Kernel l1-minimization: Application to Kernel Sparse Representation based Classification

Anupriya Gogna and Angshul Majumdar

Indraprastha Institute of Information Technology, Delhi, India
anupriyag@iiitd.ac.in and angshul@iiitd.ac.in

Abstract. The sparse representation based classification (SRC) was initially proposed for face recognition problems. However, SRC was found to excel in a variety of classification tasks. There have been many extensions to SRC, of which group SRC, kernel SRC being the prominent ones. Prior methods in kernel SRC used greedy methods like Orthogonal Matching Pursuit (OMP). It is well known that for solving a sparse recovery problem, both in theory and in practice, l_1-minimization is a better approach compared to OMP. The standard l_1-minimization is a solved problem. For the first time in this work, we propose a technique for Kernel l_1-minimization. Through simulation results we show that our proposed method outperforms prior kernelised greedy sparse recovery techniques.

Keywords: l_1-minimization, kernel machine, sparse classification

1 Introduction

In sparse recovery, the problem is to find a solution to the linear inverse problem

$$y = Ax + n \tag{1}$$

where, y is the observation, A is the system matrix, x is the solution and n is the noise assumed to be Normally distributed. The solution x is sparse, i.e. it is assumed to be have only 'k' non-zeroes. Such a problem arises in machine learning and signal processing; in fact there is a branch of signal processing called Compressed Sensing that evolves around the solution of such problems.

The exact solution to (1) is NP hard [1] and is expressed as,

$$\min_x \|y - Ax\|_2^2 \text{ such that } \|x\|_0 = k \tag{2}$$

Here the l_0-norm (not exactly a norm in the strictest sense of the term) simply counts the number of non-zeroes in the vector. There are two approaches to solve (2) – the first one is a greedy approach, where the support of x is iteratively detected and the corresponding values are estimated. The orthogonal matching pursuit (OMP) [2] is the most popular greedy technique. There are several extensions to the basic OMP approach like the stagewise orthogonal matching pursuit and the CoSamp.

However OMP is fraught with several limitations. First, the guarantees are only probabilistic; besides several strict assumptions need to be made regarding the nature of the system matrix 'A' in order for OMP to succeed (theoretically). Both in theory and in practice, a much better way to solve the sparse recovery problem is to relax the NP hard l_0-minimization problem by its closest convex surrogate the l_1-norm. This is expressed as,

$$\min_x \|y - Ax\|_2^2 \text{ such that } \|x\|_1 \leq \tau \tag{3}$$

This formulation was first proposed in Tibshirani's paper on LASSO [3]. Although convex, this (3) is a constrained optimization problem and is hard to solve; hence in [3], the unconstrained version was solved instead.

$$\min_x \|y - Ax\|_2^2 + \lambda \|x\|_1 \tag{4}$$

This is a quadratic programming problem and can be solved efficiently using iterative soft thresholding [4].

This formulation (4) is a typical linear regression problem. In this work, we are interested in kernel regression / classification problems. A typical non-linear regression is expressed as,

$$y = \varphi(A)x + n \tag{5}$$

Here the output is expressed as a linear combination of a non-linear system matrix. The Tikhonov regularized solution of (5) has a closed form solution via the kernel trick.

In this work we are interested in solving problems where a non-linear combination of the output can be expressed as linear combination of non-linear inputs, i.e.,

$$\varphi(y) = \varphi(A)x + n \tag{6}$$

Here both the input (A) and the output (y) are of non-linear forms. Such a problem does not arise in regression, where the problem is to predict the output (5) and not a non-linear version of the output (6), but it does arise in kernel sparse representation based classification [5-7]; these studies were based on modifying the OMP algorithm. Issues arising in the linear version of the OMP also persists in the non-linear version. A better approach would be modify the l_1-minimization algorithm to support kernels. This is the topic of this paper.

2 Brief review on sparse representation based classification

The SRC assumes that the training samples of a particular class approximately form a linear basis for a new test sample belonging to the same class. One can write the aforesaid assumption formally. If x_{test} is the test sample belonging to the k^{th} class then,

$$x_{test} = \alpha_{c,1}x_{c,1} + \alpha_{c,2}x_{c,2} + ... + \alpha_{c,n_k}x_{c,n_k} + n \tag{7}$$

where $x_{c,i}$ are the training samples and η is the approximation error.

In a classification problem, the training samples and their class labels are provided. The task is to assign the given test sample with the correct class label. This requires finding the coefficients $\alpha_{c,i}$ in equation (8). Equation (8) expresses the assumption in terms of the training samples of a single class. Alternately, it can be expressed in terms of all the training samples so that

$$x_{test} = X\alpha + n \tag{8}$$

where $X = [x_{1,1} | ... | x_{n,1} | ... | x_{c,1} | ... | x_{c,n_c} | ...x_{C,1} | ... | x_{C,n_C}]$ and $\alpha = [\alpha_{1,1}...\alpha_{1,n_1}...\alpha_{c,1}...\alpha_{c,n_c}...\alpha_{C,1}...\alpha_{C,n_C}]^T$.

According to the SRC assumption, only those α's corresponding to the correct class will be non-zeroes. The rest are all zeroes. In other words, α will be sparse. Therefore, one needs to solve the inverse problem (8) with sparsity constraints on the solution. This is formulated as:

$$\min_{\alpha} \left\| x_{test} - X\alpha \right\|_2^2 + \lambda \left\| x \right\|_1 \tag{9}$$

Once (9) is solved, the representative sample for every class is computed: $x_{rep}(c) = \sum_{j=1}^{n_c} \alpha_{c,j} x_{c,j}$. It is assumed that the test sample will look very similar to the representative sample of the correct class and will look very similar, hence the residual $\varepsilon(c) = \left\| x_{test} - x_{rep}(c) \right\|_2^2$, will be the least for the correct class. Therefore once the residual for every class is obtained, the test sample is assigned to the class having the minimum residual.

There are several extensions to the basic SRC; in its pristine form it is an unsupervised approach – it does not utilize information about the class labels. In [8-10] it was argued that α is supposed to be non-zero for all training samples corresponding to the correct class. The SRC assumes that the training samples for the correct class will be automatically selected by imposing the sparsity inducing l_1-norm; it does not explicitly impose the constraint that if one class is selected, all the training samples corresponding to that class should have corresponding non-zero values in α. It was claimed in [2-4] that better recovery can be obtained if selection of all the training samples within the class is enforced. This was achieved by employing a supervised $l_{2,1}$-norm instead of the l_1-norm.

$$\min_{\alpha} \left\| x_{test} - X\alpha \right\|_2^2 + \lambda \left\| \alpha \right\|_{2,1} \tag{10}$$

where the mixed norm is defined as $\left\| \alpha \right\|_{2,1} = \sum_{k=1}^{c} \left\| \alpha_k \right\|_2$.

The inner l_2-norm enforces selection of all the training samples within the class, but the sum-of- l_2-norm over the classes acts as an l_1-norm over the selection of classes and

selects very few classes. The block sparsity promoting $l_{2,1}$-norm ensures that if a class is selected, ALL the training samples within the class are used to represent the test sample.

A recent addition to the suite of sparse representation based classifiers is the group sparse representation based classifier [11]. This is a generalization of all of the above that can handle multiple kinds of datasets (like multi-modal biometrics) and multiple types of features in a single framework.

Several studies independently proposed the Kernel Sparse Representation based Classification (KSRC) approach [5-7]. KSRC is a simple extension of the SRC using the Kernel trick. The assumption here is that the non-linear function of the test-sample can be represented as a linear combination of the non-linear functions of the training samples, i.e.

$$\phi(x_{test}) = \phi(X)\alpha + n \tag{11}$$

Here $\phi(.)$ represents a non-linear function. As mentioned before, the prior studies solved this problems by modifying the Orthogonal Matching Pursuit.

3 Proposed Approach

3.1 l_1-minimization

First we will study the vanilla implementation of iterative soft thresholding algorithm. The goal is to solve (9). The derivation can be followed from [12]. The algorithm is given as follows.

Initialize: $x_0 = \min_x \|y - Ax\|_2^2$

Continue till convergence

Landweber Iteration – $b = x_{k-1} + \frac{1}{a} A^T (y - Ax_{k-1})$

Soft thresholding – $x_k = signum(b) \max\left(0, |b| - \frac{\lambda}{2a}\right)$

Here the step-size 'a' is the maximum Eigenvalue of A^TA. The iterations converge when the objective function or the value of x does not change significantly over successive iterations.

With a slight modification, one can have an iterative hard thresholding algorithm [13]. The only difference between the hard and soft thresholding algorithm is the thresholding step. In the hard thresholding only those values are kept that are greater than a pre-defined threshold. Such an algorithm is supposed to approximately solve the l0-minimization problem. In practice, it does not yield very good results.

3.2 Kernel l_1-minimization

Here we are interested in solving (6). We repeat it for the sake of convenience.

$$\varphi(y) = \varphi(A)x + n$$

If we write down the soft thresholding algorithm for the same, we get

Initialize: $x_0 = \min_x \|\varphi(y) - \varphi(A)x\|_2^2$

Continue till convergence

Landweber Iteration – $b = x_{k-1} + \frac{1}{a}\varphi(A)^T(\varphi(y) - \varphi(A)x_{k-1})$

Soft thresholding – $x_k = signum(b)\max\left(0, |b| - \frac{\lambda}{2a}\right)$

First, let us look at the initialization. The normal equations are of the form,

$$\left(\varphi(A)^T\varphi(A)\right)x_0 = \varphi(A)^T\varphi(y) \tag{12}$$

One can easily identify the kernels: $\mathrm{K}(A,A) = \varphi(A)^T\varphi(A)$ and $\mathrm{K}(A,y) = \varphi(A)^T\varphi(y)$. With the kernel trick, we can express (12) as,

$$\mathrm{K}(A,A)x_0 = \mathrm{K}(A,Y) \Rightarrow x_0 = \mathrm{K}(A,A)^{-1}\mathrm{K}(A,Y) \tag{13}$$

The inversion is guaranteed by the positive definiteness of the kernel.

Now, we look at the Landweber iteration step. One can easily see that, it can be expressed as $b = x_{k-1} + \frac{1}{a}\left(\varphi(A)^T\varphi(y) - \varphi(A)^T\varphi(A)x_{k-1}\right)$. Identifying the kernels, this is represented as,

$$b = x_{k-1} + \frac{1}{a}\left(\mathrm{K}(A,\mathrm{y}) - \mathrm{K}(A,A)x_{k-1}\right) \tag{14}$$

The soft-thresholding step does not require any change.

4 Experimental Evaluation

Once the sparse recovery problem is solved, the residual error needs to be expressed in terms of kernels. This is easily done (keeping the same notation for SRC as before).

$$\varepsilon(c) = \left\| \varphi(x_{test}) - \varphi(x_{rep}(c)) \right\|_2^2$$
$$= \left(\varphi(x_{test}) - \varphi(x_{rep}(c)) \right)^T \left(\varphi(x_{test}) - \varphi(x_{rep}(c)) \right)$$
$$= \varphi(x_{test})^T \varphi(x_{test}) + \varphi(x_{rep}(c))^T \varphi(x_{rep}(c)) - \varphi(x_{test})^T \varphi(x_{rep}(c)) - \varphi(x_{rep}(c))^T \varphi(x_{test})$$
$$= \mathrm{K}(x_{test}, x_{test}) + \mathrm{K}(x_{rep}(c), x_{rep}(c)) - 2\mathrm{K}(x_{rep}(c), x_{test})$$

4.1 Results on Benchmark Classification Tasks

In [5] the KSRC was tested on benchmark datasets from the UCI Machine Learning repository. We use the same datasets and follow the same experimental protocol here. No feature extraction or dimensionality reduction was applied on these datasets. We compare the KSRC formulation in [5] with ours; both of them use an RBF kernel. To benchmark, the results from SRC and SVM are also shown.

Table 1. Error Rate % on Benchmark Classification Tasks

Dataset	SVM	SRC	KSRC [5]	Proposed
Breast	4.09	54.57	5.78	**4.09**
Glass	30.29	33.77	32.46	**30.52**
Heart	18.7	22.8	23.8	**20.35**
Hepatitis	34.51	45.49	38.04	**34.51**
Ionosphere	5.38	8.21	13.42	**6.23**
Iris	5.63	20	**4.79**	**4.79**
Liver	31.05	35.88	32.81	**31.05**
Musk	6.84	14.75	10	**8.29**
Pima	24.67	34	30.4	**26.8**
Sonar	12.9	23.48	12.46	**12.09**
Soy	4.35	11.65	**3.41**	**3.41**
Vehicle	18.5	**18.72**	22.96	**18.72**
Vote	5.59	7.11	7.04	**5.92**
Wdbc	2.7	6.4	3.44	**3.4**
Wine	0.86	2.41	1.55	**1.55**
Wpbd	20.31	26.46	26	**22.52**

One can see from the table that in most cases SVM outperforms the SRC based methods. However comparison within SRC and its variants show that our method always yields the best results. The prior formulation of KSRC [5] had a naive implementation, therefore even with the kernel trick it was unable to improve upon the SRC which benefitted from more sophisticated optimization algorithm. In this work, our proposed method enjoys the dual benefit of non-linear kernels and better optimization; hence the results always outperform prior techniques.

4.2 Results on Hyperspectral Image Classification

In [6] it was shown that the KSRC (based on KOMP) performed exceptionally well for hyperspectral image classification problems. In this work, we show that our proposed method improves upon the prior work.

We evaluate our proposed Hyperspectral Image Classification on – 1. Indian Pines dataset which has 200 spectral reflectance bands after removing bands covering the region of water absorption and 145*145 pixels of sixteen categories; and, 2. Pavia University dataset which has 103 bands of 340*610 pixels of nine categories. The background i.e Class 0 was excluded from the second dataset. For each dataset, we randomly select 10% of the labelled data as training set and rest as testing set. Input consists of raw data of all the spectral channels pixel-wise.

In [6] a thorough study had been carried out by comparing KOMP based KSRC with SVM, SRC, KSRC etc. In [6] it was claimed that their KOMP based technique outperforms others. Therefore, in this work, we only need to show that our proposed method outperforms [6]. The results can be visualised from figure 1. One can see that our proposed method yields better results compared to the prior approach.

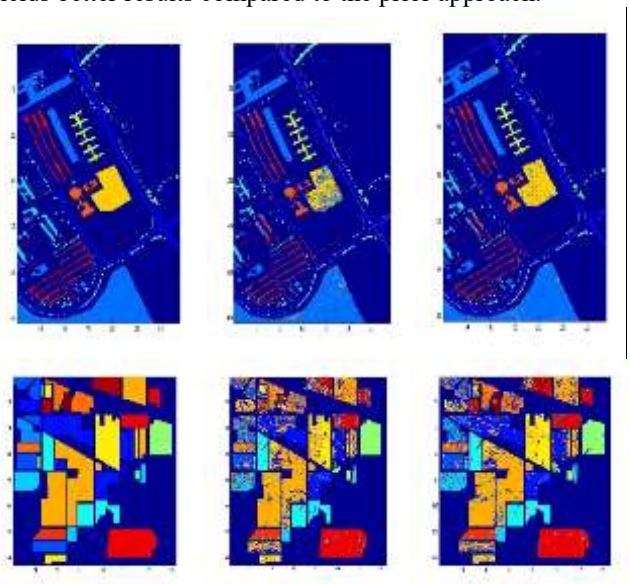

Fig. 1. Top: Pavia University; Bottom: Indian Pines. Left to Right: Groundtruth, KOMP based KSRC [6] and Proposed.

5 Conclusion

In this work, we propose a technique for solving a kernel l_1-minimization problem. To the best of our knowledge this is the first work on this topic. We start with the vanilla implementation of the l_1-minimization problem via iterative soft thresholding and show how the kernel trick can be employed on it.

The proposed kernel l_1-minimization problem is employed here to solve the kernel sparse representation based classification problem. We have compared our proposed technique on two implementations of the same – [5] and [6]. Experiments on benchmark classification datasets from the UCI machine learning repository show that our method is better than [5]. Evaluation of our proposed technique with [6] for hyperspectral imaging problems show that our method is also better than the kernel OMP based implementation [6].

In the future, we would like to extend this formulation to solve other variants of SRC like group sparse classification, robust sparse classification and robust group sparse representation based classification. We will also compare the proposed methods on a host of other real life problems.

6 References

1. B. Natarajan, "Sparse approximate solutions to linear systems", SIAM Journal on Computing, Vol. 24, 227-234, 1995.
2. J. Tropp, A. C. Gilbert, M. Strauss, "Algorithms for simultaneous sparse approximations ; Part I : Greedy pursuit", Signal Processing, Special Issue on Sparse approximations in signal and image processing, Vol.86, pp 572–588, 2006
3. R. Tibshirani, "Regression shrinkage and selection via the lasso", J. Royal. Statist. Soc B., Vol. 58 (1), pp. 267-288, 1996.
4. I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint", Communications in Pure and Applied Mathematics, Vol. 57(11):1413–1457, 2004.
5. L. Zhang, W.-D. Zhou, P.-C. Chang, J. Liu, Z. Yan, T. Wang, F.-Z. Li, "Kernel sparse representation-based classifier", IEEE Transactions on Signal Processing, Vol. 60 (4), pp. 1684–1695, 2012.
6. Y. Chen, N. Nasrabadi, T. Tran, "Hyperspectral image classification via kernel sparse representation", IEEE Transactions on Geoscience and Remote Sensing, Vol. 51 (1), pp. 217–231, 2013.
7. J. Yin, Z. Liu, Z. Jin, W. Yang, "Kernel sparse representation based classification", Neurocomputing, Vol. 77 (1), pp. 120 – 128, 2012.
8. A. Majumdar and R. K. Ward, "Robust Classifiers for Data Reduced via Random Projections", IEEE Transactions on Systems, Man, and Cybernetics, Part B, Vol. 40 (5), pp. 1359 - 1371, 2010.
9. X. T. Yuan and X. Liu and S. Yan, "Visual Classification With Multitask Joint Sparse Representation", IEEE Transactions on Image Processing, Vol. 21 (10), pp. 4349-4360, 2012
10. E. Elhamifar and R. Vidal, "Robust Classification using Structured Sparse Representation", IEEE CVPR, 2011.
11. G. Goswami, P. Mittal, A. Majumdar, R. Singh and M. Vatsa, "Group Sparse Representation based Classification for Multi-feature Multimodal Biometrics", Information Fusion.
12. Sparse signal restoration: cnx.org/content/m32168/
13. K. Bredies and D. A. Lorenz. "Iterated hard shrinkage for minimization problems with sparsity constraints", SIAM Journal on Scientific Computing, Vol. 30 (2), pp.657–683, 2008.
14. J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 31 (2), pp. 210-227, 2009.