are made to become insensitive to the privacy labels and are ordered according to their corresponding component powers.**
- **Desensitized K-means/SOM: Since the revelation of the K-means or SOM cluster structure could leak sensitive information, it will be safer perform K-means or SOM clustering on desensitized PCA subspace.**

I. Introduction

We have all grown to become dependent upon the internet and the cloud for their ubiquitous data processing services, i.e. at any time, anywhere, and for anyone. With its packet switching, bandwidth, storage, and processing capacities, the data center nowadays manages the server farm, supports extensive database, and is ready to support, on demand from clients, a variable number of machines. However, the main problem of cloud computing lies on the privacy protection. With the rapidly growing internet commerce, many of our daily activities are moving online; abundance of personal information (such as sale transactions) is being collected, stored, and circulated around the internet and cloud servers, often without the owner's knowledge. This raises concerns on the protection of sensitive and private data, known as "Online Privacy" or "Internet Privacy".

S.Y. Kung and Thee Chanyaswad are with the Princeton University, J. Morris Chang is with the Iowa State University, and Peiyuan Wu is with the Taiwan Semiconductor Manufacturing Company Limited (TSMC).

Privacy-preserving data mining and machine learning have recently become an active research field, particularly because of the advancement in internet data circulation and modern digital processing hardware/software technologies. Privacy protection can be regarded as a special technical area in the field of pattern recognition. Research and development on privacy preservation have focused on two separate fronts: one covering the theoretical aspect of machine learning for privacy protection and the other covering system design and deployment issues of privacy protection systems.

From the privacy perspective, the encryption/accessibility of data is divided into two worlds, cf. Figure 1: (1) *private sphere*: where data owners generate and process the decrypted data; and (2) *public sphere*: where cloud servers can generally access only the encrypted data, except the trusted authorities who are allowed to access the decrypted data confidentially. In this setting, however, the data become vulnerable to unauthorized leakage.

Data owner should have control over data privacy. It is safer to let data owner control the data privacy and not to take chance with the cloud servers. To achieve this goal, we must provide some owner-controlled tools to safeguard private information against intrusion. New technologies are needed to better assure that personal data uploaded to the cloud will not be diverted for malicious applications.

Compressive Privacy (CP) enables the data creator to "encrypt" data using compressive-and-lossy transformation, and hence, protects user's personal privacy while delivering the intended (classification) capability. The objective of CP is to learn what kind of compressed data may enable classification/recognition of, say, face or speech data, while concealing the original face images or speech contents from malicious attackers. For example, in an emergency such as a bomb threat, many mobile images from various sources may be voluntarily pushed to the command center for wide-scale forensic analysis. CP may be used to compute the dimension-reduced feature subspace which can (1) effectively identify the suspect(s) and (2) adequately obfuscate the face images of the innocent.

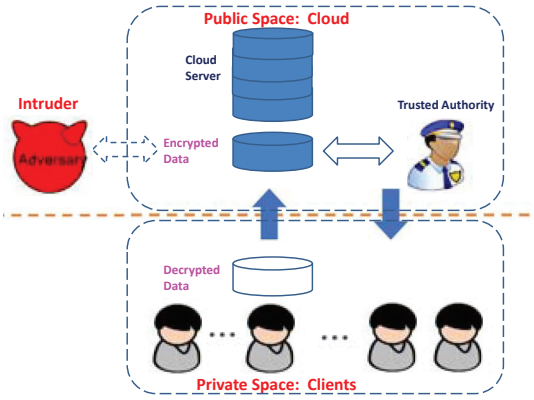


Fig. 1. From the privacy perspective, the encryption/accessibility of data is divided into two worlds: (1) *private sphere*: where data owners generate and process decrypted data; and (2) *public sphere*: where cloud servers can generally access only encrypted data, except the trusted authorities who are allowed to access decrypted data confidentially.

II. Compressive Privacy on Collaborative Machine Learning for Privacy Protection

Machine learning research embraces theories and technologies for modeling/learning a data mining system model based on the training dataset. The main function of machine learning is to convert the wealth of training data into useful knowledge by learning. The learned system is expected to be able to generalize and correctly classify, predict, or identify new input data that are previously unknown.

Collaborative learning is a method for machine learning in which users supply feature vectors to a cloud in order to collaboratively train a feature extractor and/or a classifier. In collaborative learning, the cloud aggregates samples from multiple users. Since the cloud is untrusted, the users are advised to perturb/compress their feature vectors before sending them to the cloud.

A typical Compressive Privacy on collaborative learning system is described as follows:

- **On the cloud side:** Since the cloud does not have access to original samples and, due to the lossy nature of CP methods, it cannot reconstruct recognizable face or intelligible speech, so the privacy of the participants will be protected. The reconstructed samples or the dimension-reduced feature spaces may be used for training a classifier.
- **On the owner side:** Via dimension reduction, some components are purposefully removed from the original vectors so that the original feature vectors are not easily reconstructible by others. The owner produces perturbed or dimension reduced data based on the projection matrix provided by the server.

Supervised vs. Unsupervised Learning. Collaborative learning allows supervised and unsupervised machine learning techniques to learn from the public vectors collected by the cloud servers. We shall adopt a PCA and Discriminant Component Analysis (DCA) for creating dimension-reduced subspaces useful for privacy protection in collaborative learning environments. Via PCA or DCA, individual data can be highly compressed before being uploaded to the cloud, which results in better privacy protection.

- For unsupervised learning, *principal component analysis* (PCA) is the most prominent subspace projection method. PCA is meant for mapping the originally high-dimensional (and unsupervised) training data to their low-dimensional representations.
- For supervised learning, we shall introduce the notion of Discriminant Component Analysis (DCA), an extension of PCA, to effectively exploit the known class labels accompanied with supervised training datasets.

III. Principal Component Analysis (PCA)

In unsupervised machine learning applications, the training dataset is usually a set of vectors: $\mathcal{X} = \{\ \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\ \}$, where $\mathbf{x}_i \in \mathbb{R}^M$, presumably generated under a certain underlying statistics which is unknown to the user. Pursuant to the zero-mean statistical model, it is common to first have the original vectors "center-adjusted" by its mean-value $\overrightarrow{\boldsymbol{\mu}} = \frac{\sum_{i=1}^{N} \mathbf{x}_i}{N}$, resulting in $\bar{\mathbf{x}}_i = \mathbf{x}_i - \overrightarrow{\boldsymbol{\mu}}, i = 1, \cdots, N$. This leads to a "center-adjusted" data matrix denoted as: $\bar{\mathbf{X}} = [\bar{\mathbf{x}}_1\ \bar{\mathbf{x}}_2 \cdots \bar{\mathbf{x}}_N]$. Based on $\bar{\mathbf{X}}$, a "center-adjusted" *scatter matrix* [3] may be derived as follows:

$$\bar{\mathbf{S}} \equiv \bar{\mathbf{X}}\bar{\mathbf{X}}^T = \sum_{i=1}^{N} [\mathbf{x}_i - \overrightarrow{\boldsymbol{\mu}}][\mathbf{x}_i - \overrightarrow{\boldsymbol{\mu}}]^T, \tag{1}$$

which assumes the role of the covariance matrix $\mathbf{R}$ in the estimation context. As such, denoting $\mathbf{v}_i \in \mathbb{R}^M$ as the i^{th} projection vector, its (normalized) component power is defined as

$$P(\mathbf{v}_i) \equiv \frac{\mathbf{v}_i^T \bar{\mathbf{S}} \mathbf{v}_i}{||\mathbf{v}_i||^2}, \quad i = 1, \cdots, m \tag{2}$$

A. PCA via Eigen-Decomposition of Scatter Matrix

The objective of PCA now becomes to find m ($m \leq M$) best components such that $\sum_{i=1}^{m} P(\mathbf{v}_i)$ is maximized, while $\mathbf{v}_i$ and $\mathbf{v}_j$ are orthogonal to each other if $i \neq j$.

In unsupervised learning scenarios, PCA is typically computed from the eigenvalue decomposition of $\bar{\mathbf{S}}$:

$$\bar{\mathbf{S}} = \mathbf{V}\ \boldsymbol{\Lambda}\ \mathbf{V}^{-1} = \mathbf{V}\ \boldsymbol{\Lambda}\ \mathbf{V}^T, \tag{3}$$

where $\boldsymbol{\Lambda}$ is a real-valued diagonal matrix (with decreasing eigenvalues) and $\mathbf{V}$ is a unitary matrix. It follows that The optimal PCA projection matrix can be derived from the m principal components of $\mathbf{V}$, i.e.

$$\mathbf{W}_{PCA} = \mathbf{V}_{major} = [\mathbf{v}_1\ \mathbf{v}_2 \cdots \mathbf{v}_m],$$

and the PCA-reduced feature vector can be represented by:

$$\mathbf{z} = \mathbf{W}_{PCA}^T \mathbf{x}. \tag{4}$$

B. Optimization of Power and Reconstruction Error

It is well known in the PCA literature that the mean-square-error criterion is equivalent to the maximum component power criterion. More exactly, PCA offers the optimal solution for both (1) maximum power and (2) minimal reconstruction error:

- **PCA's power associated with the principle eigenvectors: $\mathbf{V}_{major}$.** Note that λ_i equals to the power of the i-th component: $\lambda_i = P(\mathbf{v}_i)$. Consequently, the PCA solution yields the maximum total power:

$$\texttt{Max-Power} = \sum_{i=1}^{m} P(\mathbf{v}_i) = \sum_{i=1}^{m} \lambda_i. \tag{5}$$

- **PCA's reconstruction error (RE) associated with the minor eigenvectors: $\mathbf{V}_{minor}$.** Let the M-dimensional vector $\hat{\mathbf{x}}_{\mathbf{z}}$ denotes the best estimate of $\mathbf{x}$ from the m-dimensional vector $\mathbf{z}$. By PCA, $\hat{\mathbf{x}}_{\mathbf{z}} = \mathbf{W}^T\mathbf{x}$. It is well known that PCA also offers an optimal solution under the **mean-square-error** criterion:

$$\min_{\mathbf{z}\in\mathbb{R}^m} E\left[\|\mathbf{x} - \hat{\mathbf{x}}_{\mathbf{z}}\|^2\right] \tag{6}$$

where $E[\cdot]$ denotes the expected value. In unsupervised machine learning, it is a common practice to replace the covariance matrix $\mathbf{R}$ by the scatter matrix $\bar{\mathbf{S}}$. This leads to the following "Reconstruction Error"(RE):

$$\texttt{RE} \quad = \quad \sum_{i=m+1}^{M} \lambda_i, \tag{7}$$

where $\mathbf{V}_{minor}$ is formed from the $M-m$ minor columns of the unitary matrix $\mathbf{V}$.

C. Simulation Results for PCA

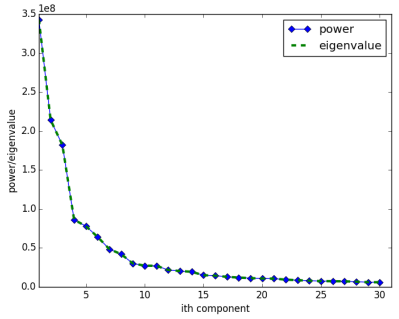
Example 1 (PCA for Privacy-preserving Face Recognition): Figure 2 shows the results from an experiment on the Yale face-image dataset. There are 165 samples (images) from 15 different classes (individuals) with 11 samples per class. Each image is 64-by-64 pixels, so the feature vectors derived from the pixel values have the dimension of 4096. To obtain the classification accuracies, one sample per class is chosen randomly to be left out for testing, so there are 15 testing samples per experiment. The other 150 samples are used for training. The experiment is repeated 30 times, which amounts to $30 \times 15 = 450$ testing samples in total. The average accuracies of the 450 testing samples are reported in Figure 2(b).

As displayed in Figure 2(a), (b), when the component eigenvalues gradually decrease, then so does the component power, further lowering the component classification accuracies. This indicates that the increased component power often implying high capacity to support the intended utility. Figure 2(c) depicts the (unsupervised) PCA eigenfaces for the face images in the Yale dataset.

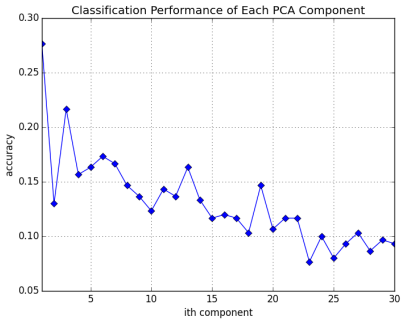
□

IV. Discriminant Component Analysis (DCA)

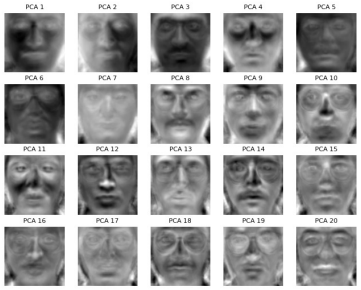
In supervised machine learning, a set of training data and their associated labels are provided to us:



(a)



(b)



(c)

Fig. 2. (a) For PCA, the component powers match exactly with the eigenvalues. (b) As the component powers decrease, the corresponding accuracies also decrease. (c) PCA eigenfaces: visualization of simulation results on the Yale dataset.

$$[\mathcal{X}, \mathcal{Y}] = \{ \quad [\mathbf{x}_1, y_1], [\mathbf{x}_2, y_2], \ldots, [\mathbf{x}_N, y_N] \quad \},$$

where the teacher values, denoted as y_i, represent the class labels of the corresponding training vectors.

A. Between-Class and Within-Class Scatter Matrices

In supervised learning, the *scatter matrix* $\bar{\mathbf{S}}$ can be further divided into two useful parts [2]:

$$\bar{\mathbf{S}} = \mathbf{S}_B + \mathbf{S}_W, \tag{8}$$

where the within-class scatter matrix $\mathbf{S}_W$ is defined as:

$$\mathbf{S}_W \quad = \quad \sum_{\ell=1}^{L} \sum_{j=1}^{N_\ell} [\mathbf{x}_j^{(\ell)} - \overrightarrow{\boldsymbol{\mu}}_\ell][\mathbf{x}_j^{(\ell)} - \overrightarrow{\boldsymbol{\mu}}_\ell]^T \tag{9}$$

where N_ℓ denoting the number of training vectors associated with the l-th class, $\overrightarrow{\boldsymbol{\mu}}_\ell$ denotes the centroid of the ℓ-th class, for $l = 1, \cdots, L$, and L denotes the number of different classes.

The between-class scatter matrix $\mathbf{S}_B$ is defined as:

$$\mathbf{S}_B = \frac{N}{2} \sum_{i=1}^{L} \sum_{j=1}^{L} f_{ij} \boldsymbol{\Delta}_{ij} \boldsymbol{\Delta}_{ij}^T, \tag{10}$$

where $f_{ij} = r_i r_j$, with $r_i \equiv \frac{N_i}{N}$, $r_j \equiv \frac{N_j}{N}$, stands for the relative frequency of involving classes i and j. Note that the greater the magnitude of $\boldsymbol{\Delta}_{ij}$, where $\boldsymbol{\Delta}_{ij} \equiv [\overrightarrow{\boldsymbol{\mu}}_i - \overrightarrow{\boldsymbol{\mu}}_j]$, the more distinguishable between the i^{th} and j^{th} classes. It is why $\mathbf{S}_B$ is also called a *Signal Matrix*.

For supervised classification, the focus is placed on *discriminant power*. Naturally, it is preferable to have a far distance between two different classes. However, a large spread of the "within-class" data will have an adverse effect. In this sense, $\mathbf{S}_B$ and $\mathbf{S}_W$ have the very opposite roles:

- The noise matrix $\mathbf{S}_W$ now plays a derogatory role in the sense that a high directional noise power for $\mathbf{S}_W$ will weaken the *discriminant power* along the same direction.
- The signal matrix is represented by the between-class scatter matrix as it is formed from the best L class-discriminating vectors learnable from the dataset.

B. Linear Discriminant Analysis (LDA)

LDA focuses on an important special case when $m = 1$ and $L = 2$ and thus,

$$\mathbf{S}_B = \frac{N_1 N_2}{N} \boldsymbol{\Delta}_{12} \boldsymbol{\Delta}_{12}^T, \tag{11}$$

Furthermore, we adopt the following denotations:

- "*signal variance*", defined as $\mathbf{w}^T\ \mathbf{S}_B\ \mathbf{w}$, is proportional to the square of d_{12}, and
- "*noise variance*", defined as $\mathbf{w}^T\ \mathbf{S}_W\ \mathbf{w}$, represents the spread of the projected data of the same class around its centroid.

In the subsequent discussion, we shall simplify the term "*signal variance*" to "*signal*" and "*noise variance*" to "*noise*", respectively. *Linear Discriminant Analysis* (LDA) [1] aims at maximizing the signal-to-noise ratio, $\text{SNR} = \frac{\textbf{signal}}{\textbf{noise}}$. More exactly,

$$\mathbf{w}_{\text{LDA}} = \underset{\{\mathbf{w} \in \mathbb{R}^M\}}{\arg\max}\ SNR(\mathbf{w}) = \underset{\{\mathbf{w} \in \mathbb{R}^M\}}{\arg\max}\ \frac{\mathbf{w}^T\ \mathbf{S}_B\ \mathbf{w}}{\mathbf{w}^T\ \mathbf{S}_W\ \mathbf{w}}, \tag{12}$$

which was originally designed as a single component analysis for binary classification. Equivalently, due to Eq. 8, we have an alternative signal-power-ratio formulation:

$$\mathbf{w}_{\text{LDA}} = \underset{\{\mathbf{w} \in \mathbb{R}^M\}}{\arg\max}\ \text{SPR}(\mathbf{w}) = \underset{\{\mathbf{w} \in \mathbb{R}^M\}}{\arg\max}\ \frac{\mathbf{w}^T\ \mathbf{S}_B\ \mathbf{w}}{\mathbf{w}^T\ \bar{\mathbf{S}}\ \mathbf{w}}, \tag{13}$$

where "SPR" stands for "signal-power-ratio".

C. Multiple Discriminant Component Analysis (MDCA)

In order to facilitate our exploration into an appropriate criterion for component analysis, we propose an optimization criterion, based on the sum of the sinal-noise-ratios pertaining to all the individual components:

$$\text{Sum of SPRs} = \sum_{i=1}^{m} \frac{s_i}{p_i} = \sum_{i=1}^{m} \frac{\mathbf{w}_i^T\ [\ \mathbf{S}_B\ ]\ \mathbf{w}_i}{\mathbf{w}_i^T\ \left[\ \bar{\mathbf{S}}\ \right]\ \mathbf{w}_i}.$$

Let the *Signal-Power-Ratio* (SPR) associated with the i-th component be defined as $\text{SPR}(\mathbf{w}_i) = \frac{s_i}{p_i}$, where

$$p_i = \mathbf{w}_i^T\ \mathbf{S}_B\ \mathbf{w}_i + \mathbf{w}_i^T\ \bar{\mathbf{S}}\ \mathbf{w}_i = s_i + n_i \text{ for } i = 1, \cdots, m$$

Thereafter, the (total) *discriminant power* is naturally defined as the sum of the individual SPR scores:

$$\text{SPR}(\mathbf{W}) \equiv \sum_{i=1}^{m} \text{SPR}(\mathbf{w}_i) = \sum_{i=1}^{m} \frac{\mathbf{w}_i^T\ \mathbf{S}_B\ \mathbf{w}_i}{\mathbf{w}_i^T\ \bar{\mathbf{S}}\ \mathbf{w}_i}\ . \tag{14}$$

In order to preserve the rotational invariance of $\text{SPR}(\mathbf{W})$, we must impose a "canonical orthonormality" constraint on the columns of $\mathbf{W}$ such that

$$\mathbf{W}^T \bar{\mathbf{S}} \mathbf{W} = \mathbf{I}. \tag{15}$$

It can be shown that [11], DCA is equivalent to PCA in the Canonical Vector Space (CVS). In fact, the component SPR in the original space is mathematically equivalent to the the component power in the CVS. The mapping from a vector $\mathbf{x}$ in the original space to its counterpart $\tilde{\mathbf{x}}$ in the "Canonical Vector Space" (CVS) is represented by $\tilde{\mathbf{x}} = \left[\bar{\mathbf{S}}\right]^{-\frac{1}{2}} \mathbf{x}$. As such, DCA may also be derived first as the PCA in CVS and transform the solution back to the original vector space.

Ridge for Numerical Robustness. We have previously assumed that $\bar{\mathbf{S}}$ is nonsingular. However, in practice, we must consider the situations (1) when $N < M$, then $\bar{\mathbf{S}}$ will be singular or (2) when $\bar{\mathbf{S}}$ is ill-conditioned. An effective remedy is to incorporate a ridge parameter ρ into the scatter matrix [9], [10], resulting the replacement of $\bar{\mathbf{S}}$ by:

$$\bar{\mathbf{S}}' = \bar{\mathbf{S}} + \rho \mathbf{I}.$$

In this case, the optimal solution can be derived as a projection matrix $\mathbf{W}^* \in M \times m$ such that [11]

$$\mathbf{W}_{DCA} = \underset{\{\mathbf{W}: \mathbf{W}^T \left[\ \bar{\mathbf{S}} + \rho \mathbf{I}\ \right] \mathbf{W} = \mathbf{I}\}}{\arg\max} \ \mathtt{tr}\left(\mathbf{W}^T\ [\ \mathbf{S}_B\ ]\ \mathbf{W}\right). \tag{16}$$

The DCA solution may be directly obtained from the first m principal eigenvectors of the regulated **Discriminant Matrix**:

$$\mathbf{D}_{DCA} \equiv \left[\bar{\mathbf{S}} + \rho \mathbf{I}\right]^{-1} \mathbf{S}_B \tag{17}$$

with the columns of the solution matrix $\mathbf{V}$ meeting the "canonical orthonormality" condition prescribed by Eq. 15. Numerically, the optimal DCA projection matrix can be de-

rived from the principal eigenvectors of[1]

$$\mathbf{eig}\left(\mathbf{S}_B\ , \bar{\mathbf{S}} + \rho\mathbf{I}\right) \tag{18}$$

It follows that the (dimension-reduced) DCA representation is

$$\mathbf{z} = \mathbf{W}_{DCA}^T \mathbf{x}. \tag{19}$$

D. Ranking of Signal-Subspace Components

The DCA eigenspace $\mathbf{V}$ is primarily spanned by $\mathbf{V}_{major} \in \mathbb{R}^{M\times(L-1)}$. Thus, the primary focus of DCA is placed on maximizing the SPR-type utility function via adapting $\mathbf{V}_{major}$. More exactly, in the eigen-transformed vectors space, the *Modified SPR* (SPR$'$) is exactly the same as its corresponding eigenvalue, for any positive integer i, i.e.

$$\lambda_i = \frac{\mathbf{v}_i^T\ \mathbf{S}_B\ \mathbf{v}_i}{\mathbf{v}_i^T\ \left[\bar{\mathbf{S}} + \rho\mathbf{I}\right]\mathbf{v}_i} = \text{SPR}_i' \tag{20}$$

After the modification, the total SPR$'$ is

$$\text{SPR}'(\mathbf{W}_{DCA}) = \sum_{i=1}^{m} \text{SPR}_i' = \sum_{i=1}^{m} \lambda_i. \tag{21}$$

Note also that there are only $L-1$ nonzero eigenvalues. As such, we can extract at most $L-1$ useful components for now. For extraction of additional and useful components, see Section V.

E. Simulation Results: Utility-Driven Applications

To best illustrate the idea, let us provide two examples: (1) an illustrative *Double-Income Problem* (DIP) and (2) a privacy-preserving face recognition (PPFR) problem based on the Yale face dataset. [12]

Example 2 (PCA/DCA for Double-Income Problem):

In the Double-Income Problem (DIP) training dataset, each family is represented by a four-dimensional feature vector. The first two features, x_1 and x_2, are the two individual incomes of a couple. Suppose that a query is intended for assessing the couple's total income, i.e. $\mathbf{u} = \mathbf{u}(\mathbf{x}) = x_1\ +\ x_2$, i.e. the financial condition of the family. From the privacy perspective, the query should not pry into the income disparity within the family, i.e. the privacy function is set as: $\mathbf{p} = \mathbf{p}(\mathbf{x}) = x_1 - x_2$, i.e. who is the bread earner of the family. In our current DIP study, two other related features are also acquired, making it a four-dimensional vector $\mathbf{x} = [x_1\ x_2\ x_3\ x_4]^T$.

Suppose that we are given a training dataset: $\{\ \mathcal{X}\ \} =$

$$\left[\begin{array}{c}11\\7\\1\\2\end{array}\right]\left[\begin{array}{c}18\\8\\2\\-1\end{array}\right]\left[\begin{array}{c}17\\5\\-1\\-1\end{array}\right]\left[\begin{array}{c}4\\10\\-1\\-4\end{array}\right]\left[\begin{array}{c}5\\6\\2\\2\end{array}\right]\left[\begin{array}{c}4\\7\\1\\-1\end{array}\right]\left[\begin{array}{c}1\\2\\-1\\1\end{array}\right]\left[\begin{array}{c}4\\1\\1\\-1\end{array}\right],$$

with the utility/privacy teacher labels, denoted by $\{\ \mathcal{Y}\ \} =$

$$\left[\begin{array}{c}H\\+\end{array}\right]\left[\begin{array}{c}H\\+\end{array}\right]\left[\begin{array}{c}M\\+\end{array}\right]\left[\begin{array}{c}M\\-\end{array}\right]\left[\begin{array}{c}M\\-\end{array}\right]\left[\begin{array}{c}M\\-\end{array}\right]\left[\begin{array}{c}L\\-\end{array}\right]\left[\begin{array}{c}L\\+\end{array}\right]$$

where "H/M/L" denotes the three (High/Middle/Low) *utility classes* (i.e. family income) and "+/-" denotes the two *privacy classes* (i.e. who-earns-more between the couple).

PCA: We first compute the the scatter matrix:

$$\bar{\mathbf{S}} = \left[\begin{array}{cccc}296 & 42 & 11 & -2\\42 & 63.5 & 3 & -15.75\\11 & 3 & 12 & 7.5\\-2 & -15.75 & 7.5 & 27.875\end{array}\right]$$

This yields the following PCA eigenvalues or, equivalently, the eigen-component powers:

$$\{\lambda_1 = 303.87\ \ \lambda_2 = 62.82\ \ \lambda_3 = 25.25\ \ \lambda_4 = 7.44\}$$

The two principle eigenvectors are

$$\mathbf{f}_1 = \left[\begin{array}{c}0.984\\0.174\\0.039\\-0.016\end{array}\right] \text{ and } \mathbf{f}_2 = \left[\begin{array}{c}0.163\\-0.899\\0.042\\0.405\end{array}\right]$$

As demonstrated by Figure 3(a) and (b), with a query marked as "♡", we note that while PCA fails to identify the utility class (i.e. total income classification), neither does it leak private information (on income disparity).

DCA: For DCA, the utility-driven signal matrix, denoted as $\mathbf{S}_{B_U}$, can be learned from the training data and their respective utility labels (i.e. High/Middle/Low), via Eq. 10:

$$\mathbf{S}_{B_U} = \left[\begin{array}{cccc}146 & 67 & 19 & 8.5\\67 & 48.5 & 6.5 & -3.26\\19 & 6.5 & 2.75 & 2.00\\8.5 & -3.26 & 2.00 & 3.38\end{array}\right]$$

The ridge set for the scatter matrix is $\rho = 1$. The generalized eigen-decomposition of $\mathbf{eig}\left(\mathbf{S}_{B_U}\ , \bar{\mathbf{S}} + \rho\mathbf{I}\right)$ yields the following eigenvalues or, equivalently, the component SPR$'$:

$$\{\lambda_1 = 0.966\ \ \lambda_2 = 0.264\ \ \lambda_3 = 0\ \ \lambda_4 = 0\ \}$$

Thereafter, the two principle eigenvectors corresponding to the two nonzero eigenvalues are, cf. Eq. 18:

$$\mathbf{f}_1 = \left[\begin{array}{c}0.204\\0.839\\0.245\\0.443\end{array}\right] \text{ and } \mathbf{f}_2 = \left[\begin{array}{c}0.221\\-0.535\\0.733\\0.357\end{array}\right]$$

As demonstrated by Figure 3(c) and (d), with a query marked as "♡", DCA can confidently identify the utility label, but it also leaks sensitive information on the privacy label.

□

Example 3 (DCA for PPFR Applications): The experimental setup basically follows that of Example 1. The difference is that DCA is used in place of PCA here. DCA components are derived for the Yale dataset with $\rho = .02 \times max(eig(\bar{S}))$. Figure 4(a) shows that there are $L-1$ nonzero eigenvalues, pursuant closely to their corresponding SPR$'$. Thus, DCA can

[1]For DCA, all the eigenvalues are generically distinct, therefore, all the columns of $\mathbf{V}$ are canonically orthogonal to each other. [6]

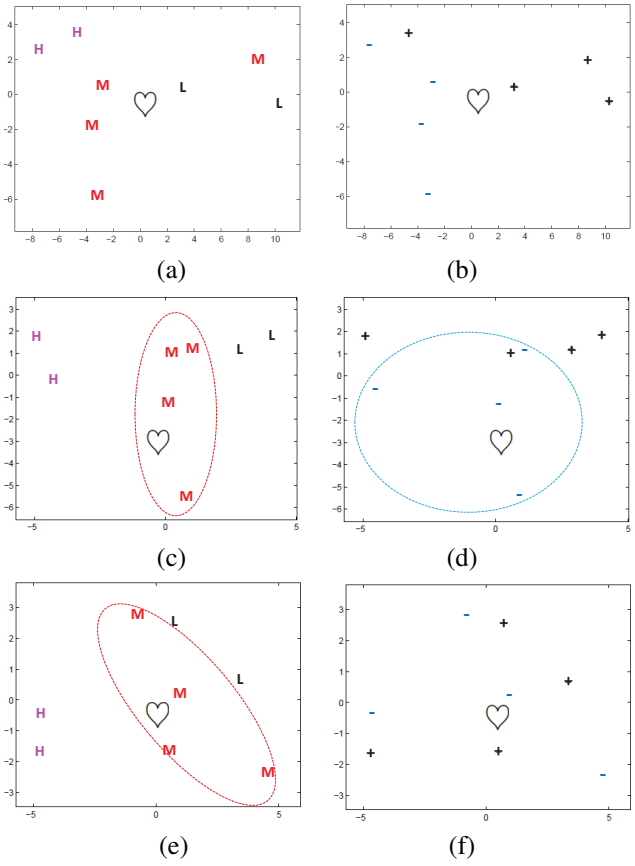


Fig. 3. Visualization of a query, marked as "♡", mapped to the optimal two-dimensional PCA, DCA, and Ridge DCA subspaces. The family income (utility) class can be confidently assessed as the "M"-class. (d) The privacy label on the income disparity remains clueless, as both classes ("+" or "-") have the equal claim. (a) PCA visualization with utility explicitly labeled; (b) PCA visualization with privacy explicitly labeled; (c) DCA visualization with utility explicitly labeled; (d) DCA visualization with privacy explicitly labeled; (e) Ridge DCA visualization with utility explicitly labeled; and (f) Ridge DCA visualization with privacy explicitly labeled. When the query may be identified with sufficient confidence, then a dashed ellipse(s) will be shown. On the other hand, no ellipse(s) will be shown when the association is deemed to be ambiguous. Based on the confident identification shown by the dashed ellipse(s), the learning results are summarized as follows: (1) While PCA fails to identify the utility class (i.e. total income classification), it leaks no private information (on income disparity); (2) DCA is the one which most effectively identifies the utility label, but it also leaks the privacy label; and (3) Ridge DCA is the only one which simultaneously identifies the utility label and protects the privacy label.

extract $L-1$ rank-ordered principal components to best serve the purpose of face recognition (FR). As shown in Figure 4(b), the first 14 eigen-components are most discriminative for face recognition, with per-component accuracy around 23%. In contrast, the next sixteen noise-subspace eigen-components (i.e. $15^{th}-30^{th}$) are basically noise ridden and carry little useful information, with a low accuracy around 6%, par the random guess. This implies that these noise-subspac components contain no useful information for face recognition (FR). □

V. Desensitized PCA via Ridge DCA

In the previous section, DCA is applied to utility-driven machine learning applications. Now we shall address a DCA variant tailored for privacy-driven principal component analysis, i.e. desensitized PCA. An exemplifying application scenario is the so-called *Anti-Recognition Utility Maximization* (ARUM), in which the the privacy intruder's objective is face recognition itself. As such, the goal of Compressive Privacy is to find a representation that may prevent the identity of the faces from being correctly classified. To this end, we first extract the desensitized PCAs and then apply either supervised classification, such as SVM [4], or unsupervised clustering, such as K-means or SOM [5]. The overall flow diagram of the desensitizing system structure is depicted in Figure 5.

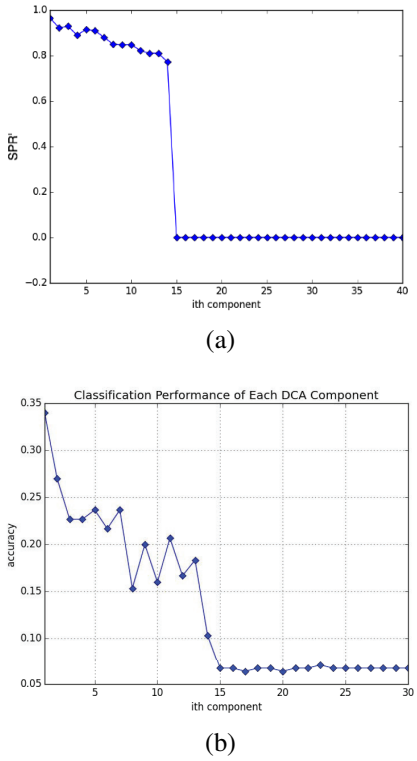


Fig. 4. For the signal-subspace component analysis. (a) The figure shows the SPR$'$ of the components of DCA. (DCA is equivalent to RDCA with $\rho'=0$.) (b) While each signal-subspace component yields a relatively higher accuracy around 23%, the 16 noise-subspace components yields a low accuracy around 6%, par the random guess. This implies that these components contain no useful information for face recognition (FR).

A. Incorporated A Negative Ridge into Signal Matrix

The Ridge DCA incorporates yet another ridge parameter ρ' to regulate the signal matrix, i.e. the between-class scatter matrix:

$$\mathbf{S}'_B = \mathbf{S}_B - \rho'\mathbf{I}.$$

The optimization formulation now searches for a projection matrix $\mathbf{W}^* \in M \times m$ such that

$$\mathbf{W}_{RDCA} = \underset{\{\mathbf{W}:\mathbf{W}^T\left[\ \bar{\mathbf{S}}+\rho\mathbf{I}\ \right]\mathbf{W}=\mathbf{I}\}}{\arg\max} \quad \mathtt{tr}\left(\mathbf{W}^T\left[\ \mathbf{S}_B-\rho'\mathbf{I}\ \right]\mathbf{W}\right). \quad (22)$$

Numerically, the optimal Ridge DCA solution can be derived

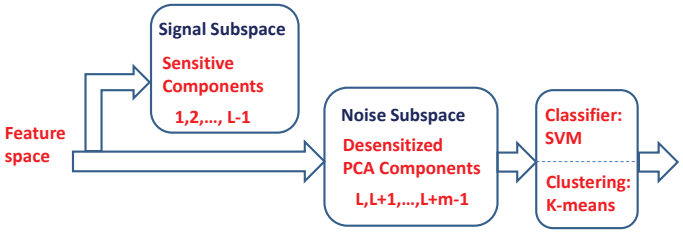


Fig. 5. The privacy-driven DCA system structure for "Desensitized PCA" (cf. Section V) and/or "Desensitized K-means" (cf. Section VI).

from the principal eigenvectors of

$$\mathbf{eig}\left(\mathbf{S}_B - \rho'\mathbf{I}\ ,\bar{\mathbf{S}} + \rho\mathbf{I}\right) \tag{23}$$

By slightly modifying Eq. 20, we obtain the following eigenvalue analysis:

$$\lambda_i = \ \mathrm{SPR}'_i - \frac{\rho'}{P(\mathbf{v}_i) + \rho} \tag{24}$$

B. Eigenvalues of Eigen-Components of Ridge DCA

Now let us elaborate the implication of Eq. 24:

- **Signal-Subspace Components, i.e. when $i < L$:** With a very small value of ρ', the eigenvalues for such eigen-components can be approximately expressed in terms of their corresponding SPR′ (SPR'_i):

$$\lambda_i \ \approx \ \mathrm{SPR}'_i \tag{25}$$

  For the ARUM application scenario, such eigen-components are potentially most intrusive and it is why they are filtered out in our desensitizing system shown in Figure 5.
- **Noise Subspace Components, i.e. when $i \geq L$:** By assuming an extremely small positive value of ρ', it can be shown that

$$\mathbf{v}_i^T\ \mathbf{S}_B\ \mathbf{v}_i \simeq 0, \quad \text{for all } i \geq L$$

  In this case, the corresponding eigenvalues (λ_i) and the component powers ($P(\mathbf{v}_i)$) are closely related as follows:

$$\lambda_i \ \approx -\frac{\rho'}{P(\mathbf{v}_i) + \rho}, \ \text{for } i \geq L. \tag{26}$$

  where the (normalized) component power, defined in Eq. 2. It implies that the eigen-component powers can be sorted by their corresponding eigenvalues:

$$P(\mathbf{v}_i) \ \approx -\frac{\rho'}{\lambda_i} - \rho, \ \text{for } i \geq L \tag{27}$$

  just like PCA. This is why the Ridge DCA is also named *Desensitized PCA*.

C. Simulation Results

Let us now illustrate the application of desensitized PCA by exploring two examples: (1) a toy application example on the double-income problem (DIP) and (2) Anti-Recognition Utility Maximization (ARUM) based on the Yale face dataset.

Example 4 (Ridge DCA for DIP):

Let us revisit the DIP example. For Ridge DCA, via Eq. 10, the privacy-driven signal matrix, denoted as $\mathbf{S}_{B_P}$, can be learned from the training data and their respective privacy labels (i.e. "+/-").

$$\mathbf{S}_{B_P} = \begin{bmatrix} 162 & -18 & 9 & 4.5 \\ -18 & 2 & -1 & -0.5 \\ 9 & -1 & 0.5 & 0.25 \\ 4.5 & -0.5 & 0.25 & 0.13 \end{bmatrix}.$$

The ridge for the scatter matrix is set as $\rho = 1$, and the ridge for the signal matrix is set as $\rho' = .01 * max(eig(\mathbf{S}_{B_P}))$.

The eigenvalues for the Ridge DCA can be computed from the generalized eigen-decomposition of $\mathbf{eig}\left(\mathbf{S}_{B_P} - \rho'\mathbf{I}\ ,\bar{\mathbf{S}} + \rho\mathbf{I}\right)$, yielding

$$\{\lambda_1 = 0.7293 \ \ \lambda_2 = -0.0201 \ \ \lambda_3 = -0.0611 \ \ \lambda_4 = -0.1885\}$$

According to Eq. 27, the (latter) three (decreasing) noise-component eigenvalues

$$\{\lambda_2 = -0.0201 \quad \lambda_3 = -0.0611 \quad \lambda_4 = -0.1885\}$$

correspond to the following (decreasing) component powers

$$\{P(\mathbf{v}_2) = 80.9 \quad P(\mathbf{v}_3) = 25.94 \quad P(\mathbf{v}_4) = 7.73\}$$

The two eigenvectors correspond to the highest component powers, i.e. $P(\mathbf{v}_2)$ and $P(\mathbf{v}_3)$, will be adopted as the two desensitized PCAs, cf. Eq. 23:

$$\mathbf{f}_1 = \begin{bmatrix} 0.107 \\ 0.941 \\ -0.027 \\ -0.320 \end{bmatrix} \ \text{and} \ \mathbf{f}_2 = \begin{bmatrix} -0.019 \\ 0.303 \\ 0.457 \\ 0.836 \end{bmatrix}$$

With reference to Figures 3(a), (b) and (c), we can summarize our simulation results as follows ("♡" represents the query):

- While PCA fails to identify the utility class (i.e. total income classification), it leaks no private information (on income disparity).
- DCA can most confidently identify the utility label, but it fails to safely protect the privacy label.
- Ridge DCA is the only one which simultaneously identifies the utility label and protects the privacy label.

□

For the conventional PPFR problem, DCA can be used to produce $L-1$ most discriminative components. Now let us consider an alternative application scenario, *Anti-Recognition Utility Maximization* (ARUM), which is in a sharp contrasting to the conventional PPFR application. Let us now discuss how to apply ridge DCA to ARUM problems. Briefly, the Ridge DCA starts with removing the first $L-1$ eigen-components, to desensitize the feature vectors, and subsequently sorts the

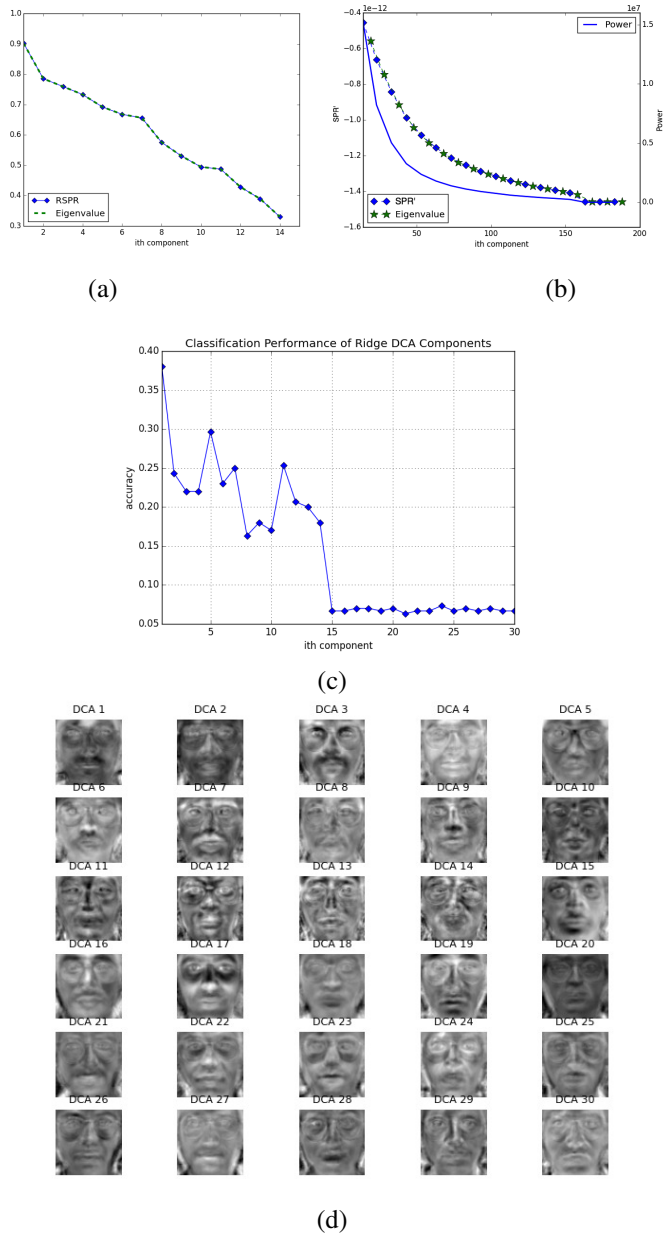


Fig. 6. Rank-ordered eigenvalues of RDCA, when $\rho' = 0.00001$, for (a) *Signal-subspace component analysis:* The figure (diamond vs dash line) confirms that the component's SPR$'$ is dictated by the eigenvalues in a way consistent with the theoretical prediction given in Eq. 25. (b) *Noise-subspace component analysis:* The figure confirms that the desensitized-PCA component power is a monotonic function of the eigenvalue as theoretically predicted in Eq. 27 (star vs solid line). Moreover, the component's SPR$'$ is dictated by the eigenvalues as predicted in Eq. 26 (diamond vs star). (c) Each of the 16 desensitized eigenfaces yields a low accuracy around 6%, par the random guess. (d) The first 14 principal DCA-eigenfaces are not very different from DCA. However, in a sharp contrast, the next 16 desensitized eigenfaces, representing the principal PCA components, are potentially more informative for the other intended utility.

remaining components based on their component powers, just like PCA.

Example 5 (Ridge DCA for ARUM):

For ARUM, our objective is to extract an optimal subspace which may prevent the person from being recognized by the compressed face image. By performing en experiment on Yale

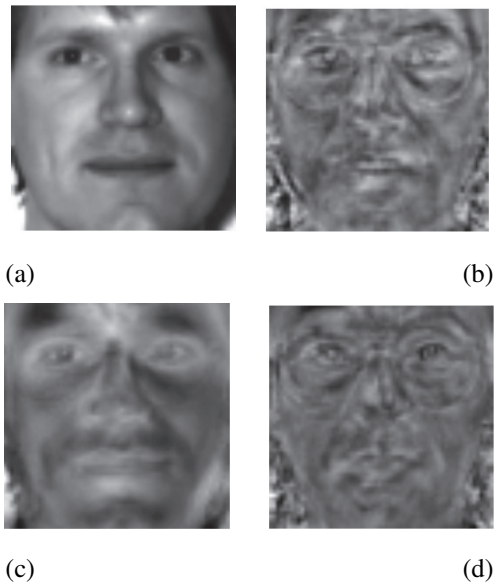


Fig. 7. Original and reconstructed face images using 160 dimensions from the Yale dataset. (a) The original face. (b) DCA with zero ridge (c) Ridge DCA with a negative ridge applied to the privacy-driven signal matrix. (d) Ridge DCA with reversely rank ordered eigen-components. (Courtesy from [12])

dataset in a similar setup to Example 1, the following results are achieved.

- Figure 6(a) confirms that the signal-subspace component's SPR$'$ is dictated by the eigenvalues in a way consistent with the theoretical prediction given in Eq. 25.
- Figure 6(b) confirms that the component powers of the desensitized-PCA components can be expressed in terms of their corresponding eigenvalues pursuant to Eq. 26.
- Figure 6(c) shows that, as theoretically predicted, each of the desensitized-PCA eigenfaces yields a low accuracy around 6%, par the random guess.
- Figure 6(d) shows that the first 14 principal DCA-eigenfaces should be cast away because they are the most privacy-intrusive. On the other hand, the desensitized components (from 15^{th} to 30^{th}) now have highest component powers, just like PCA, and so they may contain information possibly useful for the other utility function.

□

The following example provides a preliminary comparison of face reconstructions between PPFR versus ARUM. It may shed some light on how the desensitized PCA may facilitate privacy protection in the ARUM-type scenarios.

Example 6 (Face Reconstructions: PPFR vs. ARUM):

Figure 7(a) shows an original face from the Yale dataset. Figures 7(b) and (c) depict the reconstructed face image via DCA and Ridge DCA, respectively. Suppose that the intended utility is, say, to distinguish (1) smiling versus sad faces or (2) faces with versus without eyeglasses, then we observe (somewhat subjectively) that Figure 7(c) (with desensitized PCAs) compares favorably with (b) or (d). This suggests that, for ARUM, the desensitized PCA may indeed better facilitate utility maximization while offering the same privacy protection as DCA.

	Random guess (no training)	Before desensitization	After desensitization
Utility accuracy	0.500	0.983	0.955
Privacy accuracy	0.143	0.976	0.444

TABLE I
UTILITY AND PRIVACY ACCURACY PERFORMANCE OF DESENSITIZED PCA ON THE GLASSES DATASET.

An experiment on the in-house Glasses dataset, which is derived from Yale and Olivetti databases, confirms that desensitized PCA is effective for ARUM. The dataset consists of 50 samples (images) chosen from Yale and Olivetti databases such that each individual in the dataset has 50% of his/her images with glasses on and the other 50% without glasses on. There are images of seven individuals in the dataset, so the utility and privacy are defined as followed:

- Utility is the classification of whether the face wears glasses, so there are two utility classes. Obviously, higher classification accuracy means better utility gain.
- Privacy is person identification from the face image, i.e. face recognition. There are seven individuals in the dataset, so the number of privacy classes is seven. In this case, higher privacy accuracy means more privacy loss/leakage.

The experimental setup is the following. In each trial of the experiment, five samples are randomly chosen to be left out, while the other 45 are used to train the classifiers. Then, one of the five left-out samples is chosen randomly for testing. The performance of the classification on the data both before and after PCA desensitization is collected for comparison, and the experiment is repeated for 1000 trials. This experiment is specifically conducted for identifying whether the face wears glasses or not for utility, and identifying the person among possible seven individuals for privacy. SVM is used as the classifier.

Table I summarizes the results from the experiment. Briefly, among the 1000 trials:

- In terms of utility, the trained classifiers correctly predict glasses classes 983 versys 955 times before and after desensitization, respectively.
- In terms of privacy, the trained classifiers correctly predict the person's identification 976 versus 444 times before and after desensitization, respectively.

The results show that our desensitization has substantially reduced the privacy accuracy from 97.6% to 44.4%, while compromising the utility accuracy only by 2.8% down from 98.3% to 95.5%. This suggests that the desensitized PCA is promising for ARUM-type appications.

□

VI. DESENSITIZED K-MEANS VIA RIDGE DCA

Note that K-means (or SOM) by itself has a natural role for privacy preservation. By substituting the original vector by its nearest cluster centroid, there is a built-in natural perturbation or protection. Despite such perturbation, for some applications, the substitutes themselves may have adequately covered essential information for the intended purpose of classification. However, there is an accompanied risk that the revelation of the K-means (or SOM) cluster structure may inadvertently leak sensitive information exploitable by malicious intruder. As a remedy, the desensitized PCA may be performed prior to the K-means (or SOM) clustering, cf. Figure 5. The process contains two stages:

- First, extract desensitized PCA components via the Ridge DCA.
- Second, apply K-means (or SOM) to the lower-dimension and desensitized PCA vectors.

Since the data vectors are desensitized, the cluster structure formed by K-means should contain little or no sensitive information.

Fig. 8. Visualization of the 30 centroids of the desensitized K-means.

Example 7 (Desensitized K-means with Yale dataset):
Figure 8 shows the visualization of 30 K-means centroids derived from desensitized Yale dataset. Since the data vectors are already desensitized, it can be expected that the cluster structure formed by K-means (or for that matter, SOM) should leak little sensitive information.

□

VII. CONCLUSION AND FURTHER EXTENSION

Compressive Privacy (CP) aims at finding the optimal subspace of the original vector space for the purpose of privacy-preserving data mining (PPDM) and, more generally, privacy-preserving utility maximization (PPUM).

Extension to Kernel RDCA. Both DCA and Ridge DCA may be further extended to kernel DCA and kernel RDCA. More exactly, the optimal query vector in the empirical space, say $\mathbf{a}$, can be derived from the kernel-DCA optimizer:

$$\text{argmax}_{\mathbf{a}} \frac{\mathbf{a}^T\ [\mathbf{K}_B - \rho'\bar{\mathbf{K}}]\mathbf{a}}{\mathbf{a}^T[\bar{\mathbf{K}}^2 + \rho\bar{\mathbf{K}}]\mathbf{a}}$$

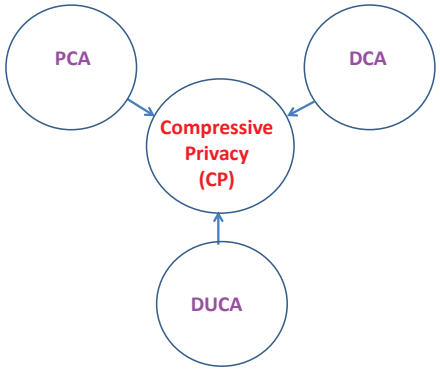


Fig. 9. Three promising subspace projection methods for CP are PCA, DCA, and DUCA.

which enables the query to be optimized in the much expanded nonlinear space so as to further enhance RDCA. For more detail, see [11], [14].

Extension to DUCA. Briefly, DUCA stands for differential ultility/cost advantage and is an extension of DCA. DUCA is based on the joint optimization of utility and privacy. DUCA is built upon the theoretical foundation of information and estimation theory, with intended applications to data mining and other machine learning problems. In short, as depicted in Figure 9, PCA, DCA, and DUCA, represent three promising subspace projection methods for compressive privacy (CP). For more detail, see [13].

Acknowledgement: This material is based upon work supported in part by the Brandeis Program of the Defense Advanced Research Project Agency (DARPA) and Space and Naval Warfare System Center Pacific (SSC Pacific) under Contract No. 66001-15-C-4068. The author wishes to thank Mert Al, Chang Chang Liu, and Artur Filipowicz from the Princeton University for invaluable discussion and assistances.

REFERENCES

[1] R. A. Fisher, “The use of multiple measurements in taxonomic problems,” *Annals of Eugenics*, vol. 7, pp. 179–188, 1936.
[2] C. R. Rao, ”The utilization of multiple measurements in problems of biological classification”, Journal of the Royal Statistical Society, Series B 10 (2): 159203, 1948.
[3] Duda, R.O. and Hart, P.E., ”Pattern Classification and Scene Analysis,” ”Wiley”, ”New York”, ”1973”. (See also “Classification,” Wiley, 2001.)
[4] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.
[5] T. Kohonen, *Self-Organization and Associative Memory*. New York: Springer-Verlag, 1984.
[6] Parlett, B. N., The Symmetric Eigenvalue Problem. Prentice-Hall Series in Computational Mathematics. Englewood Cliffs, N.J. 07 632. Prentice-Hall, Inc. 1980.
[7] H. Hotelling. Analysis of a complex of statistical variables into principal components.
[8] G. Golub and C. F. Van Loan. Matrix Computations, 3rd edition. Battimore, MD: Johns Hopkins University Press, 1996.
[9] Hoerl A. E. and Kennard R. W. , *Ridge Regression: Biased Estimation for Nonorthogonal Problems.* Technometrics, vol. 12, No. 1, pp. 55-67 Feb., 1970.
[10] A. N. Tychonoff. On the stability of inverse problems. Dokl. Akad. Nauk SSSR, 39(5):pp.195-198, 1943.
[11] S.Y. Kung, ”Discriminant component analysis for privacy protection and visualization of big data”, J. of Multimedia Tools & App., 2015.
[12] Thee Chanyaswad, J. Morris Chang, Prateek Mittal, S.Y. Kung, “Discriminant-Component Eigenfaces for Privacy-Preserving Face Recognition”, submitted to MLSP2016.
[13] S.Y. Kung, Compressive Privacy: From Information/Estimation Theory to Machine Learning, to appear on IEEE Signal Processing Magazine, submitted 2016.
[14] Kung S.Y., *Kernel Methods and Machine Learning*. Cambridge University Press, 2014.