2011-01-01

A Sparse Representation Technique For Classification Problems

Reinaldo Sanchez Arias
University of Texas at El Paso, rsanchezarias@miners.utep.edu

A SPARSE REPRESENTATION TECHNIQUE FOR CLASSIFICATION PROBLEMS

REINALDO SANCHEZ ARIAS

Program in Computational Science

APPROVED:

Miguel Argáez, Ph.D., Chair

Leticia Velázquez, Ph.D.

Rodrigo Romero, Ph.D.

Patricia Witherspoon, Ph.D.
Dean of the Graduate School

A mi amada madre Alid,

mi padre Reinaldo, y mi hermano Juan Camilo

que son la luz de mi vida.

A SPARSE REPRESENTATION TECHNIQUE FOR CLASSIFICATION PROBLEMS

by

REINALDO SANCHEZ ARIAS

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

Program in Computational Science

THE UNIVERSITY OF TEXAS AT EL PASO

May 2011

Acknowledgements

I would like to express my sincere gratitude to my mentors Dr. Argáez and Dr. Velázquez for their guidance and encouragement. I am thankful for the opportunity they have given to me to work with them and for their support and advice while living this experience away from home. I also want to acknowledge Dr. Rodrigo Romero for kindly accepting being part of my thesis committee, and for his valuable observations.

Thanks to my amazing friends Paula, Carlos, Anibal, Ron, Javier, Clemente, and all the other ones that have made me feel as part of their family. I am forever thankful for having the chance to share this adventure with them, for their patience, trust, help and encouragement. Thank you all.

Infinitely many thanks to my family for their endless support. To my dear mother Alid for her unlimited love and support in every moment; to my brother Juan Camilo for his motivation and kindness; and to my dad Reinaldo for always lighting my way and for taking care of us from Heaven. This is for you.

I also want to thank the Computational Science Program and Department of Mathematical Sciences professors and staff for all their hard work and dedication. This work was supported by the Department of the Army ARL Grant No. W911NF-07-2-0027.

Abstract

In pattern recognition and machine learning, a classification problem refers to finding an algorithm for assigning a given input data into one of several categories. Many natural signals are sparse or compressible in the sense that they have short representations when expressed in a suitable basis. Motivated by the recent successful development of algorithms for sparse signal recovery, we apply the selective nature of sparse representation to perform classification. In order to find such sparse linear representation, we implement an ℓ_l-minimization algorithm. This methodology overcomes the lack of robustness with respect to outliers. In contrast to other classification algorithms such as Support Vector Machines (SVM), no model selection dependence is involved. The minimization algorithm is a convex relaxation-like algorithm that has been proven to efficiently recover sparse signals. To study its performance, the proposed method is applied to six tumor gene expression datasets with a large number of features but few samples. Our numerical results compare favorably with various SVM methods. We also test the effectiveness of our classification algorithm in the Fisher's Iris dataset where a large number of samples but a small number of features are available.

Since the process and techniques for acquiring and analyzing data advance every day at high rates, we need to manage and analyze large amounts of data for several different scientific problems. Future work aims to study the performance of our classification method when dimensionality reduction techniques are applied, including feature selection and feature extraction strategies.

Table of Contents

Chapter 1

Introduction

Several engineering and science applications involve solving linear inverse problems that are usually ill-conditioned and for which the use of regularization techniques is required to be able to propose useful solutions. Recently, regularization via *sparsity* constraints has become very popular, where we look for an approximate solution to a linear system of equations, with the requirement that it has as few nonzero components as possible. This kind of problems can be found in several applications in machine learning, image and signal processing, and coding and information theory among others. Moreover, it has been proven that sparse signals can effectively approximate compressible signals [4].

In machine learning and pattern recognition, the term "classification" refers to an algorithm/technique for assigning a given set of input data into one of a given number of categories. An example would be assigning a given email into "spam" or "non-spam" classes, or assigning a diagnosis to a given patient as described by observed characteristics of the patient (gender, blood pressure, presence or absence of certain symptoms, etc.). An algorithm that implements classification is referred to as a classifier.

Many natural signals are sparse or compressible in the sense that they have short representations when expressed in a suitable basis. Motivated by the recent successful development of algorithms for sparse signal recovery [11, 17, 21, 26], we apply the selective nature of sparse representation to perform classification. Any test sample is represented in an overcomplete dictionary with the training sample as base elements. In case we have sufficient training samples available for each class; test samples can be expressed as a linear combination of only those training samples belonging to the same class, therefore providing a naturally sparse representation. In order to find the sparsest linear representation we propose an algorithm

based on ℓ_1-minimization that allows us to overcome the lack of robustness to outliers [3]. Sparse representations of signals have received a great deal of attention in recent years. The sparse representation problem consists in searching for the most compact representation of a signal in terms of a linear combination of *atoms* in an overcomplete *dictionary*. Research has focused on *pursuit methods* for solving the optimization problem, such as matching pursuit [28], orthogonal matching pursuit [31], basis pursuit [11], and also on the *applications* of a sparse representation for different tasks, such as signal separation, denoising, and coding. This thesis is organized as follows:

Chapter 2 presents the mathematical background in the theory of compressed sensing that gave rise to the development of efficient optimization algorithms for sparse signal recovery. We explain the formulation for the sparse representation problem, the ideas guaranteeing the recovery of sparse signals via ℓ_1 minimization, and some of the strategies to solve this problem.

Chapter 3 explains the classification problem we aim to solve using a sparse representation approach. A general description of the mathematical formulation and the strategies used are presented.

Chapter 4 includes a short description of the ℓ_1-minimization algorithm we propose to use for solving the classification problem. The algorithm is presented in pseudo-code form and we explain its capabilities.

Chapter 5 presents numerical results of the classification technique we propose in this work for different datasets. We explain the experimental design, describe the datasets used, and present a comparison of our results with commonly used algorithms for classification.

Chapter 6 includes the conclusions of our work and the future research directions we have in mind to improve our technique. We carefully describe the strategies that will be used to enhance our classification algorithm and discuss their viability.

Chapter 2

Sparse Solution of Linear Inverse Problems

The problem of sparse representation consists in representing a given signal as a linear combination of as few "base" elements as possible from a fixed collection. That is, we aim to identify a sparse vector x such that the target signal b can be represented by $Ax \approx b$, where A is a known matrix. In this chapter we formulate the problem that must be solved in order to obtain approximate sparse solutions to linear systems of equations, and discuss the strategies that have been proposed in recent years.

2.1 Problem Formulation

Consider a real matrix $A \in \mathbb{R}^{m \times n}$ whose columns a_j have unit Euclidean norm, that is $\|a_j\|_2 = 1$, for $j = 1, ..., n$. We will often refer to this type of matrix as the *dictionary*. We say that a vector (signal) $x \in \mathbb{R}^n$ is *k-sparse* if $\|x\|_0 \leq k$, where the counting function $\|\cdot\|_0 \colon \mathbb{R}^n \to \mathbb{R}$, known as the ℓ_0 "norm" [16], gives the number of nonzero elements in its argument. In other words,

$$\|x\|_0 = \operatorname{card}\{i \colon x_i \neq 0\}. \tag{2.1}$$

Even though we call it the ℓ_0-norm, one can easily verify that it does not satisfy the positive homogeneity (positive scalability) property in the definition of a norm, namely we have that $\|\lambda x\|_0 \neq |\lambda| \|x\|_0$, for any given nonzero scalar λ.

A signal x is said to be *nearly sparse* if the rearranged entries of x, decay exponentially when sorted in decreasing order of magnitude [4]. Since compressible signals are well

approximated by sparse ones, the framework of sparse approximation applies to this class too.

Given that we are looking for the sparsest vector x satisfying the linear system of equations $Ax = b$, we are interested in solving the following optimization problem:

$$\begin{aligned} \min \quad & \|x\|_0 \\ \text{subject to} \quad & Ax = b, \end{aligned} \tag{2.2}$$

assuming that the matrix $A \in \mathbb{R}^{m \times n}$ is short and fat, that is $m \ll n$, in order to find the vector x with the fewest nonzero components among all the solutions to the system. Unfortunately, problem (2.2) is a combinatorial minimization problem and NP-hard (nondeterministic polynomial-time) [30]. Therefore any algorithm that is intended to solve (2.2) given the matrix A and the vector u, will be computationally intractable. Thus, strategies to overcome this difficulty had to be developed, which gave rise to different algorithmic approaches with remarkable results in different applications.

2.2 Algorithmic Approaches

During the last decade, several strategies have been proposed to find approximate solutions to problem (2.2). These different approaches include:

- ***Convex Relaxation.*** In this case, the objective function in problem (2.2) is replaced by a convex function, overcoming the combinatorial nature of the problem [11].

- ***Nonconvex Optimization.*** The idea consists in relaxing the ℓ_0 norm with a related nonconvex function, and attack the problem by identifying the corresponding stationary points. The use of ℓ_q quasi-norms ($0 < q < 1$) has been studied in [8].

- ***Greedy Pursuit.*** Iterative refinement of a sparse solution is proposed, by successively identifying those entries in the vector producing the greatest improvement [28].

In this work, we focus on developing a Convex Relaxation technique for finding an approximate solution to the sparse representation problem. The strategy uses an ℓ_1 relaxation of the ℓ_0 norm, through which successful recovering of sparse signals has been shown.

2.3 ℓ_1-minimization problem

A practical alternative to problem (2.2) is the ℓ_1 minimization approach, which consists in finding the solution to the problem

$$\begin{aligned} \min \quad & \|x\|_1 \\ \text{subject to} \quad & Ax = b, \end{aligned} \tag{2.3}$$

where $\|x\|_1 = \sum_{i=1}^{n} |x_i|$. We now have an optimization problem whose objective function is convex, unlike the ℓ_0-norm in problem (2.2). However, we must have special conditions on the matrix A and on the sparsity of x in order to guarantee that the solution of problem (2.3) will lead us to find the solution of the original problem.

The motivation for this approach comes from studying the theory of compressed sensing (compressive sampling) which has been a research topic of interest in the last years. The work in this area initiated in late 2004 by Emmanuel Candès, Justin Romberg and Terence Tao [4], and independently by David Donoho [13]. The general theme aims to answer the question: *How much information is necessary to accurately reconstruct a signal?*. One can reconstruct *sparse* or *compressible* signals accurately from a very limited number of measurements. We wish to recover an object $x \in \mathbb{R}^n$, using information from a collection of m linear measurements $b_i = \langle a_i, x \rangle$ for $i = 1, ..., m$. In matrix notation, we can write this as, $b = Ax$ where $A \in \mathbb{R}^{m \times n}$ with the vectors a_i as rows. We will assume that $m \ll n$ and the *measurement matrix* A has full rank.

2.3.1 Restricted Isometry Property

We will say that a matrix A satisfies the *restricted isometry property* (RIP) with parameters (r, δ) if (see [5])

$$(1-\delta)\|x\|_2^2 \leq \|Ax\|_2^2 \leq (1+\delta)\|x\|_2^2, \quad \text{for all } r \text{ sparse vectors } x. \tag{2.4}$$

The restricted isometry constant δ_r of a matrix A is the smallest number satisfying (2.4). This property essentially requires that every set of columns with cardinality less than r approximately behaves like an orthonormal system [4].

2.3.2 The Null Space Property

A matrix $A \in \mathbb{R}^{m\times n}$ satisfies the *null space property* (NSP) of order r with constant $\gamma \in (0,1)$ if (see [9])

$$\|v_S\|_1 \leq \gamma\|v_{S^c}\|_1, \tag{2.5}$$

for all sets $S \subset \{1, ..., n\}$ with $\#S \leq r$, and $v \in \ker(A)$. Here S^c is the complement of S in the set $\{1, ..., n\}$. It can be shown that if a matrix A satisfies the *restricted isometry property* (2.4) then it also satisfies the *null space property* (see [9]).

2.3.3 Sparse Recovery Result

Let $A \in \mathbb{R}^{m\times n}$ be a matrix satisfying the NSP of order r with constant $\gamma \in (0,1)$. Let x^* be the solution of the ℓ_1-minimization problem (2.3). If $x \in \mathbb{R}^n$ and $Ax = b$, then

$$\|x - x*\|_1 \leq \frac{2(1+\gamma)}{(1-\gamma)}\sigma_x, \tag{2.6}$$

where σ_x is a quantity that depends on the sparsity of x. If the vector x is r-sparse then $x = x^*$.

Proof Since $Ax = Ax^* = b$, then the vector $v = x - x^*$ is in $\ker(A)$. Also, since x^* solves (2.3), then $\|x^*\|_1 \leq \|x\|_1$. Let S be the set of the r largest components of x in absolute value.

We have

$$\|x_S^*\|_1 + \|x_{S^c}^*\|_1 \quad \leq \|x_S\|_1 + \|x_{S^c}\|_1.$$

Notice also that (use triangle inequality)

$$\|x_S\|_1 - \|v_S\|_1 + \|v_{S^c}\|_1 - \|x_{S^c}\|_1 \leq \quad \|x_S\|_1 + \|x_{S^c}\|_1,$$

so we get

$$\begin{aligned} \|v_{S^c}\|_1 \quad &\leq \|v_S\|_1 + 2\|x_{S^c}\|_1, \\ &\leq \gamma\|v_{S^c}\|_1 + 2\sigma_x. \end{aligned}$$

Therefore, $\|v_{S^c}\|_1 \leq \frac{2}{(1-\gamma)}\sigma_x$. Since $v = x - x^*$, then

$$\begin{aligned} \|x - x^*\| \quad &= \|v_S\| + \|v_{S^c}\| \\ &\leq (\gamma + 1)\|v_{S^c}\| \\ &\leq \tfrac{2(1+\gamma)}{(1-\gamma)}\sigma_x. \end{aligned}$$

In case the vector x is r sparse, then $\|x_{S^c}\|_1 = \sigma_x = 0$, so we get $x = x^*$.

Thus, the notion of ℓ_1 minimization is an effective technique for finding the sparsest solution x^* of a linear system of equations $Ax = b$.

2.4 Convex Relaxation Strategies

The ℓ_1 convex relaxation approach has been proven to sucessfully find sparse solutions to linear system of equations. In the following, we briefly describe some state-of-the-art algorithms developed for finding approximate solutions of the sparse representation problem based on ℓ_1 optimization techniques.

2.4.1 Donoho, Saunders et al. - Basis Pursuit (BP)

In their work [11], Donoho et al. proposed to reformulate problem (2.3) as a linear programming problem of the form

$$\begin{aligned} \min_{x} \quad & \textstyle\sum_{i=1}^{n} u_i \\ \text{s.t} \quad & -u_i \leq x_i \leq u_i, \\ & Ax = b. \end{aligned} \tag{2.7}$$

They were able to solve linear programs of size 8192 by $212,992$. They obtained reasonable success with a primal-dual logarithmic barrier method and a conjugate gradient solver. It is easy to check that problem (2.3) is equivalent to

$$\begin{aligned} \min \quad & c^T z \\ \text{subject to} \quad & \Phi z = f,\ z \geq 0, \end{aligned} \tag{2.8}$$

by letting $\Phi = [A, -A]$, $f = b$, $c = (\mathbb{1}; \mathbb{1})$, $z = (u, v)$ and $x = u - v$.

Even though the approach provides strong guarantees and stability, it relies on Linear Programming, whose methods do not yet have strong polynomially bounded runtimes. It is worthwhile to mention, that the work by the authors of [11] was done several years before the results Candès and Tao proved on the recovery of sparse signals via the ℓ_1 minimization approach. In [6], Candès and Tao characterized the conditions that must be satisfied for finding the actual solution to the original problem (2.2), when using the ℓ_1-minimization alternative.

A natural variation to the basis pursuit problem (2.3) consists in relaxing the linear constraint in order to consider an error tolerance, say $\epsilon \geq 0$, for the situation when the signal is contaminated with some additive noise. More specifically, the following problem is considered:

$$\begin{aligned} \min \quad & \|x\|_1 \\ \text{subject to} \quad & \|Ax - b\|_2 \leq \epsilon. \end{aligned} \tag{2.9}$$

The work in [4] claims that the convex relaxation approach (2.9) is also effective in finding an approximated solution of the sparse problem (2.2) whenever the observations are contaminated with a bounded additive noise.

2.4.2 Boyd, Lustig et al.

Boyd and his research group [26] proposed to solve a generalized version of (2.3) that allows certain degree of noise, given by the unconstrained minimization problem

$$\min_{x} \quad \lambda\|x\|_1 + \|Ax - b\|_2^2. \tag{2.10}$$

where the parameter $\lambda > 0$ is used as a penalization parameter balancing the tradeoff between error and sparsity.

First, problem (2.10) is posed as the following constraint problem

$$\begin{aligned} \min \quad & \lambda \textstyle\sum_{i=1}^{n} u_i + \|Ax - b\|_2^2 \\ \text{s.t} \quad & -u_i \leq x_i \leq u_i. \end{aligned} \tag{2.11}$$

Secondly, using the notions of interior-point method (log-barrier method) they designed an algorithm to find a solution of the dual problem of (2.11). Their method makes use of a preconditioned conjugate gradient (PCG) to accelerate convergence and stabilize the algorithm. They also showed the application of their algorithm on a magnetic resonance imaging (MRI) data set. One drawback of their approach is that each step would require the solution of a Newton system of the form $H\Delta x = g$, where $H \in \mathbb{R}^{2n\times 2n}$ is the Hessian matrix and g is the gradient at the current iterate. To overcome this difficulty, they compute a search direction of an approximate Newton system using a PCG. This alternative is commonly known as the Truncated Newton Method. The truncation rule for the PCG provides the condition for terminating the algorithm. The total number of PCG iterations required by the truncated Newton interior-point method depends on the value of the regularization parameter λ and a given relative tolerance ϵ. An implementation of their algorithm is available at `http://www.stanford.edu/~boyd/l1_ls/`

2.4.3 Figueiredo, Wright et al.

Figueiredo et al. [17] studied the unconstrained problem

$$\min_x \quad \lambda\|x\|_1 + \tfrac{1}{2}\|b - Ax\|_2^2, \tag{2.12}$$

as an alternative to find the sparsest solution x of the system $Ax = b$. They posed (2.12) as a quadratic programming problem of the form

$$\begin{aligned} \min \quad & z^T Bz + c^T z \\ \text{s.t} \quad & z \geq 0, \end{aligned} \tag{2.13}$$

and their algorithm follows the *gradient projection* methodology.

The Gradient Projection for Sparse Reconstruction (GPSR) algorithm is based on the well-known projected gradient step technique

$$v^{(k+1)} \quad = v^{k-1} - \alpha_k \nabla F(v^k),$$

where F is the function to be minimized. In this case

$$F(v) \quad = \lambda \mathbb{1}^T v + \tfrac{1}{2}\|b - [A, -A]v\|_2^2,$$

with $\mathbb{1}$ the vector of ones, and $v = [v_1, v_2]$ with $v[i] \geq 0$ for all i. The step-length α_k is chosen following a backtracking technique.

Notice that

$$\|x\|_1 \quad = \mathbb{1}^T \begin{bmatrix} v_1 \\ v_2 \end{bmatrix},$$

if we let $(v_1)_i = (x_i)_+$ and $(v_2)_i = (-x_i)_+$ where $(\cdot)_+$ denotes the positive part, $(x)_+ = \max\{0, x\}$. Therefore we can formulate the quadratic programming problem:

$$\begin{aligned} \min \quad & \mathbf{c}^T v + \tfrac{1}{2} v^T \mathbf{B} v \\ s.t \quad & v \geq 0, \end{aligned} \tag{2.14}$$

where $x = v_1 - v_2$ and

$$\mathbf{b} = A^T b, \quad \mathbf{c} = \lambda \mathbb{1} + \begin{bmatrix} -\mathbf{b} \\ \mathbf{b} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} A^T A & -A^T A \\ -A^T A & A^T A \end{bmatrix}.$$

An implementation of the algorithm is available at `http://www.lx.it.pt/~mtf/GPSR/`. One of the issues of the GPSR algorithm is that the formulation of the problem in (2.13) doubles the dimension of the variables involved in the original problem (2.12). Any matrix operation involving the matrix B must then take special care of its structure with respect to A and A^T.

2.4.4 Zhang et al.

The group from Rice University leaded by Y. Zhang, developed an algorithm to solve the problem

$$\min_x \quad \|x\|_1 + \tfrac{\mu}{2}\|Ax - b\|_2^2, \tag{2.15}$$

using a Fixed Point Continuation (FPC) method [21]. The main idea described in [21] consists on deducing a fixed point equation of the form $w = F(w)$ which holds at the solution, making use of subgradient optimality conditions. To guarantee convergence, appropriate parameters are chosen so that F is a contraction, and therefore $x_{k+1} = F(x_k)$ converges. Solutions of (2.15) are also fixed points of where τ is a fixed constant and shrink $\left(c, \frac{\tau}{\mu}\right)$ is the *shrinkage* operator

$$\text{shrink}\left(c, \frac{\tau}{\mu}\right) \quad = \text{sgn}(c) \circ \max\left\{|c| - \tfrac{\tau}{\mu}, 0\right\}, \tag{2.16}$$

which is the explicit solution of

$$\min_x \quad \|x\|_1 + \tfrac{\mu}{2\tau}\|x - c\|_2^2. \tag{2.17}$$

The authors in [21] proved that the fixed-point iterations

$$x^{k+1} \quad = \text{sgn}\left(x^k - \tau g(x^k)\right) \circ \max\left\{|x^k - \tau g(x^k)| - \tfrac{\tau}{\mu}, 0\right\} \tag{2.18}$$

where $g(x) = (A^T(Ax - b))$, converge to a solution of (2.15) as long as $0 < \tau < 2$. The convergence rate is accelerated by letting μ be small, in which case $\frac{\tau}{\mu}$ is large producing a solution x^* very sparse.

The FPC algorithm can be found at `http://www.caam.rice.edu/~optimization/L1/fpc/`

2.4.5 M. Argáez et al.

The research group led by M. Argáez has been working on the basis pursuit ℓ_1-minimization problem (2.3). In [2] we propose to find a solution to (2.3) by solving a sequence of problems of the form

$$\begin{aligned} \min_x \sum_{i=1}^{n} \quad & (x_i^2 + \mu)^{1/2} \\ \text{s.t} \quad & Ax = b, \end{aligned} \tag{2.19}$$

as the parameter μ tends to 0. This approach leads to a path-following method to find the solution x of the ℓ_1-minimization problem, by solving a sequence of linear equality constrained multiquadric problems that depend on a regularization parameter that converges to zero. We have developed a homotopic principle for solving large-scale ℓ_1 underdetermined problems. Numerical experimentation has shown that our algorithm is capable of recovering sparse signals, with results comparing favorably - in both accuracy and CPU running time - with the state-of-the-art algorithms mentioned before, as reported in [2]. The MATLAB implementation of the path-following algorithm can be found at `http://www.math.utep.edu/Student/rsanchez/` The Path Following Signal Recovery (PFSR) algorithm will be used as a tool to solve classification problems. In Chapter 4 we give a more detailed description of the ideas behind the PFSR algorithm and the methodology followed in [2].

Chapter 3

Classification Problem

In pattern recognition and machine learning, a *classification problem* refers to finding an algorithm for assigning a given input data into one of several categories. Since many natural signals are sparse or compressible, in the sense that they have short representations when expressed in an appropriate basis, we propose to apply the *selective nature* of sparse representation to perform classification. As studied in the previous chapter, ℓ_1-minimization techniques provide a satisfactory methodology to solve sparse representation problems. We propose a classifier based on the solution of an ℓ_1-minimization problem for classification.

3.1 Description

Machine Learning is a research area concerned with the design of systems that can *learn* from provided input. Usually, such systems are designed to use learned knowledge to handle similar input in the future. For instance, an email spam-detecting system, where a given set of emails are marked as spam or not-spam, learns the common features of spam emails to be able to identify future email messages as either spam or not-spam. This technique is known as supervised statistical classification. Supervised because the system is first trained using already classified training data. A supervised learning system performing classification is commonly called a *classifier*. Formally, given an input dataset, $\mathbf{W} = \{\mathbf{w}_1, ..., \mathbf{w}_n\}$, a set of labels/classes $\mathbf{T} = \{t_1, ..., t_n\}$, and a training dataset $\mathbf{D} = \{(\mathbf{x}_i, t_i) : i = 1, ..., n\}$ such that t_i is the label/class of the sample $\mathbf{x}_i$, a classifier is a mapping f from $\mathbf{W}$ to $\mathbf{T}$, assigning the correct label t to a given input $\mathbf{w}$, that is, $f(\mathbf{D}, \mathbf{w}) = t$.

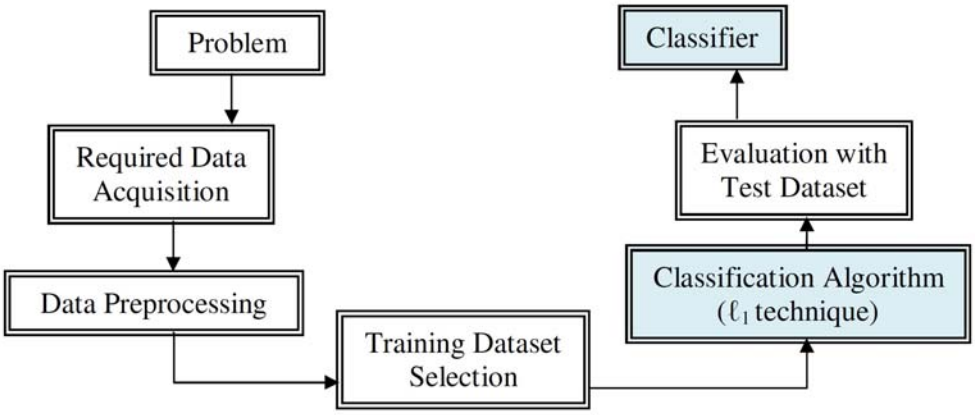

Figure 3.1: The process of classification

3.2 Mathematical Formulation

Let us consider a *training data set* $\{(\mathbf{x}_i, t_i) : i = 1, ..., n\},\ \mathbf{x}_i \in \mathbb{R}^d,\ t_i \in \{1, 2, ..., N\}$, where n is the number of samples and N the number of classes. The vector $\mathbf{x}_i \in \mathbb{R}^d$, represents the ith sample (for instance containing "gene expression" values), and t_i denotes the label of the ith sample.

The sparse representation problem is formulated as follows: For a testing sample $\mathbf{y} \in \mathbb{R}^d$, find the sparsest vector $\mathbf{c} = [c_1, c_2, ..., c_n]^T$ such that

$$\mathbf{y} \quad = c_1\mathbf{x}_1 + c_2\mathbf{x}_2 + ... + c_n\mathbf{x}_n. \tag{3.1}$$

We show that indeed a valid test sample can be represented using only the training samples from the same class, therefore inducing a natural sparse representation. Let us rearrange the given n_i training samples from the same i-th class as the columns of a submatrix $A_i = [\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, ..., \mathbf{x}_{i,n_i}] \in \mathbb{R}^{d\times n_i}$. That is, we group all of those samples with the same label into a matrix A_i. In case we have sufficient training samples of the i-th class, any test sample $\mathbf{y}$ from the same class will be represented as a linear combination of the training samples associated with class i :

$$\mathbf{y} \quad = c_{i,1}\mathbf{x}_{i,1} + c_{i,2}\mathbf{x}_{i,2} + ... + c_{i,n_i}\mathbf{x}_{i,n_i}, \tag{3.2}$$

for some values of $c_{i,j} \in \mathbb{R}, j = 1, ..., n_i$. Now, making use of the whole training dataset, we define a $d \times n$ matrix A by concatenating all of the n training samples of the different N classes, that is $A = [A_1, A_2, ..., A_N]$. Then, the linear representation of the test sample $\mathbf{y}$ that belongs to class i is written by:

$$\mathbf{y} = A\mathbf{c}, \tag{3.3}$$

where $\mathbf{c} = [0, ..., 0, c_{i,1}, c_{i,2}, ..., c_{i,n_i}, 0, ...0]^T \in \mathbb{R}^n$. Therefore, the test sample $\mathbf{y}$ is expressed by a sparse linear representation. This motivates us to formulate the following problem:

$$\begin{aligned} \min \quad & \|\mathbf{c}\|_0 \\ \text{s.t} \quad & A\mathbf{c} = \mathbf{y}. \end{aligned} \tag{3.4}$$

To obtain a sparse linear representation for the test sample $\mathbf{y}$, we propose to solve problem (3.4) using a convex relaxation technique via ℓ_1 minimization.

In this work, we consider an error vector $\mathbf{e}$ associated to the problem, so any sample is written as:

$$\mathbf{y} = A\mathbf{c} + \mathbf{e},$$

which is equivalent to $\mathbf{y} = B\mathbf{d}$, with

$$B = [A \ I], \quad \mathbf{d} = [\mathbf{c}, \ \mathbf{e}]^T. \tag{3.5}$$

Here I represents a $d \times d$ identity matrix, and $B \in \mathbb{R}^{d \times (d+n)}, \ \mathbf{d} \in \mathbb{R}^{n+d}$.

Now, the sparse linear representation for the test sample $\mathbf{y}$ is obtained by solving the following ℓ_1-minimization problem

$$\begin{aligned} \min \quad & \|\mathbf{d}\|_1 \\ \text{s.t} \quad & B\mathbf{d} = \mathbf{y}. \end{aligned} \tag{3.6}$$

We propose to solve this problem using the ℓ_1-minimization algorithm introduced by Argáez et al. [2] and described in subsection 2.4.5. One of the advantages of our formulation is that lack of robustness with respect to outliers can be overcome. Also, and we do not need to care for model selection as in support vector machine approaches.

3.3 Discriminant Functions and Classifier

Once the sparse representation vector $\hat{\mathbf{d}} = [\hat{\mathbf{c}}, \hat{\mathbf{e}}]^T$ has been found as a solution to (3.6), we identify the class to which the testing sample $\mathbf{y}$ belongs. The approach consists in associating the nonzero entries of $\hat{\mathbf{c}}$ with the columns of A corresponding to those training samples having the same category of the testing sample $\mathbf{y}$. The solution vector $\hat{\mathbf{c}}$ is decomposed as the sum of d-dimensional vectors $\hat{\mathbf{c}}_k$, where $\hat{\mathbf{c}}_k$ is obtained by keeping only those entries in $\hat{\mathbf{c}}$ associated with category k and assigning zeros to all the other entries. Then, we define the N *discriminant* functions

$$g_k(\mathbf{y}) \quad = \|\mathbf{y} - A\hat{\mathbf{c}}_k\|_2, \qquad k = 1, ..., N. \tag{3.7}$$

Thus, g_k represents the approximation error when $\mathbf{y}$ is assigned to category k. Finally, we classify $\mathbf{y}$ in the category with the smallest approximation error. That is,

$$\hat{\mathbf{t}} \quad = \arg\min_{k=1,...,N} \{g_k(\mathbf{y})\}. \tag{3.8}$$

In this manner, we identify the class of the test sample $\mathbf{y}$ based on how effectively the coefficients associated with the training samples of each class recreate $\mathbf{y}$.

3.4 Support Vector Machines (SVM)

We will be comparing the results of our proposed method for classification problems, with the well known Support Vector Machines (SVM) method, that has been commonly used in different pattern recognition and machine learning applications.

Support vector machines (SVMs) are a set of related supervised learning methods that analyze data and recognize patterns, used for classification and regression analysis. The original SVM algorithm was invented by Vladimir Vapnik and the current standard implementation was proposed by Corinna Cortes and Vladimir Vapnik [10]. Standard SVM takes a set of input data, and predicts, for each given input, which of two possible classes the input is a member of, which makes the SVM a non-probabilistic binary linear classifier.

Since an SVM is a classifier, then given a set of training examples, each marked as belonging to one of a set of specific categories, an SVM training algorithm builds a model that predicts whether a new example falls into one category or the other. Intuitively, an SVM model is a representation of the samples as points in space, mapped so that the samples of the separate categories are divided by a clear gap that is as wide as possible.

In the original SVM approach, an N-class classification problem is converted into N two-class problems (binary classification), and in the j-th two-class problem, the optimal decision function that separates class j from the remaining classes is determined. If more than one decision function classify a data point into a definite class, the data point is not classifiable. Slow training is also a possible drawback of support vector machines approaches. This has to do with the fact that support vector machines are *trained* by solving quadratic programming problems where the number of variables are equal to the number of samples in the training data set. When a large number of training data is available, the training process might turn slow. More information about the different strategies used in SVM for classification problems are described in [1, 36]. Techniques for accelerating the training process, specialized methods for multiclass problems, and nonlinear separation of class data, among other SVM related topics, are also discussed in [1].

Chapter 4

Solving the ℓ_1 Optimization Problem

We propose to solve the sparse representation problem for classification described in the previous chapter, following a pure basis pursuit problem (2.3) formulation with the convex relaxation technique described by Argáez et al. in [2].

4.1 Algorithmic Approach

In this work we propose to solve the sparse representation problem, applying the Path Following Signal Recovery (PFSR) algorithm introduced in [2]. This algorithm is a convex relaxation basis pursuit algorithm that has been shown to effectively recover sparse solutions to underdetermined linear systems of equations, with results comparing favorably with some of the state-of-the-art algorithms [17, 21, 26] in both reconstruction error and CPU running time.

In [2] we consider a homotopic principle for solving large-scale and dense ℓ_1 underdetermined problems of the type (2.3). The idea consists in obtaining the solution of the problem by solving a sequence of linear equality constrained multiquadric problems that depends on a regularization parameter μ that converges to zero. The procedure generates a central path that converges to a point on the solution set of the ℓ_1-underdetermined problem. More specifically, we solve a sequence of subproblems of the form

$$\min_x \sum_{i=1}^{n} \left(x_i^2 + \mu\right)^{1/2} \qquad \text{subject to } Ax + \nu = b, \tag{4.1}$$

so we can apply the formulation of the classification problem proposed in equation (3.5),

where we also consider the construction error vectors in the data set. Namely, the classification problem requires solving the optimization problem of finding the sparsest solution of an underdetermined system of linear equations:

$$\begin{aligned} \min \quad & \|\mathbf{d}\|_1 \\ \text{s.t} \quad & B\mathbf{d} = b, \end{aligned} \tag{4.2}$$

where $B = [A\ I]$, and $\mathbf{d} = [\mathbf{c},\ \mathbf{e}]^T$. Here I represents a $d \times d$ identity matrix, and $B \in \mathbb{R}^{d\times(d+n)}$, $\mathbf{d} \in \mathbb{R}^{n+d}$.

Several experiments and results in different applications had shown that the PFSR algorithm is capable of successfully recover sparse signals. Applications in seismic reflection, speech separation, and magnetic resonance imaging (MRI) via compressed sensing, are presented in [2].

4.2 Algorithm Description and Methods

In Algorithm 1 we describe the steps followed by the PFSR algorithm in order to find an approximate solution of the basis pursuit problem (2.3).

This PFSR algorithm generates two sequences of iterates. The first sequence (inner loop) generates a series of iterates for obtaining an approximate solution of subproblem (4.1) for a fixed regularization parameter $\mu > 0$. The second sequence (outer loop) generates a series of approximate solutions for the subproblems (4.1) that converges to an optimal solution of the ℓ_1 minimization problem

$$\begin{aligned} \min \quad & \|x\|_1 \\ \text{subject to} \quad & Ax + \nu = b, \end{aligned} \tag{4.3}$$

for a sequence of decreasing regularization parameters $\mu > 0$. The initialization parameters σ, τ, μ, and ϵ_1 are used for defining the tolerance and regularization parameter within the algorithm.

Algorithm 1 Path Following Signal Recovery

The PFSR Algorithm

Task: Find an approximate solution x to the problem

$\min_x \|x\|_1$ subject to $Ax + \nu = b$ and $\|\nu\| \leq \epsilon$

Parameters: We are given the matrix A and the vector b

Step 1. **Initialization:** Set: $\sigma,\ \tau,\ \mu,\ \epsilon_1$

Step 2. Initial approximate solution $x = A^T b$

Step 3. **Outer Loop** : **for** $j = 1, ..., maxiter$

Step 4. **Inner Loop** : Set $x_- = x$

Step 5. Update weight matrix: $D_\mu(x_-) = diag(x_-^2 + \mu)$

Step 6. Solve the fixed-point equation:

$$\begin{bmatrix} D_\mu(x_-)^{-1/2} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ b \end{bmatrix}$$

Step 7. **Check proximity to the central path:**

if $\frac{\|x - x_-\|}{1+\|x\|} \geq \sqrt{\mu}$ **go to** step 4

Step 8. Set $\tilde{x} = |x|$, $w = A^T y$, $\tilde{z} = (\mathbb{1} - |w|)$

Step 9. Compute $error_{primal} = \frac{\|Ax-b\|}{1+\|b\|}$, $gap = \frac{\tilde{x}^T \tilde{z}}{n}$

Step 10. **Stopping criteria for the problem:**

if $error_{primal} > \epsilon$ or $gap > \epsilon_1$

Update $\mu = \min\{\sigma gap, \tau\mu\}$, **go to** step 3

else

display 'x is an optimal solution'

In Step 6 of Algorithm 1, y represents the Lagrange multiplier associated to the equality constraint in problem 4.3. It is important to notice that in our algorithm, Step 6 is reformulated and solved using a specially designed Conjugate Gradient (CG) algorithm. Specifically, for a current point x_-, the first block of equations of the system in Step 6, gives $x + D_\mu(x_-)^{1/2}A^T y = 0$, and since $Ax + \nu = b$, we obtain the *weighted normal equation*

$$AD_\mu(x_-)^{1/2}A^T \quad = -b + \nu. \tag{4.4}$$

In order to solve (4.4), we apply a CG method and then compute the new approximation for x as in $x = -D_\mu(x_-)^{1/2}A^T y$. Taking into account that the values of $A^T y$ characterize the optimality set of problem (4.3), we formulate a conjugate gradient algorithm that finds an approximation of $A^T y$ rather than just y in equation (4.4). Notice that for a matrix $A \in \mathbb{R}^{d\times n}$, the linear system of equations in 4.4 is of size $d \times d$, so our methodology solves a small system of equations at each step whenever $d < n$.

Chapter 5

Experiment Design and Numerical Experimentation

In this chapter we describe the method used for testing the effectiveness of our classification approach. As performance metric, we evaluate the accuracy of the proposed sparse representation technique for classification in a *10-fold stratified cross-validation* experiment.

We test our method in two different type of datasets: one kind of datasets where we have a small number of samples each with a large number of features; and the other one where we consider a large number of samples with few features. In the former case, we use six datasets available at the Gene Expression Model Selection (GEMS) library for cancer classification using gene expression data. The GEMS software includes a graphical user interface and can be freely downloaded at `http://www.gems-system.edu/`. This software was also used in [33] for studying the performance of multicategory classifiers on gene expression cancer diagnosis. Our results are compared with the Support Vector Machines (SVM) technique, which has been successfully applied in gene profile classification. For the kind of datasets where we have a large number of samples with a few features, we test the performance of our method using the classic Fisher's Iris dataset [18]. This dataset, also known as the Iris flower dataset, consists of samples from each of the three classes of Iris flowers. The samples, whic are included in the MATLAB Statistic Toolbox, can be easily accessed and used for classification purposes.

All experiments were performed on a PC with an Intel(R) Core (TM) 2 Duo 2.20 GHz processor, 4 GB of memory, and MATLAB R2009b under Windows 7.

5.1 K-fold cross validation

Classifier performance is commonly measured by the classifier's error rate on the entire population. Cross Validation is a statistical method for evaluating machine learning algorithms in which the data is divided in two sets: one used for the training stage, and the second one used for testing (validation). These two training and testing sets should cross-over in consecutive rounds in such a way that each sample in the data set has a chance of being validated.

In the case of K-fold cross validation, a K-fold partition of the dataset is created by splitting the data into K equally (nearly equal) sized subsets (folds), and then for each of the K experiments, $K-1$ folds are used for *training* and the remaining one for *testing*. Therefore, each of the K subsamples is used exactly once as the validation data. One of the advantages of K-fold cross validation is that, eventually, all samples in the dataset are used for both testing and training.

If a large number of folds is used, the bias of the true error will be small though the method might be computationally expensive. A common choice for K-Fold Cross Validation is $K = 10$. The work in [27], compares several approaches for estimating accuracy, and recommends stratified 10-fold cross-validation as the best model selection method because it provides less biased estimation of the actual accuracy.

Given a dataset with N elements and a classifier algorithm, say $\mathcal{F}$, the averaged cross-validation accuracy of the classifier $\mathcal{F}$ on these N samples, can be considered as an estimate for the accuracy of $\mathcal{F}$ on *unseen* data when the classifier is trained with all the different samples.

5.2 Large number of features and few samples

In this numerical experimentation we use 6 different datasets from the GEMS library that are freely available in MATLAB `.mat` format at the webpage `http://www.gems-system.edu/`.

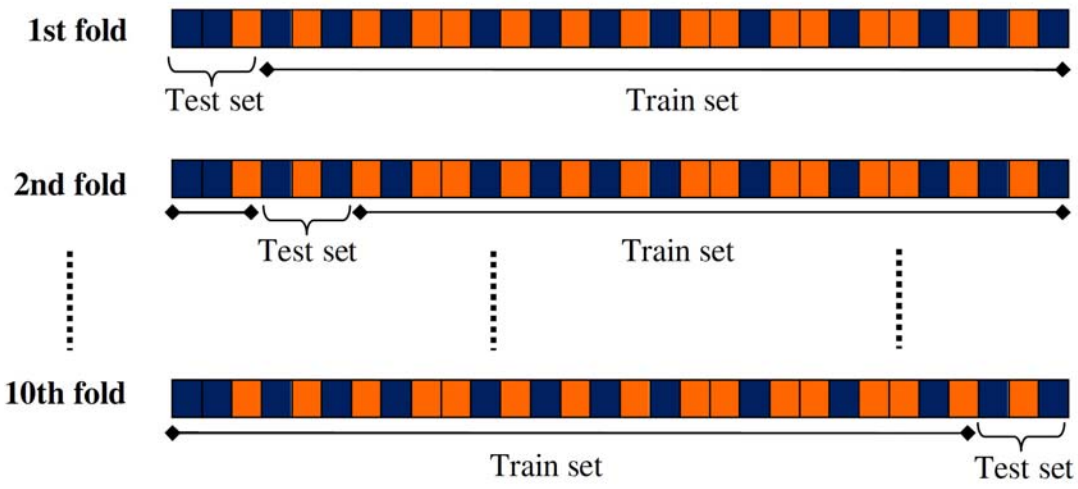

Figure 5.1: A 10-fold cross validation partition example

We also compare our numerical results, with the ones obtained using the Support Vector Machine (SVM) implementation available in the GEMS software.

5.2.1 Dataset Description

A short description of the datasets used follows:

- `9_Tumor`. The dataset comes from a study of 9 human tumor types: NSCLC, colon, breast, ovary, leukemia, renal, melanoma, prostate, and CNS.
- `11_Tumors`. Consists of gene expression data of 11 various human tumor types: ovary, bladder/ureter, breast, colorectal, gastro-esophagus, kidney, liver, prostate, pancreas, adeno lung, and squamous lung.
- `Prostate_Tumor`. Binary dataset contains gene expression data of prostate tumor and normal tissues.
- `Lung_Cancer`. Dataset of 4 lung cancer types and normal tissues.
- `SRBCT`. Small, round blue cell tumors (SRBCT) of childhood.

- `Brain_Tumor`. Dataset from a study of 5 human brain tumor types: medulloblastoma, malignant glioma, AT/RT, normal cerebellum, and PNET.

In the following table, the number of samples and genes for each dataset is described.

Table 5.1: Dataset sizes

Dataset	# Samples	# Genes	# Classes
`9_Tumors`	60	5726	9
`11_Tumors`	174	12533	11
`Prostate_Tumor`	102	10509	2
`Lung_Cancer`	203	12600	5
`SRBCT`	83	2308	4
`Brain_Tumor`	90	5920	5

5.2.2 Numerical Results

We solve the classification problem as posed in (3.6), that is, for each test we look for a solution of the ℓ_1 minimization problem:

$$\begin{aligned} \min \quad & \|\mathbf{d}\|_1 \\ \text{s.t} \quad & B\mathbf{d} = \mathbf{y}, \end{aligned} \tag{5.1}$$

where B is an augmented matrix of the form $B = [A\ I]$, and $\mathbf{d} = [\mathbf{c},\ \mathbf{e}]^T$. The matrix A is just the matrix built using the dataset elements, and for our numerical experiment we normalize the columns of A in such a way that they all have unit norm, i.e. $\|e_i^T a_i\|_2 = 1$, with e_i the i-th canonical basis vector and a_i being the i-th column of A.

The PFSR algorithm (Algorithm 1) is applied to solve each of the problems of the form (5.1) that are needed at every iteration of a 10-fold cross-validation test. The PFSR algorithm and the complete validation experiment for each dataset are implemented in MATLAB. Notice that even though we use the augmented matrix $B = [A\ I]$ in our problem formulation,

the PFSR algorithm only requires matrix-vector multiplication operations. Thus, we do not need to store the complete matrix B but only A since,

$$B\mathbf{d} = A\mathbf{c} + \mathbf{e}, \tag{5.2}$$

$$B^T\mathbf{y} = [A^T\mathbf{y}, \mathbf{y}]^T, \tag{5.3}$$

so we can implement in a fast way the matrix-vector multiplications required by the PFSR algorithm.

We compare our results using the Sparse Representation (SR) approach proposed in this work, with the classification method of Support Vector Machines (SVM). In order to perform this comparison, we use the software GEMS which has implemented several multiclass SVMs including one-versus-rest (OVR), one-versus-one (OVO), and directed acyclic graph (DAG). Polynomial and Radial Basis Functions (RBF) kernels are used for SVMs.

In Table 5.2 we show the performance measure results for each of the datasets tested in this experimentation and we compare the classifier's error rate is computed and compared.

Table 5.2: Performance of Classifier: sparse representation (SR) and SVM

Dataset	# Samples	# Genes	SR	SVM
`9_Tumors`	60	5726	66.67%	67.01%
`11_Tumors`	174	12533	96.55%	94.99%
`Prostate_Tumor`	102	10509	94.12%	93.27%
`Lung_Cancer`	203	12600	95.07%	96.05%
`SRBCT`	83	2308	100%	100%
`Brain_Tumor`	90	5920	91.11%	90.00%

In Figure 5.2, the sparse representation for the last cross-validation test on the binary dataset `Prostate_Tumor` is presented. One can notice the contrast between the large coefficients and the small ones, suggesting that the given test sample belongs to exactly one of the two classes in this dataset. This fact confirms the idea of expressing any test sample as a linear combination of only those training samples belonging to the same class.

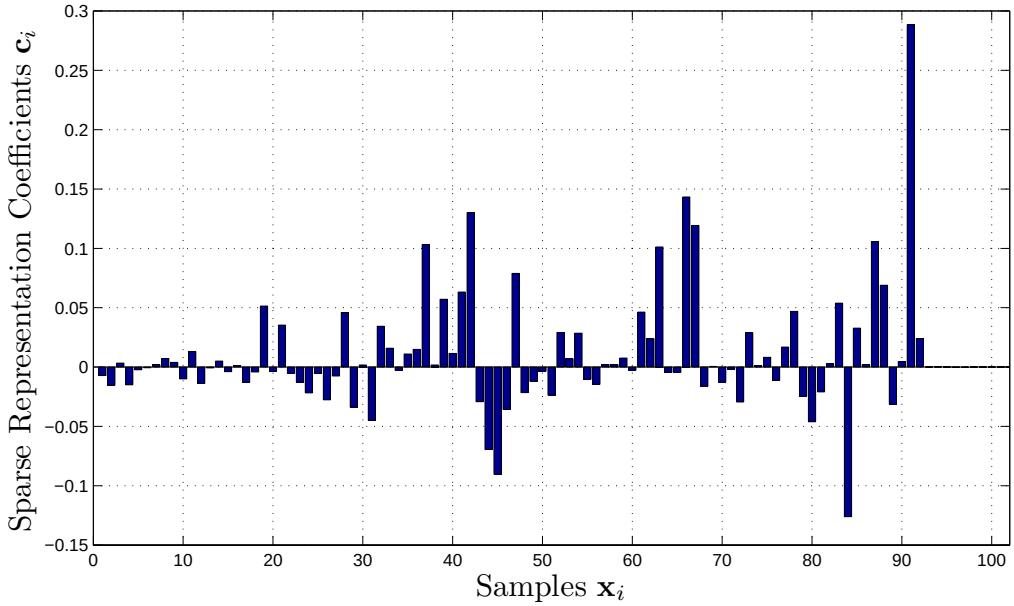

Figure 5.2: Sparse Representation of a test sample $\mathbf{y}$ in the binary dataset `Prostate_Tumor`

Discussion

The Sparse Representation (SR) results reported in Table 5.2 are encouraging when compared with the SVM approach for classification problems. We see that SR meets the best performance reached by SVMs method. In this work we reported the best performance we obtained from SVM methods using the GEMS software, which corresponds to using SVM one-versus-rest approach option available. Our SR technique in fact performs better than the SVM implementation for most of the datasets tested in this work, in terms of accuracy of the classifier according to the 10-fold cross-validation experiment studied for performance quantification.

From Table 5.2 we also see the "low" rate of accuracy for the dataset `9_Tumors`, which is probably related to the number of samples available, since from a total of 60 samples only 2 are available for category 7 corresponding to the prostate tumor case. This contrasts with the 9 samples available for non-small cell lung cancer (NSCL); 8 samples for breast, renal

and melanoma cases; 7 for colon and 6 for ovarian, leukemia, and central nervous system (CNS) cases. Therefore, in the situation when the only two samples available for category 7 happen to be in the testing dataset, generated by the random 10-fold cross validation stage, we will not have any samples of this category for training, i.e., these samples do not have any natural sparse linear representation using those elements in the training dataset.

Definitely one of the advantages of the SR technique based on ℓ_1 optimization is that we do not need to care for model selection as with SVM. Also, robustness with respect to outliers and noise in the dataset is gained when using the ℓ_1 norm.

5.3 Large number of samples and few features

We investigate the performance of the sparse representation approach for classification on the classic Fisher's Iris dataset [18]. Fisher developed a linear discriminant model to distinguish one species from another based on the combination of four different features. MATLAB has incorporated this dataset under the name of `fisheriris` as part of its Statistics Toolbox.

5.3.1 Dataset Description

This dataset was introduced in 1936 by Sir Ronald Aylmer Fisher and consists of 50 samples from each of the three classes of Iris flowers: *iris setosa*, *iris versicolor*, and *iris virginica*. The dataset contains 4 different feature measurements (in centimeters): the sepal length, sepal width, petal length, and petal width of 150 iris specimens.

We are interested in studying the effectiveness of the sparse representation approach on this dataset since it represents a dataset where we have more samples than features per sample. Specifically, we have $d = 4$, $n = 150$, and our formulation of the problem can be written as

$$\begin{aligned} \min \quad & \|\mathbf{c}\|_1 \\ \text{s.t} \quad & A\mathbf{c} = \mathbf{y}, \end{aligned} \tag{5.4}$$

giving rise to a highly underdetermined linear system where the matrix $A \in \mathbb{R}^{d \times n}$ and we can avoid working with the augmented matrix $B = [A\ I]$. This contrasts with the high number of features (genes) available when we work in the classification of tumors using DNA microarray information, where the number of samples is very small compared with the number of genes on each sample.

In Figure 5.3 and 5.4 we show how the sepal and petal measurements differ from class to class. We use a scatter plot to show the values of width and length for both sepal and petal.

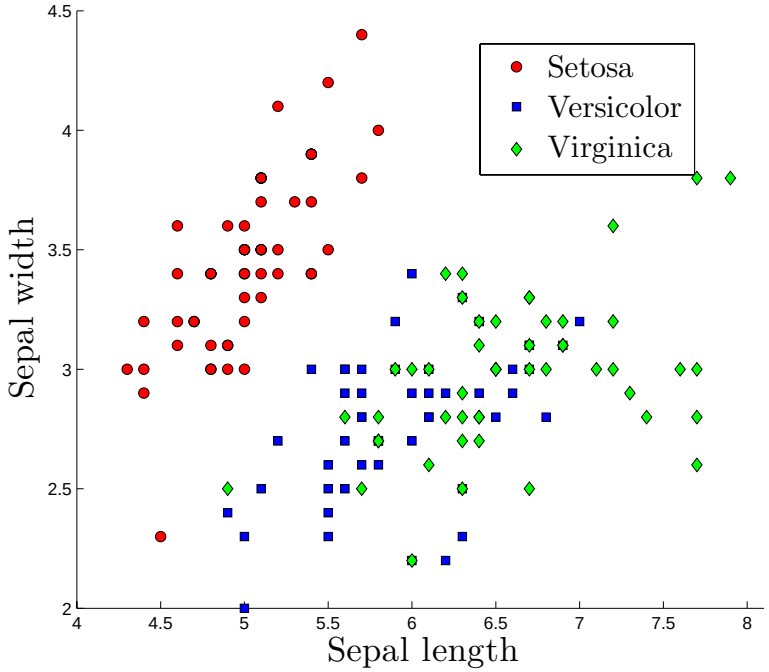

Figure 5.3: Sepal length and width

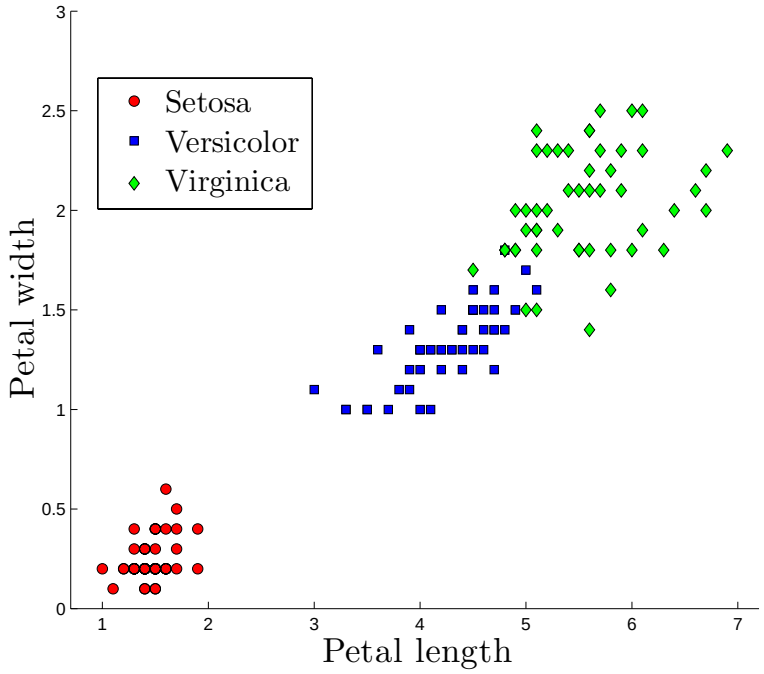

Figure 5.4: Petal length and width

5.3.2 Numerical Results

We set up the same type of experiment described in the six datasets for cancer classification using gene expression data example with a 10-fold cross-validation test. Our ℓ_1 sparse representation for classification technique was able to accurately predict the class of every test sample in the dataset with a 96.36% effective rate. In Figure 5.5 we show how an iris virginica test sample (class 3) is sparsely represented by only those samples in the training data set with the label for iris virginica class.

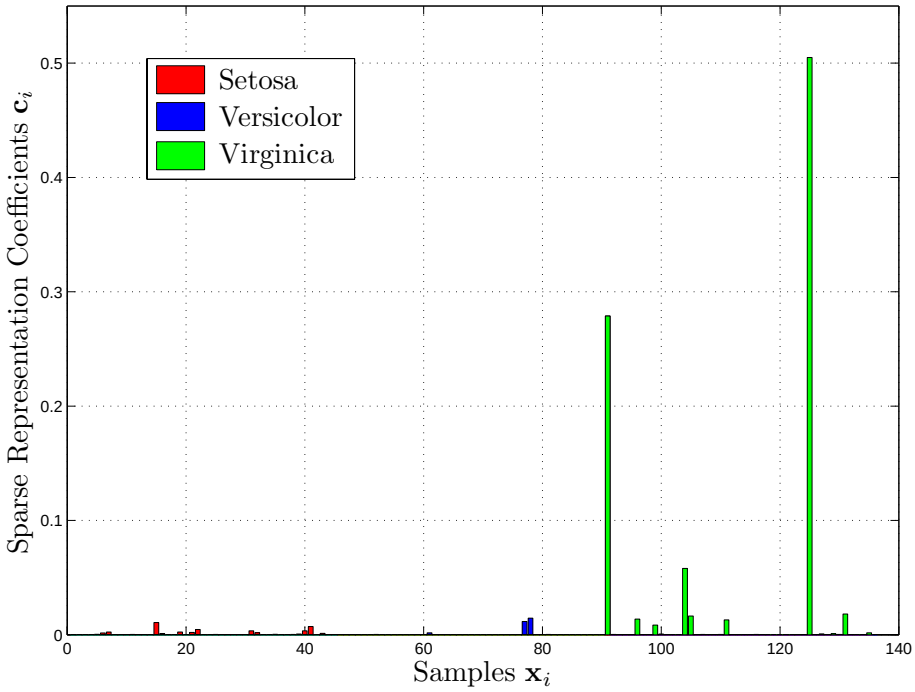

Figure 5.5: Sparse Representation of a virginica test sample $\mathbf{y}$ in the dataset `fisheriris`

Discussion The method proposed is very effective for classification in this case too, and the fact that the resulting linear system of equations for sparse representation is highly underdetermined helps to have classification results in a faster time. We take advantage of the relation between measurements and features, namely $d \ll n$, inspired by the theory of Compressed Sensing, where our PFSR algorithm solves a system of size $d \times d$ at each iteration using a conjugate gradient method and boosting the performance of the classification procedure.

Chapter 6

Future Research and Conclusions

6.1 Sparse Representation Capabilities

The results reported for the numerical experimentation described in this work show the effectiveness of the sparse representation technique proposed for solving classification problems. We have also mentioned two of the advantages of using sparse representation via ℓ_1-optimization for classification: the first one being that no model selection dependence is involved, and the second one highlighting that the lack of robustness with respect to outliers in the data can be overcome when using the ℓ_1 norm.

Our results also show that the performance of the Sparse Representation approach can be as accurate and often higher than other classification techniques such as Support Vector Machines (SVM).

6.2 Further Research

Since the process and techniques for acquiring and analyzing data advances every day at high rates, we are exposed to manage and study large amounts of data for many different scientific problems. High-dimensional data analysis definitely represents a key factor for classification problems, and the *"curses and blessings of dimensionality"* [15] can give us different ideas in order to improve our techniques. [1]

[1] Donoho [15] uses the term "curse of dimensionality" to refer to the apparent intractability of systematically searching through a high-dimensional space. "Blessings of dimensionality" is the phenomenon where statements about high-dimensional settings could be made where moderate dimensions would be complicated.

We will focus in *dimensionality reduction* via *feature selection* and *feature extraction* in general. For instance, in the case of gene expression data, several filtering techniques have already been studied. *Gene selection* is often applied to classification processes producing encouraging accuracy results [22], and reducing dramatically the number of genes used in classification. In particular, the Kruskal-Wallis non-parametric one-way ANOVA (KW) and the ratio of between classes to within class sums of square (BW) can be applied for classification purposes [32] when using gene expression data.

6.2.1 Dimensionality Reduction

We are interested in techniques for reducing the amount of features on each sample data to be considered, so the dimension of the problem can be decreased without losing important information affecting the classification rates. This will alleviate the large number of data that must be handled in some of the datasets, as well as produce quicker results when solving the sparse representation problems needed for classification.

Nowadays, there are several and effective techniques to collect data. This process accumulates data at high speed, and *preprocessing* is an important part of successful data mining and machine learning techniques. For instance, the number of genes responsible for a given type of disease may be small, so the original samples might be downsized in certain cases.

Feature Selection refers to those approaches with the goal of finding optimal subsets of the original variables (attributes) [19]. A common strategy in these type of dimensionality reduction is *filtering*. *Feature Extraction* methods aim to transform the given data into a lower dimensional space through a linear transformation, in which case Principal Component Analysis (PCA) has been widely studied, or via nonlinear transformations that have also been successfully developed in recent years [20].

In this research, for *feature selection* we plan to study how some filtering techniques can improve our classification algorithm in general. The dimensionality reduction of the data set produced by the filtering process can also help us identify which features/attributes of the original samples actually influence in the sparse representation so the selective nature of

the sparsity technique recognizes those more important variables easily, faster, and with less error. Correlation, entropy, and mutual information are three of the common filter metrics that can be tested with our classification algorithm.

In *feature extraction*, the purpose is to find an appropriate mapping of the original high-dimensional data onto a lower dimensional space. In this case, all the original features are used and the transformed attributes might be expressed as linear combinations of the original ones. We will focus in two approaches:

1. One of the advantages of the sparse representation approach, motivated by the results in the area of compressed sensing, is that even *random features* contain enough information to recover the sparse representation and hence correctly classify any test image [37]. As mentioned in the work by Wright et al. [37] the projection from the original dataset space to the feature space can be represented as a matrix $T \in \mathbb{R}^{q \times d}$ with $q \ll d$. So applying T to equation (3.3) we get

$$\hat{\mathbf{y}} = T\mathbf{y} \quad = TAv \in \mathbb{R}^d, \tag{6.1}$$

   and the dimension q of the feature space is chosen to be much smaller than d. The linear system of equation (6.1) remains underdetermined and the sparsity of the solution v, permits to find the solution to the sparse representation problem via ℓ_1 minimization. The advantage of using *random features* is that the transformation T would be independent of the training dataset one has for the specific problem.

2. We will also study the *Principal Component Analysis* (PCA) approach within our classification method. PCA applies a linear mapping of the dataset onto a lower dimensional space in such a way that the variance of the samples in the new low dimensional representation is maximized [25]. This technique uses an orthogonal transformation to convert the set of observations of possibly correlated variables into a set of uncorrelated variables called principal components. The number of principal components is less than the number of original variables. We will make use of the MATLAB specialized routines for PCA in order to test the effectiveness of this dimensionality reduction

technique. This will be added as a preprocessing step in the sparse representation approach for classification that we are proposing.

6.2.2 Sparse Representation Technique Alternative

We propose an alternate method for classification problems also based on sparse representation. For a given dataset with N different classes, the approach consists in solving N different *binary classification* problems that are independent one from the other.

Binary Classification

In Binary Classification we aim to classify the elements of a given set into two different groups characterized by a certain property. The problem of binary classification considers a training dataset $\{(\mathbf{x}_i, t_i) : i = 1, ..., n\}$, $\mathbf{x}_i \in \mathbb{R}^d$, $t_i \in \{-1, 1\}$. [2]. We describe the two different classes using a linear model of the form

$$t = \quad [\mathbf{x}^T, 1]\mathbf{w}, \tag{6.2}$$

for any given sample $\mathbf{x}$, where $\mathbf{w} \in \mathbb{R}^{d+1}$ (weight vector) characterizes the normal vector of the separating hyperplane [1]. Therefore, for each of the elements in the dataset, we have

$$t_i = \quad [\mathbf{x}_i^T, 1]\mathbf{w}, \quad i = 1, ..., n. \tag{6.3}$$

We write these n linear equations as the linear system $\mathbf{X}\mathbf{w} = \mathbf{t}$, where $\mathbf{X} \in \mathbb{R}^{n\times(d+1)}$ is the matrix whose i-th row is given by $[\mathbf{x}_i^T, 1]$. The vector $\mathbf{t}$ contains all the different labels t_i for each sample. A sparse classifier aims to differentiate between classes, identifying a small number of relevant features. For a sparse separating hyperplane, this means that the weight vector $\mathbf{w}$ has few nonzero elements [1].

If we assume that $n < d$, the system $\mathbf{X}\mathbf{w} = \mathbf{t}$, is underdetermined and we look for the

[2]The choice of labels -1,1 instead of 1,2 is merely for convenience in the formulation of the problem

solution of the ℓ_1-minimization problem

$$\begin{aligned} \min \quad & \|\mathbf{w}\|_1 \\ \text{s.t} \quad & \mathbf{X}\mathbf{w} = \mathbf{t}. \end{aligned} \tag{6.4}$$

The sparse solution $\mathbf{w}^*$ of (6.4) is then used to define the label of any other input sample $\mathbf{x}$ by computing

$$t = \quad \text{sgn}\left([\mathbf{x}^T, 1]\mathbf{w}^*\right), \tag{6.5}$$

where sgn is the sign function. Now, if $\mathbf{x}$ belongs to the class with label 1, then the sign is positive, otherwise is negative.

Multicategory Classification

Now we describe a multicategory classification technique using the ideas behind binary classification, based on the one-versus-rest (OVR) approach for support vector machines and linear classifiers [1]. The idea consists in solving a series of binary classification subproblems, in order to obtain the correct label t for a given test sample $\mathbf{x}$.

Consider the multicategory dataset $\{(\mathbf{x}_i, t_i) : i = 1, ..., n\},\ \mathbf{x}_i \in \mathbb{R}^d,\ t_i \in \{1, 2, ..., N\}$, with $\mathbf{x}_i \in \mathbb{R}^d$ and N as the number of categories. For each class k, we determine a binary classifier, separating class k from the rest of the classes. Therefore we define N linear models of the form:

$$\mathbf{t}_k \quad = \mathbf{X}\mathbf{w}_k, \quad k = 1, ..., N \tag{6.6}$$

where the labels vector $\mathbf{t}_k$ is constructed by changing to 1 all of the labels of samples belonging to class k and setting the rest to -1. Therefore, the matrix $\mathbf{X} \in \mathbb{R}^{n\times(d+1)}$ has $[\mathbf{x}_i^T, 1]^T$, as the i-th row. as the i-th row. In the same way as in the binary case described previously, we can look for the solution of the underdetermined system of equations (6.6) by solving the N

constrained minimization problems:

$$\begin{aligned} \min \quad & \|\mathbf{w}_k\|_1 \\ \text{s.t} \quad & \mathbf{X}\mathbf{w}_k = \mathbf{t}_k, \quad k = 1, ..., N. \end{aligned} \tag{6.7}$$

Notice that the subproblems of the form (6.7) are independent, and therefore a parallel implementation of this approach can be studied. Different processors can look for the solution of problem (6.7) in an independent manner, since all we will need for the classification process are the different N solutions $\mathbf{w}_k$.

Once we have computed the N different solution vectors to problem (6.7), we determine the label of any given test sample $\mathbf{x}$ by computing

$$\hat{\mathbf{t}}_k \quad = \arg\max_{k=1,...,N} \left\{ [\mathbf{x}^T, 1]\mathbf{w}_k \right\}. \tag{6.8}$$

In this manner, an original multicategory classification problem is solved by a series of binary classification subproblems.

References

[1] S. Abe, "Support Vector Machines for Pattern Classification", *Springer Verlag London*, Advances in Pattern Recognition, 2005

[2] M. Argáez, C. Ramirez, R. Sanchez, "An ℓ_1-algorithm for underdetermined systems and applications", *To appear in North American Fuzzy Information Processing Society* , IEEE Conference Proceedings, 2011.

[3] S. Boyd and L. Vandenberghe, Convex Optimization, Cambridge, U.K.: Cambridge University Press, 2004.

[4] E. Candès, "Compressive sampling", *Proceedings of the International Congress of Mathematicians*, Madrid, Spain, 2006.

[5] E. Candès, "The restricted isometry property and its implications for compressed sensing", *Compte Rendus de l'Academie des Sciences*, Paris, Series I, 346, pp. 589-592, 2008.

[6] E. Candès, T. Tao, "Decoding by linear programming", *IEEE Trans. Inform. Theory 51* (12) (December 2005) 42034215.

[7] E. Candès, J. Romberg and T. Tao, "Robust Uncertainty Principles: Exact Signal Reconstruction from Highly Incomplete Frequency Information", *IEEE Transactions on Information Theory*, Vol. 52, pp. 489-509, 2006.

[8] R. Chartrand, "Exact reconstruction of sparse signals via nonconvex minimization", *IEEE Signal Processing Letters*, Vol. 14, pp. 707-710, 2007.

[9] A. Cohen, W. Dahmen and R. DeVore, "Compressed sensing and best k-term approximation", *J. Amer. Math. Soc. 22* pp. 211231, 2009.

[10] C. Cortes and V. Vapnik, "Support Vector Networks ", *Machine Learning*, Vol. 20, No. 3, pp. 273-297, 1995.

[11] S. Chen, D. Donoho, M. Saunders, "Atomic Decomposition by Basis Pursuit", *SIAM Review*, Vol. 43 No 1, pp. 129-159, 2001.

[12] D. Donoho, X. Huo, "Uncertainty principles and ideal atomic decomposition", *IEEE Trans. Inform. Theory* Vol 47, pp. 2845-2862, 2001.

[13] D. Donoho, "Compressed sensing", *IEEE Transactions on Information Theory*, Vol. 52, No. 4, pp. 1289-1306, 2006.

[14] D. Donoho, "For most large underdetermined systems of linear equations, the minimal ℓ_1 solution is also the sparsest solution", *Communication on pure and applied Mathematics*, Vol. 59, No. 7, pp. 907934, 2006.

[15] D. Donoho, "High-dimensional Data Analysis: The Curses and Blessings of Dimensionality" , *International conference of mathematicians*, Paris, August 2000.

[16] M. Elad, "Sparse and redundant representations", Springer 2010.

[17] M. Figueiredo, R. Nowak, and S. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems", *IEEE Selected topics in signal processing*, Vol. 1, No. 4, pp. 586-597, 2007.

[18] R. A. Fisher, "The Use of Multiple Measurements in Taxonomic Problems", *Annals of Eugenics*, Vol. 7, pp. 179-188, 1936.

[19] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection" *Journal of Machine Learning Research*, Vol. 3, pp. 1157-1182, 2003.

[20] I. Guyon, S. Gunn, M. Nikravesh and L. Zadeh, "Feature Extraction, Foundations and Applications" *Series Studies in Fuzziness and Soft Computing*, Physica-Verlag, Springer, 2006.

[21] E. Hale, W. Yin, and Y. Zhang, "A fixed-point continuation method for ℓ_1-regularized minimization with applications to compresses sensing", Technical Report TR07-07, Department of Computational and Applied Mathematics, Rice University, Houston, TX, USA, 2007.

[22] X. Hang, F. Wu, "Sparse Representation for Classification of Tumors Using Gene Expression Data", *Journal of Biomedicine and Biotechnology*, Vol. 2009, Article ID 403689.

[23] X. Hang, F. Wu, "ℓ_1 least square for cancer diagnosis using gene expression data", *Journal of Computer Science and System Biology*, Vol. 2009, pp. 167-173.

[24] S. Jokar, M. Pfetsch, "Exact and approximate sparse solutions of underdetermined linear equations", *SIAM Journal on Scientific Computing*, Vol. 31, No. 1, pp. 23-44, 2008.

[25] I.T Jolliffe, "Principal Component Analysis", *Springer Series in Statistics*, 2nd ed., Springer, NY, 2002.

[26] S. Kim, K. Koh, M. Lustig, S. Boyd, D. Gorinvesky, "An interior-point method for large-scale ℓ_1-regularized least squares", *IEEE Selected topics in signal processing*, Vol. 1, No. 4, pp. 606-617, 2007.

[27] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection", *Proceedings of International Joint Conference on AI* , pp 1137-1145, 1995.

[28] S. Mallat and Z. Zhang, Matching pursuits with time-frequency dictionaries", *IEEE Trans. on Signal Processing*, Vol. 41, pp. 33973415, 1993.

[29] C. Miosso, R. Von-Borries, M. Argáez, L. Velázquez, C. Quintero, C. Potes, "Compressed sensing reconstruction with prior information using penalized reweighted normal equations", *IEEE Transactions on Signal Processing*, Vol. 52, No. 4, pp. 1289-1306, 2009.

[30] B. K. Natarajan, "Sparse approximate solutions to linear systems", *SIAM Journal on computing*, Vol. 24, pp.227-234, 1995.

[31] Y. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition", *27th Annual Asilomar Conference on Signals, Systems, and Computers*, 1993.

[32] Y. Saeys, I. Inza, and P. Larranaga, "A review of feature selection techniques in bioinformatics", *Bioinformatics*, Vol.23, No. 19, pp. 25072517, 2007.

[33] A. Statnikov, C.F Aliferis, I. Tsamardinos, D. Hardin and S. Levy, "A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis", *Bioinformatics*, Vol. 21, No. 5, pp. 631-643, 2005.

[34] H. L. Taylor, S. C. Banks, J. F. McCoy. "Deconvolution with the ℓ_1 norm", *Geophysics*, Vol. 44, pp. 39-52, 1979.

[35] J. Tropp, "Greed is good: Algorithmic results for sparse approximation", *IEEE Transactions on Information Theory*, Vol. 50, No. 10, pp. 22312242, 2004.

[36] L. Wang (Editor), "Support Vector Machines: Theory and Applications", *Springer-Verlag Berlin*, 2005.

[37] J. Wright, Y. Yang, A. Ganesh, S. S. Sastry and Y. Ma, "Robust Face Recognition via Sparse Representation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 61, No. 2, pp 210-227, 2009.

[38] W. Yin, Y. Zhang, "Extracting salient features from less data via ℓ_1-minimization", *SIAG/Optimization Views and News* Vol. 9, 2008.

Curriculum Vitae

Reinaldo Sanchez Arias was born in Cali, Colombia on February 17, 1986. He is the first son of Reinaldo Sanchez and Alid Arias. His interest in mathematics and science dates back to his early childhood, inspired by his father who was a mathematician, and the constant search for knowledge inculcated by his mother. Reinaldo is the brother of Juan Camilo Sanchez Arias, a current Medical School student. In Spring 2008 he obtained his Bachelor's title majoring in Mathematics from Universidad del Valle in Cali, Colombia, under the direction of Dr. Jairo Duque. Later that year, he met Dr. Leticia Veláquez and Dr. Miguel Argáez during their visit to Colombia for attending a conference. Advised by them, he deciced to pursuit a doctoral degree in Computational Science in the United States. In Fall 2008 he arrived at El Paso, and started his doctoral studies at The University of Texas at El Paso. During his first semester he was a teaching assistant at the Mathematical Sciences Department, and after that became a research assistant under the direction of principal investigators Drs. Argáez and Veláquez, in a work with the Army High Performance Computing Research Center (AHPCRC) funded by an ARL grant. He is currently a member of the Society of Industrial and Applied Mathematics (SIAM).

The University of Texas at El Paso
Program in Computational Science
500 West University Ave. Bell Hall
El Paso, Texas 79968-0514
rsanchezarias@miners.utep.edu
reinaldosanar@gmail.com

Home address: 3500 Sun Bowl Dr No 86 El Paso, Texas 79912-4927