Efficient Spectral Feature Selection with Minimum Redundancy

Zheng Zhao
Computer Science and Engineering
Arizona State University
Tempe, AZ, USA
zhaozheng@asu.edu

Lei Wang
College of Engineering and Computer Science, Australian National University
Canberra, ACT, Australia
lei.wang@anu.edu.au

Huan Liu
Computer Science and Engineering
Arizona State University
Tempe, AZ, USA
huan.liu@asu.edu

Abstract

Spectral feature selection identifies relevant features by measuring their capability of preserving sample similarity. It provides a powerful framework for both supervised and unsupervised feature selection, and has been proven to be effective in many real-world applications. One common drawback associated with most existing spectral feature selection algorithms is that they evaluate features individually and cannot identify redundant features. Since redundant features can have significant adverse effect on learning performance, it is necessary to address this limitation for spectral feature selection. To this end, we propose a novel spectral feature selection algorithm to handle feature redundancy, adopting an embedded model. The algorithm is derived from a formulation based on a sparse multi-output regression with a $L_{2,1}$-norm constraint. We conduct theoretical analysis on the properties of its optimal solutions, paving the way for designing an efficient path-following solver. Extensive experiments show that the proposed algorithm can do well in both selecting relevant features and removing redundancy.

Introduction

Handling high-dimensional data represents one of the most challenging problems for learning. Given huge number of features, learning algorithms can overfit data and become less comprehensible. Feature selection is one effective means to reduce dimensionality by removing irrelevant and redundant features (Guyon and Elisseeff 2003; Liu and Motoda 1998). In recent years, researchers designed spectral feature selection algorithms (He, Cai, and Niyogi 2005; Zhao and Liu 2007) to identify relevant features through evaluating features' capability on preserving sample similarity. Given m features, and a similarity matrix $\mathbf{S}$ of the samples, the idea of spectral feature selection is to select features that align well with the leading eigenvectors of $\mathbf{S}$. Since the leading eigenvectors of $\mathbf{S}$ contain structure information of sample distribution and group similar samples into compact clusters (von Luxburg 2007), features aligning better to them will have stronger capability on preserving sample similarity (Zhao and Liu 2007). There exist many spectral feature selection algorithms: Laplacian score (He, Cai, and Niyogi 2005), Fisher score (Duda, Hart, and Stork 2001), trace ratio (Nie et al. 2008), relief and reliefF (Sikonja and Kononenko 2003), SPEC (Zhao and Liu 2007), and HSIC (Song et al. 2007). These algorithms demonstrated excellent performance in both supervised and unsupervised learning. However, since the algorithms evaluate features individually, they cannot handle redundant features. Redundant features increase dimensionality unnecessarily (Kearns and Vazirani 1994), and worsen learning performance when facing shortage of data. It is also shown empirically that removing redundant features can result in significant performance improvement (Hall 1999; Ding and Peng 2003; Gabrilovich and Markovitch 2004; Yu and Liu 2004; Appice et al. 2004; Duangsoithong 2009). Note that none of these redundant feature removal algorithms are based on spectral analysis.

In this work, we address the limitation of existing spectral feature selection algorithms in handling redundant features, and propose a novel spectral feature selection algorithm of an embedded model, which evaluates the utility of a set of features jointly and can efficiently remove redundant features. The algorithm is derived from a formulation based on multi-output regression (Hastie, Tibshirani, and Friedman 2001), and feature selection is achieved by enforcing sparsity through applying $L_{2,1}$-norm constraint on the solutions (Obozinski, Wainwright, and Jordan 2008; Argyriou, Evgeniou, and Pontil 2008). We analyze its capability on redundancy removal and study the properties of its optimal solutions, which paves the way for an efficient path-following solver. By exploiting the necessary and sufficient conditions for the optimal solutions, our solver can automatically adjust its parameters to generate a solution path for selecting a specific number of features efficiently. We conduct extensive empirical study on the proposed algorithm in both supervised and unsupervised learning to demonstrate that it can select relevant features with low redundancy.

Spectral Feature Selection with Sparse Multi-output Regression

Let $\mathbf{X} \in \mathbb{R}^{n \times m}$ be the data matrix, where n and m are the number of samples and features, respectively. Given a sample similarity matrix $\mathbf{S}$ specifying the similarity among sam-

ples, spectral feature selection aims to select features that preserve the sample similarity specified by $\mathbf{S}$. Given a feature $\mathbf{f}$, different spectral feature selection algorithms can be formulated in a common way:

$$SC(\mathbf{f},\mathbf{S}) = \hat{\mathbf{f}}^\top \hat{\mathbf{S}}\, \hat{\mathbf{f}} = \sum_{i=1}^{n} \hat{\lambda}_i \left(\hat{\mathbf{f}}^\top \hat{\boldsymbol{\xi}}_i\right)^2. \quad (1)$$

In the equation, $\hat{\mathbf{f}}$ and $\hat{\mathbf{S}}$ are the normalized $\mathbf{f}$ and $\mathbf{S}$ generated by certain normalization operators. $\hat{\lambda}_i$ and $\hat{\boldsymbol{\xi}}_i$ are the i-th eigenvalue and eigenvector of the $\hat{\mathbf{S}}$, respectively. Different spectral feature selection algorithms adopt different ways to define $\mathbf{S}$ and use different criteria to normalize $\mathbf{f}$ and $\mathbf{S}$ to achieve the certain effect, such as noise removal. Due to the space limit, we refer readers to the literature for details on spectral feature selection algorithms. Eq. (1) shows that existing spectral feature selection algorithms evaluate features individually. Therefore they cannot identify redundant features, which forms a common drawback of them.

Sparse Multi-output Regression Formulation

To identify feature redundancy, features must be evaluated jointly. To this end, given $\mathbf{y}_i = \lambda_i^{1/2}\boldsymbol{\xi}_i$, where λ_i and $\boldsymbol{\xi}_i$ are the i-th eigenvalue and eigenvector of $\mathbf{S}$, instead of looking for one feature which closely aligns $\mathbf{y}_i$, as formulated in Eq. (1), we propose to find a set of l features, such that their linear span is close to $\mathbf{y}_i$. The idea can be formulated as:

$$\arg\min_{\mathcal{A},\mathbf{w}_{i,\mathcal{A}}} \|\mathbf{y}_i - \mathbf{X}_{\mathcal{A}}\mathbf{w}_{i,\mathcal{A}}\|_2^2.$$

In the equation, $\mathcal{A} = \{i_1,\ldots,i_l\} \subseteq \{1,\ldots,m\}$, $\mathbf{X}_{\mathcal{A}} = (\mathbf{f}_{i_1},\ldots,\mathbf{f}_{i_l})$ and $\mathbf{w}_i \in \mathbb{R}^{l\times 1}$. Note, to facilitate the subsequent formulations, in the above equation, we use L_2 norm on the difference of two vectors to measure the closeness among vectors. When all λ_i and $\boldsymbol{\xi}_i$ are considered, their joint optimization can be formulated as:

$$\arg\min_{\mathcal{A},\mathbf{w}_{i,\mathcal{A}}} \sum_{i=1}^{n} \|\mathbf{y}_i - \mathbf{X}_{\mathcal{A}}\mathbf{w}_{i,\mathcal{A}}\|_2^2 = \|\mathbf{Y} - \mathbf{X}_{\mathcal{A}}\mathbf{W}_{\mathcal{A}}\|_F^2. \quad (2)$$

In the equation, $\mathbf{Y}$=$(\mathbf{y}_1,\ldots,\mathbf{y}_n)$, $\mathbf{W}_{\mathcal{A}}$=$(\mathbf{w}_{1,\mathcal{A}},\ldots,\mathbf{w}_{n,\mathcal{A}})$. Assume $\mathbf{S} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{U}^\top$ is the SVD of $\mathbf{S}$, we have $\mathbf{Y}$=$\mathbf{U}\boldsymbol{\Sigma}^{1/2}$. Note, when $\mathcal{A}$ contains only one feature, the formulation reduces to searching for features that maximize the Eq. (1).

Given $\mathbf{Y}$ and $\mathbf{X}_{\mathcal{A}}$, $\mathbf{W}_{\mathcal{A}}$ can be obtained in a closed form. However, feature selection needs to find the optimal $\mathcal{A}$, which is a combinatorial problem of NP-hard. To efficiently solve the problem, we propose the following formulation:

$$\begin{aligned} &\arg\min_{\mathbf{W},c} \|\mathbf{Y} - \mathbf{X}\mathbf{W}\|_F^2 \quad (3)\\ s.t.\quad &\|\mathbf{W}\|_{2,1} \le c\\ &\mathcal{A} = \{i : \|\mathbf{w}^i\|_2 > 0\},\ Card\,(\mathcal{A}) = l \end{aligned}$$

Here $\mathbf{w}^i$ denotes the ith row of $\mathbf{W}$, and $\|\mathbf{W}\|_{2,1}$ is the $L_{2,1}$-norm which is defined in the following way:

$$\|\mathbf{W}\|_{2,1} = \sum_{i=1}^{m} \|\mathbf{w}^i\|_2 \quad (4)$$

When applied in regression, the $L_{2,1}$-norm constraint is equivalent to applying Laplace prior (Seeger 2008) on $\mathbf{w}^i$, which tends to force many rows in $\mathbf{W}$ to be $\mathbf{0}^\top$, resulting sparse solution. The advantages of the formulation presented in Eq. (3) are three folds.

First, it can find a set of features that jointly preserve the sample similarity specified by $\mathbf{S}$.

Theorem 1 *Let* $\mathbf{Y} = \mathbf{U}\boldsymbol{\Sigma}^{1/2}$ *and* $\boldsymbol{\Omega} = \mathbf{Y} - \mathbf{X}\mathbf{W}$. *We have:*

$$\left\|\mathbf{X}\mathbf{W}\mathbf{W}^\top\mathbf{X}^\top - \mathbf{S}\right\|_F \le 2(\|\mathbf{Y}\|_F + \|\boldsymbol{\Omega}\|_F)\,\|\boldsymbol{\Omega}\|_F$$

The proof is straightforward, given the fact that $\|\mathbf{A}\mathbf{B}\|_F \le \|\mathbf{A}\|_F\|\mathbf{B}\|_F$. In the theorem, $\mathbf{X}\mathbf{W}$ is a new representation of samples obtained by linearly combining the selected features[1]. And $\mathbf{X}\mathbf{W}\mathbf{W}^\top\mathbf{X}^\top$ returns the pairwise similarity among samples measured by their inner product under the new representation. The theorem shows that by minimizing $\|\boldsymbol{\Omega}\|_F$, we also minimize $\|\mathbf{X}\mathbf{W}\mathbf{W}^\top\mathbf{X}^\top - \mathbf{S}\|_F$, which ensures the selected features can jointly preserve the sample similarity specified by $\mathbf{S}$.

Second, by jointly evaluating a set of features, it tends to select non-redundant features. Assume two features $\mathbf{f}_p$ and $\mathbf{f}_q$ satisfy the following conditions: (1) they are equally correlated to $\mathbf{Y}$, i.e. $\mathbf{f}_p^\top\mathbf{Y}$=$\mathbf{f}_q^\top\mathbf{Y}$; (2) $\mathbf{f}_q$ is highly correlated to $\mathbf{f}_d$, i.e. $\mathbf{f}_q^\top\mathbf{f}_d \to 1$. And $\mathbf{f}_q$ is less correlated to $\mathbf{f}_d$, i.e. $\mathbf{f}_p^\top\mathbf{f}_d > \mathbf{f}_q^\top\mathbf{f}_d$. Without loss of generality, we assume both $\mathbf{f}_p$ and $\mathbf{f}_q$ are positively correlated to $\mathbf{f}_d$; (3) they are equally correlated to other features, i.e. $\mathbf{f}_p^\top\mathbf{f}_i$=$\mathbf{f}_q^\top\mathbf{f}_i$, $\forall i \in \{1,\ldots,m\},\ i \ne d$. We have:

Theorem 2 *Assume the above assumptions hold, and* $\mathbf{f}_d$ *have been selected by an optimal solution of Eq. (3), then feature* $\mathbf{f}_q$ *has higher priority than* $\mathbf{f}_p$ *to be selected in the optimal solution.*

The theorem can be proved by contradiction through assuming $\mathbf{f}_d$ and $\mathbf{f}_p$ are in the optimal solution but $\mathbf{f}_q$ is not. The theorem shows that the formulation in Eq. (3) tends to select features that are less correlated to the already selected ones, which ensures the selection of non-redundant features.

Third, it is tractable. Given a value for c, the problem:

$$\arg\min_{\mathbf{W},c} \|\mathbf{Y} - \mathbf{X}\mathbf{W}\|_F^2 \quad s.t.\ \ \|\mathbf{W}\|_{2,1} \le c \quad (5)$$

can be solved by applying a general solver (Obozinski, Wainwright, and Jordan 2008; Argyriou, Evgeniou, and Pontil 2008; Liu, Ji, and Ye 2009). And given l, a proper c value, which results in the selection of about l features, can be found by applying either a grid search or a binary search based on the observation that, a smaller c value usually results in selecting fewer features. However, for a given l, this approach may require to run a solver many times for searching the c value, which is computationally inefficient.

[1]Note that although $\mathbf{W} \in \mathbb{R}^{m\times k}$, many of its rows are $\mathbf{0}^\top$. Therefore, the representation is generated by using only a small set of selected features

MRSF, An Efficient Solver for Eq. (3)

We propose an efficient path-following solver for the problem in Eq. (3). It can automatically detect the points when new features enter its "active set", and update its parameters accordingly. It can efficiently generate a solution path to select the specified number of features. We start by deriving the necessary and sufficient conditions for a feature to be selected in an optimal solution of Eq. (5).

The Lagrangian for Eq. (5) has the following form:

$$\mathcal{L}(\mathbf{W}, \lambda) = \|\mathbf{Y} - \mathbf{X}\mathbf{W}\|_F^2 - \lambda \left(c - \|\mathbf{W}\|_{2,1}\right) \quad (6)$$

$\mathcal{L}(\mathbf{W}, \lambda)$ is convex. According to the convex optimization theorem (Boyd and Vandenberghe 2004), $\mathbf{W}_*$ minimizes $\mathcal{L}(\mathbf{W}, \lambda)$ if and only if $\mathbf{0} \in \partial_{\mathbf{w}^i} \mathcal{L}(\mathbf{W}, \lambda)|_{\mathbf{W}=\mathbf{W}_*}$, $i = 1, \ldots, m$. Here, $\partial_{\mathbf{w}^i} \mathcal{L}(\mathbf{W}, \lambda)$ is the subdifferential of $\mathcal{L}(\mathbf{W}, \lambda)$ corresponding to $\mathbf{w}^i$, and has the following form:

$$\partial_{\mathbf{w}^i} \mathcal{L}(\mathbf{W}, \lambda) = \mathbf{f}_i^\top (\mathbf{Y} - \mathbf{X}\mathbf{W}) + \lambda \mathbf{v}_i \quad (7)$$

$$\mathbf{v}_i = \frac{\mathbf{w}^i}{\|\mathbf{w}^i\|}, \text{ if } \mathbf{w}^i \neq 0$$

$$\mathbf{v}_i \in \left\{\mathbf{u} | \mathbf{u} \in \mathbb{R}^{1\times k}, \|\mathbf{u}\|_2 \leq 1\right\}, \text{ if } \mathbf{w}^i = 0$$

Therefore, $\mathbf{W}_*$ is an optimal solution if and only if:

$$-\lambda \mathbf{v}_i = \mathbf{f}_i^\top (\mathbf{Y} - \mathbf{X}\mathbf{W})|_{\mathbf{W}=\mathbf{W}_*}, \ \forall i \in \{1, \ldots, m\} \quad (8)$$

Base on this observation, we give the necessary conditions, and the necessary and sufficient conditions for $\mathbf{W}$ to be optimal with the following two propositions.

Proposition 1 *Assume $\mathbf{w}^i$ is the i-th row of $\mathbf{W}$, the necessary conditions for $\mathbf{W}$ to be optimal are: $\forall i \in \{1, \ldots, m\}$:*

$$\begin{aligned} \mathbf{w}^i \neq \mathbf{0} \quad &\Rightarrow \quad \|\mathbf{f}^\top (\mathbf{Y} - \mathbf{X}\mathbf{W})\|_2 = \lambda \\ \mathbf{w}^i = \mathbf{0} \quad &\Rightarrow \quad \|\mathbf{f}^\top (\mathbf{Y} - \mathbf{X}\mathbf{W})\|_2 \leq \lambda \end{aligned} \quad (9)$$

Proposition 2 *Assume $\mathbf{w}^i$ is the i-th row of $\mathbf{W}$, the necessary and sufficient conditions for $\mathbf{W}$ to be optimal are: $\forall i$*

$$\begin{aligned} \mathbf{w}^i \neq \mathbf{0} \quad &\Rightarrow \quad \mathbf{f}^\top (\mathbf{Y} - \mathbf{X}\mathbf{W}) = -\lambda \frac{\mathbf{w}^i}{\|\mathbf{w}^i\|_2} \\ \mathbf{w}^i = \mathbf{0} \quad &\Rightarrow \quad \|\mathbf{f}^\top (\mathbf{Y} - \mathbf{X}\mathbf{W})\|_2 \leq \lambda \end{aligned} \quad (10)$$

Based on the two propositions, we propose an efficient solver for Eq. (3), and its pseudo code can be found in Algorithm 1. In the algorithm, $\mathcal{A}_i$ is the "active set" in the i-th run, which contains the features selected in that run. Algorithm 1 contains two major steps. (1) Lines 4-10, the algorithm determines the direction for updating $\mathbf{W}^{[i]}$ (Line 4), and the step size (Lines 5-8), by which, it updates the active set and the λ (Line 10). (2) Lines 11-18, the algorithm finds an optimal solution corresponding to the λ obtained in step 1. Given λ, it first solves an $L_{2,1}$-norm regularized regression problem using a general solver based on the Nesterov's method (Liu, Ji, and Ye 2009) (Line 11). Note, this problem is of small scale, since it is only based on the features in the current active set, but not the whole set. $\hat{\mathbf{W}}$ is used as a starting point to ensure fast convergence of the Nesterov solver. It then checks whether the obtained solution is also optimal on the whole data (Line 13). If it is true, the algorithm records the current optimal solution and goes to Line 4 to start next run (Line 14). Otherwise, it adjusts the active set and goes back to Line 11 (Line 17).

Algorithm 1: MRSF

Input: $\mathbf{X}$, $\mathbf{Y}$, k
Output: $\mathbf{W}$

1 $\mathbf{W}^{[0]} = 0$, $i = 1$ and $\mathbf{R}^{[0]} = \mathbf{Y}$;
2 Compute the initial "active set" $\mathcal{A}_1$:
$\mathcal{A}_1 = \arg\max_j \|\mathbf{f}_j^\top \mathbf{R}^{[0]}\|_2^2$;
3 **while** $i \leq k$ **do**
4 Compute the walking direction $\gamma_{\mathcal{A}_i}$:
$\gamma_{\mathcal{A}_i} = \left(\mathbf{X}_{\mathcal{A}_i}^\top \mathbf{X}_{\mathcal{A}_i}\right)^{-1} \mathbf{X}_{\mathcal{A}_i}^\top \mathbf{R}^{[i-1]}$;
5 **for** *each $j \notin \mathcal{A}_i$ and an arbitrary $t \in \mathcal{A}_i$* **do**
6 Compute the step size α_j can be taken in direction $\gamma_{\mathcal{A}_i}$, before $\mathbf{f}_j$ enters the active set.
$\|\mathbf{f}_j^\top \left(\mathbf{R}^{[i-1]} - \alpha_j \mathbf{X}_{\mathcal{A}_i} \gamma_{\mathcal{A}_i}\right)\|_2$
$= (1 - \alpha_j)\|\mathbf{f}_t^\top \mathbf{R}^{[i-1]}\|_2$;
7 **end**
8 $j^* = \arg\min_{j \notin \mathcal{A}_i} \alpha_j$;
9 $\hat{\mathbf{W}} = \left(\left(\mathbf{W}^{[i-1]} + \alpha_{j^*} \gamma_{\mathcal{A}_i}\right)^\top, \mathbf{0}\right)^\top$;
10 $\hat{\mathcal{A}} = \mathcal{A}_i \bigcup \{j^*\}$, $\lambda = (1 - \alpha)\|\mathbf{f}_t^\top \mathbf{R}^{[i-1]}\|_2$;
11 Solve $\min_{\tilde{\mathbf{W}}} \|\mathbf{Y} - \mathbf{X}_{\hat{\mathcal{A}}} \tilde{\mathbf{W}}\|_F^2 + \lambda \|\tilde{\mathbf{W}}\|_{2,1}$ using Nesterov's method, with $\hat{\mathbf{W}}$ as the starting point;
12 $\tilde{\mathbf{R}} = \mathbf{Y} - \mathbf{X}_{\hat{\mathcal{A}}} \tilde{\mathbf{W}}$;
13 **if** $\forall i \notin \hat{\mathcal{A}}$, $\|\mathbf{f}_i^\top \tilde{\mathbf{R}}\|_2 \leq \lambda$ **then**
14 $i = i + 1$, $\mathcal{A}_i = \hat{\mathcal{A}}$, $\mathbf{W}^{[i-1]} = \tilde{\mathbf{W}}$, $\mathbf{R}^{[i-1]} = \tilde{\mathbf{R}}$;
15 **else**
16 $\hat{\mathcal{A}} = \left\{i : \|\tilde{\mathbf{w}}^i\| \neq 0\right\} \bigcup \left\{\arg\max_j \|\mathbf{f}_j^\top \tilde{\mathbf{R}}\|_2\right\}$;
17 Remove $\tilde{\mathbf{w}}^i$ from $\tilde{\mathbf{W}}$, if $\|\tilde{\mathbf{w}}^i\| = 0$,
$\hat{\mathbf{W}} = \left(\tilde{\mathbf{W}}^\top, \mathbf{0}\right)^\top$, **Goto** line 11;
18 **end**
19 **end**
20 Extend $\mathbf{W}^{[k]}$ to $\mathbf{W}$ by adding empty rows to $\mathbf{W}^{[k]}$;
21 **return** $\mathbf{W}^{[k]}$;

Theorem 3 *(1) Given $\mathbf{W}^{[i-1]}$ is the current optimal solution, the $\hat{\mathbf{W}}$ generated in step 1 (Line 9) satisfies the necessary condition for an optimal solution specified in Proposition 1. (2) And the $\tilde{\mathbf{W}}$ in Line 14 of step 2 is an optimal solution on $\mathbf{X}$ corresponding to the current λ.*

Proof: To prove the first point, it is sufficient to show that $\mathbf{f}_i^\top \mathbf{X}_{\mathcal{A}_i} \left(\mathbf{X}_{\mathcal{A}_i}^\top \mathbf{X}_{\mathcal{A}_i}\right)^{-1} \mathbf{X}_{\mathcal{A}_i}^\top = \mathbf{f}_i^\top$, $\forall\, i \in \mathcal{A}_i$. And the second point of the theory can be simply verified by applying the necessary and sufficient conditions for the optimal solutions developed in Proposition 2. ■

Algorithm 1 is very efficient. In each run, in step 1 it increases the size of its active set and decreases the λ accordingly. At the same time, it generates a tentative solution, which satisfies the necessary conditions for the opti-

Table 1: Summary of the benchmark data sets

Data Set	# Features	# Instances	# Classes
AR10P	2400	130	10
PIE10P	2400	210	10
PIX10P	10000	100	10
ORL10P	10000	100	10
TOX	5748	171	4
CLL-SUB	11340	111	3

mal solution. And in step 2, it generates an optimal solution corresponding to λ for the whole data by working on features in the active set only. Since the tentative solution is usually very close to a true optimal, step 2 often converges in just a few iterations. Our analysis shows that when n features are selected, Algorithm 1 has a time complexity of $O(n^3k + mn^2)$. We omit the analysis due to space limit.

Experimental Study

We now empirically evaluate the performance of the proposed algorithm in both supervised and unsupervised learning. We name it MRSF, since it is proposed to Minimize the feature Redundancy for Spectral Feature selection.

Experiment Setup

In the experiments, we choose eight representative feature selection algorithms for comparison. For supervised learning, six feature selection algorithms are chosen as baselines: reliefF, Fisher score, trace ratio, HSIC, mRMR (Ding and Peng 2003) and AROM-SVM (Weston et al. 2003). The first four are existing spectral feature selection algorithms. And the last two are the-state-of-the-art feature selection algorithms for removing redundant features. For unsupervised learning, four algorithms are used for comparison: Laplacian score, SPEC, trace ratio, and HSIC. They are all spectral feature selection algorithms. For MRSF, in supervised learning, $\mathbf{S}$ is calculated by $\mathbf{S}_{ij} = 1/n_l$, if $y_i = y_j = l$, otherwise $\mathbf{S}_{ij} = 0$; and in unsupervised learning, $\mathbf{S}$ is calculated by the Gaussian RBF kernel function. Six high dimensional data sets are used in the experiment. There are four image data: AR10P, PIE10P, PIX10, and ORL10P. And Two Microarray data: TOX and CLL-SUB. Detailed information of the data sets is listed in Table 1.

Assume $\mathbf{F}$ is the set of selected features, and $\mathbf{X}_\mathbf{F}$ only containing features in $\mathbf{F}$. In the supervised learning context, algorithms are compared on (1) **classification accuracy** and (2) **redundancy rate**. The redundancy rate is measured by:

$$\mathrm{RED}\left(\mathbf{F}\right) = \frac{1}{m(m-1)} \sum_{f_i, f_j \in \mathbf{F}, i>j} \rho_{i,j},$$

where, $\rho_{i,j}$ returns the correlation between the ith and the jth features. A large value of $\mathrm{RED}\left(\mathbf{F}\right)$ indicates that many selected features are strongly correlated and thus redundancy is expected to exist in $\mathbf{F}$. For unsupervised case, two measurements are used: (1) the **redundancy rate** as defined above; and (2) the **Jaccard score** computed by:

$$\mathrm{JAC}\left(\mathbf{S}_\mathbf{F}, \mathbf{S}, k\right) = \frac{1}{n} \sum_{i=1}^{n} \frac{NB\left(i, k, \mathbf{S}_\mathbf{F}\right) \cap NB\left(i, k, \mathbf{S}\right)}{NB\left(i, k, \mathbf{S}_\mathbf{F}\right) \cup NB\left(i, k, \mathbf{S}\right)},$$

where, $\mathbf{S}_\mathbf{F} = \mathbf{X}_\mathbf{F}\mathbf{X}_\mathbf{F}^\top$ is the similarity matrix computed from the selected features using inner product; and $NB\left(i, k, \mathbf{S}\right)$ returns the k nearest neighbors of the i-th sample according to $\mathbf{S}$. The Jaccard score measures the averaged overlapping of the neighborhoods specified by $\mathbf{S}_\mathbf{F}$ and $\mathbf{S}$. A high Jaccard score indicates that sample similarity are well preserved.

For each data set, we randomly sample 50% samples as the training data and the remaining are used for test. The process is repeated for 20 times and results in 20 different partitions. Different algorithms are evaluated on each partition. The results are recorded and averaged to generate the final results. Linear SVM is used for classification. The parameters in feature selection algorithms and SVM are tuned via cross-validation if necessary. Student's t-test is used to evaluate the statistical significance with a threshold of 0.05.

Study of Supervised Cases

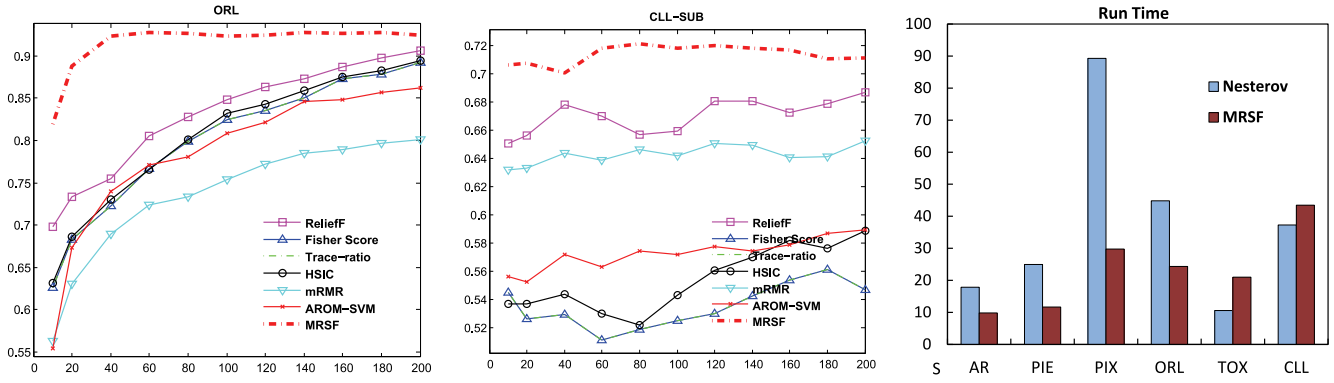
Accuracy: The classification accuracy results are shown in Figure 1-(a,b) and Table 2. Figure 1-(a,b) contains the plots of the accuracy achieved by the SVM classifier when uses the top 10, 20, . . ., 200 features selected by each algorithm. Due to the space limit, we only plot the results from ORL and CLL data. Table 2 shows the "aggregated accuracy" of different algorithms on each data set. The aggregated accuracy is obtained by averaging the averaged accuracy achieved by SVM using the top 10, 20, . . ., 200 features selected by each algorithm. The value in the parentheses is the p-Val. In Figure 1 and Table 2, we can observe that MRSF produces superior classification performance comparing to the baseline algorithms. The averaged value for aggregated accuracy achieved by the baseline algorithms is 0.78, which is 11% lower than that achieved by MRSF.

Redundancy rate: Table 3 presents the averaged redundancy rates of the top n features selected by different algorithms, where n is the number of samples. We choose n, since when the number of selected features is larger than n, any feature can be expressed by a linear combination of the remaining ones, which will introduces unnecessary redundancy in evaluation. In the table, the boldfaced values are the lowest redundancy rates or the ones without significant difference to the lowest. The results show that MRSF attains very low redundancy, which suggests that the redundancy removal mechanism in MRSF is effective.

Study of Unsupervised Cases

Jaccard score: Tables 4 present the averaged Jaccard score achieved by different algorithms. Results show that MRSF achieves significant better results on all data sets comparing to the baseline algorithms, which demonstrates its strong capability on selecting good features for similarity preserving.

Redundancy rate: Table 5 shows the averaged redundancy rates achieved with the top n features selected by different algorithms. The results show that the features selected by MRSF contains much less redundancy comparing with the



(a) (b) (c)

Figure 1: Plots (a) and (b): Study of supervised cases, accuracy (Y axis) vs. different numbers of selected features (X axis). The higher the accuracy the better. Plot (c), the Running time of MRSF and the Nesterov method on each data set.

Table 2: Study of supervised cases: **aggregated accuracy** with p-Val. (The higher the better.)

Algorithm	ORL	PIX	AR	PIE	TOX	CLL-SUB	AVE
ReliefF	0.83 (0.00)	0.93 (0.00)	0.80 (0.00)	0.94 (0.00)	0.77 (0.03)	0.67 (0.00)	0.82
Fisher Score	0.80 (0.00)	0.92 (0.00)	0.77 (0.00)	0.93 (0.00)	0.72 (0.00)	0.54 (0.00)	0.78
Trace-ratio	0.80 (0.00)	0.92 (0.00)	0.77 (0.00)	0.93 (0.00)	0.72 (0.00)	0.54 (0.00)	0.78
HSIC	0.80 (0.00)	0.93 (0.00)	0.77 (0.00)	0.94 (0.00)	0.73 (0.00)	0.55 (0.00)	0.79
mRMR	0.73 (0.00)	0.87 (0.00)	0.70 (0.00)	0.95 (0.00)	0.70 (0.00)	0.64 (0.00)	0.76
AROM-SVM	0.78 (0.00)	0.86 (0.00)	0.63 (0.00)	0.94 (0.02)	0.64 (0.00)	0.57 (0.00)	0.74
MRSF	**0.91 (1.00)**	**0.96 (1.00)**	**0.84 (1.00)**	**0.98 (1.00)**	**0.79 (1.00)**	**0.71 (1.00)**	**0.86**

Table 3: Study of supervised cases: **averaged redundancy rate** with p-Val. (The lower the better.)

Algorithm	ORL	PIX	AR	PIE	TOX	CLL-SUB	AVE
ReliefF	0.92 (0.00)	0.79 (0.00)	0.77 (0.00)	0.36 (0.00)	0.34 (0.00)	0.59 (0.00)	0.63
Fisher Score	0.79 (0.00)	0.83 (0.00)	0.67 (0.00)	0.37 (0.00)	0.56 (0.00)	0.76 (0.00)	0.66
Trace-ratio	0.79 (0.00)	0.83 (0.00)	0.67 (0.00)	0.37 (0.00)	0.56 (0.00)	0.76 (0.00)	0.66
HSIC	0.79 (0.00)	0.83 (0.00)	0.67 (0.00)	0.37 (0.00)	0.56 (0.00)	0.76 (0.00)	0.66
mRMR	**0.25 (0.29)**	0.33 (0.00)	0.26 (0.00)	0.29 (0.00)	0.26 (0.00)	0.26 (0.00)	0.27
AROM-SVM	**0.25 (0.44)**	**0.26 (1.00)**	0.25 (0.00)	0.32 (0.00)	**0.15 (1.00)**	0.59 (0.00)	0.31
MRSF	**0.25 (1.00)**	**0.35 (0.17)**	**0.21 (1.00)**	**0.24 (1.00)**	**0.16 (0.40)**	**0.21 (1.00)**	**0.24**

Table 4: Study of unsupervised cases: **averaged Jaccard score** with p-Val. (The higher the better.)

Algorithm	ORL	PIX	AR	PIE	TOX	CLL-SUB	AVE
NB = 1							
Laplacian Score	0.07 (0.00)	0.05 (0.00)	0.07 (0.00)	0.04 (0.00)	0.10 (0.00)	0.06 (0.00)	0.07
SPEC	0.15 (0.00)	0.05 (0.00)	0.09 (0.00)	0.05 (0.00)	0.12 (0.00)	0.05 (0.00)	0.09
Trace-Ratio	0.06 (0.00)	0.05 (0.00)	0.08 (0.00)	0.03 (0.00)	0.12 (0.00)	0.08 (0.00)	0.07
HSIC	0.08 (0.00)	0.05 (0.00)	0.07 (0.00)	0.04 (0.00)	0.12 (0.00)	0.10 (0.00)	0.08
MRSF	**0.56 (1.00)**	**0.53 (1.00)**	**0.41 (1.00)**	**0.41 (1.00)**	**0.31 (1.00)**	**0.17 (1.00)**	**0.40**
NB = 5							
Laplacian Score	0.16 (0.00)	0.11 (0.00)	0.13 (0.00)	0.08 (0.00)	0.17 (0.00)	0.16 (0.00)	0.13
SPEC	0.28 (0.00)	0.11 (0.00)	0.16 (0.00)	0.11 (0.00)	0.19 (0.00)	0.14 (0.00)	0.17
Trace-Ratio	0.15 (0.00)	0.11 (0.00)	0.14 (0.00)	0.08 (0.00)	0.18 (0.00)	0.17 (0.00)	0.14
HSIC	0.16 (0.00)	0.13 (0.00)	0.14 (0.00)	0.10 (0.00)	0.18 (0.00)	0.16 (0.00)	0.14
MRSF	**0.57 (1.00)**	**0.63 (1.00)**	**0.41 (1.00)**	**0.38 (1.00)**	**0.34 (1.00)**	**0.24 (1.00)**	**0.43**

Table 5: Study of unsupervised cases: **averaged redundancy rate** with p-Val. (The lower the better.)

Algorithm	ORL	PIX	AR	PIE	TOX	CLL-SUB	AVE
Laplacian Score	0.88 (0.00)	0.97 (0.00)	0.82 (0.00)	0.84 (0.00)	0.57 (0.00)	0.65 (0.00)	0.68
SPEC	0.72 (0.00)	0.97 (0.00)	0.75 (0.00)	0.77 (0.00)	0.47 (0.00)	0.59 (0.00)	0.61
Trace-Ratio	0.88 (0.00)	0.97 (0.00)	0.81 (0.00)	0.87 (0.00)	0.57 (0.00)	0.67 (0.00)	0.68
HSIC	0.88 (0.00)	0.97 (0.00)	0.80 (0.00)	0.82 (0.00)	0.57 (0.00)	0.64 (0.00)	0.67
MRSF	**0.35 (1.00)**	**0.32 (1.00)**	**0.32 (1.00)**	**0.29 (1.00)**	**0.27 (1.00)**	**0.37 (1.00)**	**0.27**

baseline algorithms. This is expected, since the latter cannot remove redundant features in feature selection.

Study of Efficiency

Figure 1-(c) presents the running time of MRSF and a solver for Eq. (5) proposed in (Liu, Ji, and Ye 2009). Since that solver is based the Nesterov's method, we call it Nesterov in this paper. As shown in (Liu, Ji, and Ye 2009), Nesterov is one of the fastest solvers for solving Eq. (5). The running time is obtained in the following way. We first run MRSF to select 100 features on each data set and record the obtained $c = \|W\|_{2,1}$ and the time it used. We then run Nesterov on each data using the c we obtained from MRSF and record its running time. The precision of the Nesterov is set to 10^{-6}. Note that Nesterov is also used in Line 11 of MRSF, where its precision is set to 10^{-6}, too. The results show that on the six data sets, MRSF achieved an average running time of 23.33s, which compares to the 37.46s of Nesterov. Note that if the grid search or binary search is applied to determine c, Nesterov may have a running time which is much longer, even with the warm start strategy (Liu, Ji, and Ye 2009). The results demonstrate the high efficiency of the proposed MRSF algorithm.

The experiment results from both supervised and unsupervised learning cases show consistently that MRSF can very efficiently select features containing less redundancy and producing excellent learning performance.

Conclusion

In this work, we propose a novel spectral feature selection algorithm based on sparse multi-output regression with $L_{2,1}$-norm constraint. We study the properties of its solutions, and design an efficient solver following our formulation. The algorithm improves existing spectral feature selection algorithms by overcoming a common drawback in handling feature redundancy. As illustrated by extensive experimental study, the proposed algorithm can effectively remove redundant features and achieve superior performance in both supervised and unsupervised learning. In our study, we find that our formulation for spectral feature selection can be linked to a wide range of learning models, such as SVM and LDA through their least square formulations (Suykens and Vandewalle 1999; Sun, Ji, and Ye 2009). We will investigate these connections to gain more insights on spectral feature selection for further research.

Acknowledgments

This work is, in part, supported by NSF Grant (0812551).

References

Appice, A.; Ceci, M.; and *et al*. 2004. Redundant feature elimination for multi-class problems. In *ICML*.

Argyriou, A.; Evgeniou, T.; and Pontil, M. 2008. Convex multi-task feature learning. *Mach. Learning* 73:243–272.

Boyd, S., and Vandenberghe, L. 2004. *Convex Optimization*. Cambridge University Press.

Ding, C., and Peng, H. 2003. Minimum redundancy feature selection from microarray gene expression data. In *CSB'03*, 523–529.

Duangsoithong, R. 2009. Relevant and redundant feature analysis with ensemble classification. In *ICAPR '09*.

Duda, R.; Hart, P.; and Stork, D. 2001. *Pattern Classification*. John Wiley & Sons, New York, 2 edition.

Gabrilovich, E., and *et al*. 2004. Text categorization with many redundant features: using aggressive feature selection to make svms competitive with c4.5. In ICML'04.

Guyon, I., and Elisseeff, A. 2003. An introduction to variable and feature selection. *JMLR* 3:1157–1182.

Hall, M. 1999. *Correlation Based Feature Selection for Machine Learning*. Ph.D. Dissertation, University of Waikato, Dept. of Computer Science.

Hastie, T.; Tibshirani, R.; and Friedman, J. 2001. *The Elements of Statistical Learning*. Springer.

He, X.; Cai, D.; and Niyogi, P. 2005. Laplacian score for feature selection. In *NIPS*.

Kearns, M., and Vazirani, U. 1994. *An Introduction to Computational Learning Theory*. The MIT Press.

Liu, H., and Motoda, H. 1998. *Feature Selection for Knowledge Discovery and Data Mining*. Boston: Kluwer Academic Publishers.

Liu, J.; Ji, S.; and Ye, J. 2009. Multi-task feature learning via efficient l2,1-norm minimization. In *UAI'09*.

Nie, F.; Xiang, S.; Jia, Y.; Zhang, C.; and Yan, S. 2008. Trace ratio criterion for feature selection. In *AAAI*.

Obozinski, G.; Wainwright, M. J.; and Jordan, M. I. 2008. Highdimensional union support recovery in multivariate regression. In *Neural Information Processing Systems*.

Seeger, M. 2008. Bayesian inference and optimal design for the sparse linear model. *JMLR* 9:759–813.

Sikonja, M. R., and Kononenko, I. 2003. Theoretical and empirical analysis of Relief and ReliefF. *Machine Learning* 53:23–69.

Song, L.; Smola, A.; Gretton, A.; Borgwardt, K.; and Bedo, J. 2007. Supervised feature selection via dependence estimation. In *ICML*.

Sun, L.; Ji, S.; and Ye, J. 2009. A least squares formulation for a class of generalized eigenvalue problems in machine learning. In *ICML*.

Suykens, J., and Vandewalle, J. 1999. Least squares support vector machine classifiers. *Neural Processing Letters* 9(3):1370–4621.

von Luxburg, U. 2007. A tutorial on spectral clustering. Tech. report, Max Planck Inst. for Biological Cybernetics.

Weston, J.; Elisseff, A.; Schoelkopf, B.; and Tipping, M. 2003. Use of the zero norm with linear models and kernel methods. *JMLR* 3:1439–1461.

Yu, L., and Liu, H. 2004. Efficient feature selection via analysis of relevance and redundancy. *JMLR* 5:1205–1224.

Zhao, Z., and Liu, H. 2007. Spectral feature selection for supervised and unsupervised learning. In *ICML*.