Collaborative PCA/DCA Learning Methods for Compressive Privacy

SUN-YUAN KUNG and THEE CHANYASWAD, Princeton University
J. MORRIS CHANG, University of South Florida
PEIYUAN WU, Taiwan Semiconductor Manufacturing Company Limited

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 ensure our privacy, it is safer to let the data owner control the data to be uploaded to the network as opposed to taking chance with data servers or third parties. To this end, we propose *compressive privacy*, a privacy-preserving technique 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 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 the supervised learning environment, it is more effective to adopt a supervised PCA, known as discriminant component analysis (DCA), 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), whereas the noise subspace components of DCA are tightly coupled with recoverability and/or privacy protection. This article presents 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 is safer to perform K-means or SOM clustering on a desensitized PCA subspace.

Categories and Subject Descriptors: H.3.5 [**Online Information Services**]: Data sharing; K.4.1 [**Public Policy Issues**]: Privacy

General Terms: Algorithms, Design, Security

Additional Key Words and Phrases: DCA, PCA, compressive privacy, K-means, face-recognition, KDCA

This material is based on work that was 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 66001-15-C-4068.
Authors' addresses: S.-Y. Kung and T. Chanyaswad, Department of Electrical Engineering, Engineering Quadrangle, 41 Olden St, Princeton, NJ 08544 USA; J. Morris Chang, Department of Electrical Engineering, University of South Florida, Tampa, FL 33647 USA; P. Wu, Taiwan Semiconductor Manufacturing Company Limited, 8 LiHsin Rd. 6, Hsinchu Science Park, Hsinchu 300-78, Taiwan, R.O.C.

DOI: http://dx.doi.org/10.1145/2996460

ACM Reference Format:
Sun-Yuan Kung, Thee Chanyaswad, J. Morris Chang, and Peiyuan Wu. 2017. Collaborative PCA/DCA learning methods for compressive privacy. ACM Trans. Embed. Comput. Syst. 16, 3, Article 76 (July 2017), 18 pages.
DOI: http://dx.doi.org/10.1145/2996460

1. INTRODUCTION

We have all grown to become dependent on the Internet and the cloud for their ubiquitous data processing services, available at any time, anywhere, and for anyone. With its packet switching, bandwidth, storage, and processing capacities, the today's data center manages the server farm, supports an extensive database, and is ready to support, on demand from clients, a variable number of machines. However, the main problem of cloud computing lies in privacy protection. With rapidly growing Internet commerce, many of our daily activities are moving online; an abundance of personal information (e.g., 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.

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 (Figure 1): the *private sphere*, where data owners generate and process the decrypted data, and the *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 owners should have control over data privacy. It is safer to let the data owner control the data privacy and not take chances with cloud servers. To achieve this goal, we must provide some owner-controlled tools to safeguard private information against intrusion. New technologies are needed to ensure that personal data uploaded to the cloud will not be diverted to malicious applications.

Compressive privacy (CP) enables the data creator to "encrypt" data using compressive-and-lossy transformation and hence protects the 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 content 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.

2. CP 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.

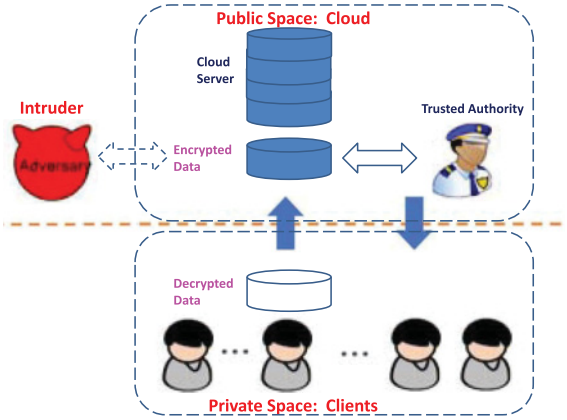


Fig. 1. From the privacy perspective, the encryption/accessibility of data is divided into two worlds: the *private sphere*, where data owners generate and process decrypted data, and the *public sphere*, where cloud servers can generally access only encrypted data, except the trusted authorities who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

Collaborative learning is a method for machine learning in which users supply feature vectors to a cloud 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 CP on a 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 versus unsupervised learning. Collaborative learning allows supervised and unsupervised machine learning techniques to learn from public vectors collected by cloud servers. We adopt principal component analysis (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, 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 introduce the notion of DCA, an extension of PCA, to effectively exploit the known class labels accompanied by supervised training datasets.

3. PRINCIPAL COMPONENT ANALYSIS

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 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 $\bar{\mathbf{x}}_i = \mathbf{x}_i - \overrightarrow{\boldsymbol{\mu}}, i = 1, \ldots, N$. 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* [Duda and Hart 1973] 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

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

3.1. PCA via Eigen-Decomposition of the Scatter Matrix

The objective of PCA now becomes finding the m ($m \leq M$) best components such that $\sum_{i=1}^{m} P(\mathbf{v}_i)$ is maximized, whereas $\mathbf{v}_i$ 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 $\bar{\mathbf{S}}$:

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

where $\mathbf{\Lambda}$ 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 $\mathbf{V}$—that is,

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

and the PCA-reduced feature vector can be represented by

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

3.2. Optimization of Power and Reconstruction Error

It is well known in 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 (RE):

—**PCA's power associated with the principal 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 RE associated with the minor eigenvectors: $\mathbf{V}_{minor}$.** Let the M-dimensional vector $\hat{\mathbf{x}}_\mathbf{z}$ denote 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:

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

where $E[\cdot]$ denotes the expected value. In an unsupervised machine learning, it is common practice to replace the covariance matrix $\mathbf{R}$ by the scatter matrix $\bar{\mathbf{S}}$. This

leads to the RE:

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

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

3.3. Simulation Results for PCA

Example 3.1 (*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 64×64 pixels, so the feature vectors derived from the pixel values have the dimension of 4096. To obtain 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) and (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 implies a high capacity to support the intended utility. Figure 2(c) depicts the (unsupervised) PCA eigenfaces for the face images in the Yale dataset.

4. DISCRIMINANT COMPONENT ANALYSIS

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

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

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

4.1. 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

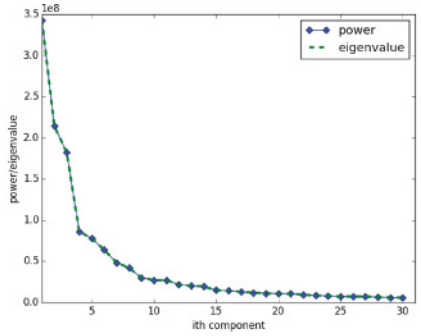
$$\mathbf{S}_W \;=\; \sum_{\ell=1}^{L} \sum_{j=1}^{N_\ell} \big[\mathbf{x}_j^{(\ell)} - \overrightarrow{\boldsymbol{\mu}}_\ell\big]\big[\mathbf{x}_j^{(\ell)} - \overrightarrow{\boldsymbol{\mu}}_\ell\big]^T, \tag{9}$$

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

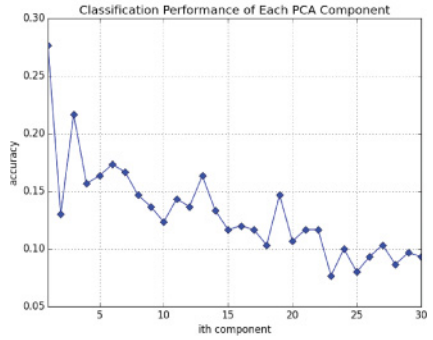
The between-class scatter matrix $\mathbf{S}_B$ is defined as

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

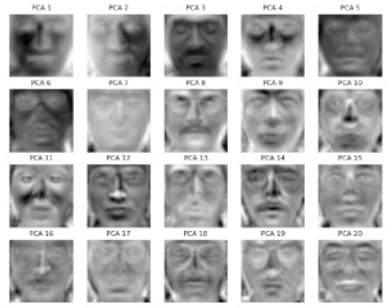
where $f_{ij} = r_i r_j$, with $r_i \equiv \frac{N_i}{N}$, $r_j \equiv \frac{N_j}{N}$, stands for the relative frequency of involving classes i and j. Note that the greater the magnitude of $\boldsymbol{\Delta}_{ij}$, where $\boldsymbol{\Delta}_{ij} \equiv [\overrightarrow{\boldsymbol{\mu}}_i - \overrightarrow{\boldsymbol{\mu}}_j]$, the



(a)



(b)



(c)

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

more distinguishable between the i-th and j-th classes. This 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 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.

4.2. Linear Discriminant Analysis

Linear discriminant analysis (LDA) focuses on an important special case when $m = 1$ and $L = 2$, and thus

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

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 simplify the terms *signal variance* to *signal* and *noise variance* to *noise*, respectively. LDA [Fisher 1936] aims to maximize the signal-to-noise ratio (SNR): $\text{SNR} = \frac{\textbf{signal}}{\textbf{noise}}$. More exactly,

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

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

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

4.3. Multiple Discriminant Component Analysis

To facilitate our exploration into an appropriate criterion for component analysis, we propose an optimization criterion based on the sum of the SPRs pertaining to all individual components:

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

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

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

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

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

To preserve the rotational invariance of 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 in Kung [2015], 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 component power in the CVS. The mapping from a vector $\mathbf{x}$ in the original space to its counterpart $\tilde{\mathbf{x}}$ in the CVS is represented by $\tilde{\mathbf{x}} = [\bar{\mathbf{S}}]^{-\frac{1}{2}} \mathbf{x}$. 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 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 [Tikhonov 1943; Hoerl and Kennard 1970], resulting in 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 as that in Kung [2015]:

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

The DCA solution may be directly obtained from the first m principal eigenvectors of the regulated discriminant matrix:

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

with the columns of the solution matrix $\mathbf{V}$ meeting the canonical orthonormality condition prescribed by Equation (15). Numerically, the optimal DCA projection matrix can be derived from the principal eigenvectors of[1]

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

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

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

4.4. 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—that is,

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

After the modification, the total SPR′ is

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

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 5.

4.5. 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 [Chanyaswad et al. 2016].

[1]For DCA, all eigenvalues are generically distinct, and therefore all columns of $\mathbf{V}$ are canonically orthogonal to each other [Parlett 1980].

Example 4.1 (*PCA/DCA for the DIP*). In the 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., $\mathbf{u} = \mathbf{u}(\mathbf{x}) = x_1 + x_2$, the financial condition of the family). From the privacy perspective, the query should not pry into the income disparity within the family. For instance, the privacy function is set as $\mathbf{p} = \mathbf{p}(\mathbf{x}) = x_1 - x_2$, which determines the bread winner 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 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 principal 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 in Figure 3(a) and (b), with a query marked as "♡", we note that although 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 Equation (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 $\mathbf{eig}(\mathbf{S}_{B_U}, \bar{\mathbf{S}} + \rho\mathbf{I})$ yields the following eigenvalues or, equivalently, the component SPR′:

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

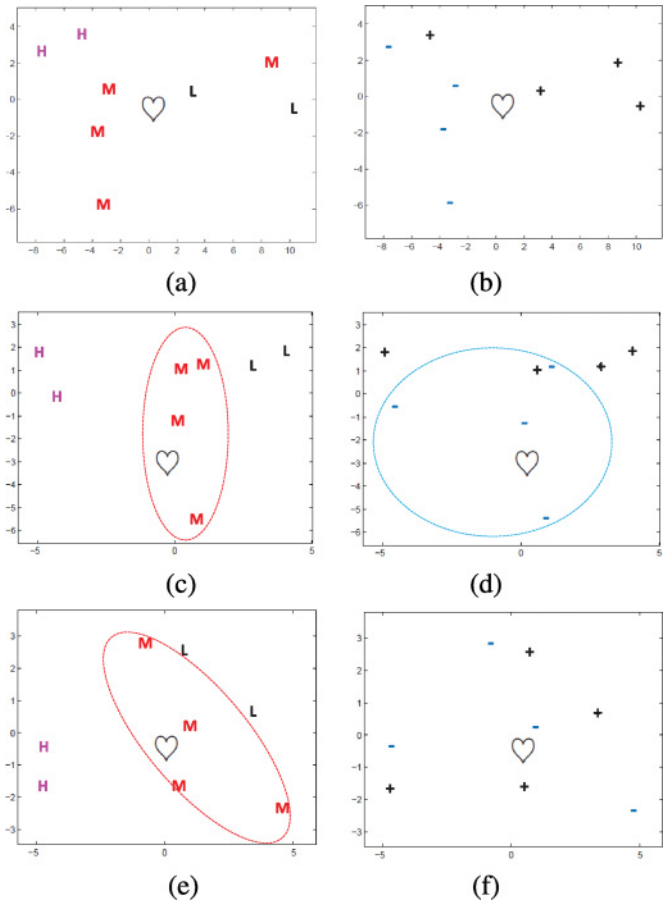
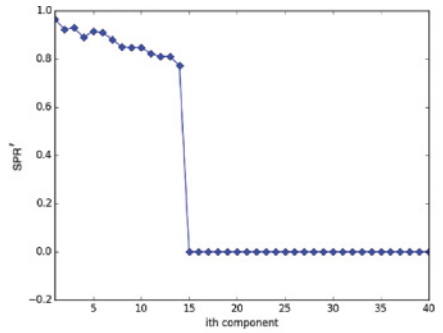


Fig. 3. Visualization of a query, marked as "♡", mapped to the optimal two-dimensional PCA, DCA, and ridge DCA (RDCA) subspaces. The family income (utility) class can be confidently assessed as the "M"-class. Shown are (a) PCA visualization with utility explicitly labeled, (b) PCA visualization with privacy explicitly labeled, (c) DCA visualization with utility explicitly labeled, (d) DCA visualization with privacy explicitly labeled, (e) RDCA visualization with utility explicitly labeled, and (f) RDCA 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) although 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 that most effectively identifies the utility label, but it also leaks the privacy label; and (3) RDCA is the only one that simultaneously identifies the utility label and protects the privacy label. The privacy label on the income disparity remains clueless, as both classes ("+" or "−") have equal claim (see (f)).

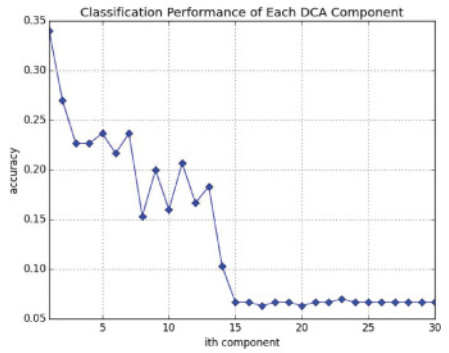
There after, the two principal eigenvectors corresponding to the two nonzero eigenvalues are (see Equation (18)):

$$\mathbf{f}_1 = \begin{bmatrix} 0.204 \\ 0.839 \\ 0.245 \\ 0.443 \end{bmatrix} \text{ and } \mathbf{f}_2 = \begin{bmatrix} 0.221 \\ -0.535 \\ 0.733 \\ 0.357 \end{bmatrix}.$$

As demonstrated in 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.



(a)



(b)

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

Example 4.2 (*DCA for PPFR Applications*). The experimental setup basically follows that of Example 3.1. 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 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 FR, with per-component accuracy around 23%. In contrast, the next 16 noise-subspace eigen-components (i.e., 15th through 30th) are basically noise ridden and carry little useful information, with a low accuracy around 6%, par random guess. This implies that these noise-subspace components contain no useful information for FR.

5. DESENSITIZED PCA VIA RIDGE DCA

In the previous section, DCA is applied to utility-driven machine learning applications. Now we address a DCA variant tailored for privacy-driven PCA (i.e., desensitized PCA). An exemplifying application scenario is the so-called antirecognition utility maximization (ARUM), in which the privacy intruder's objective is FR itself. As such, the goal of CP 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 [Vapnik 1995], or unsupervised

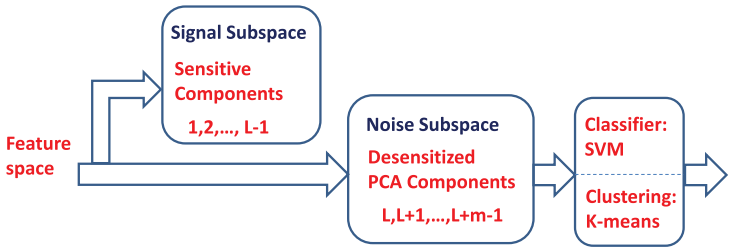


Fig. 5. The privacy-driven DCA system structure for desensitized PCA and/or desensitized K-means (see Section 6).

clustering, such as K-means or SOM [Kohonen 1984]. The overall flow diagram of the desensitizing system structure is depicted in Figure 5.

5.1. Incorporating a Negative Ridge into the Signal Matrix

Ridge DCA (RDCA) 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

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

Numerically, the optimal RDCA solution can be derived from the principal eigenvectors of

$$\mathbf{eig}(\mathbf{S}_B - \rho'\mathbf{I}, \tilde{\mathbf{S}} + \rho\mathbf{I}). \tag{23}$$

By slightly modifying Equation (20), we obtain the following eigenvalue analysis:

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

5.2. Eigenvalues of Eigen-Components of RDCA

Now let us elaborate on the implication of Equation (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, which 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

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

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 is defined in Equation (2). This 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 RDCA is also called *desensitized PCA*.

5.3. Simulation Results

Let us now illustrate the application of desensitized PCA by exploring two examples: (1) a toy application example on the DIP and (2) ARUM based on the Yale face dataset.

Example 5.1 (*RDCA for DIP*). Let us revisit the DIP example. For RDCA, via Equation (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 RDCA can be computed from the generalized eigen-decomposition of $\mathbf{eig}(\mathbf{S}_{B_P} - \rho'\mathbf{I}, \tilde{\mathbf{S}} + \rho\mathbf{I})$, yielding

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

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

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

correspond to the following (decreasing) component powers:

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

The two eigenvectors correspond to the highest component powers—that is, $P(\mathbf{v}_2)$ and $P(\mathbf{v}_3)$ will be adopted as the two desensitized PCAs (see Equation (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 Figure 3(a) through (c), we can summarize our simulation results as follows ("♡" represents the query):

—Although 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.
—RDCA is the only one that 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 ARUM, an alternative application scenario that is in sharp contrast to the conventional PPFR application. Let us now discuss how to apply RDCA to ARUM problems. Briefly, RDCA starts with removing the first

$L-1$ eigen-components, to desensitize the feature vectors, and subsequently sorts the remaining components based on their component powers, just like PCA.

Example 5.2 (*RDCA for ARUM*). For ARUM, our objective is to extract an optimal subspace that may prevent a person from being recognized in the compressed face image. By performing an experiment on the Yale dataset in a similar setup as Example 3.1, the following results are achieved:

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

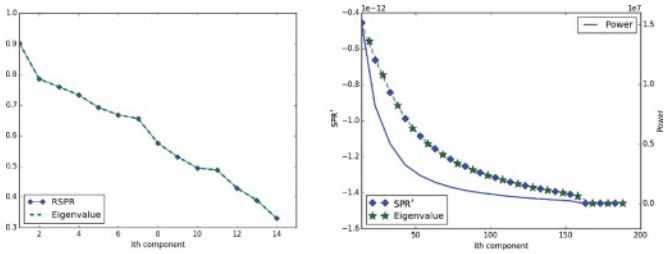
The following example provides a preliminary comparison of face reconstructions with PPFR versus ARUM. It may shed some light on how the desensitized PCA may facilitate privacy protection in the ARUM-type scenarios.

Example 5.3 (*Face Reconstructions: PPFR vs. ARUM*). Figure 7(a) shows an original face from the Yale dataset. Figure 7(b) and (c) depict the reconstructed face image via DCA and RDCA, respectively. Suppose that the intended utility is, say, to distinguish (1) smiling faces versus sad faces or (2) faces with eyeglasses with versus faces 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.

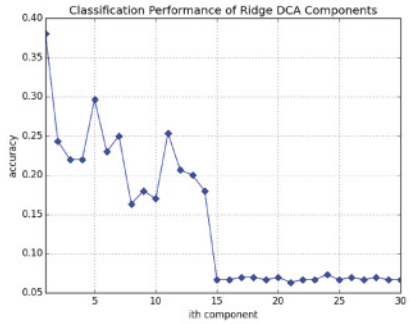
An experiment on the in-house Glasses dataset confirms that desensitized PCA is effective for ARUM. The dataset consists of 50 samples (images) chosen from Yale and Olivetti databases such that each individual in the dataset has 50% of his or her images with glasses on and the other 50% without glasses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

—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., FR). 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.

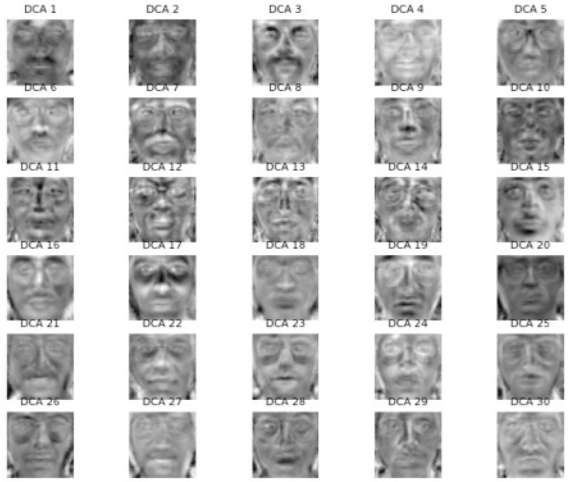
The experimental setup is as follows. In each trial of the experiment, 5 samples are randomly chosen to be left out, whereas the other 45 samples are used to train the classifiers. Then one of the 5 left-out samples is chosen randomly for testing. The performance of the classification on the data both before and after PCA desensitization is collected for comparison, and the experiment is repeated for 1,000 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.



(a) (b)



(c)



(d)

Fig. 6. Rank-ordered eigenvalues of RDCA, when $\rho' = 0.00001$. (a) *Signal-subspace component analysis*: The figure (diamond vs. dashed line) confirms that the component's SPR′ is dictated by the eigenvalues in a manner consistent with the theoretical prediction given in Equation (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 Equation (27) (star vs. solid line). Moreover, the component's SPR′ is dictated by the eigenvalues as predicted in Equation (26) (diamond vs. star). (c) Each of the 16 desensitized eigenfaces yields a low accuracy around 6%, par 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.

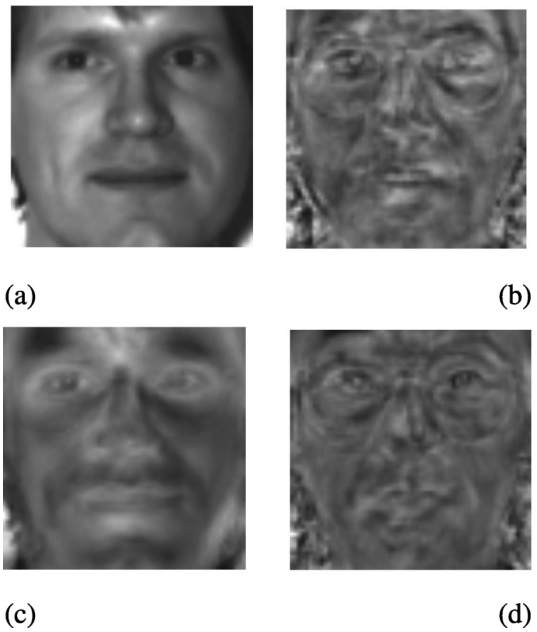


Fig. 7. Original and reconstructed face images using 160 dimensions from the Yale dataset. (a) The original face. (b) DCA with zero ridge. (c) RDCA with a negative ridge applied to the privacy-driven signal matrix. (d) RDCA with reversely rank-ordered eigen-components. (Courtesy of Chanyaswad et al. [2016]).

Table I. Utility and Privacy Accuracy Performance of Desensitized PCA on the Glasses Dataset

	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 summarizes the results from the experiment. Briefly, among the 1,000 trials:

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

6. DESENSITIZED K-MEANS VIA RDCA

Note that K-means (or SOM) by itself has a natural role in 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 a malicious intruder. As a remedy, the desensitized PCA may be performed prior to the K-means (or SOM) clustering (see Figure 5). The process contains two stages:

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

—First, extract desensitized PCA components via RDCA.
—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.

Example 6.1 (*Desensitized K-means with the Yale Dataset*). Figure 8 shows the visualization of 30 K-means centroids derived from the 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.

7. CONCLUSION AND FURTHER EXTENSION

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 RDCA 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:

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

which enables the query to be optimized in the much expanded nonlinear space to further enhance RDCA. For more detail, see the work of Kung [2014, 2015] and Chanyaswad et al. [2017].

Extension to differential utility/cost advantage. Briefly, the differential utility/cost advantage (DUCA) is an extension of DCA. DUCA is based on the joint optimization of utility and privacy. DUCA is built on the theoretical foundation of information and estimation theory, with intended applications to data mining and other machine learning problems. As depicted in Figure 9, PCA, DCA, and DUCA represent three promising subspace projection methods for CP. For more detail, see Kung [2017].

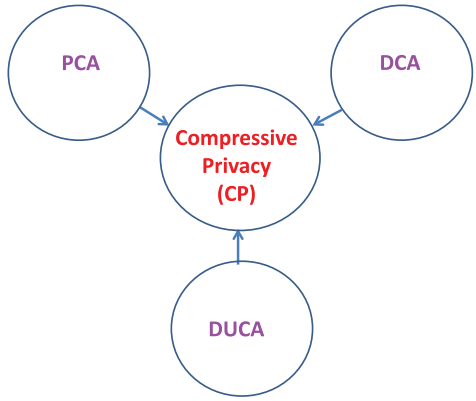


Fig. 9. Three promising subspace projection methods for CP are PCA, DCA, and DUCA.

ACKNOWLEDGMENTS

The author wishes to thank Mert Al, Changchang Liu, and Artur Filipowicz of Princeton University for invaluable discussion and assistance.

REFERENCES

Thee Chanyaswad, J. Morris Chang, and Sun-Yuan Kung. 2017. A compressive multi-kernel method for privacy-preserving machine learning. In *Proceedings of the 2017 IEEE International Joint Conference on Neural Networks (IJCNN'17)*.

Thee Chanyaswad, J. Morris Chang, Prateek Mittal, and Sun-Yuan Kung. 2016. Discriminant-component eigenfaces for privacy-preserving face recognition. In *Proceedings of the 2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP'16)*. IEEE, Los Alamitos, CA, 1–6.

Richard O. Duda and Peter E. Hart. 1973. *Pattern Recognition and Scene Analysis*. Wiley.

Ronald A. Fisher. 1936. The use of multiple measurements in taxonomic problems. *Annals of Eugenics* 7, 2, 179–188.

Arthur E. Hoerl and Robert W. Kennard. 1970. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* 12, 1, 55–67.

Teuvo Kohonen. 1984. *Self-Organization and Associative Memory*. Springer-Verlag, New York, NY.

Sun-Yuan Kung. 2014. *Kernel Methods and Machine Learning*. Cambridge University Press, Cambridge, England.

Sun-Yuan Kung. 2015. Discriminant component analysis for privacy protection and visualization of big data. *Multimedia Tools and Applications* 76, 3, 3999–4034.

Sun-Yuan Kung. 2017. Compressive privacy: From information\/estimation theory to machine learning [lecture notes]. *IEEE Signal Processing Magazine* 34, 1, 94–112.

Beresford N. Parlett. 1980. *The Symmetric Eigenvalue Problem*. Prentice-Hall Series in Computational Mathematics. Prentice Hall.

Andrey Nikolayevich Tikhonov. 1943. On the stability of inverse problems. *Comptes Rendus (Doklady) de l'Academie des Sciences de l'URSS* 39, 195–198.

Vladimir Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, NY.

Received April 2016; revised August 2016; accepted Septmber 2016