Representative Selection with Structured Sparsity

Hongxing Wang[†,‡,§], Yoshinobu Kawahara[♮,♯], Chaoqun Weng[§], Junsong Yuan[§]

[†]*Key Laboratory of Dependable Service Computing in Cyber Physical Society Ministry of Education (Chongqing University), Chongqing 400044, PR China*

[‡]*School of Software Engineering, Chongqing University, Chongqing 401331, PR China*

[§]*School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798*

[♮]*The Institute of Scientific and Industrial Research, Osaka University, Osaka 567-0047, Japan*

[♯]*Center for Advanced Integrated Intelligence Research, RIKEN, Saitama 351-0198, Japan*

E-mail: ihxwang@cqu.edu.cn, ykawahara@sanken.osaka-u.ac.jp, weng0018@e.ntu.edu.sg, jsyuan@ntu.edu.sg

Abstract

We propose a novel formulation to find representatives in data samples via learning with structured sparsity. To find representatives with both diversity and representativeness, we formulate the problem as a structurally-regularized learning where the objective function consists of a reconstruction error and three structured regularizers: (1) group sparsity regularizer, (2) diversity regularizer, and (3) locality-sensitivity regularizer. For the optimization of the objective, we propose an accelerated proximal gradient algorithm, combined with the proximal-Dykstra method and the calculation of parametric maximum flows. Experiments on image and video data validate the effectiveness of our method in finding exemplars with diversity and representativeness and demonstrate its robustness to outliers.

Keywords: representative selection, structured sparsity, diversity

1. Introduction

Representative selection is the problem of finding exemplar samples from a data collection, where the selected exemplars serve to summarize the data collection. This problem has recently been actively discussed in data analysis and processing because it holds several advantages over analyzing the dataset as a whole. First, the selected exemplars are expected to be more interpretable than the entire dataset. Second, the memory cost for storing information on the data can be significantly reduced, and the computational efficiency for data modeling, such as classifier training, and for the application of the model can also be improved. For example, in computer vision, representative selection has been used for video anomaly detection and video summarization [1, 2, 3, 4].

While knowing class labels of training data, a variety of selection strategies have been proposed to reduce the number of training data for some specific classifiers [5, 6, 7, 8, 9, 10, 11]. In addition to this family of methods that require extra knowledge for representative selection [12, 13, 14], there has also been increasing interest in unsupervised approaches to find representatives.

As for unsupervised schemes, one naïve approach is the application of the k-medoids algorithm [15] to find k medoid centers as representatives. The selected representatives achieve the minimum total distance from all samples. With such a method, however, it is difficult to determine the optimum number of centers, and the performance heavily depends on the initialization of these centers. To alleviate these issues, a soft variant of k-medoids algorithm has been proposed in [16, 17]. Instead of expressing one sample using one center as in the k-medoids algorithm, it allows us to represent one sample using a sparse group of centers. Similar to the k-medoids algorithm, Affinity Propagation [18, 19] is another approach to clustering-based exemplar selection methods that does not need to initialize the

center positions or specify the number of centers. These clustering-based methods usually perform well when the data are distributed to groups around a few centers.

Another strategy is the one based on a low-rank representation of data. For example, the Rank Revealing QR approaches [20, 21] aim to find k columns from the data matrix such that the column submatrix is well conditioned and approximates the original data matrix in a projection sense. The data samples corresponding to the k columns are then selected as representatives. It is also feasible to use randomized or greedy methods [22, 23, 24, 25] to produce the well-conditioned column submatrix from a low-rank data matrix. For these methods, the number of representatives k usually still needs to be preset.

Recently, relying on dictionary selection, some methods have taken the original dataset as a dictionary of atoms so that representatives are required to approximately express all the data by linear combination [1, 26, 3, 27]. To avoid setting up the number of representatives, sparse constraints have been exploited in most methods, which has led to some promising results in anomaly detection and visual summarization. To make sure selected samples meet some specific properties, some methods have been proposed using auxiliary information and additional constraints. For example, affine constraints have been applied for translation invariance of representatives in [2, 28]. In [29], a temporal location based weight matrix has been designed for each given video to prevent too close frames being selected simultaneously, thus guaranteeing diverse locations of extracted key-frames. To summarize videos into key objects, the proposed method in [4] can incorporate prior knowledge of objects with existing object proposal approaches such as Edge Boxes [30] and Adobe Boxes [31].

Similar to traditional sparse dictionary selection, exemplars are required to sparsely reconstruct original data by leveraging a row sparsity regularizer in our method. In addition, we design a novel regularizer for representative selection

to encourage dissimilar samples to be selected simultaneously, thus enforcing a diverse selection of exemplars. It is worth noting that, however, we do not mean to determine diverse locations of video frames in temporal domain as mentioned in [29], but to select samples with diverse features. To further guarantee the representativeness of selected exemplars, we also develop a locality-sensitivity regularizer to consider the dependent information among similar data samples. We thus formulate our objective as a dictionary selection problem with structured sparse regularization, which aims to find a selection matrix with the properties of low data reconstruction error, row sparsity, diversity and locality-sensitivity. Because this formulation involves the calculation of a difficult optimization, we propose an accelerated proximal gradient algorithm combined with the proximal-Dykstra method [32] and the efficient calculation of parametric maximum flows. Finally, we demonstrate the effectiveness and robustness of the proposed method on several real-world datasets.

Notation. Scalars are denoted by lowercase italic Roman or Greek letters (*e.g.*, x, α). Boldface letters refer to matrices if written in uppercase ($\mathbf{X}$) and vectors otherwise ($\mathbf{x}$). Given a matrix $\mathbf{A} \in \mathbb{R}^{r\times c}$, we decompose $\mathbf{A}$ into columns of vectors $[\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_c]$ or rows of vectors $[\mathbf{a}^1, \mathbf{a}^2, \ldots, \mathbf{a}^r]^{\mathrm{T}}$. Matrix and vector elements are represented by lowercase italic letters with indexing subscripts (a_{ij}, x_i). In addition, we denote by $\|\mathbf{s}\|_q$ the l_q-norm of $\mathbf{s}$ for $q \geq 1$.

2. Problem Statement

Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$ be the data matrix of n samples in $\mathbb{R}^d$. Many works on dictionary learning focus on decomposing $\mathbf{X}$ into the product of a dictionary matrix $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_m] \in \mathbb{R}^{d\times m}$ of m atoms and a coding matrix $\mathbf{A} = (a_{ij}) = [\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_n] \in \mathbb{R}^{m\times n}$, which enables us to reconstruct each

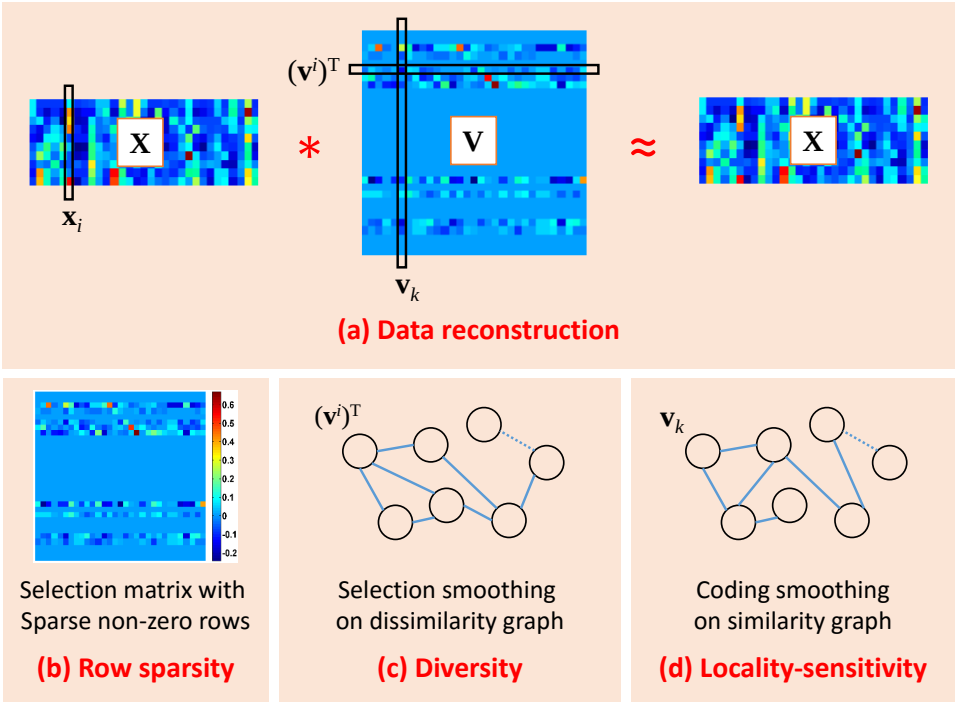


Figure 1: Structured representative selection using a selection matrix $\mathbf{V}$ with the properties of (a) low data reconstruction error, (b) row sparsity, (c) diversity and (d) locality-sensitivity.

sample $\mathbf{x}_i$ using the learned atoms in such a way that $\mathbf{x}_i \approx \mathbf{D}\mathbf{a}_i = \sum_{k=1}^{m} a_{ki}\mathbf{d}_k$. In sparse dictionary learning, there can only be sparse non-zero entries in the coding coefficients $a_{1i}, a_{2i}, \cdots, a_{mi}$, which will finally determine a few atoms to linearly express $\mathbf{x}_i$. Inspired by selecting atoms from a dictionary to reconstruct each single data, it is of great interest to select a universal set of atoms for the whole dataset. Furthermore, we can regard the original data to be a dictionary so that representative data will be selected to compress and summarize the entire dataset [1, 2].

To this end, we can compose the loss function with a regularized reconstruction

error:

$$\ell(\mathbf{V}) = \mathcal{E}(\mathbf{V}) + \mathcal{R}(\mathbf{V}) = \frac{1}{2}\|\mathbf{X} - \mathbf{X}\mathbf{V}\|_{\mathrm{F}}^2 + \mathcal{R}(\mathbf{V}), \tag{1}$$

where $\mathbf{V} \in \mathbb{R}^{n\times n}$ is the coding matrix used to linearly combine the data in $\mathbf{X}$, $\|\cdot\|_{\mathrm{F}}^2$ is the squared Frobenius norm to define the reconstruction error $\mathcal{E}(\mathbf{V}) = \frac{1}{2}\sum_{i=1}^{n}\|\mathbf{x}_i - \mathbf{X}\mathbf{v}_i\|_2^2 = \frac{1}{2}\|\mathbf{X} - \mathbf{X}\mathbf{V}\|_{\mathrm{F}}^2$, and $\mathcal{R}(\mathbf{V})$ is the regularizer that ensures $\mathbf{V}$ can select representative samples.

The reader may notice that the loss function (1) is similar to those of subspace clustering, e.g., [33] and [34], where the reconstruction error is minimized to generate a self-representation matrix $\mathbf{V}$, and the regualrizer is designed to build a meaningful affinity matrix from $\mathbf{V}$ for the input of spectral clustering [35]. However, we do not target at obtaining an affinity matrix for data clustering, but selecting representative samples by $\mathbf{V}$ from a data collection.

To be concrete, when minimizing the loss function (1), we will use $\mathbf{X}\mathbf{V} = \sum_{i=1}^{n}\mathbf{x}_i\mathbf{v}^i$ to reconstruct $\mathbf{X}$, as shown in Fig. 1 (a). Thus, a subset $\{\mathbf{x}_i|\mathbf{v}^i \neq \mathbf{0}\}$ can be used as a set of exemplar samples to represent the whole dataset $\{\mathbf{x}_i|i = 1, 2, \cdots, n\}$. If $\mathcal{R}(\mathbf{V}) = 0$, then a trivial solution to the problem of minimizing $\ell(\mathbf{V})$ will be $\mathbf{V} = \mathbf{I}$ (identity matrix), which gives rise to zero loss, i.e., $\ell(\mathbf{V}) = 0$. In such a case, each data sample can be perfectly reconstructed by itself. In other words, all the data samples are selected as representatives. In contrast, a well-designed $\mathcal{R}(\mathbf{V}) \neq 0$ will make $\mathbf{V}$ row sparse, allowing us to select representative samples.

Here, we design the regularizer $\mathcal{R}(\mathbf{V})$ to select exemplars with sparisity, diversity, and locality-sensitivity. It consists of three sub-regularizers: the regularizer for row sparsity $\mathcal{R}_1(\mathbf{V})$, the regularizer for diversity $\mathcal{R}_2(\mathbf{V})$, and the regularizer for locality-sensitivity $\mathcal{R}_3(\mathbf{V})$. We combine these sub-regularizers

together in the following:

$$\mathcal{R}(\mathbf{V}) = \sum_{i=1}^{3} \lambda_i \mathcal{R}_i(\mathbf{V}), \tag{2}$$

where each $\lambda_i \geq 0$ is a parameter to balance the reconstruction error and regularizers.

2.1. Row Sparsity Regularizer

First, to construct the matrix $\mathbf{V}$ that will be used to select exemplar samples, we encourage some rows of $\mathbf{V}$ to be zero vectors, *i.e.*, let $\mathbf{V}$ be row sparse. The columns of $\mathbf{X}$ corresponding to the nonzero rows of $\mathbf{V}$ will be used to recover $\mathbf{X}$. To this end, we follow Sparse Dictionary Selection (SDS) [1] and Sparse Modeling Representative Selection (SMRS) [2] to define the l_1/l_q-norm regularizer of $\mathbf{V}$:

$$\mathcal{R}_1(\mathbf{V}) = \|\mathbf{V}\|_{1,q} = \sum_{i=1}^{n} \|\mathbf{v}^i\|_q, \tag{3}$$

where $q > 1$. Throughout this paper, we set $q = 2$.

Because of the sparsity-inducing property of the l_1-norm [36], $\mathcal{R}_1(\mathbf{V})$ will make the vector $(\|\mathbf{v}^1\|_2, \|\mathbf{v}^2\|_2, \cdots, \|\mathbf{v}^n\|_2)^{\mathrm{T}}$ sparse. As a result, $\mathbf{V}$ will have a sparse non-zero row entries.

2.2. Diversity Regularizer

Considering the sample, $\mathbf{x}_i$, which is selected as an exemplar, it is difficult to represent $\mathbf{x}_j$ that is very dissimilar to $\mathbf{x}_i$. Therefore, for a diverse selection, it is preferred to also select $\mathbf{x}_j$ as an exemplar. To this end, we design the following diversity regularizer over $\mathbf{V}$:

$$\mathcal{R}_2(\mathbf{V}) = \sum_{i,j}^{n} \theta_{ij} \left\|\mathbf{v}^i - \mathbf{v}^j\right\|_1 \tag{4}$$

with penalty weights

$$\theta_{ij} = \begin{cases} \mathrm{dSim}(\mathbf{x}_i, \mathbf{x}_j), & \text{if } \mathrm{dSim}(\mathbf{x}_i, \mathbf{x}_j) \geq \max(\mathrm{dSim}_{i,l}, \mathrm{dSim}_{j,l}) \\ 0, & \text{otherwise} \end{cases}, \tag{5}$$

where $\mathrm{dSim}(\mathbf{x}_i, \mathbf{x}_j)$ measures the dissimilarity of selected samples, and $\mathrm{dSim}_{i,l}$ denotes the l^{th} largest element in $\{\mathrm{dSim}(\mathbf{x}_i, \mathbf{x}_j)|j = 1, 2, \cdots, n\}$. In practice, we define $\mathrm{dSim}(\mathbf{x}_i, \mathbf{x}_j)$ using the squared Euclidean distance d_{ij} between $\mathbf{x}_i$ and $\mathbf{x}_j$. We can further normalize the corresponding dissimilarities by $\mathrm{dSim}(\mathbf{x}_i, \mathbf{x}_j) = d_{ij}/\max\{d_{ij}|\, 1 \leq i, j \leq n\}$ to make the range of dissimilarities between 0 and 1. For l, we fix it to 3 in our paper expect for the parameter analysis experiment.

Actually, (5) defines a dissimilarity graph, where the nodes are $\{\mathbf{v}^i|i = 1, 2, \cdots, n\}$ and the edge weights are $\{\theta_{ij}|i, j = 1, 2, \cdots, n\}$ (Fig. 1 (c)). This graph enforces a smooth selection on dissimilar samples, and encourage them to be selected together. Concretely, for similar samples $\mathbf{x}_i$ and $\mathbf{x}_j$, θ_{ij} will be zero, and thus $\theta_{ij}\left\|\mathbf{v}^i - \mathbf{v}^j\right\|_1 = 0$ for any $\mathbf{v}^i$ and $\mathbf{v}^j$. But for very dissimilar samples $\mathbf{x}_i$ and $\mathbf{x}_j$, θ_{ij} will be a large weight. To ensure $\mathcal{R}_2(\mathbf{V})$ is small, $\theta_{ij}\left\|\mathbf{v}^i - \mathbf{v}^j\right\|_1$ should be small, which implies that $\mathbf{v}^i$ and $\mathbf{v}^j$ are similar. Therefore, $\mathbf{x}_i$ and $\mathbf{x}_j$ have similar chances getting selected. In other words, sample $\mathbf{x}_i$ is likely to be selected when its dissimilar sample $\mathbf{x}_j$ is selected, and vise versa. Overall, minimizing $\mathcal{R}_2(\mathbf{V})$ will enforce a diverse selection, thus enhancing the representativeness of selected exemplars.

2.3. Locality-sensitivity Regularizer

To make sample coding natural and meaningful, a common rule is to enable the coding vectors to satisfy the locality-sensitivity property such that similar samples generate similar codes [37]. We thus design the following locality-sensitivity regularizer over $\mathbf{V}$:

$$\mathcal{R}_3\left(\mathbf{V}\right) = \sum_{i,j}^{n} \rho_{ij} \|\mathbf{v}_i - \mathbf{v}_j\|_1 \tag{6}$$

with the weights

$$\rho_{ij} = \begin{cases} \mathrm{Sim}(\mathbf{x}_i, \mathbf{x}_j), & \text{if } \mathrm{Sim}(\mathbf{x}_i, \mathbf{x}_j) \geq \max(\mathrm{Sim}_{i,s}, \mathrm{Sim}_{j,s}) \\ 0, & \text{otherwise} \end{cases}, \tag{7}$$

where $\mathrm{Sim}(\mathbf{x}_i, \mathbf{x}_j)$ measures the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$, and $\mathrm{Sim}_{i,s}$ denotes the s^{th} largest element in $\{\mathrm{Sim}(\mathbf{x}_i, \mathbf{x}_j)|j = 1, 2, \cdots, n\}$. We usually define $\mathrm{Sim}(\mathbf{x}_i, \mathbf{x}_j) = \exp\{-d_{ij}\}$, where $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2^2$. For s, we fix it to 3 in our paper expect for the parameter analysis experiment.

A similarity graph is defined by (7), where the nodes are $\{\mathbf{v}_i|i = 1, 2, \cdots, n\}$ and the edge weights are $\{\rho_{ij}|i, j = 1, 2, \cdots, n\}$ (Fig. 1 (d)). Unlike dissimilarity graph (5), this graph acts on very similar samples $\mathbf{x}_i$ and $\mathbf{x}_j$, and prefers smoothing $\mathbf{v}_i$ and $\mathbf{v}_j$ to make them similar. Overall, $\mathcal{R}_3(\mathbf{V})$ will leverage the locally dependent information among similar data to avoid a trivial solution of $\mathbf{V}$, thus guaranteeing the representativeness of selected exemplars.

By defining the above three sub-regularizers, we can rewrite the loss function (1) and formulate our objective as follows:

$$\begin{aligned} \min_{\mathbf{V} \in \mathbb{R}^{n \times n}} & \frac{1}{2} \|\mathbf{X} - \mathbf{X}\mathbf{V}\|_{\mathrm{F}}^2 + \lambda_1 \|\mathbf{V}\|_{1,2} \\ & + \lambda_2 \sum_{i,j}^{n} \theta_{ij} \|\mathbf{v}^i - \mathbf{v}^j\|_1 + \lambda_3 \sum_{i,j}^{n} \rho_{ij} \|\mathbf{v}_i - \mathbf{v}_j\|_1. \end{aligned} \tag{8}$$

As is shown, it contains four complex terms, thus challenging the optimization. Despite that, we will show that it can be deemed as a proximal gradient problem, which can be solved by the fast iterative shrinkage thresholding algorithm (FISTA) through the Proximal-Dykstra method involving the calculation of parametric maximum flows.

3. Optimization

It is worth noting that the objective in (8) includes four convex terms, the first one is smooth, and the others are nonsmooth. Hence it is difficult to apply classical gradient algorithm. Fortunately, however, we can convert it into a proximal gradient problem to analyze and obtain the solution.

3.1. Proximal Gradient

The proximal gradient method optimizes (1) by iteratively solving

$$\text{prox}_{\mathcal{R}}\left(\mathbf{Z}\right) = \underset{\mathbf{V}\in\mathbb{R}^{n\times n}}{\arg\min} \frac{1}{2}\left\|\mathbf{V}-\mathbf{Z}\right\|_{\text{F}}^{2} + \frac{1}{L}\mathcal{R}\left(\mathbf{V}\right), \tag{9}$$

where

$$\mathbf{Z} = \mathbf{V} - \frac{1}{L}\frac{\partial}{\partial \mathbf{V}}\mathcal{E}(\mathbf{V}) = \mathbf{V} + \frac{1}{L}\mathbf{X}^{\text{T}}\left(\mathbf{X}-\mathbf{X}\mathbf{V}\right) \tag{10}$$

and L is an upper bound of the Lipschitz constant of $\frac{\partial}{\partial \mathbf{V}}\mathcal{E}(\mathbf{V})$. Since $\frac{\partial}{\partial \mathbf{V}}\mathcal{E}(\mathbf{V})$ is a linear function of $\mathbf{V}$, it is easy to compute the Lipschitz constant [38] as

$$L = \left\|\mathbf{X}\mathbf{X}^{\text{T}}\right\|_{\text{F}}. \tag{11}$$

To accelerate the proximal gradient procedure, we leverage the FISTA method, which is known to converge to a true solution with a fast convergence rate $O(1/k^2)$ in k iterations [39]. We show the FISTA procedure in Algorithm 1. Although we use FISTA, the nonsmooth terms still wait to be optimized as shown in (9). Considering that the nonsmooth terms intertwine with each other, which is intractable to optimize simultaneously, we next split (9) into two proximal subproblems using the Proximal-Dykstra method.

Algorithm 1 FISTA for Problem (8).

Input: $\mathbf{X}$

Output: $\mathbf{V}$

$L \leftarrow \left\|\mathbf{X}\mathbf{X}^{\mathrm{T}}\right\|_{\mathrm{F}}$

▷ Lipschitz constant (Equation (11))

$\mathbf{V} \leftarrow \mathbf{0}$, $\mathbf{W} \leftarrow \mathbf{V}$, $t \leftarrow 1$

▷ Initialization

repeat

  $\mathbf{Z} \leftarrow \mathbf{W} + \frac{1}{L}\mathbf{X}^{\mathrm{T}}(\mathbf{X} - \mathbf{X}\mathbf{W})$

  ▷ Equation (10)

  $\mathbf{U} \leftarrow \mathbf{V}$, $\mathbf{V} = \mathrm{prox}_{\mathcal{R}}(\mathbf{Z})$

  ▷ Problem (9)⇐ Proximal-Dykstra (Alg. 2)

  $s = t - 1$, $t \leftarrow (1 + \sqrt{1 + 4t^2})/2$

  $\mathbf{W} \leftarrow \mathbf{V} + s(\mathbf{V} - \mathbf{U})/t$

until convergence

3.2. Proximal Splitting

To solve (9), we expand it into

$$\begin{aligned}\mathrm{prox}_{\mathcal{R}}(\mathbf{Z}) = \underset{\mathbf{V}\in\mathbb{R}^{n\times n}}{\arg\min}\ & \frac{1}{2}\|\mathbf{V} - \mathbf{Z}\|_{\mathrm{F}}^2 + \hat{\lambda}_1\|\mathbf{V}\|_{1,2} \\ & + \hat{\lambda}_2 \sum_{i,j}^{n} \theta_{ij}\|\mathbf{v}^i - \mathbf{v}^j\|_1 + \hat{\lambda}_3 \sum_{i,j}^{n} \rho_{ij}\|\mathbf{v}_i - \mathbf{v}_j\|_1,\end{aligned} \tag{12}$$

where $\hat{\lambda}_i$ denotes λ_i/L. Since this includes row-wise and column-wise fused terms in addition to a group regularization term, it is not straightforward to solve. We thus consider it as a proximal splitting problem and obtain the solution by the Proximal-Dykstra method [32], as shown in Algorithm 2. According to the Proximal-Dykstra

method, we are required to solve (9) when $\hat{\lambda}_2 = 0$,

$$\begin{aligned}\text{prox}_{\hat{\lambda}_1\mathcal{R}_1+\hat{\lambda}_3\mathcal{R}_3}(\mathbf{Z}) = \underset{\mathbf{V}\in\mathbb{R}^{n\times n}}{\arg\min}\ &\frac{1}{2}\|\mathbf{V}-\mathbf{Z}\|_{\text{F}}^2 \\ &+ \hat{\lambda}_1\|\mathbf{V}\|_{1,2} + \hat{\lambda}_3\sum_{i,j}^{n}\rho_{ij}\|\mathbf{v}_i - \mathbf{v}_j\|_1,\end{aligned} \tag{13}$$

and when $\hat{\lambda}_1 = \hat{\lambda}_3 = 0$,

$$\text{prox}_{\hat{\lambda}_2\mathcal{R}_2}(\mathbf{Z}) = \underset{\mathbf{V}\in\mathbb{R}^{n\times n}}{\arg\min}\ \frac{1}{2}\|\mathbf{V}-\mathbf{Z}\|_{\text{F}}^2 + \hat{\lambda}_2\sum_{i,j}^{n}\theta_{ij}\|\mathbf{v}^i - \mathbf{v}^j\|_1, \tag{14}$$

As is well known, the Proximal-Dykstra method has a linear convergence rate to an exact solution [32, 40]. To apply Algorithm 2, we need to solve (13) and (14), which are still complex and non-trivial to solve. We will continue to decompose Problems (13) and (14), and finally optimize the problems by parametric flow maximization.

3.3. Proximal Decomposition

Following the strategy in [43], we can equivalently decompose Problem (13) into n pairs of proximal operators, i.e., for $i = 1, 2, \cdots, n$,

$$\mathbf{h}^i = \underset{\mathbf{q}\in\mathbb{R}^n}{\arg\min}\ \frac{1}{2}\|\mathbf{q}-\mathbf{z}^i\|_2^2 + \hat{\lambda}_3\sum_{i,j}^{n}\rho_{ij}|q_i - q_j|, \tag{15}$$

and

$$\mathbf{v}^i = \underset{\mathbf{q}\in\mathbb{R}^n}{\arg\min}\ \frac{1}{2}\|\mathbf{q}-\mathbf{h}^i\|_2^2 + \hat{\lambda}_1\|\mathbf{q}\|_2. \tag{16}$$

As a result, we have $\text{prox}_{\hat{\lambda}_1\mathcal{R}_1+\hat{\lambda}_3\mathcal{R}_3}(\mathbf{Z}) = [\mathbf{v}^1, \mathbf{v}^2, \cdots, \mathbf{v}^n]^{\text{T}}$. Similarly, for Problem (14), we equivalently have $\text{prox}_{\hat{\lambda}_2\mathcal{R}_2}(\mathbf{Z}) = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_n]$, in which

$$\mathbf{v}_i = \underset{\mathbf{q}\in\mathbb{R}^n}{\arg\min}\ \frac{1}{2}\|\mathbf{q}-\mathbf{z}_i\|_2^2 + \hat{\lambda}_2\sum_{i,j}^{n}\theta_{ij}|q_i - q_j|. \tag{17}$$

Algorithm 2 Proximal-Dykstra Algorithm used in Algorithm 1.

Input: $\mathbf{Z}$, $\{\hat{\lambda}_i\}_{i=1}^3$

Output: $\mathbf{V}$

$\mathbf{V} \leftarrow \mathbf{Z}$, $\mathbf{P} \leftarrow \mathbf{0}$, $\mathbf{Q} \leftarrow \mathbf{0}$

▷ Initialization

repeat

  $\mathbf{Y} \leftarrow \text{prox}_{\hat{\lambda}_1\mathcal{R}_1+\hat{\lambda}_3\mathcal{R}_3}(\mathbf{V}+\mathbf{P})$

  ▷ Problem (13) ⇐ parametric max-flow ([41, 42])

  $\mathbf{P} \leftarrow \mathbf{V}+\mathbf{P}-\mathbf{Y}$

  $\mathbf{V} \leftarrow \text{prox}_{\hat{\lambda}_2\mathcal{R}_2}(\mathbf{Y}+\mathbf{Q})$

  ▷ Problem (14) ⇐ parametric max-flow ([41, 42])

  $\mathbf{Q} \leftarrow \mathbf{Y}+\mathbf{Q}-\mathbf{V}$

until convergence

Since Problem (16) is a (non-overlapping) group lasso signal approximator, it can be exactly solved by soft-thresholding [44, 45]:

$$\mathbf{v}^i = \mathbf{h}^i \max\{(1-\frac{\hat{\lambda}_1}{\|\mathbf{h}^i\|_2}), 0\}. \tag{18}$$

For Problems (15) and (17), they have the same configurations except for the different arguments, which can be converted into solving parametric maximum flows[1]. In our implementation, we use the efficient GGT algorithm which can find an exact solution of parametric flow maximization [42, 46]. In the worst case, the time costs to solve (15) and (17) are $O(\frac{s+4}{2}n^2\log(\frac{2n}{s+4}))$ and $O(\frac{l+4}{2}n^2\log(\frac{2n}{l+4}))$, respectively. The main time cost to solve (13) and (14) in each iteration is then $O(\frac{n^3}{2}((s+4)\log(\frac{2n}{s+4})+(l+4)\log(\frac{2n}{l+4})))$. For k_1

[1]The details of solving (15) and (17) are shown in Appendix.

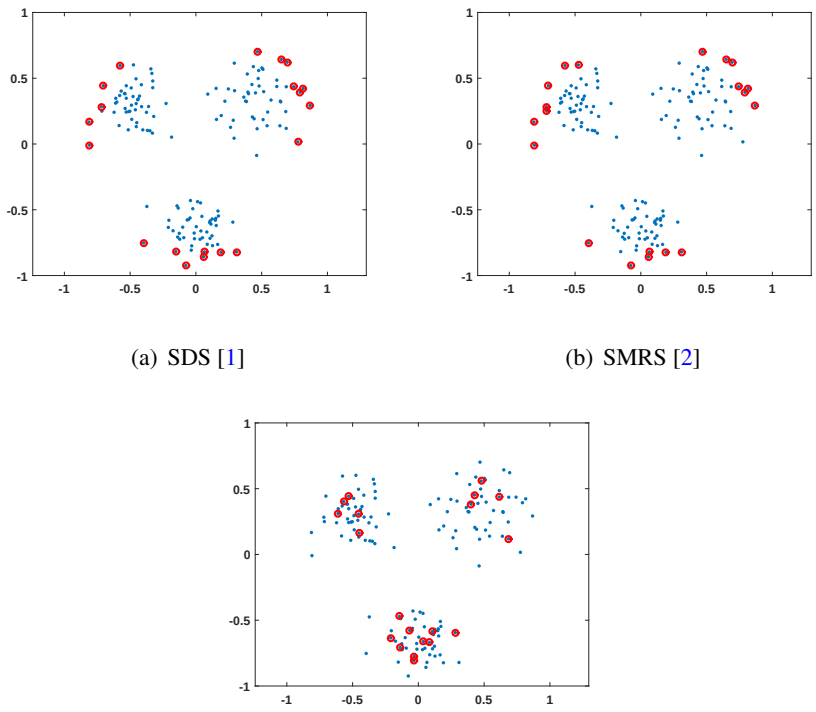
iterations of Proximal-Dykstra and k_2 iterations of FISTA, the time complexity is $O(\frac{k_1 k_2}{2} n^3((s+4)\log(\frac{2n}{s+4}) + (l+4)\log(\frac{2n}{l+4})))$, which is approximately different between $\frac{k_1 k_2}{2}((s+4)\log(\frac{2n}{s+4}) + (l+4)\log(\frac{2n}{l+4}))$ and k from SDS [1] and SMRS [2] that have the computational complexity of $O(kn^3)$ in k iterations.

4. Parameter setting

In Eq. (5), we introduce parameter l to control the diversity of the selected representatives. A small l tends to only select highly dissimilar samples while a large l may tolerate similar ones. We used a small value of l (i.e., 3) in the experiments. Another parameter s in Eq. (7) encourages similar samples to share similar sparse codes. A small s will only force highly similar samples to have similar codes while a large s may encode dissimilar samples similarly. We thus used a small value of s (i.e., 3) in the experiments.

It is data dependent to set λ_i for $i = 1, 2, 3$. In fact, they are related to the Lipschitz constant L as in Eq. (11). To facilitate the parameter tuning, we analyze the relationship between λ_i and the Lipschitz constant to decouple the dependency between them, acquiring $\hat{\lambda}_i = \lambda_i / L$ as in Eq. (12). We further introduce a hyper-parameter α, as in [2], to compute $\hat{\lambda}_1 = \lambda_0 / \alpha$, where λ_0 is revealed by analyzing the data samples themselves [2]. To reduce the number of free parameters, we set $\hat{\lambda}_i = \lambda_0 / \alpha$ for each $i = 2, 3$. In our experiments, we set α in the interval $[2, 30]$ for our method.

It is worth noting that SDS [1] can be viewed as a special case of our method when $\lambda_2 = \lambda_3 = 0$, and SMRS [2] evolves from SDS by enforcing an affine constraint, $\mathbf{1}^{\mathrm{T}}\mathbf{V} = \mathbf{1}^{\mathrm{T}}$. Therefore, we choose SDS and SMRS as the baselines to justify the effectiveness of our proposed method. Both SDS and SMRS can be tuned by setting α for λ_1.



(a) SDS [1] (b) SMRS [2]

(c) SSDS

Figure 2: Data points in three clusters (blue dots) and the representatives (red circles) found by (a) SDS [1] (b) SMRS [2] and (c) our proposed SSDS.

5. Experimental Results

5.1. Experiments on Synthetic Data

To evaluate the effectiveness of our method in diverse selection on synthetic data, we conduct experiments compared with Sparse Dictionary Selection (SDS) [1] and its variation, Sparse Modeling Representative Selection (SMRS) [2]. We refer to our proposed method as Structured Sparse Dictionary Selection (SSDS). Similar to our method, each compared method can generate a selection matrix. We use $\mathbf{V} = \{v_{ij}\}$ to denote the selection matrix, where the absolute value of v_{ij}

depicts the ability of the i^{th} data point reconstructing the j^{th} data point. Hence, exemplars are ranked and selected according to the l_2 norms of the rows of $\mathbf{V}$ as used by SDS and SMRS. Before performing the ranking, we also adjust $\mathbf{V}$ by $\mathbf{V} \leftarrow \mathbf{V}./(\mathbf{V}^{\mathrm{T}}+\epsilon)$ (i.e., $v_{ij} \leftarrow v_{ij}/(v_{ji}+\epsilon)$), where $./$ and $+$ denote element-wise operations of division and addition, and ϵ is a small number to avoid dividing by zero. In this way, a data point is likely to be an exemplar if that point has a large reconstruction coefficient to others, while other points are difficult to reconstruct that point.

We consider the data shown in Fig. 2, which consists of data points in three clusters. We show top 20 representatives by each method in Figs. 2 (a)-(c). As can be seen, to ensure a lower linear reconstruction error, both SDS and SMRS select points at the border of the convex hull of the dataset. In contrast, by considering the diversity regularizer in (4) and locality-sensitivity regularizer in (6) for dictionary selection, our proposed SSDS can select points in the clusters and diversify them between clusters. For our method, we use $l = 3$ and $s = 3$ to build our diversity and locality-sensitivity regularizers, and set α to 10 to combine the regularizers. We notice that no matter how to set the parameters of SDS and SMRS, they prefer keeping the vertices of convex hull of the given dataset. Hence, we let SDS and SMRS produce sufficient number of exemplars for ranking by parameter tuning, so that we can observe the distribution of selected points and compare them with our proposed method.

5.2. Evaluation by Classification

In this section, we conduct experiments on the 10 categories of U.S. Postal Service (USPS) handwritten digit dataset [47] and 8 classes of scene categorization dataset [48] to justify the effectiveness of our regularizers (4) and (6) upon the row sparsity regularizer (3). To define (4) and (6), we measure the dissimilarity of

	Number of Representatives	DT	RF	NN	SVM
All	1000	80.94%	88.30%	94.39%	92.32%
SDS [1]	100	61.40%	69.74%	83.50%	79.54%
SMRS [2]	100	**65.61%**	69.73%	83.26%	82.73%
SSDS	100	62.29%	**71.07%**	**84.30%**	**85.67%**

Table 1: Classification results on USPS digit data.

pairwise samples using normalized squared Euclidean distance, and the similarity using the exponential of negative squared Euclidean distance. When performing representative selection, we describe each digit using its 16×16 pixel values. Each scene image is described by a 512-dimensional GIST feature [48]. For each digit/scene class, we randomly take 100 samples from their respective datasets. Then, we obtain 1000 samples in the digit dataset for representative selection, while in the sence dataset, the number of samples is 800. We next train multiple classifiers on the selected exemplars and report in Table 1 the classification accuracy on the remaining 4649 test digit samples in the digit dataset, and in Table 2 the classification accuracy on the remaining 1888 image samples in the scene dataset. The classification methods considered are Decision Tree (DT) [49], Random Forest (RF) [50] of 15 trees with maximum depth of 50, Nearest Neighbor (NN) [51] and linear SVM [52, 53] with 5-fold cross validation.

For a fair comparison, we select 100 exemplars by each compared method. Regarding the performance, those after representative selection will be inferior to those without representative selection because of the decrease of the number of training samples. However, by contrasting SDS, SMRS and the proposed SSDS, the results in Table 1 and 2 show that this issue has less of an effect on our method. Thanks to our new formulation for representative selection in

	Number of Representatives	DT	RF	NN	SVM
All	800	55.14%	70.71%	70.82%	80.14%
SDS [1]	100	44.65%	52.44%	**64.78%**	65.63%
SMRS [2]	100	**46.50%**	55.99%	62.76%	69.23%
SSDS	100	44.39%	**56.09%**	60.49%	**69.65%**

Table 2: Classification results on scene categorization data.

Equation (8), for either the digit dataset or the scene dataset, our method achieves the best performance by combining with the SVM classifier. Especially, for the digit dataset, our method outperforms SDS and SMRS on most of the used classifiers. This further validates that, by applying the constraints of diversity (Equation (4)) and locality-sensitivity (Equation (6)), the selected exemplars using our method show the qualities of both diversity and representativeness, thus suitable for classifier training. In this experiment, we set $l = 3$ and $s = 3$ for both digit and scene datasets in building our diversity and locality-sensitivity regularizers. For each of the compared method, the hyper-parameter is tested on a range of $\alpha \in \{5, 10, 15, 20, 25, 30\}$. After obtaining the selection matrix $\mathbf{V}$ of each method, we rank and select exemplars according to the l_2 norm of the rows of $\mathbf{V}$ and $\mathbf{V}./(\mathbf{V}^{\mathrm{T}} + \epsilon)$. The best classification accuracy result is eventually reported for each classification method.

Besides comparing with other methods, we also evaluate our method with different parameters on the USPS datastet. The classification accuracies are shown in Fig. 3, where we train a SVM classifier using representatives selected by our method for each setting of parameters. The top figure shows the performance when fixing l and s to 3, but changing α from 5 to 30, while the bottom figure shows the performance when fixing α to 10, but changing l and s together from 3 to 24. Since

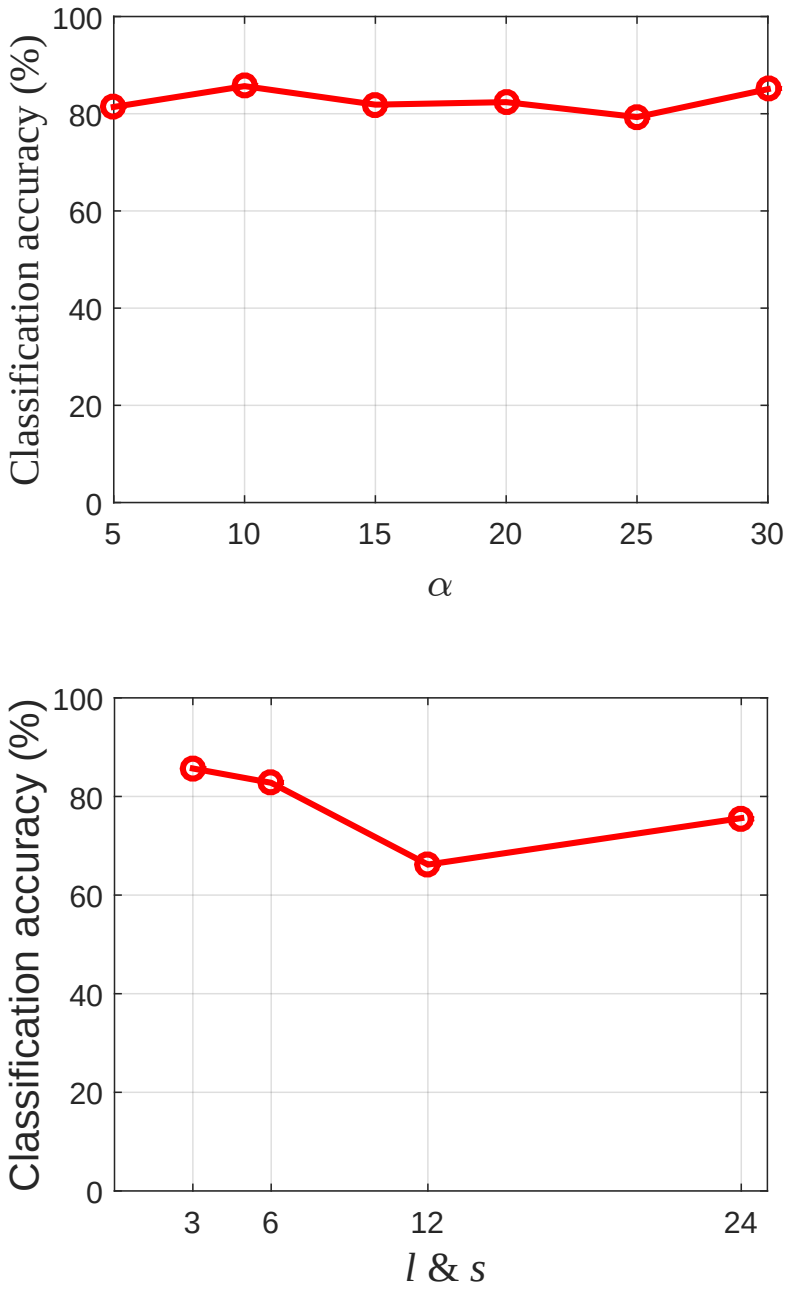


Figure 3: Performance of SVM classifier trained by selected representatives using our method with different parameters: (top) $\alpha \in [5, 30]$, $l = 3$, $s = 3$; (bottom) $\alpha = 10$, $l = s \in [3, 24]$.

we have partly decoupled the dependency between regularization parameters and data dependent information by introducing the hyper-parameter α in Section 4, as can be seen from Fig. 3 (top), different values of α do not influence much for classifier training. Fig. 3 (bottom) suggests that our method prefers a smaller value of l and s, which conforms to our analysis in Section 4. The best performance is achieved by $\alpha = 10, s = 3, l = 3$, which is 85.67%.

5.3. Robustness to outliers

We will in this section discuss and investigate the robustness of our proposed SSDS to the existence of outliers in data compared with existing methods. Representative selection by SDS [1] and SMRS [2] both prefer to select the outliers. This is because these outliers contribute the smallest reconstruction errors to other samples using the smallest coding coefficients, which also results in the smallest cost of row sparsity regularizer (3). That is, with these selected outliers, SDS and SMRS can minimize their objective function, i.e., the total cost of reconstruction error and row sparsity regularizer. In contrast, our method makes a trade-off between selecting and rejecting outliers using the diversity regularizer defined by (4), which prefers very dissimilar data to contribute similar coding coefficients to other samples. As is known, outliers are usually dissimilar to other samples. Hence, by reducing the cost of (4), our SSDS can reject the outliers by assigning their coding coefficients by zero.

To evaluate this point empirically, we use all the 435 face images in Caltech 101 dataset [54] and regard Google downloaded images in Caltech 101 as outliers. Each image is described by a 512-dimensional GIST feature [48]. We randomly select outliers with $\{20\%, 30\%, 40\%, 50\%\}$ of the number of face images. As the number of outliers increases, we show in Fig. 4 (top) the percentage of outliers among the selected representatives. As can be seen, our proposed SSDS consistently results

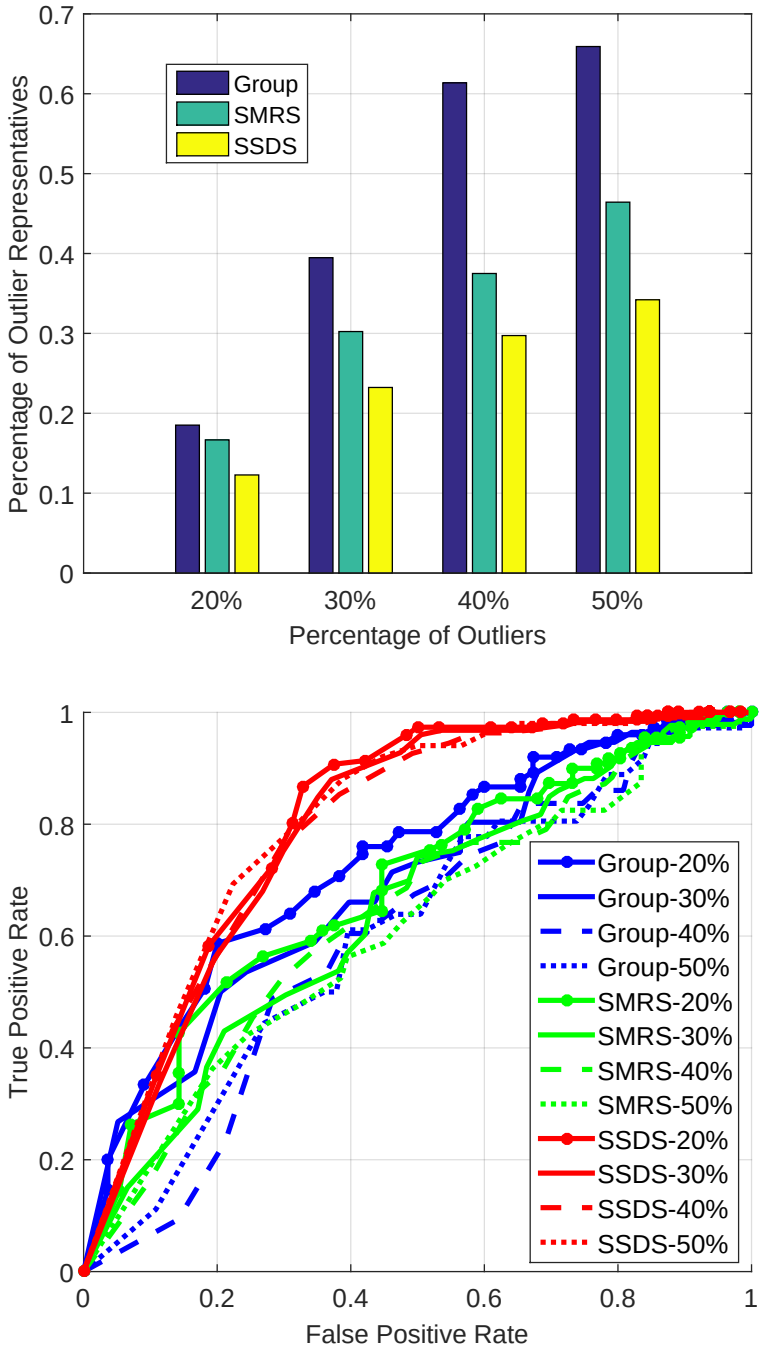


Figure 4: Comparison of different approaches using (top) percentage of selected outliers and (bottom) ROC curves.

in less outliers than the SDS and SMRS. We also show the ROC curves [55] for comparison in Fig. 4 (bottom). For different numbers of outliers, our SSDS can generally outperform SDS and SMRS with a higher true positive rate and a lower false positive rate. In the experiments, to build our diversity and locality-sensitivity regularizers, we define the dissimilarity and similarity of pairwise samples using the normalized squared Euclidean distance and the exponential of negative squared Euclidean distance as described in Eqs. (4) and (6), and set $l = 3$ and $s = 3$. In comparison, we use the same hyper-parameter $\alpha = 15$ for the compared methods to generate the results in Fig. 4. The ranking of selected samples is based on the l_2 norm of the rows of selection matrix.

5.4. Video Summarization

For a further evaluation, we apply our method to video summarization, which is compared with the methods of SDS [1] and SMRS [2]. In video summarization, it is expected that each activity is less repeatedly selected. Thus, for each method, we further follow [2] to prune selected frames from those with too-close appearance features. We used a commercial video clip taken from YouTube.com [56], which is about the Beneful dog food including 8 continuous activities/events. After video sampling at 2 frames/s, we generated 51 key frames. To describe the key frame images, we reduce the size of each frame to a descriptor of length $32 \times 32 \times 3$ [57].

In Fig. 5, we show the result by our method, where we fix $l = 3$ and $s = 3$, and use the normalized squared Euclidean distance and the exponential of negative squared Euclidean distance to measure the dissimilarity and similarity of pairwise samples. For the hyper-parameter α, we set it to 10. To select exemplar frames, we rank the l_2 norm of the rows of selection matrix. As can be seen from the result, the representatives can cover all the 8 activities without redundancy.

We show the comparison results in Fig. 6, including those obtained by SDS,

Figure 5: Video frames from a Beneful commercial video clip. The highlighted frames are the representatives selected by our method.

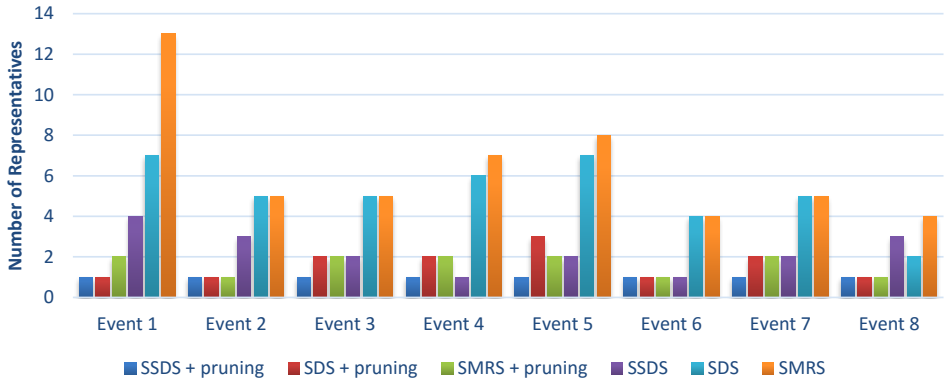
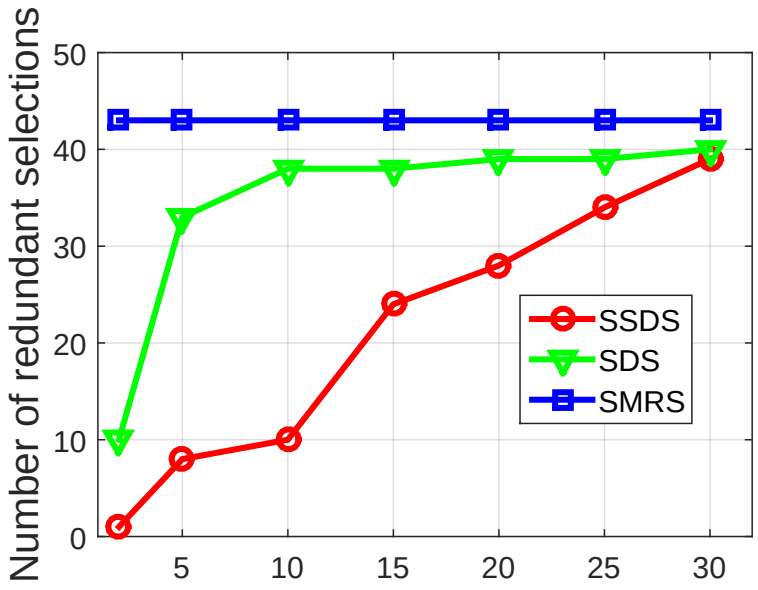


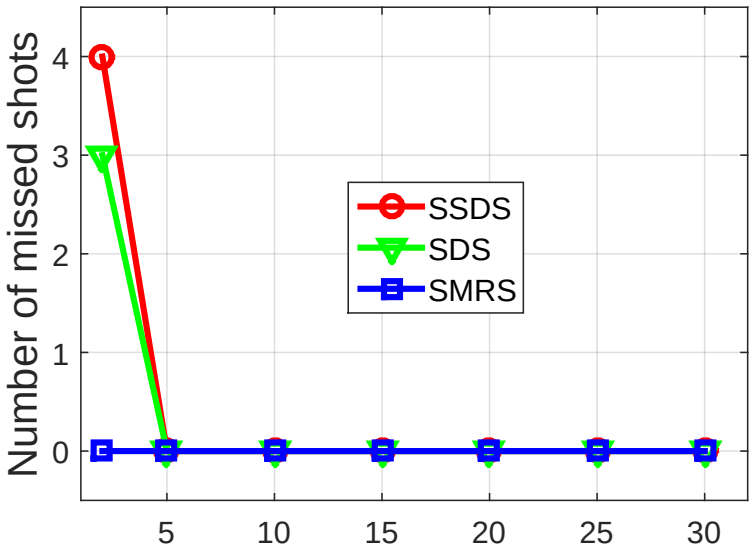
Figure 6: Comparison of number of selected representatives for each shot in the Beneful commercial video.

SMRS and our SSDS before and after pruning similar frames. For SDS and SMRS, we show the best result with the hyper-parameter $\alpha \in \{2, 5, 10, 15, 20, 25, 30\}$. We can see that, no matter whether there is pruning or not, our method can choose a smaller number of representatives to cover the whole activities. Even without pruning, our method is comparable to the pruning results of SDS and SMRS in some events, e.g., Events $3 \sim 7$.

To study how hyper-parameter α influences the summarization result before pruning in comparison with SDS [1] and SMRS [2], we conducted summarization



(a) Number of redundant selection



(b) Number of missed shots

Figure 7: Comparison of representative selection from the Beneful commercial video in (a) number of redundant selections and (b) number of missed shots on different hyper-parameters.

experiments on video frames when α equals to one of $\{2, 5, 10, 15, 20, 25, 30\}$. We again followed [56] to use the Beneful commercial video.

We show the results in Fig. 7, where the number of redundant selections means the total number of representatives exceeding 1 in an event while the number of missed shots stands for how many shots do not appear in the selected representatives. We can see from Fig. 7, with increased α, there is increase in the number of redundant selections and decrease in the number of missed shots for SDS and ours. This is because a larger α results in a weaker constraint on the row sparsity regularizer (Eq. (3)), leading a less sparse and more complete selection. As SMRS adds an affine constraint on the selection matrix, i.e., $\mathbf{1}^{\mathrm{T}}\mathbf{V} = \mathbf{1}^{\mathrm{T}}$, beyond the objective of the SDS method, it usually cannot allow the selection matrix $\mathbf{V}$ to be a matrix with many zero rows. It hence nearly always selects the vertices in the convex hull spanned by input data [2], regardless of the choice of α. Although without missed shots, SMRS results in a more redundant selection than SDS and our SSDS method.

6. Conclusion

Selecting a subset of data samples to represent the whole dataset, i.e., representative selection, is a fundamental problem for data analysis. In this paper, we proposed a novel formulation to find representatives in data samples via learning with structured sparsity. For the selection of representatives with both diversity and representativeness, we formulated the problem as a structurelly-regularized learning, where the objective function consists of a reconstruction error and three structured regularizers: (1) group sparsity regularizer, (2) diversity regularizer, and (3) locality-sensitivity regularizer. The results in image classification, outlier removal, and video summarization seem to show the advantages of our new

formulation.

Appendix

In this appendix, we show how to transform Problems (15) and (17) into minimum-norm-point problems under submodular constraints, and efficiently obtain the solutions by parametric flow maximization. As the two problems have the same configurations except for the different arguments, we take Problem (17) as an example for the below discussion.

Minimum-Norm-Point

We first show the relation between the penalty $\hat{\lambda}_2 \sum_{i,j}^{n} \theta_{ij}|q_i - q_j|$ in (17) and a cut function by Lovász extension. Then we apply the submodular property of cut functions to transform Problem (17) into a Minimum-Norm-Point (MNP) problem under submodular constraints.

To begin with, we denote a finite set by $\mathcal{V} = \{1, 2, \cdots, n\}$, where a cut function of a set $S \subseteq \mathcal{V}$ is defined on a set of non-negative weights $\omega : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}_+$:

$$f_c(S) = \sum_{i \in S, j \in \mathcal{V} \setminus S} \omega_{ij}$$

We next apply the following definition and lemma of Lovász extension.

Definition 1 ([58]). *Given any set-function f on the set $\mathcal{V}$ such that $f(\emptyset) = 0$, the Lovász extension F: $\mathbb{R}^n \rightarrow \mathbb{R}$ is defined as: $F(\mathbf{q}) = \int_{-\infty}^{+\infty} f(\{k \in \mathcal{V} \mid q_k \geq a\})da$.*

Lemma 1. *The term of $\hat{\lambda}_2 \sum_{i,j}^{n} \theta_{ij}|q_i - q_j|$ in (17) is equivalent to the Lovász extension of a cut function.*

Proof. Let $\omega_{ij} = \hat{\lambda}_2 \theta_{ij}$, then we have

$$\hat{\lambda}_2 \sum_{i,j}^{n} \theta_{ij} |q_i - q_j| = \sum_{i,j}^{n} \omega_{ij} |q_i - q_j|. \tag{19}$$

It has been shown in [58] that a function of the form of Equation (19) corresponds to the Lovász extension of the cut function:

$$f_c(S) = \sum_{i \in S, j \in \mathcal{V} \setminus S} \omega_{ij}, \ S \subseteq \mathcal{V}$$

□

With this Lemma, we define a submodular function by adding a modular term to the cut function f_c and obtain

$$g(S) = f_c(S) - \mathbf{z}_i(S),$$

where $\mathbf{z}_i(S) = \mathbf{z}_i^{\mathrm{T}} \mathbf{1}_S$, and $\mathbf{1}_S$ is the indicator vector of the set S. Next, we let $\mathcal{B}(g)$ be the base polyhedron of g:

$$\mathcal{B}(g) = \{\mathbf{x} \in \mathbb{R}^n : \forall S \subseteq \mathcal{V}, x(S) \leqslant g(S); x(\mathcal{V}) = g(\mathcal{V})\}.$$

Following [59], we have the following proposition.

Proposition 1 ([59]). *Let $\mathbf{t}^*$ be a minimizer of the Minimum-Norm-Point problem on the base polyhedron $\mathcal{B}(g)$:*

$$\mathbf{t}^* = \arg\min_{\mathbf{t} \in \mathbb{R}^n} \{ \|\mathbf{t}\|_2^2 \, | \mathbf{t} \in \mathcal{B}(g) \}. \tag{20}$$

Then a minimizer of Problem (17) is obtained by $\mathbf{v}_i^* = -\mathbf{t}^*$.

To solve Problem (20), we will follow [60] to apply an efficient parametric flow algorithm [42].

Parametric max-flow

As shown in [61], parametric flow algorithms can be applied to separable convex minimization problems under non-decreasing submodular function constraints. Problem (20) is a separable convex minimization problem, as the squared l_2-norm is convex. In addition, $g(S) = f_c(S) - \mathbf{z}_i(S)$ is submodular, though not necessarily non-decreasing. We further apply Lemmas 2 and 3 to meet the non-decreasing requirement.

Lemma 2 ([60]). *For any $\gamma \in \mathbb{R}$ and any submodular function f, $\mathbf{t}^*$ is an optimal solution to $\min_{t \in \mathcal{B}(f)} \|\mathbf{t}\|_2^2$ if and only if $\mathbf{t}^* + \gamma \mathbf{1}$ is an optimal solution to $\min_{t \in \mathcal{B}(f+\gamma \mathbf{1})} \|\mathbf{t}\|_2^2$.*

Lemma 3 ([60]). *For a submodular function f, set $\gamma = \max_{i=1,2,\cdots,n}\{0, f(\mathcal{V} \setminus \{i\}) - f(\mathcal{V})\}$. Then $f + \gamma \mathbf{1}$ is a nondecreasing submodular function.*

After applying Lemma 3 to $g(S)$, we obtain the non-decreasing submodular function

$$g'(S) = f_c(S) - \mathbf{z}_i(S) + \gamma \mathbf{1}(S).$$

Then for the parametric set function

$$g'_\beta(S) = g'(S) - \beta \mathbf{1}(S) = f_c(S) - \mathbf{z}_i(S) + (\gamma - \beta)\mathbf{1}(S), \beta \geqslant 0, S \subseteq \mathcal{V}, \quad (21)$$

there are $l + 1 (\leqslant n)$ subsets

$$(\emptyset =) S_0 \subseteq S_1 \subseteq \cdots \subseteq S_l \, (= \mathcal{V}) \quad (22)$$

and $l + 1$ intervals

$$R_0 = [0, \beta_1), \cdots, R_j = [\beta_j, \beta_{j+1}), \cdots, R_l = [\beta_l, +\infty)$$

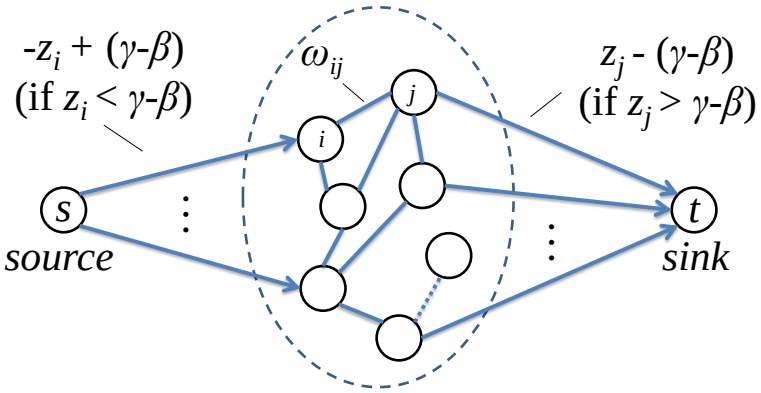


Figure 8: Graph construction for the parametric set function (21).

such that $\forall j \in \{0, 1, \cdots, l\}$, S_j is the maximal minimizer of $g'_\beta(S)$ for all $\alpha \in R_j$ [61]. Then the unique optimal solution $\mathbf{o}^* = \arg\min\{\|\mathbf{o}\|_2^2 \,|\mathbf{o} \in \mathcal{B}(g')\}$, is determined by, for $i \in V \cap (S_{j+1} \backslash S_j)$ $(j \in \{0, 1, \cdots, l-1\})$,

$$o_i^* = \frac{\mathcal{G}(S_{j+1}) - \mathcal{G}(S_j)}{\text{card}(S_{j+1} - S_j)}, \tag{23}$$

where $\text{card}(S)$ denotes the cardinality of the set S. If the collection $S^* = \{S_0, S_1, \cdots, S_l\}$ is computed, Equation (23) and Lemma 2 together imply that Problem (20) can be immediately solved by

$$\mathbf{t}^* = \mathbf{o}^* - \gamma \mathbf{1}. \tag{24}$$

Proposition 2 ([41]). *For any cut function f_c, minimizing the set function (21) with a given β is equivalent to solving the max-flow problem with the s/t graph defined in Fig. 8.*

As is known, the GGT algorithm [42, 46] can be applied to find an exact solution to the parametric max-flow problem (21) for all $\beta \geqslant 0$, thus finding all

solutions to Equation (22). Finally, we can obtain the solution of Problem (20) by applying Equations (23) and (24). The GGT algorithm has the worst-case complexity of $O(d|E|\log(d^2/|E|))$, where d is the number of nodes, and $|E|$ is the number of edges [42].

Fig. 8 indicates that, for Problem (17), we have $d = n + 2$, $|E| \approx \frac{ln}{2} + 2n = (\frac{l+4}{2})n$. Then, we have $d|E|\log(d^2/|E|) \approx n(\frac{l+4}{2})(n+2)\log((n+2)^2/((\frac{l+4}{2})n)) = (\frac{l+4}{2})n(n+2)\log(\frac{2(n+2)^2}{(l+4)n})$. Thus, in the worst-case, the time complexity is $O(\frac{l+4}{2}n^2\log(\frac{2n}{l+4}))$. Similarly, for Problem (15), the worst-case time complexity is $O(\frac{s+4}{2}n^2\log(\frac{2n}{s+4}))$.

Acknowledgments

This work is supported in part by National Natural Science Foundation of China under Grant 61602069, Chongqing Research Program of Basic Research and Frontier Technology (No. cstc2016jcyjA0468), JSPS KAKENHI Grant Numbers JP16H01548 and JP26280086, JSPS-NTU joint grant M4080882, and Singapore Ministry of Education Academic Research Fund Tier 2 MOE2015-T2-2-114.

References

[1] Y. Cong, J. Yuan, J. Liu, Sparse reconstruction cost for abnormal event detection, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 3449–3456.

[2] E. Elhamifar, G. Sapiro, R. Vidal, See all by looking at a few: Sparse modeling for finding representative objects, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 1600–1607.

[3] Y. Cong, J. Yuan, J. Luo, Towards Scalable Summarization of Consumer Videos Via Sparse Dictionary Selection, IEEE Transactions on Multimedia 14 (1) (2012) 66–75.

[4] J. Meng, H. Wang, J. Yuan, Y. Tan, From Keyframes to Key Objects: Video Summarization by Representative Object Proposal Selection, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[5] Y. Li, L. Maguire, Selecting critical patterns based on local geometrical and statistical information, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (6) (2011) 1189–1201.

[6] S. García, J. R. Cano, F. Herrera, A memetic algorithm for evolutionary prototype selection: A scaling up approach, Pattern Recognition 41 (8) (2008) 2693–2709.

[7] J. Calvo-Zaragoza, J. J. Valero-Mas, J. R. Rico-Juan, Improving kNN multi-label classification in Prototype Selection scenarios using class proposals, Pattern Recognition 48 (5) (2015) 1608–1622.

[8] E. Leyva, A. González, R. Pérez, Three new instance selection methods based on local sets: A comparative study with several approaches from a bi-objective perspective, Pattern Recognition 48 (4) (2015) 1523–1537.

[9] E. Pekalska, R. P. Duin, P. Paclík, Prototype selection for dissimilarity-based classifiers, Pattern Recognition 39 (2) (2006) 189–208.

[10] P. Hernandez-Leal, J. A. Carrasco-Ochoa, J. F. Martínez-Trinidad, J. A. Olvera-Lopez, InstanceRank based on borders for instance selection, Pattern Recognition 46 (1) (2013) 365–375.

[11] E. Z. Borzeshi, M. Piccardi, K. Riesen, H. Bunke, Discriminative prototype selection methods for graph embedding, Pattern Recognition 46 (6) (2013) 1648–1657.

[12] D. R. Wilson, T. R. Martinez, Reduction techniques for instance-based learning algorithms, Machine learning 38 (3) (2000) 257–286.

[13] J. A. Olvera-López, J. A. Carrasco-Ochoa, J. F. Martínez-Trinidad, J. Kittler, A review of instance selection methods, Artificial Intelligence Review 34 (2) (2010) 133–143.

[14] S. Garcia, J. Derrac, J. R. Cano, F. Herrera, Prototype selection for nearest neighbor classification: Taxonomy and empirical study, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (3) (2012) 417–435.

[15] L. Kaufman, P. Rousseeuw, Clustering by means of medoids, in: Y. Dodge (Ed.), Statistical Data Analysis Based on the L1 Norm and Related Methods, North-Holland, 1987, pp. 405–416.

[16] E. Elhamifar, G. Sapiro, R. Vidal, Finding Exemplars from Pairwise Dissimilarities via Simultaneous Sparse Recovery, in: Proceedings of Advances in Neural Information Processing Systems, 2012, pp. 19–27.

[17] E. Elhamifar, G. Sapiro, S. S. Sastry, Dissimilarity-based sparse subset selection, IEEE Transactions on Pattern Analysis and Machine Intelligence, DOI: 10.1109/TPAMI.2015.2511748 (2016) .

[18] B. J. Frey, D. Dueck, Mixture modeling by affinity propagation, in: Proceedings of Advances in Neural Information Processing Systems, 2005, pp. 379–386.

[19] B. J. Frey, D. Dueck, Clustering by passing messages between data points, Science 315 (5814) (2007) 972–976.

[20] C. Boutsidis, M. W. Mahoney, P. Drineas, An improved approximation algorithm for the column subset selection problem, in: Proceedings of ACM-SIAM Symposium on Discrete Algorithms, 2009, pp. 968–977.

[21] T. F. Chan, Rank revealing QR factorizations, Lin. Algebra. Appl. 88 (1987) 67–82.

[22] J. A. Tropp, Column subset selection, matrix factorization, and eigenvalue optimization, in: Proceedings of ACM-SIAM Symposium on Discrete Algorithms, 2009, pp. 978–986.

[23] L. Balzano, R. Nowak, W. U. Bajwa, Column subset selection with missing data, in: NIPS workshop on Low-Rank Methods for Large-Scale Machine Learning, 2010.

[24] J. Bien, Y. Xu, M. W. Mahoney, CUR from a sparse optimization viewpoint, in: Proceedings of Advances in Neural Information Processing Systems, 2010, pp. 217–225.

[25] S. Wang, Z. Zhang, Improving CUR matrix decomposition and the Nyström approximation via adaptive sampling, Journal of Machine Learning Research 14 (1) (2013) 2729–2769.

[26] C. Yang, J. Shen, J. Peng, J. Fan, Image collection summarization via dictionary learning for sparse representation, Pattern Recognition 46 (3) (2013) 948–961.

[27] Y. Cong, J. Yuan, J. Liu, Abnormal event detection in crowded scenes using sparse representation, Pattern Recognition 46 (7) (2013) 1851 – 1864.

[28] F. Dornaika, I. K. Aldine, Decremental Sparse Modeling Representative Selection for Prototype Selection, Pattern Recognition 48 (11) (2015) 3714–3727.

[29] H. Liu, Y. Liu, Y. Yu, F. Sun, Diversified Key-Frame Selection Using $L_{2,1}$ Structured Optimization, IEEE Transactions on Industrial Informatics 10 (3) (2014) 1736–1745.

[30] C. L. Zitnick, P. Dollr, Edge Boxes: Locating Object Proposals from Edges, in: Proceedings of European Conference on Computer Vision, 2014, pp. 391–405.

[31] Z. Fang, Z. Cao, Y. Xiao, L. Zhu, J. Yuan, Adobe Boxes: Locating Object Proposals Using Object Adobes, IEEE Transactions on Image Processing 25 (9) (2016) 4116–4128.

[32] P. L. Combettes, J.-C. Pesquet, Proximal splitting methods in signal processing, in: Fixed-point algorithms for inverse problems in science and engineering, Springer, 2011, pp. 185–212.

[33] E. Elhamifar, R. Vidal, Sparse Subspace Clustering: Algorithm, Theory, and Applications, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (11) (2013) 2765–2781.

[34] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, Y. Ma, Robust Recovery of Subspace Structures by Low-Rank Representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (1) (2013) 171–184.

[35] U. von Luxburg, A tutorial on spectral clustering, Statistics and Computing 17 (4) (2007) 395–416.

[36] R. Tibshirani, Regression shrinkage and selection via the lasso, Journal of the Royal Statistical Society. Series B (Methodological) (1996) 267–288.

[37] S. Gao, I. W. Tsang, L.-T. Chia, P. Zhao, Local features are not lonely–laplacian sparse coding for image classification, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 3555–3561.

[38] K. Eriksson, D. Estep, C. Johnson, Applied Mathematics: Body and Soul, Volumn 2: Integrals and Geometry in $\mathbb{R}^n$, Springer Berlin Heidelberg, 2004.

[39] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM Journal on Imaging Sciences 2 (1) (2009) 183–202.

[40] F. Deutsch, Dykstra's Cyclic Projections Algorithm: The Rate of Convergence, in: S. P. Singh (Ed.), Approximation Theory, Wavelets and Applications, Springer Netherlands, 1995, pp. 87–94.

[41] B. Xin, Y. Kawahara, Y. Wang, W. Gao, Efficient Generalized Fused Lasso with its Application to the Diagnosis of Alzheimer's Disease, in: Proceedings of AAAI Conference on Artificial Intelligence, 2014, pp. 2163–2169.

[42] G. Gallo, M. D. Grigoriadis, R. E. Tarjan, A fast parametric maximum flow algorithm and applications, SIAM Journal on Computing 18 (1) (1989) 30–55.

[43] J. Zhou, J. Liu, V. A. Narayan, J. Ye, Modeling disease progression via fused sparse group lasso, in: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2012, pp. 1095–1103.

[44] M. Yuan, Y. Lin, Model selection and estimation in regression with grouped variables, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 68 (1) (2006) 49–67.

[45] J. Friedman, T. Hastie, H. Höfling, R. Tibshirani, Pathwise coordinate optimization, The Annals of Applied Statistics 1 (2) (2007) 302–332.

[46] M. Babenko, J. Derryberry, A. Goldberg, R. Tarjan, Y. Zhou, Experimental Evaluation of Parametric Max-Flow Algorithms, in: C. Demetrescu (Ed.), Experimental Algorithms, Springer Berlin Heidelberg, 2007, pp. 256–269.

[47] J. J. Hull, A database for handwritten text recognition research, IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (5) (1994) 550–554.

[48] A. Oliva, A. Torralba, Modeling the shape of the scene: A holistic representation of the spatial envelope, International Journal of Computer Vision 42 (3) (2001) 145–175.

[49] D. Coppersmith, S. J. Hong, J. R. Hosking, Partitioning nominal attributes in decision trees, Data Mining and Knowledge Discovery 3 (2) (1999) 197–217.

[50] L. Breiman, Random forests, Machine Learning 45 (1) (2001) 5–32.

[51] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Transactions on Information Theory 13 (1) (1967) 21–27.

[52] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (3) (1995) 273–297.

[53] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, C.-J. Lin, LIBLINEAR: A Library for Large Linear Classication, Journal of Machine Learning Research 9 (2008) 1871–1874.

[54] L. Fei-Fei, R. Fergus, P. Perona, Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories, Computer Vision and Image Understanding 106 (1) (2007) 59–70.

[55] T. Fawcett, An introduction to ROC analysis, Pattern Recognition Letters 27 (8) (2006) 861–874.

[56] G. Zhao, J. Yuan, J. Xu, Y. Wu, Discovering the thematic object in commercial videos, IEEE MultiMedia 18 (3) (2011) 56–65.

[57] A. Torralba, R. Fergus, W. T. Freeman, 80 million tiny images: A large data set for nonparametric object and scene recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (11) (2008) 1958–1970.

[58] F. Bach, Shaping level sets with submodular functions, in: Proceedings of Advances in Neural Information Processing Systems, 2011, pp. 10–18.

[59] F. Bach, Structured sparsity-inducing norms through submodular functions, in: Proceedings of Advances in Neural Information Processing Systems, 2010, pp. 118–126.

[60] K. Nagano, Y. Kawahara, Structured convex optimization under submodular constraints, in: Proceedings of Conference on Uncertainty in Artificial Intelligence, 2013, pp. 459–468.

[61] K. Nagano, K. Aihara, Equivalence of convex minimization problems over base polytopes, Japan Journal of Industrial and Applied Mathematics 29 (3) (2012) 519–534.