DIFFERENTIAL MUTUAL INFORMATION FORWARD SEARCH FOR MULTI-KERNEL DISCRIMINANT-COMPONENT SELECTION WITH AN APPLICATION TO PRIVACY-PRESERVING CLASSIFICATION

Thee Chanyaswad Mert Al J. Morris Chang S. Y. Kung
Princeton University Princeton University University of South Florida Princeton University
Princeton, NJ, USA Princeton, NJ, USA Tampa, FL, USA Princeton, NJ, USA

ABSTRACT
In machine learning, feature engineering has been a pivotal stage in building a high-quality predictor. Particularly, this work explores the multiple Kernel Discriminant Component Analysis (mKDCA) feature-map and its variants. However, seeking the right subset of kernels for mKDCA feature-map can be challenging. Therefore, we consider the problem of kernel selection, and propose an algorithm based on Differential Mutual Information (DMI) and incremental forward search. DMI serves as an effective metric for selecting kernels, as is theoretically supported by mutual information and Fisher’s discriminant analysis. On the other hand, incremental forward search plays a role in removing redundancy among kernels. Finally, we illustrate the potential of the method via an application in privacy-aware classification, and show on three mobile-sensing datasets that selecting an effective set of kernels for mKDCA feature-maps can enhance the utility classification performance, while successfully preserve the data privacy. Specifically, the results show that the proposed DMI forward search method can perform better than the state-of-the-art, and, with much smaller computational cost, can perform as well as the optimal, yet computationally expensive, exhaustive search.

Index Terms— Compressive Privacy, Differential Mutual Information (DMI), incremental forward search, Kernel Discriminant Component Analysis (KDCA), kernel selection, multi-kernel learning.

1. INTRODUCTION
In machine learning, non-linear feature mapping has been effective in building predictive features, especially for classification. There have been several methods to produce effective feature mappings such as the kernel method [1], subspace projection [1], and artificial neural networks [2]. With a wide range of possible feature-maps, using multiple feature-maps can be beneficial as seen in multi-kernel learning [3], which combines feature-maps derived from many kernel functions to build an effective predictor. Conversely, using too many feature-maps altogether can cost the performance [4], as will also be demonstrated in this work. Therefore, there is merit in effectively choosing a subset of feature-maps to be used for a certain application.

Meanwhile, as machine learning tools become more powerful, they can be a double-edged sword. As more data are collected and analyzed daily, they can divulge sensitive information about individuals. Hence, the concern of data privacy arises, and the relationship between machine learning and data privacy is inevitable. For this reason, in this work, we consider the application in privacy-preserving classification. To this end, we bring in the regime of Compressive Privacy [5, 6], which uses lossy compression as the primary tool for privacy protection. The conjecture is that compressed data would only be useful for the intended utility task, and, hence, provide protection against any other sensitive privacy leakage. One particular type of feature-maps that has been shown to be effective for privacy-preserving classification under Compressive Privacy is the multiple Kernel Discriminant Component Analysis (mKDCA) feature-map and its variants [3]. More specifically, mKDCA employs the utility-optimizing compression scheme called the Kernel Discriminant Component Analysis (KDCA) [1, 7, 3], which allows very high compression rate, while optimizing the classification capability of the data.

However, to effectively utilize the mKDCA feature-map, the selection of kernels has to be done carefully since using too many kernels can actually hurt the performance, as will be shown in our experiments. Therefore, we consider the kernel selection problem for KDCA feature-maps, and propose a method based on the Differential Mutual Information (DMI) metric and incremental forward search. DMI is used as the relevance metric to evaluate the quality of the select mKDCA feature-map. Meanwhile, the incremental forward search is used to reduce redundancy among the select KDCA feature-maps. Both parts of the algorithm play integral roles in efficiently finding an effective subset of kernels. To summarize, we propose the DMI forward search method to select a subset of KDCA feature-maps that, once used collectively, is effective for the privacy-preserving classification task.
We evaluate the method on three real-world datasets – MHEALTH, HAR, and REALDISP. All three are mobile sensing datasets, so they all have privacy ramifications in the real-world setting. The results show that the proposed method can perform better than the state-of-the-art both in terms of utility and privacy. Specifically, for utility, the proposed method can outperform the state-of-the-art by as much as 3.33%, while for privacy, it achieves almost or equal to the privacy level of random guess across all experiments. Additionally, without the burden of computational cost or infeasibility, the proposed incremental forward search can perform as well as the optimal exhaustive search method.

2. RELATED WORKS

2.1. Kernel Selection

The kernel selection problem in our setting has a tie to the traditional feature selection problem. Specifically, kernel selection for KDCA feature-maps extends the feature selection problem to select disjoint groups of features. Although the extension may seem trivial, it requires careful inspection of the theoretical aspect, as well as the algorithmic implementation, of existing feature selection framework. For example, for Fisher Discriminant Ratio (FDR)-based algorithm [1, 8], the FDR score has to be modified to accommodate the multi-feature nature of the problem; for the mutual-information-based methods such as Minimal-Redundancy-Maximal-Relevance (mRMR) [9], the mutual information estimation has to, similarly, be modified to consider multiple features simultaneously.

The closest work to our problem is possibly the Group Lasso framework [10, 11, 12]. This is an extension to the Lasso feature selection by enforcing coefficients of all features in the same group to be zero together. Although Group Lasso gives non-zero coefficients for the features in the select groups, which can be used directly as a classifier, these non-zero coefficients can also simply be used as a mask vector for group selection. In order to compare the performance gain via the kernel selection method only, this is how Group Lasso will be applied to serve as the state-of-the-art in our work.

2.2. Compressive Privacy

Since our work concerns the application of data privacy, we adopt the regime of Compressive Privacy [6, 5]. Compressive Privacy aims at protecting privacy via utility-optimizing lossy encoding by transforming the data such that they become information-rich with respect to the utility task, but information-lossy with respect to any other tasks, including privacy-sensitive ones.

Another important principle in Compressive Privacy is the separation of the public and private spheres. Compressive Privacy emphasizes leaving the task of privacy protection to the data owner. Therefore, the private sphere is where the lossy-encoding is carried out by the data owner before the privacy-preserving version of the data are shared in the public sphere for utility purposes, which may be carried out by other – possibly untrusted – entities.

3. MKDCA FEATURE-MAP FOR PRIVACY-PRESERVING CLASSIFICATION

Since we aspire to build a combination of feature-maps that is indicative of the utility classification, but not of the privacy classification under Compressive Privacy [6, 5], we utilize a privacy-preserving utility-optimizing feature mapping scheme called the multiple Kernel Discriminant Component Analysis (mKDCA), which has been shown to be effective for the application of privacy-aware classification [1, 7, 13]. This scheme combines the frameworks of multi-kernel learning and the Kernel Discriminant Component Analysis (KDCA) [1, 7, 3] subspace projection to form a combination of feature-maps that is very informative of the given utility task, but not of anything else.

Specifically, given a kernel function \( k(x_i, x_j) \) and a supervised training dataset, KDCA aims at finding a subspace that maximizes the discriminant power via the optimization:

\[
A_{KDCA} = \arg \max_{A: A^T [K - \rho I] A = 1} \text{tr}(A^T K_B A)
\]

where \( \rho \) is the ridge parameter, and \( K, K_B \) are defined as follows.

\[
K = C_N K_N
\]

\[
K_B = C_N \left[ \sum_{l=1}^{L} N_l (k(\mu_l) - k(\mu)) (k(\mu_l) - k(\mu))^T \right] C_N
\]

where \( k(\mu_l) = \sum_{x_j \in C_l} \frac{k(x_j)}{N_l} \); \( k(\mu) = \sum_{j=1}^{N} \frac{k(x_j)}{N} \); \( C_N = \left[ I - \frac{1}{N} ee^T \right] \); \( e = [1, 1, \ldots, 1]^T \). \( K \) is the kernel matrix; and \( k(x_j) \in \mathbb{R}^N \) is the kernel vector of \( x_j \). More importantly, the optimal subspace of KDCA has the dimension of \( L - 1 \), where \( L \) is the number of classes. This property allows KDCA to have very high compression rate, while optimizing the classification power of the features. Additionally, each mapping from the original feature space to a KDCA subspace via a different kernel is a possible feature-map. Therefore, we define the KDCA feature-map derived from a given kernel as:

\[
\phi^{(j)}_i = A_{K_{DCA}}^j k_j(x_i) \in \mathbb{R}^{L-1}
\]

where \( A_{K_{DCA}} \in \mathbb{R}^{N \times (L-1)} \), and \( \phi^{(j)}_i \) is the KDCA feature-maps of \( x_i \) based on the kernel \( k_j(\cdot, \cdot) \).

By varying the choice of kernels, many individual KDCA feature-maps can be derived, and by concatenating these multiple KDCA feature-maps together, we form the desired mKDCA feature-map. More formally, given \( Q \) KDCA feature-maps, \( \{ \phi^{(1)}_i, \phi^{(2)}_i, \ldots, \phi^{(Q)}_i \} \), the mKDCA feature-map for \( x_i \) is [3], \( f(x_i) = \phi_i = [\phi^{(1)}_i, \phi^{(2)}_i, \ldots, \phi^{(Q)}_i]^T \).
To address the problem defined in Section 4, we propose the **DMI forward search** method. The method consists of two components – the DMI metric, and the incremental forward search algorithm. The former component functions as the metric to predict the quality of the kernels with respect to the utility classification task. On the other hand, as it is well-studied that redundancy across features can hurt the collective performance [14, 15, 9, 1], the latter component is used to reduce redundancy among the KDCA feature-maps selected.

### 5.1. Differential Mutual Information

We consider the classification task, so we choose the metric to determine the quality of the kernels based on Fisher’s discriminant power [7, 8]. To this end, we adopt the **Differential Mutual Information (DMI)** metric [5] that has been shown to be effective in predicting the inter-class separability of kernels [3]. The theoretical foundation of DMI is as follows.

From Theorem 1 of [5], under the Gaussian distribution assumption, the maximum differential mutual information between the utility function and the query – which is the KDCA feature-maps in our case – is given by the **Utility-Driven Differential Information Maximization (DMI)**:

$$\max_{\beta} f^T \Sigma_B \beta$$

To apply this metric to the classification setting, we follow the procedure in [5]. First, $\Omega$ is represented by the **between-class scatter matrix** [16], $S_B = \sum_{l=1}^{L} N_l [\Phi_l^T - \bar{\Phi}^T] [\Phi_l - \bar{\Phi}]^T$, where $L$ is the number of classes, $N_l$ is the number of samples in class $l$, $\Phi_l^T = \sum_{i \in C_l} \phi_i / N_l$ is the centroid of class $l$, and $\bar{\Phi}$ is the centered training data based on the specified KDCA feature-map. Finally, the ridge parameter $\rho$ can assume the role of $\sigma^2$.

Therefore, the maximum differential mutual information becomes:

$$\max_{f \in \| \Phi \|_2 + \rho f = 1} f^T S_B f$$

For the general case when the query $f$ may be a matrix, this can be formulated as a trace-norm optimization similar to the generalization from Fisher’s LDA [8] to multiple discriminant analysis (MDA) [17], yielding:

$$\max_{f \in \| S + \rho I \|_F = 1} \text{tr}(f^T S_B f)$$

has been shown to be [5]:

$$\text{tr}([S + \rho I]^{-1} S_B)$$

Therefore, we can use this as a metric to predict the classification potential of the given KDCA feature-map. Formally, we present our **Differential Mutual Information (DMI)** metric for a given feature-map $f(X) = \Phi$ and the training utility label $y_u$ as,

$$DMI[\Phi] = \text{tr}([S + \rho I]^{-1} S_B)$$

### 5.2. **DMF Forward Search**

**DMF Forward Search** is an effective criterion to select the optimal kernel. From Fisher’s discriminant analysis [8, 19], DMI represents the sum of mutual information and Fisher’s discriminant analysis. From mutual information [18], DMI is the optimal mutual information between the query and the utility [5]. From Fisher’s discriminant analysis [8, 19], DMI represents the sum of signal-to-noise ratio under the canonical orthogonality, which has been shown to translate well to the classification performance [8, 1]. Second, as shown by Kung [20], unlike MDA, DMI takes component analysis into full account, i.e. the DMI criterion could better facilitate kernel selection than the MDA criterion. For example, it is theoretically possible that two different kernel functions could result in identical MDA but distinct DMI. In this case, DMI (instead of MDA) is a more effective criterion to select the optimal kernel.

Despite its strong prediction power, the DMI metric does not yet take into account possible redundancy among select KDCA feature-maps, so we propose a simple incremental forward search algorithm to reduce such redundancy.
We present an application of the Intuitively, \( \delta \) significantly increase DMI of the select set. In Algorithm 1, that if the new KDCA feature-map is redundant, it would not KDCA feature-map to the select set sufficiently increases the holding avoid inter-feature-map redundancy by adding one kernel to the set at a time as sorted by the decreas-

incremental forward search via the sive, if not infeasible. Therefore, we propose a compromise optimal subset, it is undoubtedly computationally very expen-

is to perform an exhaustive search on all possible subsets of One way to resolve redundancy among KDCA feature-maps

5.2. Incremental Forward Search Algorithm

One way to resolve redundancy among KDCA feature-maps is to perform an exhaustive search on all possible subsets of KDCA feature-maps. Although this method would find the optimal subset, it is undoubtedly computationally very expensive, if not infeasible. Therefore, we propose a compromise via the incremental forward search method with incremental thresholding. Algorithm 1 describes the method. The general procedure is to start with an empty select set \( \Phi' = \emptyset \), \( f_{mKDCA} = \emptyset \). Initialize \( DMI_{current} = 0 \).

2. for each \( \Phi^{(k)} \) sorted by decreasing DMI do:
   (a) \( \Phi^{new} = [\Phi^T, \Phi^{(k)}]^T \).
   (b) \( DMI_{new} = DMI[\Phi^{new}] \).
   (c) if \( DMI_{new} > (1 + \delta) \cdot DMI_{current} \) then:
      i. \( \Phi' = \Phi^{new} \); add \( f_k \) to \( f_{mKDCA} \).
      ii. \( DMI_{current} = DMI_{new} \).
3. return \( \Phi', f_{mKDCA} \)

6. EXPERIMENTS

We present an application of the DMI forward search method on the privacy-preserving classification problem in mobile sensing. More specifically, we use the information from mobile sensors as the original input features, and derive 21 KDCA feature-maps by varying the kernel functions used in the derivation. Hence, these 21 KDCA feature-maps serve as the possible input set in all experiments. Finally, the utility classification task is to predict the activity being performed from the sensing data, while the sensitive privacy classification that we do not want the data to reveal is the identification of the person performing the activity.

6.1. Datasets

We perform experiments on three datasets. In all three datasets, there are 6 utility classes and 10 privacy classes, and each has the following features/training/testing sizes – MHEALTH [21, 22]: 23/4800/900, REALDISP [23, 24]: 117/4320/1620, and HAR [25]: 561/1905/1200.

6.2. Procedure

For all experiments, we select the subset of KDCA feature-maps from the input set of 21 KDCA feature-maps, derived from the following 21 kernels:

1. Linear: \( k(x_i, x_j) = x_i^T x_j \).
2. Polynomial: \( k(x_i, x_j) = (\gamma x_i^T x_j + 1)^p; p = \{2, 3\} \).
3. RBF: \( k(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \).
4. Laplacian: \( k(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|) \).

All of these are with \( \gamma = 10^{(-3,-2, -1,0,1)} \), and since there are 6 utility classes, each KDCA feature-map has 5 dimensions. In all experiments, we use support vector machine (SVM) [26, 27] as the classifier, and select its parameters via three-fold cross-validation. We compare our approach (DMI forward search) with four other methods:

1. Using all 21 KDCA feature-maps without performing kernel selection (Full multi-kernel).
2. Applying Group Lasso \([10, 11, 12]\) to select a subset of KDCA feature-maps (Group Lasso).
3. The method which ranks the KDCA feature-maps by their individual DMI score, but without the incremental forward search algorithm (DMI filter).
4. Exhaustively searching for the subset that yields the maximum DMI (DMI exhaustive search).

The purpose of the first comparison is to show the merit in selecting kernels. The second comparison is to show that our method can improve upon the state-of-the-art. The third comparison is to illustrate that redundancy can hurt performance. Finally, the last comparison is to show that incremental forward search can reach the performance close to that achievable by the computationally expensive exhaustive search.

The hyper-parameters of Group Lasso and DMI forward search are chosen from the set of \( \alpha = 2\{-1,-2,\ldots,-10\} \), and \( \delta = 2\{-1,-2,\ldots,-10\} \), respectively. The number of select feature-maps in the exhaustive search method is chosen from the set of \( \{1,2,3,4\} \) to make the computation feasible. The value of \( \rho \) for the DMI metric is set to be: \( \rho = 10^{-3} \cdot \min_{\Phi} \lambda(S) \). Finally, note that even though Group Lasso traditionally gives the coefficients for the selected KDCA feature-maps as well, to fairly compare the kernel selection algorithms, we use these coefficients merely as a masking to select KDCA feature-maps. In other words, if the Group Lasso algorithm yields a non-zero coefficient for
<table>
<thead>
<tr>
<th>Method</th>
<th>MHEALTH</th>
<th></th>
<th>REALDISP</th>
<th></th>
<th>HAR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># select</td>
<td>feature size</td>
<td>Utility</td>
<td>Privacy</td>
<td># select</td>
</tr>
<tr>
<td></td>
<td>kernels</td>
<td></td>
<td></td>
<td></td>
<td>kernels</td>
</tr>
<tr>
<td>Full multi-kernel</td>
<td>21</td>
<td>105</td>
<td>85.55</td>
<td>19.44</td>
<td>21</td>
</tr>
<tr>
<td>Group Lasso</td>
<td>2</td>
<td>10</td>
<td>87.89</td>
<td>15.56</td>
<td>2</td>
</tr>
<tr>
<td>DMI filter</td>
<td>8</td>
<td>40</td>
<td>89.44</td>
<td>12.00</td>
<td>17</td>
</tr>
<tr>
<td>DMI exhaustive search</td>
<td>4</td>
<td>20</td>
<td>91.33</td>
<td>10.22</td>
<td>4</td>
</tr>
<tr>
<td>DMI forward search</td>
<td>4</td>
<td>20</td>
<td>91.22</td>
<td>16.56</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 1. Utility and privacy classification accuracies (%) on the three datasets: MHEALTH, REALDISP, and HAR. All methods select the optimal subset of KDCA feature-maps with respect to their criteria, as described in Section 6.2. Since each KDCA feature-map has five dimensions, the feature size is a multiple of fives. The results for DMI exhaustive search are the best testing results among all of the possible subsets of KDCA feature-maps with the subset cardinality of four or less.

the certain kernel, we select it; otherwise, we discard it. Then, we use SVM as the classifier similar to other methods. This is to eliminate possible discrepancy caused by the choice of the classifier, and focus the comparative study on the kernel selection methods.

6.3. Results

The results in Table 1 confirm the overall efficacy of the mKDCA feature-map and its variants for privacy-preserving classification. Even without any kernel selection algorithm (Full multi-kernel), mKDCA feature-map already performs respectably well. However, we discuss here how kernel selection can improve upon it.

There are two components of our results:
- Privacy Protection
- Utility Classification Performance

For the former, our privacy protection is achieved via the KDCA compression, and the results confirm it to be successful, as seen by the privacy classification accuracies close or equal to random guess in all experiments. These results match our intuition and conjecture that once the data are highly-compressed for the intended utility, they hardly leak any other information, including the specified privacy task.

On the other hand, the utility performance is more intriguing, as we propose five variants of mKDCA feature-maps based on different kernel selection algorithms, and justify the improvement and design choice stage by stage. We now elaborate them in the following subsections.

6.3.1. Full multi-kernel

It is noticeable that mKDCA without any kernel selection algorithm (Full multi-kernel) already yields respectable utility, especially in REALDISP and HAR. However, we observe that there are possible improvements with kernel selection. Specifically, DMI forward search shows improvements by 5.67%, 0.25%, and 1.66%, on the three datasets with much smaller numbers of kernels used. This confirms the merit of performing kernel selection for better utility.

6.3.2. Group Lasso

Group Lasso also performs better than the Full multi-kernel overall, which, again, confirms the merit of kernel selection. In addition, it performs almost equally well when compared to DMI forward search on REALDISP and HAR. However, on MHEALTH, DMI forward search performs considerably better than Group Lasso (by 3.33%), which signifies the potential of DMI as the criterion for kernel selection.

6.3.3. DMI filter

DMI filter is a very simple kernel selection algorithm, but even so, it still improves upon the mKDCA without kernel selection. More interestingly, the comparison between this simple algorithm, which does not consider redundancy, and the DMI forward search verifies that redundancy can hurt performance. Crucially, the gain in utility via incremental forward search is achievable by a much smaller or equal number of select kernels, which indicates that a good kernel selection algorithm can really boost the utility performance.

6.3.4. DMI exhaustive search

As opposed to the simple DMI filter, DMI exhaustive search is very computational expensive. Nevertheless, this price comes with the benefit of superior performance to the mKDCA without kernel selection. In fact, DMI exhaustive search has the best performance among all methods on MHEALTH, but not so on REALDISP and HAR. Admittedly, this implies that there exists a gap between the DMI score and the test accuracy, which may be partially explained by contending that the training data may not follow the Gaussian distribution as implicitly assumed by DMI on HAR and REALDISP, but does so on MHEALTH.

Another key observation is that DMI forward search’s performance is relatively equal to the optimal DMI exhaustive search’s in all three datasets. However, the proposed DMI forward search provides an added advantage over exhaustive search via computational saving. Hence, it can be claimed that DMI forward search provides a free-lunch in this case.
6.3.5. **DMI forward search**

From the results, DMI forward search is proven to be the best design. It performs better than the baseline without the kernel selection in all three datasets. It outperforms the state-of-the-art Group Lasso, especially on MHEALTH by 3.33%. It requires a much smaller number of kernels than DMI filter while performing better. Finally, it has the computational-saving advantage over the optimal DMI exhaustive search, while still performing generally equally well. Therefore, with all things considered, we conclude that DMI forward search has the highest potential overall.

7. **CONCLUSION AND FUTURE WORKS**

We consider the problem of kernel selection for the mKDCA framework that has been shown to be effective on privacy-preserving classification. The motivation arrives from the fact that mKDCA feature-map is formed by combining many individual KDCA feature-maps and, thus, requires a careful selection of the kernel functions to be included. To address this problem, we propose a kernel selection method called DMI forward search. Via experiments on three mobile-sensing datasets, we show that DMI forward search can outperform the state-of-the-art and yield near optimal performance with much smaller computational cost. Finally, for future works on the application of privacy-preserving machine learning, one direction is to consider incorporating the specified privacy task information into the formulation of the design. For instance, DMI may be modified to consider the privacy label $y_p$, e.g., via the Differential Utility/Cost Analysis (DUCA) [5] or the Ratio Utility and Cost Analysis (RUCA) [28].

**Acknowledgement**

This material is based on work supported in part by the Brandeis Program of Defense Advanced Research Project Agency (DARPA) and Space and Naval Warfare System Center Pacific (SSC Pacific) under Contract No. 66001-15-C-4068.

8. **REFERENCES**


