2010 IEEE International Conference on Data Mining Workshops

Batch Mode Sparse Active Learning

Lixin Shi
Institute for Theoretical Computer Science, Tsinghua University
Email: shilixinhere@gmail.com

Yuhang Zhao
Department of Computer Science and Technology, Tsinghua University
Email: zhaoyh630@gmail.com

***Abstract*—Sparse representation, due to its clear and powerful insight deep into the structure of data, has seen a recent surge of interest in the classification community. Based on this, a family of reliable classification methods have been proposed. On the other hand, obtaining sufficiently labeled training data has long been a challenging problem, thus considerable research has been done regarding active selection of instances to be labeled. In our work, we will present a novel unified framework, i.e. BMSAL(*Batch Mode Sparse Active Learning*). Based on the existing sparse family of classifiers, we define rigorously the corresponding BMSAL family and explore their shared properties, most importantly (approximate) submodularity. We focus on the feasibility and reliability of the BMSAL family: The first one inspires us to optimize the algorithms and conduct experiments comparing with state-of-the-art methods; for reliability, we give error-bounded algorithms, as well as detailed logical deductions and empirical tests for applying sparse in non-linear data sets.**

***Keywords*-batch mode sparse active learning; sparse classification; active learning; submodularity**

I. Introduction

Sparse representations from overcomplete dictionaries have been highlighted as one of the crucial principles in signal processing [8]. Under the *sparseland* model [29] where classes of signals are linear subspaces spanned by bases in this dictionary, there has been a recent surge of interest in exploring sparsity in signals. Much of the excitement comes from the discovery that sparse representation can be reduced to linear programming or second conic programming when the solution is sparse enough [10, 7, 5]. Following this thread, considerable research has been performed to explore frameworks to solve sparse coding [9, 30, 12], dictionary learning [31, 23, 1, 24], compressive sensing[18, 6], etc.

Recently, a new direction in sparse representation and pattern recognition is to combine them together, i.e. *Sparse Classification*. Developments show that sparsity is relevant to data sets in many applications, such as face recognition[32, 33], digit recognition[15], remote sensing[22] and speech recognition[14]. A key observation is that these data sets satisfy the *linear subspace* assumption[4] that samples from single class lie on a linear subspace. Sparse representations in nonlinear data sets are motivated by [33], which discussed the feasibility of sparse classification in general data sets.

However, machine learning algorithms, including sparse classification, often suffer from insufficiently labeled training data. A traditional way to overcome this problem is *active learning*[35], i.e., selecting optimal query in each iteration to be labeled. Conventional active learning methods are performed in single mode, by selecting a single instance, querying for its label and retraining in each iteration. Iterative retraining and long waiting time for the next query will be the efficiency concern; and there tends to be information overlap between selected instances in different iterations. To solve this problem, we will perform it in *batch mode*, i.e., selecting a batch of samples each iteration [17, 16].

In this paper, we propose a novel framework combining both the sparse representation and batch mode active learning, namely, *batch mode sparse active learning*(BMSAL). Although the BMSAL problem is related to the Dictionary Learning problem[20, 1], the existing methods(e.g. the K-SVD method) cannot be directly applied to the BMSAL problem, because the dictionary learning problem allows constructions and transformations(such as projection) of basis, but the BMSAL problem must select basis from a given data set otherwise the selected samples cannot be labeled.

A. Challenges and Contributions

The main challenges for batch mode sparse active learning are manifold:

- How to design well-defined objective functions for BMSAL: we formally define the "*correspondence*" relationships between BMSALs and sparse classifications, and propose a family of BMSAL methods based on it
- Reliability concern: Sparse representation based methods are proven to work well in data sets with the linear subspace guarantee. Although Yang et al. [33] demonstrates the feasibility of sparse in nonlinear data sets, they didn't provide experimental results. We will test the reliability with both analysis and extensive experiments.
- Time concern: Batch mode active learning problem itself is NP-hard, and sparse representation is time-consuming. We will perform a greedy algorithm as well as timing speedups to solve the time issue.

978-0-7695-4257-7/10 $26.00 © 2010 IEEE
DOI 10.1109/ICDMW.2010.175

To this end, three members in the BMSAL family are defined, corresponding to three closely-related sparse classification methods: NN(Nearest Neighbor [27]), NS(Nearest Subspace [3]) and L1(ℓ^1-Minimization [32]). We show that the BMSAL family naturally imply *submodularity*[25] or *approximate submodularity*[21], both of which are crucial properties to guarantee greedy algorithms. To further speed up the algorithm in large-scale data sets, specific optimization techniques are employed, such as OMP(Orthogonal Matching Pursuit [12]). We have applied this method to the document classification data set and have compared it with the results from using state-of-the-art batch mode active learning methods.

The rest of this paper is organized as follows: Section II gives a brief overview of the sparse classification family as the background of our work. Section III defines the BMSAL methods corresponding to the three sparse classification methods given in section II, proves their (approximate) submodularity and gives efficient algorithms respectively. Section IV conducts the experiments in both synthetic and real-world data sets. Section V gives concluding remarks.

II. Sparse Representation

In this section we will give a definite statement of sparse classification task and BMSAL model. There is a brief introduction to three methods in sparse classification, the NN, NS and L1 methods.

A. The Sparseland Model

The sparseland model [29] is the basis of both the sparse classification and BMSAL task. We will restate the sparseland model here in the perspective of supervised learning. We want to classify all the data instances (represented using d-dimension vectors) into classes $\{1, 2, \cdots, C\}$. For each class i, the distribution of data instances forms a *linear subspace* $\mathcal{L}_i$. These linear subspaces are disjoint, and each can be spanned using very few bases. The set of all these essential bases is called a *dictionary*, denoted using $\mathcal{D}$. Given a test sample $\boldsymbol{x}$, there exists $\boldsymbol{\alpha}$, such that $\mathcal{D}\boldsymbol{\alpha} = \boldsymbol{x}$. Particularly $\boldsymbol{\alpha}$ is very sparse for only elements corresponding to bases in the subspace of $\boldsymbol{x}$'s class could be nonzero. We define α as the *sparse representation* of $\boldsymbol{x}$.

Sparse classification is based on this representation. Suppose the samples in the dictionary $\mathcal{D}$ are given and labeled with $y_{\mathcal{D}}$. Now for each instance $\boldsymbol{x}$ in the unlabeled data set $\mathcal{T}$, we can find the sparse representation $\boldsymbol{\alpha}$. Ideally the non-zero entries in $\boldsymbol{\alpha}$ determine which class the instance is in, but practically we must find ways to measure the "sparsity" of entries for each class. Following this thread, we can define different score function $f(\mathcal{D}, y_{\mathcal{D}}, \boldsymbol{x}, c)$ where c denotes the c-th class, and the final classification is

$$y_{\boldsymbol{x}} = \underset{c\in\{1,2,\cdots,C\}}{\arg\min} f(\mathcal{D}, y_{\mathcal{D}}, \boldsymbol{x}, c)$$

Note in section we will give three different definitions of f.

Note sparseland model requires the linear subspace assumption over the structure of data sets. It is proven that data sets in face recognition[32, 33], digit recognition[15], remote sensing[22] and so on approximately satisfy this assumption but others might not. In this paper we strictly derive the BMSAL family from the sparseland model but extend the application of them to non-linear data sets, such as document classification.

Given size k, a BMSAL task aims to find an optimized dictionary $\mathcal{D}$ to be queried. We also need a score function $g(\mathcal{D})$ to measure the informativeness of selected set. Specifically the selected set is

$$\mathcal{D}^* = \underset{\mathcal{D}\subseteq\mathcal{S},|\mathcal{D}|=k}{\arg\max}\ g(\mathcal{D})$$

Table I gives a summary of variables used in our model:

Symbol	Explanation
C	The total number of classes
$\mathcal{D}$	The dictionary of sparse representation
$\mathcal{T}$	The test set, i.e. set of unlabeled data instances
$\mathcal{S}$	The set of all samples, $\mathcal{S} = \mathcal{D} \cup \mathcal{T}$
$\mathcal{L}_i$	The subspace spanned by data instances in class i
$y_{\mathcal{D}}$	The label of data instances in the dictionary $\mathcal{D}$
k	The number of instances to be selected, i.e., the size of $\mathcal{D}$
f	The score function of sparse classification tasks
g	The score function of BMSAL tasks

Table I
List of Variables

B. Sparse Classification Family

In this section we will introduce three methods of sparse classification(namely L1, NS and NN), by defining score function f in three ways.

L1 Method : Mostly there are infinite number of sparse representations given a test data instance $\boldsymbol{x}$ and dictionary $\mathcal{D}$. We seek for the sparsest solution using equation (1)[11], observing that ℓ^0-norm(denoted by $||\cdot||_0$) represents the number of non-zero entries:

$$\min ||\boldsymbol{\alpha}||_0, \text{ s.t. } \boldsymbol{x} = \mathcal{D}\boldsymbol{\alpha} \tag{1}$$

Unfortunately, the problem is NP-complete and even hard to approximate, as proven by theoretical results [2]. Alternatively we minimize the ℓ^1 norm, and this gives a reliable result when the representation is sparse enough [32, 10]. This ℓ^1-norm could be solved efficiently by recent research[9]. Define $\delta_c^{(\mathcal{D})}(\boldsymbol{\alpha})$ as the characteristic vector, such that the only non-zero entries are entries in $\boldsymbol{\alpha}$ that correspond with the indices of bases in class c. The score function is the error generated using $\delta_c^{(\mathcal{D})}(\boldsymbol{\alpha})$ as the sparse representation:

$$f_{\text{L1}}(\mathcal{D}, y_{\mathcal{D}}, \boldsymbol{x}, c) = ||\mathcal{D}\delta_c^{(\mathcal{D})}(\boldsymbol{\alpha}^*) - \boldsymbol{x}||_p \tag{2}$$

where $\boldsymbol{\alpha}^* = \underset{\boldsymbol{\alpha}}{\arg\min}\{||\boldsymbol{\alpha}||_1 : \boldsymbol{x} = \mathcal{D}\boldsymbol{\alpha}\}$ and p may be 1 or 2.

NS Method : Nearest Subspace (NS, [3]) classifier finds the class i whose subspace $\mathcal{L}_i$ has the smallest projection distance with data instance $\boldsymbol{x}$, i.e.

$$f_{\text{NS}}(\mathcal{D}, y_{\mathcal{D}}, \boldsymbol{x}, c) = \min_{\boldsymbol{\beta}} ||\mathcal{D}_c\boldsymbol{\beta} - \boldsymbol{x}||_2^2 \quad (3)$$

It can be viewed as a first-step relaxation of the L1 method, for both of the classifiers determine the same class with high probability if the data set is close to ideal condition. Under such circumstances, the sparse representation $\boldsymbol{\alpha}$ of $\boldsymbol{x}$ has only nonzero entries in class c, hence $\mathcal{D}\delta_c^{(\mathcal{D})}(\boldsymbol{\alpha}) = \boldsymbol{x}$, i.e. the projection distance of $\boldsymbol{x}$ to $\mathcal{L}_i$ is zero.

NN Method : Nearest Neighbor (NN, [27]) method classifies text data instance $\boldsymbol{x}$ into the class of the nearest labeled data instance, i.e.

$$f_{\text{NN}}(\mathcal{D}, y_{\mathcal{D}}, \boldsymbol{x}, c) = \min_{\boldsymbol{b}\in\mathcal{D}_c} ||\boldsymbol{x} - \boldsymbol{b}||_2^2 \quad (4)$$

Using similar verification method, we can show that NN and L1 classifier are guaranteed to the same output is when sparse representation α is almost 1-sparse, i.e. approximately with single nonzero entry.

NN, NS and L1 methods need increasing computational power while requiring decreasingly strong assumptions at the same time. Roughly speaking, NN assumes the sparse representation is 1-sparse, NS assumes exactly the sparse-land model, i.e., only the entry corresponding with its class can be nonzero. By contrast, L1 is more reliable for different data sets, even nonlinear data sets.

III. The Proposed Approach

A. The BMSAL Family

How to design criteria in objective function based batch mode active learning is a central problem. We want to define BMSAL score functions based on sparse classification score functions by defining *correspondence* relationships between g and f. To make the representation more sparse, we want to select the dictionary that best minimizes score function f for any possible labeling, strictly defined as:

$$g_{\text{inst}}(\mathcal{D}) = \Lambda - \max_{y_{\mathcal{D}}} \left\{ \sum_{\boldsymbol{x}\in\mathcal{S}} \min_{c\in\{1,\cdots,C\}} f_{\text{inst}}(\mathcal{D}, y_{\mathcal{D}}, \boldsymbol{x}, c) \right\} \quad (5)$$

where $\mathcal{S} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n\}$ is the set of samples, and Λ is a constant to ensure that $g(\mathcal{D}) \geq 0$ for all $\mathcal{D} \subseteq S$.

To demonstrate how the correspondence works, we first derive $g_{\text{NN}}(\mathcal{D})$ corresponding to the NN method. No matter what the labeling will be, the minimum score function of $\boldsymbol{x}$ will be

$$\min_{c\in\{1,\cdots,C\}} \min_{\boldsymbol{b}\in\mathcal{D}_c} ||\boldsymbol{x} - \boldsymbol{b}||_2^2 = \min_{\boldsymbol{b}\in\mathcal{D}} ||\boldsymbol{x} - \boldsymbol{b}||_2^2$$

We add all these minimum values together over all $\boldsymbol{x} \in \mathcal{S}$ to measure the dictionary, i.e.

$$g_{\text{NN}}(\mathcal{D}) = \Lambda_{\text{NN}} - \sum_{\boldsymbol{x}\in\mathcal{S}} \min_{\boldsymbol{b}\in\mathcal{D}} ||\boldsymbol{x} - \boldsymbol{b}||_2^2$$

Theorem 1 defines the corresponding BMSAL tasks of NN, NS and L1 methods. For the purpose of convenience, we will call the three instances **BMSAL-NN**, **BMSAL-NS** and **BMSAL-L1** respectively in our context.

Theorem 1. *The corresponding BMSAL instances of NN, NS and L1 are given as follows:*

$$g_{NN}(\mathcal{D}) = \Lambda_{NN} - \sum_{\boldsymbol{x}\in\mathcal{S}} \min_{\boldsymbol{b}\in\mathcal{D}} ||\boldsymbol{x} - \boldsymbol{b}||_2^2 \quad (6)$$

$$g_{NS}(\mathcal{D}) = \Lambda_{NS} - \sum_{\boldsymbol{x}\in\mathcal{S}} ||\boldsymbol{x} - \mathcal{D}\mathcal{D}^\dagger\boldsymbol{x}||_2^2 \quad (7)$$

$$g_{LI}(\mathcal{D}) = \Lambda_{LI} - \sum_{\boldsymbol{x}\in\mathcal{S}} \min\{||\boldsymbol{\alpha}||_1 : \mathcal{D}\boldsymbol{\alpha} = \boldsymbol{x}\} \quad (8)$$

where † denotes pseudoinverse, and we assume $p = 1$ in equation (2).

Proof: For equation (7), we can see that

$$\min_{c\in\{1,\cdots,C\}} f_{\text{NS}} \min_{\boldsymbol{\beta}} ||\mathcal{D}_c\boldsymbol{\beta} - \boldsymbol{x}||_2^2$$

is the square of distance of vector $\boldsymbol{x}$ to subspace $\mathcal{D}$, it can be written as $||\boldsymbol{x} - \mathcal{D}\mathcal{D}^\dagger\boldsymbol{x}||_2^2$ using the concept of pseudoinverse.

For the L1 classifier when $p = 1$, we have

$$||\mathcal{D}\delta_c^{(\mathcal{D})}(\boldsymbol{\alpha})||_1 \geq ||\boldsymbol{\alpha}||_1$$

On the other hand, if all the labels of $\mathcal{D}$ are from the same class c, $\delta_c^{(\mathcal{D})}(\boldsymbol{\alpha}) = \boldsymbol{\alpha}$. Therefore, in this case we have found the specific $y_{\mathcal{D}}$ that maximizes the minimum error, and the error is exactly the solution to the original ℓ^1-minimization function, i.e., equation (8) holds. ■

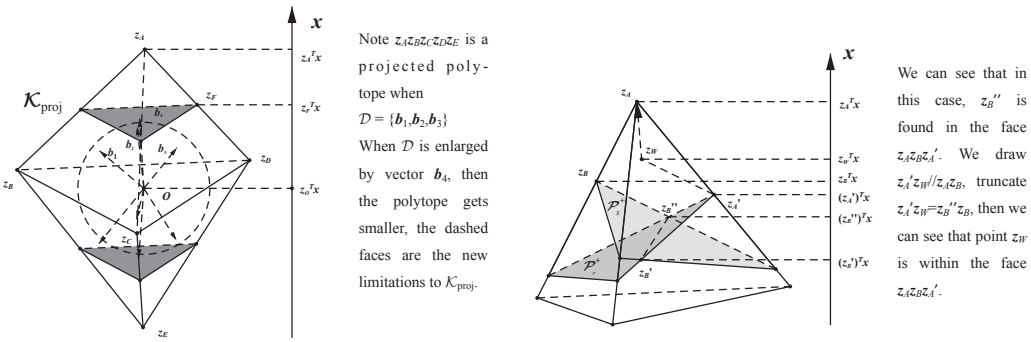
Besides the theoretical proof, we will give some intuitive explanations to the three members in BMSAL family. BMSAL-NN selects the set of data instances with the smallest summation of distances to the unselected instances; BMSAL-NS selects one set spanning a subspace with the smallest summation of projection distances with unselected instances; BMSAL-NN selects the set with smallest summation of ℓ^1-norm sparse representations.

In summary, we have deduced three corresponding BMSAL instances under a unified definition based on minimum empirical error criterion. On the other hand, when studied respectively, the objective functions have its own sense to measure the informativeness.

B. Algorithm with Reliable Approximation Bound

(Approximate) submodularity is an important property of a set function, for it guarantees the reliability of greedy algorithm [25, 21], we will explore the submodularity property in this section to give a reliable algorithm of the three BMSAL methods. A set function $g(\mathcal{D})$ is *approximately submodular* with $\epsilon \geq 0$ if for $\forall\mathcal{D}_1 \subseteq \mathcal{D}_2 \subseteq \mathcal{S}$ and $\boldsymbol{x} \in \mathcal{S} - \mathcal{D}_2$, we have

$$g(\mathcal{D}_1 \cup \{\boldsymbol{x}\}) - g(\mathcal{D}_1) \geq g(\mathcal{D}_2 \cup \{\boldsymbol{x}\}) - g(\mathcal{D}_2) - \epsilon$$



(a) A demonstration of $\mathcal{K}_{\text{proj}}$

(b) The proof when $\dim(\mathcal{D}) = 3$

Figure 1. Demonstration of polytope $\mathcal{K}$

Moreover, when $\epsilon = 0$, it's called *submodular*. There has been a famous approximation rate maximizing the submodular problem shown in lemma 1:

Lemma 1. *[34, 25]If set function $g(\mathcal{D})$ is nonnegative, monotonic and approximately submodular with ϵ, then greedy algorithm(algorithm 1) has the following guarantee:*

$$g(\mathcal{D}) \geq (1 - \frac{1}{e}) g(\mathcal{D}^*) - k\epsilon$$

where $\mathcal{D}$ is the output of algorithm 1 and $\mathcal{D}^*$ is the optimal solution subject to $|\mathcal{D}| = k$. Moreover, if $\epsilon = 0$, the approximation rate is exactly $(1 - \frac{1}{e})$.

It's not hard to check that BMSAL-NN, BMSAL-NS and BMSAL-L1 are all nonnegative and monotonically increasing functions. Here we assume $\boldsymbol{x}^T \boldsymbol{x} = 1$ and $\boldsymbol{x}^T \boldsymbol{x}' \geq 0$, $\forall \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{S}$. This assumption could be achieved by normalizing and limiting elements to be non-negative. The following lemmas 2, 3, 4 show that BMSAL-NN, BMSAL-NS and BMSAL-L1 are all submodular:

Lemma 2. *BMSAL-NN problem is submodular.*

Proof: We only have to prove that if $v_{\boldsymbol{x}}(\mathcal{D}) = \min_{\boldsymbol{b} \in \mathcal{D}} ||\boldsymbol{x} - \boldsymbol{b}||_2^2$, and $\forall \boldsymbol{x} \in \mathcal{S}$, $\forall \mathcal{D} \subseteq \mathcal{S}$, $\forall \boldsymbol{b}_1, \boldsymbol{b}_2 \in \mathcal{S} - \mathcal{D}$ such that $\boldsymbol{b}_1 \neq \boldsymbol{b}_2$,

$$v_{\boldsymbol{x}}(\mathcal{D}) - v_{\boldsymbol{x}}(\mathcal{D} \cup \{\boldsymbol{b}_1\}) \geq v_{\boldsymbol{x}}(\mathcal{D} \cup \{\boldsymbol{b}_2\}) - v_{\boldsymbol{x}}(\mathcal{D} \cup \{\boldsymbol{b}_1, \boldsymbol{b}_2\})$$

If the right side is zero, then the inequity holds trivially. Otherwise, when the left side is not zero,

$$\begin{aligned} \text{left side} &= \min_{\boldsymbol{b} \in \mathcal{D}} ||\boldsymbol{b} - \boldsymbol{x}||_2^2 - ||\boldsymbol{b_1} - \boldsymbol{x}||_2^2 \\ &\geq \min_{\boldsymbol{b} \in \mathcal{D} \cup \{\boldsymbol{b}_2\}} ||\boldsymbol{b} - \boldsymbol{x}||_2^2 - ||\boldsymbol{b_1} - \boldsymbol{x}||_2^2 \geq \text{right side} \end{aligned}$$

Otherwise, it means that $\exists \boldsymbol{b} \in \mathcal{D}$, $||\boldsymbol{x} - \boldsymbol{b}||_2^2 \leq ||\boldsymbol{x} - \boldsymbol{b}_1||_2^2$, hence the right side is also zero. ■

The proof that BMSAL-NS is approximately submodular has been covered in the work [20].

Lemma 3. *BMSAL-NS problem is $4k\mu$-approximately submodular, where μ is the incoherency of the system:*

$$\mu = \max_{\boldsymbol{b}_i, \boldsymbol{b}_j \in \mathcal{S}, \boldsymbol{b}_i \neq \boldsymbol{b}_j} \boldsymbol{b}_i^T \boldsymbol{b}_j$$

and k is the number of samples to be selected in the BMSAL task.

For BMSAL-L1 problem, the proof that it is submodular is quite lengthy, so we only provide a sketch of proof. In the proof we present a nice geometry interpretation of ℓ^1 minimization problem, this interpretation is also the basis of algorithm 3.

Lemma 4. *BMSAL-L1 problem is submodular.*

Proof: First let's consider the ℓ^1 problem

$$\min ||\boldsymbol{\alpha}||_1, \text{ s.t. } \mathcal{D}\boldsymbol{\alpha} = \boldsymbol{x}$$

We can rewrite it using second conic programming concepts:

$$\min \boldsymbol{\gamma}, \text{ s.t. } \exists \boldsymbol{\alpha}, \mathcal{D}\boldsymbol{\alpha} = \boldsymbol{x}, ||\boldsymbol{\alpha}||_1 \leq \boldsymbol{\gamma} \tag{9}$$

Since the dual cone of $||\boldsymbol{x}||_1 \leq \boldsymbol{\gamma}$ is $||\boldsymbol{x}'||_\infty \leq \boldsymbol{\gamma}'$, we can write the dual problem of (9):

$$\max \boldsymbol{x}^T \boldsymbol{z}, \text{ s.t. } ||\mathcal{D}^T \boldsymbol{z}||_\infty \leq 1 \tag{10}$$

Note $\mathcal{D}$ is a set of bases, so primal problem (9) has a solution; since obviously the dual problem (10) has a strictly feasible solution $\mathbf{0}$, by the strong duality, programming (9) and (10) has exactly the same optimal value. Let the value be $v_{\boldsymbol{x}}(\mathcal{D})$, we can see that $g_{\text{L1}}(\mathcal{D}) = \Lambda_{\text{L1}} - \sum_{\boldsymbol{x} \in \mathcal{D}} v_{\boldsymbol{x}}(\mathcal{D})$ is submodular if $\forall \boldsymbol{x} \in \mathcal{D}, \forall \mathcal{D} \subseteq \mathcal{S}, \forall \boldsymbol{b}_r, \boldsymbol{b}_s \in \mathcal{S} - \mathcal{D}$ such that $\boldsymbol{b}_r \neq \boldsymbol{b}_s$,

$$v_{\boldsymbol{x}}(\mathcal{D}) - v_{\boldsymbol{x}}(\mathcal{D} \cup \{\boldsymbol{b}_r\}) \geq v_{\boldsymbol{x}}(\mathcal{D} \cup \{\boldsymbol{b}_s\}) - v_{\boldsymbol{x}}(\mathcal{D} \cup \{\boldsymbol{b}_r, \boldsymbol{b}_s\}) \tag{11}$$

To prove (11), we introduce a nice geometry interpretation by (10). Given $\mathcal{D}$ with column vectors $\boldsymbol{b}_1, \cdots, \boldsymbol{b}_k \in \mathbb{R}^d$, we can see the solution $||\mathcal{D}^T \boldsymbol{z}||_\infty \leq 1$ forms a polytope $\mathcal{K}$ satisfying:

- For each $\boldsymbol{b}_i$ there are two hyperplanes $\dot{\mathcal{P}}_i^+$ and $\dot{\mathcal{P}}_i^-$, orthogonal to vector $\boldsymbol{b}_i$ and tangent to the hypersphere with radius one and its center at the origin. Let $\mathcal{P}_i$ denotes the space between these two planes, i.e., $\mathcal{P}_i = \{\boldsymbol{z} : \boldsymbol{b}^T\boldsymbol{z} \leq 1\}$ Then

$$\mathcal{K} = \bigcap_{\boldsymbol{b}_i \in \mathcal{D}} \mathcal{P}_i \tag{12}$$

- The solution is the projection of polytope $\mathcal{K}$ over vector $\boldsymbol{x}$.
- $\mathcal{D}_1 \subseteq \mathcal{D}_2 \Leftrightarrow \mathcal{K}_2 \subseteq \mathcal{K}_1$, because more bases in $\mathcal{D}$ mean more limitation hyperplanes of $\mathcal{K}$ due to equation (12).

Following this interpretation, the next step is to project $\mathcal{K}$ into a lower dimension space. Note we have known that columns in $\mathcal{D}$ span the whole space $\mathcal{U}$ of samples, if we project $\mathcal{K}$ through vectors in $\mathcal{U}^\perp$ onto $\mathcal{U}$, we will get a well-closed $\dim(\mathcal{U})$-polytope $\mathcal{K}_{\text{proj}}$, for the completeness of $\mathcal{D}$ in spanning $\mathcal{U}$ indicates that every vertex of $\mathcal{K}_{\text{proj}}$ is a intersection of at least $\dim(\mathcal{U})$ hyperplanes. Figure 1(a) is a demonstration of $\mathcal{K}$.

For equation (11), let $\mathcal{K}_{\text{proj}}$ denote the polytope constructed by $\mathcal{D}$ and $\mathcal{K}'_{\text{proj}}$ denote the polytype constructed by $\mathcal{D} \cup \{\boldsymbol{b}_s\}$. We can assume $\mathcal{K}_{\text{proj}} \neq \mathcal{K}'_{\text{proj}}$, otherwise (11) holds trivially. Note the optimal solution must be reached in a vertex, let that optimal vertex of $\mathcal{K}_{\text{proj}}$ be $\boldsymbol{v}_A$ and that of $\mathcal{K}_{\text{proj}}$ be $\boldsymbol{v}_B$. Then consider hyperplane $\dot{\mathcal{P}}_r$, which intersects with existing hyperplanes and restricts $\mathcal{K}_{\text{proj}}$ and $\mathcal{K}'_{\text{proj}}$ into $\mathcal{K}_{\text{new}}$ and $\mathcal{K}'_{\text{new}}$ respectively; let the optimal vertices of the two polytopes be $\boldsymbol{z}'_A$ and $\boldsymbol{z}'_B$.

Before going on, we assume $\boldsymbol{z}_A^T\boldsymbol{x} > \boldsymbol{z}_B^T\boldsymbol{x}$, $\boldsymbol{z}_A^T\boldsymbol{x} > \boldsymbol{z}_A'^T\boldsymbol{x}$ and $\boldsymbol{z}_B^T\boldsymbol{x} > \boldsymbol{z}_B'^T\boldsymbol{x}$, because if any of the inequities is violated, inequity (11) holds trivially. The following method is performed to find $\boldsymbol{z}''_B$:

$\boldsymbol{z}_A$, $\boldsymbol{z}'_A$ and $\boldsymbol{z}_B$ determine a unique 2-dimensional face of $\mathcal{K}_{\text{proj}}$, then $\boldsymbol{z}''_B$ is the intersection of this face, hyperplane $\dot{\mathcal{P}}_r^+$ and hyperplane $\dot{\mathcal{P}}_s^+$; we can easily check that $\boldsymbol{z}''_B$ must exist. Let $\boldsymbol{z}_\delta = \boldsymbol{z}_A - \boldsymbol{z}'_A + \boldsymbol{z}''_B$, since $\boldsymbol{z}_A, \boldsymbol{z}'_A, \boldsymbol{z}''_B$ and $\boldsymbol{z}_B$ are in the same 2-dimensional face of $\mathcal{K}_{\text{proj}}$, we can easily argue that point $\boldsymbol{z}_W = \boldsymbol{z}_A - \boldsymbol{z}_\delta$ is within that face as well. Figure 1(b) shows a simple demonstration when $\dim(\mathcal{U}) = 3$. This leads to the following equation:

$$\boldsymbol{z}_\delta = \sum_{i=1}^{k} \lambda_i \boldsymbol{b}_i, \text{ where } \lambda_i \geq 0 \tag{13}$$

From equation (13) as well as the observation that $\boldsymbol{z}_B'^T\boldsymbol{x} \geq \boldsymbol{z}_B''^T\boldsymbol{x}$, we have

$$\begin{aligned} & (\boldsymbol{z}_A - \boldsymbol{z}'_A + \boldsymbol{z}'_B - \boldsymbol{z}_B)^T\boldsymbol{x} \\ = \; & (\boldsymbol{z}_\delta + \boldsymbol{z}'_B - \boldsymbol{z}''_B)^T\boldsymbol{x} \\ = \; & \left(\sum_{i=1}^{k} \lambda_i \boldsymbol{b}_i^T\boldsymbol{x}\right) + (\boldsymbol{z}_B'^T\boldsymbol{x} - \boldsymbol{z}_B''^T\boldsymbol{x}) \\ \geq \; & 0 \end{aligned}$$

Therefore, equation (11) holds, BMSAL-L1 problem is submodular. ■

According to the (approximate) submodularity, greedy algorithm 1 is a reliable algorithm to implement any of the three methods. Theorem 2 shows the reliable approximation rate, as a corollary of lemmas 1, 2, 3 and 4:

Theorem 2. *Regarding approximation rates of BMSAL-NN, BMSAL-NS and BMSAL-L1 instances, we have:*

- $g_{NN}(\mathcal{D}) \geq (1 - \frac{1}{e})g_{NN}(\mathcal{D}^*)$, $g_{NN}(\mathcal{D}) \geq (1 - \frac{1}{e})g_{NN}(\mathcal{D}^*)$
- $g_{NS}(\mathcal{D}) \geq (1 - \frac{1}{e})g_{NS}(\mathcal{D}^*) - 6k^2\mu$, *where* μ *is defined in lemma 3. [20]*

Algorithm 1 Naive Greedy Algorithm Solving $g(\mathcal{D})$

Input: $\mathcal{S}, g, k$
1: **initialize:** $\mathcal{D} \leftarrow \emptyset$
2: **for** $i = 1$ to k **do**
3: $\quad \mathcal{D} \leftarrow \mathcal{D} \cup \{\boldsymbol{x}^*\}$
$\quad$ where $\boldsymbol{x}^* = \arg\max_{\boldsymbol{x} \in \mathcal{S} - \mathcal{D}} g(\mathcal{D} \cup \{\boldsymbol{x}\})$
4: **end for**
5: **return** $\mathcal{D}$

C. Speedups

In this section, we want to further speedup the BMSAL-NS and BMSAL-L1 methods based on algorithm 1. The problem comes from exhaustive search in line 3 in algorithm 1. We will give optimizations to avoid repeatedly computing pseudoinverse or ℓ^1-minimization.

BMSAL-NS Speedup : Suppose currently we have selected $\mathcal{D}_i = \{\boldsymbol{b}_1, \cdots, \boldsymbol{b}_i\}$, and we want to select $\boldsymbol{b}_{i+1}$. Then we have

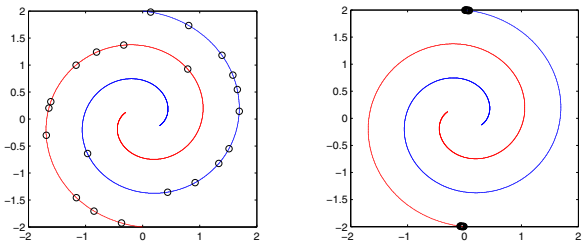
$$\begin{aligned} & \sum_{\boldsymbol{x} \in \mathcal{S}} ||\boldsymbol{x} - \mathcal{D}_{i+1}\mathcal{D}_{i+1}^\dagger \boldsymbol{x}||_2^2 \\ = \; & \sum_{\boldsymbol{x} \in \mathcal{S}} \left(||\boldsymbol{x} - \mathcal{D}_i\mathcal{D}_i^\dagger \boldsymbol{x}||_2^2 - (\boldsymbol{x} - \mathcal{D}_i\mathcal{D}_i^\dagger \boldsymbol{x})^T \boldsymbol{b}_{i+1} \right) \end{aligned}$$

This leads directly to algorithm 2, a feasible algorithm that solves BMSAL-NS. Algorithm 2 is a modification of the original OMP(Orthogonal Matching Pursuit) algorithm [30] to be applied to BMSAL tasks.

Algorithm 2 OMP Algorithm Solving BMSAL-NS

Input: $\mathcal{S}, g, k$
1: **initialize:** $\boldsymbol{r}_i \leftarrow \boldsymbol{x}_i, \forall 1 \leq i \leq n$; $\mathcal{D} \leftarrow \emptyset$
2: **for** $i = 1$ to k **do**
3: $\quad \mathcal{D} \leftarrow \mathcal{D} \cup \left\{ \arg\max_{\boldsymbol{b} \in \mathcal{S} - \mathcal{D}} \sum_{j=1}^{n} \boldsymbol{r}_j^T \boldsymbol{b} \right\}$
4: $\quad \boldsymbol{r}_j \leftarrow \boldsymbol{r}_j - \mathcal{D}\mathcal{D}^\dagger \boldsymbol{r}_j, \forall 1 \leq j \leq n$
5: **end for**
6: **return** $\mathcal{D}$

BMSAL-L1 Speedup : One possible way of speeding up is recording the optimal solution information for $\boldsymbol{x}$. In the proof of lemma 4 a nice geographical interpretation is



(a) BMSAL-NN (b) BMSAL-NS

Figure 2. Results of BMSAL in Two Spirals Data Set. The red and blue points denote two classes, and the circles are selected points.

introduced, and we use this interpretation to explain the algorithm: each time we find an optimal solution, it must be a vertex $\boldsymbol{z}_A$ of the polytope, i.e., an intersection of at least $\dim(\mathcal{U})$ faces. When a new basis $\boldsymbol{b}_i$ is introduced, it changes the optimal solution if and only if $\boldsymbol{z}_A \notin \mathcal{P}_i$, i.e.,

$$|\boldsymbol{b}_i^T \boldsymbol{z}_A| > 1 \tag{14}$$

Using this condition we can eliminate many unnecessary searches, especially when $\mathcal{D}$ gets larger, very small part of the candidates satisfy this condition. We can also record the information that which conditions are tight, i.e., the set F of faces intersects at the optimal vertex; if a new hyperplane $\dot{\mathcal{P}}_i^+$ satisfies (14), then we can calculate the $|F|$ points generated by $\dot{\mathcal{P}}_i^+$ intersecting faces in F, the simple observation is that the optimal solution must be one of them. As a summary, algorithm 3 shows the sketch of the algorithm.

Algorithm 3 Elimination Algorithm Solving BMSAL-L1

Input: $\mathcal{S}, g, k$
initialize: $\boldsymbol{z}_{Ai} \leftarrow \infty$, $F_i \leftarrow \emptyset$, $\forall 1 \le i \le n$
for $i = 1$ to k **do**
  for $\boldsymbol{b} \in \mathcal{S} - \mathcal{D}$ **do**
    initialize: $result[\boldsymbol{b}] \leftarrow 0$
    test each $\boldsymbol{x}_i$
      if $|\boldsymbol{b}^T \boldsymbol{z}_{Ai}| \le 1$ then elimination
      otherwise Calculate $\boldsymbol{z}_{Ai}$ and F_i,
        $result[\boldsymbol{b}] \leftarrow result[\boldsymbol{b}] + \boldsymbol{z}_{Ai}^T \boldsymbol{x}_i$
  end for
  $\mathcal{D} \leftarrow \mathcal{D} \cup \{\arg\max_{\boldsymbol{b}} result[\boldsymbol{b}]\}$
end for

IV. Experiments

In our experiment part, we will explore our model's performance, comparing with state-of-the-art batch mode active learning methods. At the same time, there are discussions about some practical concerns in our model.

The main goals of our experiments are twofold: first, it is to prove that the proposed BSMAL methods work well for synthetic and real-world data sets and they outperform some state-of-art methods; second, we want to explore the reliability of sparse methods, and experiment results show that BMSAL-L1 works well in data sets with non linear subspaces, or even highly nonlinear and low dimensional data sets.

A. Non-linear Synthetic Data Sets

To test the performance of the sparse-related methods in non-linear data set, we use a classical non-linear data set: the Two Spirals data set [28]. It has been shown that the two spirals data set is a challenge for linear classification methods, since the data set itself is highly non-linear. Figure 2 demonstrates the data set on a 2-D plane, the blue and red points denote the two classes respectively.

Result : Note the challenge in the Two Spirals data set for sparse-related methods is twofold: first, it's highly non-linear; second, it's in 2-D plane, but sparse assumes high-dimensional data. The result shows that BMSAL-NN and BMSAL-NS fails the task, all the results are of accuracy about 50%+. Note BMSAL-NN fails because it assumes the 1-sparse, i.e., it assumes that two near points tend to be in the same class, but for the Two Spirals it's not the case: BMSAL-NN tends to select samples in the outer ring (Figure 2(a)), but actually the inner rings are hard to classify since it's crowded with points from different classes. For BMSAL-NS, figure 2(b) shows that the selected samples are in an ill condition that they are crowded at the outer ends of the spirals. That is because the Two Spirals set is 2-dimensional, the distance to a subspace is 0 as long as there are two vectors in the subspace, hence the conclusion is that BMSAL-NS is sensitively dependent on high dimension. On the other hand, BMSAL-L1 works very well, the accuracy of 30 runs is $(98.21 \pm 1.40)\%$. Note in section III-B we have mentioned that the vectors should be normalized, so we have done a small trick before applying the algorithm, that we move the origin to a far point, e.g., (100, 100). This is roughly the *normalization* process. For linear subspace model, normalization will not lose information since only the direction of vectors matter; for non-linear model, that is crucial to guarantee a good result.

Explanation : Here we extend the explain in [33]: in non-linear system, sparse representation performs a *piecewise approximation*. In figure 3, we can see that since all of the features are (roughly) normalized, the most competing

points of two bases are the linear segment between them, so sparse achieves piecewise approximation. The red bar in figure 3 shows the error in $||\cdot||_2$ measurement. Note the piecewise argument can not only be applied to the curve structure, but also to any continuous structure, where we may use three points or more as a piece. The "normalization" is essential to the piecewise argument, otherwise the points nearer to the origin should be more competent(recall that in ℓ^1, the value of the coefficient counters, the point nearer to origin has smaller coefficients). The training set is anther crucial factor. If the training set is selected poorly, then the piecewise approximation will be erroneous. It should be distributed regularly, and denser in less smooth areas. Actually the samples BSMAL-L1 selected have these good features, hence we suggest using BMSPA-L1 and L1 at the same time.

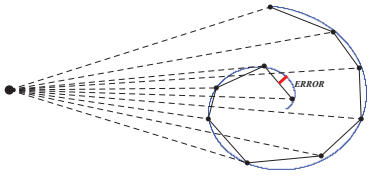


Figure 3. A Piecewise Approximation to Nonlinear Structure

B. Non-linear Real-World Data Sets

Why Document Classification Document classification doesn't have linear subspace assumption, but we want to explain here that it could work both empirically and logically, using a similar argument of [19]. First, the feature vectors are high dimensional; typically the word indices have at least thousands of entries. Then, we may find it useful to write document features like this:

$$\boldsymbol{x}_i = \boldsymbol{f}_c + \boldsymbol{e}$$

where c is the class label. It means that the feature vector of $\boldsymbol{x}$ is the shared feature vector $\boldsymbol{f}_c$ plus $\boldsymbol{e}$. And typically $\boldsymbol{e}$ should contain the specific words for a given document. For example, if the class labels are topics, such as Data Mining topic, perhaps "data", "mining" are the words from $\boldsymbol{f}_c$, while words like "sparse" might surge up in a specific document, but have no appearance in another data mining paper. Hence "sparse" should be an entry in the error item. Note a document in this class shouldn't have too many such words, so $\boldsymbol{e}$ is *sparse*. That observation inspires us to use $p = 1$ in the L1 method and expect it to have a good performance.

Setups and Results We test our proposed method in the following widely-used text categorization data sets:

UCI 20NewsGroups[13] is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. We construct different binary classification models, the first one stands for easy task, and the second one stands for hard:

- comp.sys.mac.hardware (963) vs comp.windows.x (988): totally 3338 words.
- talk.religion.misc (628) vs alt.atheism (799): totally 3360 words.

WebKB[26] contains web pages from four computer science departments, categorized into five topics: course, faculty, student, project, and staff. The webKB data set contains 877 data points and 1703 unique words.

We use two most popular batch mode active learning methods as our baseline:

SVM active learning method is a batch mode active learning method. Traditional SVM batch mode active learning samples the instances closest to the decision boundary for labeling. We use a modified version by [17], which incorporates the diversity information as well.

Fisher information matrix has been used by many batch mode active learning methods. We choose the method proposed in [16].

The results of the tests are shown in figure 4. Comparing with other batch mode active learning and BMSAL methods, BMSAL-L1 has an overall advantage. Other methods are not so stable between different data sets, for example, we may find that BMSAL-NN works quite well in figure 4(b) and figure 4(c), but in figure 4(a) it doesn't have a very good performance. BMSAL-NS also performs well and stably, but generally it doesn't have as good performance as BMSAL-NN.

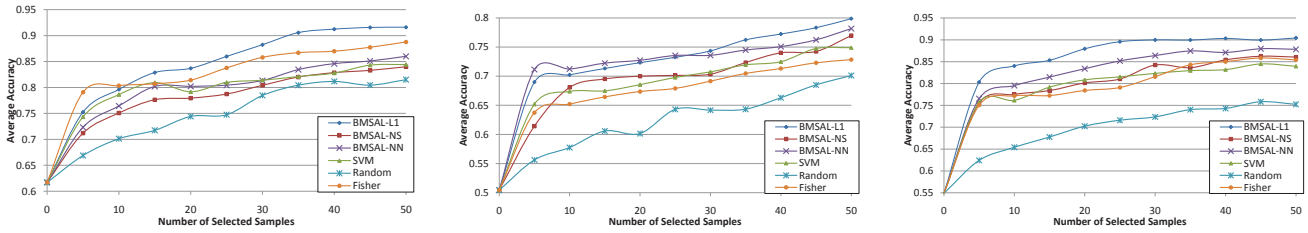
In order to compare the reliability of the methods, we compute the standard variances of the methods. Table II shows that BMSAL-L1 method is quite reliable, i.e., it tolerates outliers and non-linear properties, just as we have argued in section IV-A.

Data Set	BMSAL-L1	BMSAL-NS	BMSAL-NN	SVM	Fisher
M.v.W	0.0260	0.0375	0.0311	0.0649	0.0299
R.v.A	0.0144	0.0294	0.0268	0.0492	0.0287
WebKB	0.0164	0.0187	0.0268	0.0492	0.0287

Table II
STANDARD VARIANCES IN REAL-WORLD DATA SETS

V. CONCLUSION

In this paper, a unified framework integrating the sparse representation and batch mode active learning is proposed, namely BMSAL. The main goal is to design batch mode active learning method with high precision, efficiency and reliability. We solve the problem by designing and optimizing the greedy algorithm, extending the piecewise argument to non-linear data sets, and extensive experiments. An intriguing problem regarding our work is further speedups of BMSAL-L1 and BMSAL-NS, applying the BMSAL



(a) Result of MAC vs Windows (b) Result of Religion vs Atheism (c) Result of WebKB

Figure 4. Results of Real-World Data Sets

methods in very large scale applications is an interesting problem.

VI. Acknowledgement

This work was supported in part by the National Basic Research Program of China Grant 2007CB807900,2007CB807901,the National Natural Science Foundation of China Grant 60604033, 60553001, 61073174, 61033001 and the Hi-Tech research and Development Program of China Grant 2006AA10Z216.

References

[1] M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311 – 4322, 2006.

[2] E. Amaldi and V. Kann. On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science*, 209:237 – 260, 1998.

[3] R. Basri, T. Hassner, and L. Zelnik-Manor. Approximate nearest subspace search with applications to pattern recognition. In *CVRP'07*, 2007.

[4] R. Basri and D. W. Jacobs. Lambertian reflectance and linear subspaces. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(2):218–233, 2003.

[5] E. Candès, M. Rudelson, T. Tao, and R. Vershynin. Error correction via linear programming. In *FOCS'05*, pages 295–308, 2005.

[6] E. J. Candes and J. Romberg. Quantitative robust uncertainty principles and optimally sparse decompositions. *Found. Comput. Math.*, 6(2):227–254, 2006.

[7] E. J. Candès and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203 – 4215, 2005.

[8] E. J. Candès and M. Wakin. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, 25(2):21 – 30, 2008.

[9] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Rev.*, 43(1):129–159, 2001.

[10] D. L. Donoho. For most large underdetermined systems of linear equations the minimal ℓ_1-norm solution is also the sparsest solution. *Comm. Pure and Applied Math*, 59(6):797 – 829, 2006.

[11] D. L. Donoho and M. Elad. Optimally sparse representation in general(non-orthogonal) dictionaries via ℓ^1 minimization. In *PNAS'03*, pages 2197 – 2202, 2003.

[12] D. L. Donoho, Y. Tsaig, I. Drori, and J. Starck. Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit. Technical Report 2006-02, Standford, Department of Statistics, 2006.

[13] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[14] J. F. Gemmeke and B. Cranen. Noise reduction through compressed sensing. In *Proceedings of Interspeech*, pages 1785 – 1788, 2008.

[15] J. F. Gemmeke and T. Virtanen. Noise robust digit recognition using sparse representations. In *ISCA 2008 ITRW "Speech Analysis and Processing for knowledge discovery"*, 2008.

[16] S. C. H. Hoi, R. Jin, and M. R. Lyu. Large-scale text categorization by batch mode active learning. In *WWW '06*, pages 633–642, 2006.

[17] S. C. H. Hoi, R. Jin, J. Zhu, and M. R. Lyu. Semisupervised svm batch mode active learning with applications to image retrieval. *ACM Trans. Inf. Syst.*, 27(3):1–29, 2009.

[18] P. Indyk. Explicit constructions for compressed sensing of sparse signals. In *SODA '08*, pages 30–33, 2008.

[19] J. Wright K. Min, Z. Zhang and Y. Ma. Decomposing background topics from keywords by principal component pursuit. In *CIKM'10*, 2010.

[20] A. Krause and V. Cevher. Submodular dictionary selection for sparse representation. In *ICML'10*, 2010.

[21] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *J. Mach. Learn. Res.*, 9:235–284, 2008.

[22] J. Ma and F. L. Dimet. Deblurring from highly incomplete measurements for remote sensing. *IEEE Trans. Geoscience and Remote Sensing*, 47(3):792 – 802, 2009.

[23] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis. In *CVPR'08*, 2008.

[24] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. In *NIPS'08*, pages 1033–1040, 2008.

[25] G. Nemhauser, L. A. Wolsey, and M. L. Fischer. An analysis of the approximations for maximizing submodular set functions. *Mathematical Programming*, 14, 1978.

[26] P. Sen, G. M. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008.

[27] G. Shakhnarovich, T. Darrell, and P. Indyk. Nearest-neighbor methods in learning and vision. *IEEE Transactions on Neural Networks*, 19(2):377–377, 2008.

[28] T. R. Shultz and J. L. Elman. Analyzing cross-connected networks. In *NIPS'93*, pages 1117–1124, 1993.

[29] Z. Si and Y. N. Wu. Wavelet, active basis, and shape script: a tour in the sparse land. In *MIR'10*, pages 201–210, 2010.

[30] J. A. Tropp and A. C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 53(12):4655–4666, 2007.

[31] D. P. Wipf and B. D. Rao. Sparse bayesian learning for basis selection. *IEEE Transactions on Signal Processing*, 52(8):2153 – 2164, 2004.

[32] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(2):210–227, 2009.

[33] A. Y. Yang, J. Wright, Y. Ma, and S. S. Sastry. Feature selection in face recognition: A sparse representation perspective. Technical Report UCB/EECS-2007-99, University of California at Berkeley, 2007.

[34] J. Yang and Y. Zhang. Alternating direction algorithms for ℓ^1-problems in compressive sensing. Technical Report TR09-37, Rice University, 2009.

[35] X. Zhu, J. Lafferty, and Z. Ghahramani. Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In *ICML Workshop '03*, pages 58 – 65, 2003.