2010

Feature selection with redundancy-constrained class separability

Luping Zhou
Australian National University, leiw@uow.edu.au

Lei Wang
Australian National University

Chunhua Shen
University of Adelaide, chunhua.shen@adelaide.edu.au

Feature selection with redundancy-constrained class separability

Abstract

Scatter-matrix-based class separability is a simple and efficient feature selection criterion in the literature. However, the conventional trace-based formulation does not take feature redundancy into account and is prone to selecting a set of discriminative but mutually redundant features. In this brief, we first theoretically prove that in the context of this trace-based criterion the existence of sufficiently correlated features can always prevent selecting the optimal feature set. Then, on top of this criterion, we propose the redundancy-constrained feature selection (RCFS). To ensure the algorithm's efficiency and scalability,we study the characteristic of the constraints with which the resulted constrained 0-1 optimization can be efficiently and globally solved. By using the totally unimodular (TUM) concept in integer programming, a necessary condition for such constraints is derived. This condition reveals an interesting special case in which qualified redundancy constraints can be conveniently generated via a clustering of features. We study this special case and develop an efficient feature selection approach based on Dinkelbach's algorithm. Experiments on benchmark data sets demonstrate the superior performance of our approach to those without redundancy constraints.

Keywords

era2015

Disciplines

Engineering | Science and Technology Studies

Publication Details

Zhou, L., Wang, L. & Shen, C. (2010). Feature selection with redundancy-constrained class separability. IEEE Transactions on Neural Networks, 21 (5), 853-858.

Brief Papers

Feature Selection With Redundancy-Constrained Class Separability

Luping Zhou, Lei Wang, and Chunhua Shen

***Abstract*—Scatter-matrix-based class separability is a simple and efficient feature selection criterion in the literature. However, the conventional trace-based formulation does not take feature redundancy into account and is prone to selecting a set of discriminative but mutually redundant features. In this brief, we first theoretically prove that in the context of this trace-based criterion the existence of sufficiently correlated features can always prevent selecting the optimal feature set. Then, on top of this criterion, we propose the redundancy-constrained feature selection (RCFS). To ensure the algorithm's efficiency and scalability, we study the characteristic of the constraints with which the resulted constrained 0–1 optimization can be efficiently and globally solved. By using the totally unimodular (TUM) concept in integer programming, a necessary condition for such constraints is derived. This condition reveals an interesting special case in which qualified redundancy constraints can be conveniently generated via a clustering of features. We study this special case and develop an efficient feature selection approach based on Dinkelbach's algorithm. Experiments on benchmark data sets demonstrate the superior performance of our approach to those without redundancy constraints.**

***Index Terms*—Class separability measure, feature redundancy, feature selection, fractional programming, integer programming.**

I. Introduction

Feature selection plays an important role in pattern recognition [6], [7]. In the literature, scatter-matrix-based class separability has been widely used as a filter-type feature selection criterion and has been well discussed in text books on pattern recognition [5]. This criterion can take the form of determinants or traces of the scatter matrices. Besides its conceptual simplicity and computational efficiency, the trace-based criterion is often favored in the case of small samples, which makes the determinants zero. Recent work [13] shows that the trace-based criterion has an intrinsic relationship with the generalization error bound of support vector machines (SVMs) [1]. That work demonstrates better feature selection performances of the trace-based criterion when the small sample problem and noisy features are encountered. In another recent work [9], a global optimization algorithm is elegantly developed for two feature selection criteria, namely, the Fisher score and the Laplacian score. The former is equivalent to the trace-based class separability criterion.

Although the trace-based class separability criterion has the aforementioned advantages, it is criticized due to its incapability in dealing with the redundancy among features. Directly optimizing this criterion is prone to selecting a set of discriminative but mutually redundant features. For instance, if the most discriminative feature is duplicated several times, this criterion will select all of them. This is problematic for selecting the best set of k features because other discriminative and complementary features will be missed. For example, the redundancy problem is pronounced in a pixel-based image representation, where the intensity values of adjacent pixels often heavily correlate. A classifier with the k features selected in such a way will lead to a poor classification performance. Neither the work in [13] nor that in [9] has addressed this problem.

In this brief, for the trace-based class separability criterion, we first theoretically prove that as long as sufficiently correlated features are added into a feature set, they can always adversely affect the selection result and prevent the optimal feature set from being selected. To address this problem, we propose a redundancy-constrained feature selection (RCFS) approach. Formulating feature selection with the trace-based criterion as a 0–1 linear fractional program problem, our approach imposes extra constraints in order to avoid selecting redundant features. However, solving a constrained 0–1 program is NP-hard in general and there are no general polynomial-time algorithms to date [10]. Current optimization techniques, such as the cutting plane and the enumerative methods, become computationally expensive or even intractable for feature selection over a large number of features. To achieve efficient selection, we further study the type of constraints that enables the resulted constrained 0–1 optimization problem to be efficiently and globally solved. Based on the totally unimodular (TUM) condition in integer programming [11], we derive a necessary condition for the applicable constraints. Under this framework, we then discuss an interesting special case, in which such a type of constraints can be conveniently generated via a clustering of features. An efficient feature selection approach is developed accordingly based on the Dinkelbach's algorithm for linear fractional programming [3], [8], [12]. The proposed approach is tested on benchmark data sets with different types of features. Results show that with the features selected by our approach, an SVM classifier can achieve significant improvement on classification performances.

II. Background and Related Work

Let $(\mathbf{x}, y) \in (\mathbb{R}^n \times \mathcal{Y})$ be a sample, where $\mathbb{R}^n$ is an n-dimensional feature space and $\mathcal{Y} = \{1, 2, \ldots, s\}$ is the label set. l_i is the number of samples in the ith class, and l is the total number of samples. Let $\mathbf{x}_{ij}$ denote the jth sample in the ith class, $\boldsymbol{m}_i$ the sample mean of the ith class, and $\boldsymbol{m}$ the sample mean of all classes. The within-class scatter matrix $\mathbf{S}_W$, between-class scatter matrix $\mathbf{S}_B$, and total scatter matrix $\mathbf{S}_T$ are defined as

$$\begin{aligned}\mathbf{S}_W &= \sum_{i=1}^{s}\sum_{j=1}^{l_i}(\mathbf{x}_{ij}-\mathbf{m}_i)(\mathbf{x}_{ij}-\mathbf{m}_i)^\top \\ \mathbf{S}_B &= \sum_{i=1}^{s} l_i(\mathbf{m}_i-\mathbf{m})(\mathbf{m}_i-\mathbf{m})^\top \\ \mathbf{S}_T &= \sum_{i=1}^{s}\sum_{j=1}^{l_i}(\mathbf{x}_{ij}-\mathbf{m})(\mathbf{x}_{ij}-\mathbf{m})^\top. \end{aligned} \tag{1}$$

Large class separability means small within-class scattering and large between-class scattering. A combination of two of them can be used as

Manuscript received June 30, 2009; revised December 03, 2009; accepted February 17, 2010. Date of publication March 11, 2010; date of current version April 30, 2010. National ICT Australia is supported by the Australian Government's Backing Australia's Ability initiative, in part through the Australian Research Council.

L. Zhou and L. Wang are with the School of Engineering, The Australian National University, Canberra, A.C.T. 0200, Australia (e-mail: luping.zhou.jane@googlemail.com; Lei.Wang@anu.edu.au).

C. Shen is with the Canberra Research Laboratory, National ICT Australia (NICTA), Canberra, A.C.T. 2601, Australia (e-mail: chhshen@gmail.com).

Color versions of one or more of the figures in this brief are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNN.2010.2044189

a measure, for example, $\mathrm{tr}(\mathbf{S}_B)/\mathrm{tr}(\mathbf{S}_W)$ or $|\mathbf{S}_B|/|\mathbf{S}_W|$, where $\mathrm{tr}(\cdot)$ and $|\cdot|$ denote the trace and determinant of a matrix, respectively. For the convenience of analysis, $\mathrm{tr}(\mathbf{S}_B)/\mathrm{tr}(\mathbf{S}_T)$ is used throughout this paper. In terms of maximization, it is identical to $\mathrm{tr}(\mathbf{S}_B)/\mathrm{tr}(\mathbf{S}_W)$ because $\mathbf{S}_T = \mathbf{S}_B + \mathbf{S}_W$.

From the perspective of graph-based feature selection, the work in [9] proposes an algorithm to globally maximize the Fisher score over a $\{0,1\}$ transform matrix $\mathbf{W}$

$$\mathbf{W}^\star = \arg\max_{\mathbf{W}} = \frac{\mathrm{tr}(\mathbf{W}^\top \mathbf{B}\mathbf{W})}{\mathrm{tr}(\mathbf{W}^\top \mathbf{E}\mathbf{W})}. \tag{2}$$

Let k be the number of features to be selected. The $n \times k$ matrix $\mathbf{W}$ is expressed as $[\mathbf{w}_{i_1}, \mathbf{w}_{i_2}, \ldots, \mathbf{w}_{i_k}]$, where the column vector $\mathbf{w}_{i_j}$ has one and only one "1" at its i_jth element. The set $\{i_1, i_2, \ldots, i_k\}$ is a subset of $\{1, 2, \ldots, n\}$. Matrices $\mathbf{B}$ and $\mathbf{E}$ are defined as $\mathbf{B} = \mathbf{X}\mathbf{L}_b\mathbf{X}^\top$ and $\mathbf{E} = \mathbf{X}\mathbf{L}_w\mathbf{X}^\top$, respectively. The column vectors of $\mathbf{X}$ are the l samples. $\mathbf{L}_b$ and $\mathbf{L}_w$ are the Laplacian matrices of the weighted undirected graphs reflecting the between-class and within-class relationship of the samples. Nie *et al.* [9] solve (2) by iteratively solving the following subproblem:

$$\begin{cases} \mathbf{W}_{i+1} = \arg\max_{\mathbf{W}} \mathrm{tr}\left(\mathbf{W}^\top \mathbf{B}\mathbf{W} - \lambda_i \mathbf{W}^\top \mathbf{E}\mathbf{W}\right) \\ \lambda_{i+1} = \dfrac{\mathrm{tr}(\mathbf{W}_{i+1}^\top \mathbf{B}\mathbf{W}_{i+1})}{\mathrm{tr}(\mathbf{W}_{i+1}^\top \mathbf{E}\mathbf{W}_{i+1})}. \end{cases} \tag{3}$$

Recognizing that the optimal $\lambda^\star$ is the root of a piecewise linear function $f(\lambda) = \max\ \mathrm{tr}\left(\mathbf{W}^\top \mathbf{B}\mathbf{W} - \lambda \mathbf{W}^\top \mathbf{E}\mathbf{W}\right)$, the work in [9] proves the convergence.

Through simple manipulation, we can show that 1) $\mathbf{B}$ and $\mathbf{E}$ are essentially $\mathbf{S}_B$ and $\mathbf{S}_W$, respectively, and 2) $\mathrm{tr}(\mathbf{W}^\top \mathbf{B}\mathbf{W})$ exactly selects k diagonal elements of $\mathbf{B}$ and computes its sum, and this observation also applies to $\mathrm{tr}(\mathbf{W}^\top \mathbf{E}\mathbf{W})$. Hence, the Fisher score part in [9] is exactly the same as the feature selection with the trace-based class separability. However, the work in [9] aims to develop an efficient and global optimization algorithm and does not consider the feature redundancy. In what follows, we address this important issue.

III. The Feature Redundancy Problem in Feature Selection

The following gives a theoretical analysis of the feature redundancy problem. From the definition in (1), the tth diagonal elements of $\mathbf{S}_B$ and $\mathbf{S}_T$ can be obtained as

$$\begin{cases} f_t = \sum_{i=1}^{s} l_i (m_{it} - m_t)^2 \geq 0 \\ g_t = \sum_{i=1}^{s} \sum_{j=1}^{l_i} (x_{ijt} - m_t)^2 \geq 0 \end{cases} \tag{4}$$

where x_{ijt} is the tth feature of $\mathbf{x}_{ij}$, and m_{it} and m_t are the tth features of $\mathbf{m}_i$ and $\mathbf{m}$, respectively. f_t and g_t characterize the between-class scattering and total scattering information from the tth feature. Let $\boldsymbol{\omega} \in \{0,1\}^n$ denote a binary selector. Selecting k out of n features based on the trace-based class separability criterion can thus be formulated as a 0–1 linear fractional optimization problem. Note that $g_t > 0$ can always be ensured by removing constant features

$$\boldsymbol{\omega}^\star = \arg\max_{\boldsymbol{\omega}} \frac{f_1\omega_1 + \cdots + f_n\omega_n}{g_1\omega_1 + \cdots + g_n\omega_n} = \arg\max_{\boldsymbol{\omega}} \frac{\mathbf{f}^\top \boldsymbol{\omega}}{\mathbf{g}^\top \boldsymbol{\omega}}$$
$$\text{subject to } \boldsymbol{\omega} \in \{0,1\}^n, \qquad \mathbf{1}^\top \boldsymbol{\omega} = k. \tag{5}$$

As can be seen, feature selection here is to find k features $x_{i_1}, x_{i_2}, \ldots, x_{i_k}$ that give the maximal value of $(f_{i_1} + \cdots + f_{i_k})/(g_{i_1} + \cdots + g_{i_k})$.

Case I: In the presence of duplicated features, we obtain the following result.[1]

Theorem 1: Let (g_p, f_p) satisfy $f_p/g_p \geq f_i/g_i,\ \forall i = 1, \ldots, n$. If the feature x_p is duplicated for k times, it will be repeatedly selected for k times.

Case II: In the presence of correlated features, we obtain similar results although the analysis is slightly complicated.

Lemma 1: In feature selection with the trace-based class separability criterion, the feature x_i with the largest f_i/g_i value must be selected.

Lemma 2: For any two features x_i and x_j, if $\exists\ \epsilon > 0$ such that their correlation coefficient $|\rho(x_i, x_j)| \geq 1 - \epsilon$, then

$$\left| \frac{f_i}{g_i} - \frac{f_j}{g_j} \right| \leq 2\sqrt{2} l \sqrt{\epsilon}$$

where l is the total number of samples.

This lemma indicates that for two sufficiently correlated features, their f/g values will be sufficiently close to each other. With these lemmas, we have the following result.

Theorem 2: In feature selection with the trace-based class separability criterion, introducing features that are sufficiently correlated to the most discriminative feature can always prevent the optimal feature subset from being selected.

Proof: Without loss of generality, denote the optimal k features selected by the trace-based criterion as $x_1, \ldots, x_k$, and x_1 has the largest f/g value. Then, the optimal (maximum) criterion value is

$$h^0 = \frac{f_1 + f_2 + \cdots + f_k}{g_1 + g_2 + \cdots + g_k}.$$

Separate the k features into two disjoint sets $\mathcal{S}_1$ and $\mathcal{S}_2$, where $\mathcal{S}_1 = \{x_i | f_i/g_i > h^0\}$ and $\mathcal{S}_2 = \{x_i | f_i/g_i \leq h^0\}$. In general, none of $\mathcal{S}_1$ and $\mathcal{S}_2$ is empty for $k \geq 2$. By Lemma 1, x_1 must be selected. Now let $\eta = f_1/g_1 - h^0$ and $\epsilon = (1/8l^2)\eta^2$. Introduce features $\tilde{x}_{1i}$, $i = (1, \ldots, |\mathcal{S}_2|)$, which satisfies that $|\rho(x_1, \tilde{x}_{1i})| > 1 - \epsilon$. By Lemma 2, it can be obtained that

$$\left| \frac{f_{1i}}{g_{1i}} - \frac{f_1}{g_1} \right| < \eta \Longrightarrow \frac{f_{1i}}{g_{1i}} > h^0.$$

Define $\mathcal{S}_3 = \mathcal{S}_1 \cup \{\tilde{x}_{1i} | i = 1, \ldots, |\mathcal{S}_2|\}$. Since $f_j/g_j > h^0$ for any feature $x_j \in \mathcal{S}_3$, it then holds that

$$\frac{\sum_{x_j \in \mathcal{S}_3} f_j}{\sum_{x_j \in \mathcal{S}_3} g_j} > h^0.$$

Thus, the original optimal k features will not be entirely selected any more.

Note that for both Cases I and II, the obtained results can be readily extended to the features other than the most discriminative one. The extension is omitted here.

IV. The Proposed RCFS Approach

A. Basic Problem

To address the above problem, we impose extra constraints to prevent selecting redundant features. Maximizing $\mathbf{f}^\top \boldsymbol{\omega}/\mathbf{g}^\top \boldsymbol{\omega}$ in (5) is a linear fractional programming (LFP) problem. It can be solved by Dinkelbach's algorithm which is a general algorithm for optimizing $\phi(\boldsymbol{\omega})/\psi(\boldsymbol{\omega})$ with $\psi(\boldsymbol{\omega}) > 0$. It converts the problem to a sequence of subproblems of optimizing $\phi(\boldsymbol{\omega}) - \lambda\psi(\boldsymbol{\omega})$. As long as the subproblem

[1]The proofs for some lemmas and theorems can be found from http://users.cecs.anu.edu.au/~wanglei/

is globally solvable in polynomial time, the global optimum of the original problem can be found in polynomial time.

In our case, let $\mathbf{\Omega}$ be the set of newly added constraints. Problem (5) becomes

$$\boldsymbol{\omega}^\star = \arg\max_{\boldsymbol{\omega}} \frac{\mathbf{f}^\top \boldsymbol{\omega}}{\mathbf{g}^\top \boldsymbol{\omega}} \quad \text{subject to } \boldsymbol{\omega} \in \{0,1\}^n,\ \mathbf{1}^\top \omega = k, \text{and } \boldsymbol{\omega} \in \mathbf{\Omega}. \tag{6}$$

Dinkelbach's algorithm iteratively solves the subproblem

$$z(\lambda) \triangleq \max_{\boldsymbol{\omega}} \left(\mathbf{f}^\top \boldsymbol{\omega} - \lambda \mathbf{g}^\top \boldsymbol{\omega}\right) \quad \text{subject to } \boldsymbol{\omega} \in \{0,1\}^n,\ \mathbf{1}^\top \boldsymbol{\omega} = k,\ \boldsymbol{\omega} \in \mathbf{\Omega} \tag{7}$$

where $\lambda = \mathbf{f}^\top \boldsymbol{\omega}^- / \mathbf{g}^\top \boldsymbol{\omega}^-$, with $\boldsymbol{\omega}^-$ being the solution of last iteration. The function $z(\lambda)$ is a piecewise linear, convex, and strictly decreasing function. When solving (7) leads to $z(\lambda) = 0$, the obtained $\boldsymbol{\omega}^\star$ will also be the optimal solution of (6). The work in [9] essentially solves (6) and (7) without the constraint of $\boldsymbol{\omega} \in \mathbf{\Omega}$ from another perspective. To maintain efficient feature selection, the subproblem in (7) has to be sufficiently easy to solve. However, in the presence of $\boldsymbol{\omega} \in \{0,1\}^n$, arbitrarily adding constraints will make the subproblem very difficult, even if we restrict ourselves to linear constraints.

B. Integer Linear Programming With TUM Condition

When $\mathbf{\Omega}$ contains linear constraints only, (7) is an integer linear programming (ILP) problem. ILP is much more difficult to solve than LP, and there are no general polynomial-time algorithms so far. However, the situation will be different if the coefficient matrix of an ILP problem fortunately satisfies the TUM condition. In this case, solving its LP relaxed version solves the original ILP. An LP problem can be trivially solved with off-the-shelf packages. For our problem, by relaxing $\boldsymbol{\omega} \in \{0,1\}^n$ to $\boldsymbol{\omega} \in [0,1]^n$, the feasible region of (7) becomes

$$R(\boldsymbol{\omega}) = \{\boldsymbol{\omega} \in \mathbb{R}^n : \boldsymbol{\omega} \in [0,1]^n,\ \mathbf{1}^\top \boldsymbol{\omega} = k,\ \boldsymbol{\omega} \in \mathbf{\Omega}\}.$$

Since all the constraints are linear, the feasible region can be written in a matrix form as

$$R(\boldsymbol{\omega}) = \{\boldsymbol{\omega} \in \mathbb{R}^n : \mathbf{A}\boldsymbol{\omega} \leq \mathbf{b}, \boldsymbol{\omega} \geq 0\}. \tag{8}$$

Geometrically, $R(\boldsymbol{\omega})$ is a polyhedron. According to Hoffman and Kruskal's theorem [11], for each integral vector $\mathbf{b}$, $R(\boldsymbol{\omega})$ is an *integral* polyhedron if and only if $\mathbf{A}$ is TUM. Because the optimal solution of an LP problem is always at one of the vertexes of the polyhedron, solving the LP relaxed version will obtain the optimal integral solution for the ILP problem. Hence, for efficient feature selection, $\mathbf{A}$ in (8) has to be TUM after the extra redundancy constraints are imposed.

By definition, a TUM matrix is a matrix with the determinants of all of its square submatrices equaling either $+1$, -1, or 0. In the literature, a set of sufficient and necessary conditions has been given to check whether a matrix is TUM. Besides, a general polynomial-time algorithm has been developed to do this job when the above conditions cannot be conveniently checked. The following highlights several properties of TUM [11] that will be used in this paper.

- P1) TUM is preserved under the operations of permuting rows or columns or taking transpose.
- P2) TUM is preserved under the operations of multiplying a row or column by -1 or repeating a row or column.
- P3) If a matrix $\mathbf{A}$ is TUM, then the matrix $[\mathbf{A}\ \mathbf{I}]$ is TUM, where $\mathbf{I}$ denotes an identity matrix.

C. TUM Condition in Feature Selection Problems

As shown in the subproblem (7), there is a specific constraint $\mathbf{1}^\top \boldsymbol{\omega} = k$ due to feature selection. It is expressed as $\mathbf{1}^\top \boldsymbol{\omega} \leq k$ and $(-\mathbf{1})^\top \boldsymbol{\omega} \leq -k$, inducing one row of "$+1$" and one row of "$-1$" in $\mathbf{A}$. The constraint $\boldsymbol{\omega} \leq \mathbf{1}$ induces an identity matrix $\mathbf{I}$ in $\mathbf{A}$. If $\mathbf{\Omega}$ is not imposed, $\mathbf{A}$ simply satisfies the TUM condition due to P2) and P3). This explains, from the perspective of integer programming, why the case without constraints (the subproblem in [9]) can be conveniently solved. Now we study how the presence of $\mathbf{\Omega}$ restricts $\mathbf{A}$ and the following result is obtained.

Theorem 3: With the existence of the constraint $\mathbf{1}^\top \boldsymbol{\omega} = k$, a necessary condition for $\mathbf{A}$ to be TUM is that there is no -1 and $+1$ appearing in the same row of $\mathbf{A}$.

This means that each row of $\mathbf{A}$ can only contain 1) 0 and/or $+1$ or 2) 0 and/or -1. This indicates that the redundancy constraints can only take the following two forms:

$$\sum_{x_i \in \mathcal{S}} \omega_i \leq b \quad \text{or} \quad \sum_{x_i \in \mathcal{S}'} (-\omega_i) \leq b'$$

where $\mathcal{S}$ and $\mathcal{S}'$ are two subsets of $\{x_1, \ldots, x_n\}$. This result is important. It gives a clear idea about the linear constraints that could be used to constrain feature redundancy and maintain the TUM property of $\mathbf{A}$.

Although the above necessary condition is restrictive, constraints can still be designed to effectively avoid selecting redundant features. Here, we consider the *interval* matrix, which contains 0 and 1 elements only and has consecutive "1"s in each row like

$$(0, \ldots, 0, 1, 1, \ldots, 1, 0, \ldots, 0).$$

Every interval matrix is TUM [11], and it satisfies the necessary condition in Theorem 3. Furthermore, we consider a special interval matrix where the "1"s in different rows have different positions (Or, equally, each column of this matrix has one and only one "1"), and it is called *partition* matrix in this work. Clearly, for our problem, each partition matrix uniquely defines an exhaustive and mutually exclusive partition of n features. This links the sophisticated TUM condition to feature clustering technique in textbooks [4]. We now study this interesting special case and develop an efficient redundancy-constrained selection algorithm.

D. Our Redundancy-Constrained Selection Algorithm

Let $\{x_1, x_2, \ldots, x_n\}$ be the n features. We define $d(x_i, x_j)$ as the "distance" between x_i and x_j reflecting their independence or complementarity. $d(x_i, x_j)$ can be defined based on correlation coefficient, mutual information, or any criterion characterizing feature redundancy. This work simply uses the correlation coefficient and defines $d(x_i, x_j) = 1 - |\rho(x_i, x_j)|$. Let $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_m$ be the obtained m clusters

$$\{x_1, x_2, \ldots, x_n\} = \cup_{i=1}^m \mathcal{C}_i \quad \text{and} \quad \mathcal{C}_i \cap \mathcal{C}_j = \emptyset.$$

Their sizes are denoted by $c_1, c_2, \ldots, c_m$. Since the features within the same cluster are regarded as being sufficiently correlated, a natural constraint for avoiding selecting redundant features is to enforce that at most p_i features are selected from $\mathcal{C}_i$. Let $x_{i_1}, x_{i_2}, \ldots, x_{i_{c_i}}$ be the features in $\mathcal{C}_i$. This enforces that

$$\omega_{i_1} + \omega_{i_2} + \cdots + \omega_{i_{c_i}} \leq p_i. \tag{9}$$

TABLE I
REDUNDANCY-CONSTRAINED FEATURE SELECTION (RCFS)

Input: l training samples $\{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$ and the value of k,
Output: optimal binary selector $\boldsymbol{\omega}$.
Initialization:
  hierarchically cluster n features with a predefined distance,
  establish linear constraints $\boldsymbol{\Omega}$ accordingly,
  compute g_i and f_i $(i = 1, 2, \cdots, n)$ for each feature,
  initialize k components of $\boldsymbol{\omega}$ as "1" and the remaining as "0",
Feature selection on each level with the Dinkelbach's algorithm:
  (1) **Set** $\lambda = \mathbf{f}^\top \boldsymbol{\omega}^- / \mathbf{g}^\top \boldsymbol{\omega}^-$,
  (2) **Solve** the maximization problem in (7)
  (3) If $\mathbf{f}^\top \boldsymbol{\omega} - \lambda \mathbf{g}^\top \boldsymbol{\omega} < \xi$ (e.g., 10^{-4}), $\boldsymbol{\omega}$ is optimal.
    Otherwise, set $\boldsymbol{\omega}^- = \boldsymbol{\omega}$ and go to (1).
Cross-validation to identify the best feature selection.

Accordingly, $\mathbf{A}\boldsymbol{\omega} \leq \mathbf{b}$ in (8) can be explicitly written as

$$\begin{pmatrix} & \mathbf{1}_{1\times n} & & \\ & -\mathbf{1}_{1\times n} & & \\ -- & -- & -- & -- \\ \mathbf{1}_{1\times c_1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1}_{1\times c_2} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1}_{1\times c_m} \\ -- & -- & -- & -- \\ & \mathbf{I}_{n\times n} & & \end{pmatrix} \begin{pmatrix} \omega_{r1} \\ \omega_{r2} \\ \vdots \\ \omega_{rn} \end{pmatrix} \leq \begin{pmatrix} k \\ -k \\ -- \\ p_1 \\ \vdots \\ p_m \\ -- \\ \mathbf{1}_{n\times 1} \end{pmatrix}. \quad (10)$$

Note that for convenience of exposition, $\boldsymbol{\omega}$ has been rearranged as $(\omega_{r1}, \ldots, \omega_{rn})^\top$ according to the order of $x_1, \ldots, x_n$ appearing in $\mathcal{C}_1, \ldots, \mathcal{C}_m$. $\mathbf{1}_{1\times c_i}$ is a row vector of "1"s with length c_i. The middle part of $\mathbf{A}$ is a partition matrix satisfying the TUM condition. Incorporating the $\mathbf{1}_{1\times n}$ in first row makes it an interval matrix, still being TUM. Hence, the whole $\mathbf{A}$ is TUM due to P1)–P3). Thus, the resulting subproblem can be efficiently solved because of the equivalence of ILP and LP in this case.

A practical issue of (10) is that there are many algorithmic parameters, including m and $p_1, \ldots, p_m$. Optimally presetting them is impractical. We alleviate this problem by applying agglomerative hierarchical clustering. Starting with the n features at the bottom level, two features (or subclusters later) are merged at each level until only k clusters are left, where k is the number of features to be selected. This gives a hierarchy of $n - k + 1$ levels, each of which is a partition of the n features. Then, feature selection is performed at each level by setting all of $p_1, \ldots, p_m$ as 1. Multifold cross validation is used to identify the best selection from the $n - k + 1$ levels. The advantages of using the hierarchical clustering are as follows. 1) We need not set m. Instead, features are clustered with different degree of redundancy in this hierarchy. 2) We only need to set all p_i as 1. Because each cluster at a given level is formed by multiple clusters at the preceding levels, the case of $p_i > 1$ can be implicitly approximated by a group of $p_j = 1$ in the preceding levels. 3) The identity matrix $\mathbf{I}$ in $\mathbf{A}$ can be ignored. 4) This will not significantly prolong feature selection because only LP problems are solved and Dinkelbach's algorithm usually terminates in several iterations. The whole algorithm is summarized in Table I.

Finally, it is worth noting that with the particular constraints in (10), we can even solve (7) with simple sorting operations, as the case without constraints in [9]. However, this is only a special case and the inclusion of one more constraint may make LP techniques have to be used. By characterizing (7) from the perspective of TUM, we provide a more essential and general observation for this subproblem.

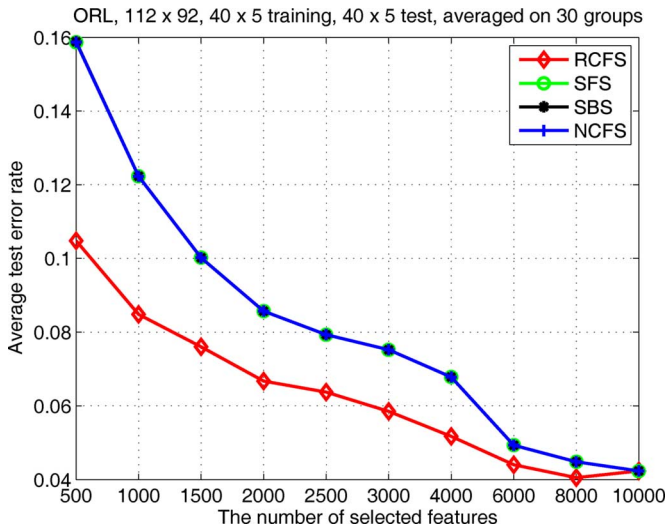

Fig. 1. Comparison of the SVM classifiers using the features selected by each of the four approaches on ORL (112×92).

V. EXPERIMENTS

The experiments compare the RCFS with the nonconstrained feature selection (NCFS), the sequential forward selection (SFS), and the sequential backward selection (SBS). Given a training set, the correlation coefficient ρ_{ij} is computed between each pair of features and is used to hierarchically cluster all features to generate the redundancy constraints. The four selection approaches are applied to the training set respectively, and a linear SVM classifier with the k selected features is trained. After that, the SVM classifier is tested on an independent test set. The approach achieving a lower test error rate is considered as the better one. To ensure fair comparison, the hyperparameter of the SVM classifier is optimally tuned for each approach via fivefold cross validation. Besides comparing test error rates, the McNemar test (with significance level $\alpha = 0.05$) [2] is conducted to check whether the difference of two SVM classifiers is statistically significant. Four benchmark data sets with strongly correlated features are used, including ORL (112×92 pixels), United States Postal Service (USPS), and the Vehicle and Dermatology data sets from the University of California at Irvine (UCI) Machine Learning Repository.[2] They have different types of features and involve the problems in different areas.

A. ORL Facial Image Data

This database consists of 40 subjects, each of which has ten gray-level facial images of size 112×92, leading to 10 304 features. By randomly sampling five images from each subject, 400 images are split into 30 pairs of training/test subsets with equal size of 200. Each pair forms a 40-class classification problem. The test error rate of each SVM classifier is averaged on the 30 pairs and plotted in Fig. 1. The classifier with RCFS-selected features consistently achieves the best performance. NCFS, SFS, and SBS lead to (almost identically) poorer performances, indicating that feature selection without constraining feature redundancy is inferior regardless of the search strategy. To verify the improvement, the McNemar test is conducted between RCFS and NCFS on each test subset. The result is summarized in Table II. Each number given in this table is a summary of McNemar tests on the 30 test sets. The number is generated through two steps: 1) check whether the χ^2 value of the test statistic is larger than 3.8415 ($\alpha = 0.05$) on each of the 30 test sets; and 2) if yes, compare the test error rates of the SVM classifiers using the features selected by RCFS and NCFS.

[2]http://archive.ics.uci.edu/ml/datasets.html

TABLE II
SIGNIFICANCE TEST OF SVM TEST ERRORS (WITH $\alpha = 0.05$)

	Number of selected features (ORL (112×92))						
McNemar Test	0.5K	1K	1.5K	2K	3K	4K	6K
# RCFS better	8	9	5	4	2	2	0
# RCFS worse	0	0	0	0	0	0	0
# No difference	22	21	25	26	28	28	30
	Number of selected features (USPS)						
McNemar Test	16	32	64	100	128	160	192
# RCFS better	1	1	1	1	1	0	0
# RCFS worse	0	0	0	0	0	0	0
# No difference	0	0	0	0	0	1	1
	Number of selected features (Vehicle)						
McNemar Test	3	5	8	10	12	15	
# RCFS better	76	47	67	59	17	47	
# RCFS worse	0	0	0	0	1	0	
# No difference	24	53	33	41	82	53	
	Number of selected features (Dermatology)						
McNemar Test	5	10	15	20	25	30	
# RCFS better	100	96	95	61	8	1	
# RCFS worse	0	0	0	0	1	0	
# No difference	0	4	5	39	92	99	

If the test error rate of RCFS is lower, RCFS is declared "better" and "worse" otherwise. As shown in Table II, when 500 features are selected, RCFS is significantly better than NCFS on eight out of 30 test sets and is comparable on the remaining 22. With different number of selected features, RCFS always gives feature selection that is better than or comparable to that of NCFS, particularly when a small number of features are selected. This confirms the advantage of RCFS on the ORL data set.

To further investigate the efficacy of RCFS in dealing with redundancy, we visually show which pixels are selected as well. The result of the first training/test pair with $k = 500$ is used. The optimal binary selector $\boldsymbol{\omega}^\star$ is reshaped to a 112×92 matrix and displayed in Fig. 5. Each black dot is a "1" in $\boldsymbol{\omega}^\star$, indicating that the corresponding pixel is selected. The pixels selected by NCFS roughly form three compact clusters: two upper corners and the forehead area. The selection of two corners may be because the background of facial images is identical for the same subject but changes across different subjects.[3] Adjacent pixels are intensively selected by NCFS although they are strongly correlated. The pixels selected by RCFS are well scattered in the whole image. This shows the effect of redundancy constraints. It avoids repeatedly selecting correlated features and allows more information to be brought in.

B. USPS, Vehicle, and Dermatology Data

The USPS data set contains predefined 7291 training and 2007 test handwritten digit images. They form ten classes of digits from "0" to "9." Each digit image is characterized by 256 features by reshaping a 16×16 gray-level image. The Vehicle data set is to classify four types of vehicles based on their 2-D silhouettes. Each sample is represented by 18 shape features. The 846 samples are randomly split into 100 training and test pairs with equal size. The Dermatology data set is to discriminate six types of erythemato–squamous diseases, which is a real problem in dermatology. It has 366 samples (patients), each of which is represented by 34 features. Again, the data set is randomly split into 100 training and test pairs with equal size. The four approaches are compared as before. As shown in Figs. 2–4, RCFS is superior or comparable to the other three and has never performed worse. The most significant improvement is attained on the Dermatology data set, reaching about 30% when five features are selected. In general, the smaller the number of selected features, the more significant is the improvement. The statistical test result in Table II confirms the above observation.

[3]See the images shown at http://www.cl.cam.ac.uk/research/dtg/attarchive/facesataglance.html

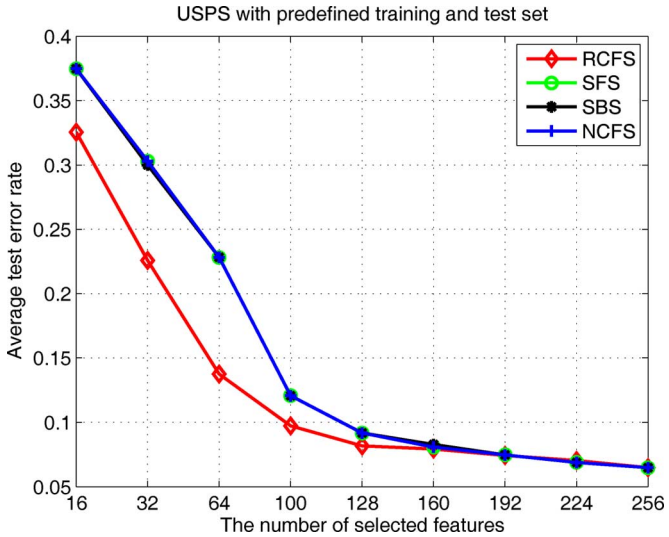

Fig. 2. Comparison of the SVM classifiers using the features selected by each of the four approaches on USPS data set.

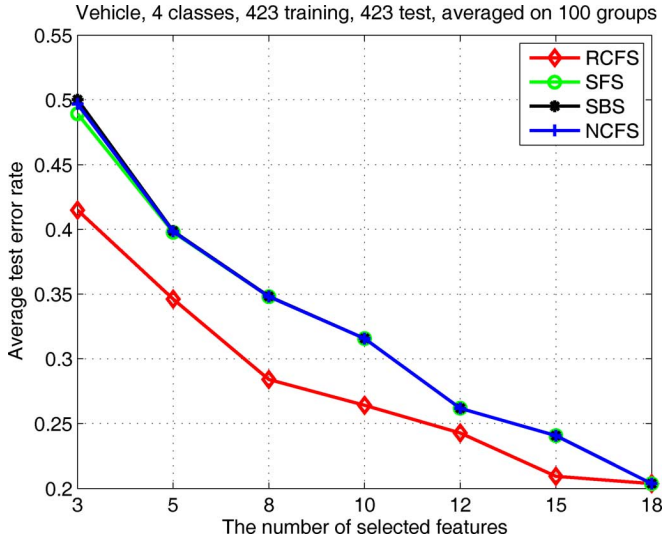

Fig. 3. Comparison of the SVM classifiers using the features selected by each of the four approaches on Vehicle data set.

C. Computational Cost

Take the ORL (112×92) data set as an example, which selects 500 out of 10 304 features. A Linux server with 2.8-GHz central processing unit (CPU) and 4.0 GB memory is used. All approaches are implemented in Matlab. The average feature selection time of NCFS, SFS, and SBS on 30 groups is 2.7, 3.1, and 17.3 s, respectively. Given redundancy constraints, RCFS averagely takes 4.1 s to solve (10) to select the optimal features. Certainly, RCFS will need more time if the time used by hierarchical clustering (for generating redundancy constraints, which takes 28.4 min) and cross validation (for identifying the best selection result, which takes 7.1 min) is also taken into account. Nevertheless, the extra computations of RCFS can well be justified by the significant improvement on feature selection performance. In addition, because we focus on analyzing and solving the constrained integer programming problem, basic hierarchical clustering and k-fold cross

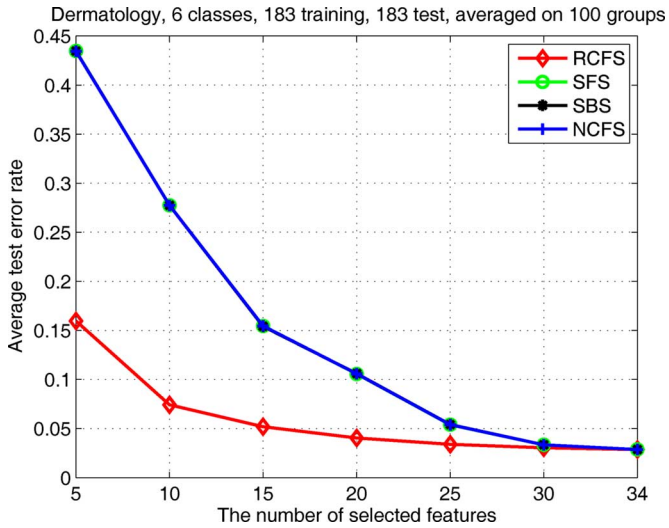

Fig. 4. Comparison of the SVM classifiers using the features selected by each of the four approaches on Dermatology data set.

(a) (b)

Fig. 5. Pixels selected by (a) NCFS and (b) RCFS (black dots, $k = 500$).

validation of SVMs are simply employed. More efficient ways can be explored to speed up clustering and cross validation in the future work.

In summary, the experimental result verifies the advantages of the proposed RCFS. It indicates that feature correlation can happen to different feature representations and different problems. Carefully addressing this issue can lead to significant improvement on classification performance.

VI. Conclusion

This brief studies feature selection with the trace-based class separability criterion in the presence of feature redundancy. We theoretically show the adverse affect of feature redundancy to feature selection. Moreover, we discuss the redundancy constraints that can guarantee optimal and efficient feature selection based on the TUM condition in integer programming. A special case is then studied, in which feature clustering is used to generate the constraints naturally satisfying the TUM condition. Experimental results demonstrate the effectiveness and advantages of the proposed RCFS.

References

[1] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge, U.K.: Cambridge Univ. Press, 2000.

[2] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural Comput.*, vol. 10, pp. 1895–1923, 1998.

[3] W. Dinkelbach, "On nonlinear fractional programming," *Manage. Sci.*, vol. 13, no. 7, pp. 492–498, Mar. 1967.

[4] R. O. Duda, D. G. Stork, and P. E. Hart, *Pattern Classification*, 2nd ed. New York: Wiley, 2001.

[5] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. New York: Academic, 1990.

[6] A. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 2, pp. 153–158, Feb. 1997.

[7] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 4, pp. 491–502, Apr. 2005.

[8] T. Matsui, Y. Saruwatari, and M. Shigeno, "An analysis of Dinkelbach's llgorithm for 0–1 fractional programming problems," Dept. Math. Eng. Inf. Phys., Univ. Tokyo, Tokyo, Japan, METR92-14, 1992.

[9] F. Nie, S. Xiang, Y. Jia, C. Zhang, and S. Yan, "Trace ratio criterion for feature selection," in *Proc. 23rd AAAI Conf. Artif. Intell.*, Jul. 2008, pp. 671–676.

[10] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. New York: Dover, 1998.

[11] A. Schrijver, *Theory of Linear and Integer Programming*. New York: Wiley, 1986.

[12] C. Shen, H. Li, and M. J. Brooks, "Supervised dimensionality reduction via sequential semidefinite programming," *Pattern Recognit.*, vol. 41, no. 12, pp. 3644–3652, 2008.

[13] L. Wang, "Feature selection with kernel class separability," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 9, pp. 1534–1546, Sep. 2008.

A Convolutional Learning System for Object Classification in 3-D Lidar Data

Danil Prokhorov

***Abstract*—In this brief, a convolutional learning system for classification of segmented objects represented in 3-D as point clouds of laser reflections is proposed. Several novelties are discussed: 1) extension of the existing convolutional neural network (CNN) framework to direct processing of 3-D data in a multiview setting which may be helpful for rotation-invariant consideration, 2) improvement of CNN training effectiveness by employing a stochastic meta-descent (SMD) method, and 3) combination of unsupervised and supervised training for enhanced performance of CNN. CNN performance is illustrated on a two-class data set of objects in a segmented outdoor environment.**

***Index Terms*—Convolutional neural network (CNN), multiview input, stochastic meta-descent (SMD), unsupervised and supervised learning.**

I. Introduction

Convolutional learning systems and, in particular, convolutional neural networks are loosely inspired by biological vision systems. They have been applied successfully to a variety of problem in computer vision [5], [6].

Manuscript received August 11, 2009; revised December 07, 2009 and February 22, 2010; accepted February 23, 2010. Date of publication March 29, 2010; date of current version April 30, 2010.

The author is with the Toyota Research Institute NA, Ann Arbor, MI 48105 USA (e-mail: dvprokhorov@gmail.com).

Color versions of one or more of the figures in this brief are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNN.2010.2044802