RC24801 (W0906-038) June 9, 2009
Computer Science

IBM Research Report

Isometry-enforcing Data Transformations for Improving Sparse Model Learning

Avishy Carmi
Signal Processing and Communications Laboratory
University of Cambridge
UK

Irina Rish, Guillermo Cecchi, Dimitri Kanevsky, Bhuvana Ramabhadran
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598
USA

Research Division
Almaden - Austin - Beijing - Cambridge - Haifa - India - T. J. Watson - Tokyo - Zurich

Isometry-enforcing Data Transformations for Improving Sparse Model Learning

Avishy Carmi[1], Irina Rish[2], Guillermo Cecchi[2], Dimitri Kanevsky[2] and Bhuvana Ramabhadran[2]

[1] Signal Processing and Communications Laboratory, University of Cambridge, UK.
[2] IBM T. J. Watson, Yorktown, NY.

Abstract. Imposing sparsity constraints (such as l_1-regularization) on the model parameters is a practical and efficient way of handling very high-dimensional data, which also yields interpretable models due to embedded feature-selection. Compressed sensing (CS) theory provides guarantees on the quality of sparse signal (in our case, model) reconstruction that relies on the so-called restricted isometry property (RIP) of the sensing (design) matrices. This, however, cannot be guaranteed as these matrices form a subset of the underlying data set. Nevertheless, as we show, one can find a distance-preserving linear transformation of the data such that any transformed subspace of the data satisfies the RIP at some level. We demonstrate the effects of such RIP-enforcing data transformation on sparse learning methods such as sparse and compressed Random Fields, as well as sparse regression (LASSO), in the context of classifying mental states based on fMRI data.

1 Introduction

Sparse modeling techniques, such as sparse regression, sparse graphical models, and sparse component analysis, just to name a few, became a popular topic of research in the recent few years. The rise of interest in sparse learning is motivated by an increasing number of practical applications where the dimensionality of the problem is significantly larger than then the number of available samples (e.g., in bio-informatics and medical imaging), and thus an efficient regularization is required. A popular l_1-norm regularization seems to prevent overfitting quite well while yielding tractable optimization problems. Moreover, sparsity-enforcing l_1-regularization (and in general, l_q-regularization with $0 \leq q \leq 1$) facilitates "embedded" feature-selection which has an important benefit of interpretability, the property of a model that is often as important as (or even more important than) its predictive accuracy.

This paper proposes a novel approach to improving the performance of sparse classification models, such as sparse Random Fields, Markov Networks, and LASSO-based classifiers, via linear transformation of the data that aims at enforcing a property known as the Restricted Isometry Property (RIP) in Compressed Sensing (CS) literature [1, 2]. RIP is an important condition on the design (data) matrix that facilitates accurate recovery of sparse signals (i.e.,

models). In this paper, we provide both theoretical analysis of the proposed transformation, proving that it indeed enforces the RIP at certain level on an arbitrary data matrix, and empirical evaluation on real-life datasets from medical imaging domain, specifically, from functional MRI (fMRI) studies.

Moreover, our approach can be viewed as a semi-supervised technique, since it assumes availability of unlabeled test data at the training phase. A combination of such transductive setting with the RIP-enforcing transformation can lead to dramatic improvements in prediction accuracy of certain sparse methods, such as compressed Random Fields, and often improve the performance of other sparse methods such as sparse Markov networks (sparse MRFs) and sparse linear regression (LASSO).

2 Problem Formulation

Let $\{v(1), \ldots, v(n)\}$ be a set of features (predictive variables), and let Y be the class label (response variable) taking values $\{+1, -1\}$[3]. We assume a given set of m training data samples, and a set of l test samples. We will use $V = [v(1), \ldots, v(n)]$, $v(i) \in \mathbb{R}^m$ to denote the training set, and $V' = [v'(1), \ldots, v'(n)]$, $v'(i) \in \mathbb{R}^l$ to denote the test set. Given (V, Y) and V', we are interested in predicting the response vector Y' associated with V'. It is important to note that we will focus on the *transductive* setting rather than the classification setting, i.e. we will assume that unlabeled test data can be used by a learner at the training phase.

3 Sparse and Compressible Model Learning

In many practical applications (e.g., computational biology, medical imaging, etc.), the number of features n can be much larger than the number of training samples m. Thus, some form of regularization is necessary in order to prevent overfitting. One common approach is to follow Occam's razor principle and seek the simplest model that adequately describes the data. Simple models are often more interpretable as well.

Suppose for a moment that Y is a real-valued response vector. Following the usual regression set-up one may write

$$Y = V\beta + \zeta \tag{1}$$

where ζ and $\beta \in \mathbb{R}^n$ denote a noise random vector and the model parameters, respectively. It can be recognized that the ordinary least-squares (OLS) solution is infeasible in case of $n > m$, owing to the rank deficiency of $V^T V$. Nevertheless, an accurate and sometimes even exact solution can be obtained by assuming that β is sufficiently sparse, i.e., most of its entries vanish.

[3] In this work we restrict ourselves to the binary classification, although our approach can be easily extended to the multi-class case.

Definition 1. *A vector x is said to be s-sparse if the number of its non-zero entries equals s, that is $\| x \|_0 := \#\{\text{supp}(x)\} = s$, where $\text{supp}(x)$ denotes the support of x, and $\| x \|_0$ is called l_0-norm of x.*

It has already been shown that in the noiseless case where $Y = V\beta$, a unique and exact solution exists assuming β is at most m-sparse [1, 3]. However, the problem of finding the sparsest model β is known to be NP-hard in general. Its noisy version which is associated with (1) is given by

$$\min_{\hat{\beta}} \| \hat{\beta} \|_0 \ \text{ s.t. } \ \| Y - V\hat{\beta} \|_2 \leq \epsilon \tag{2}$$

An alternative approach utilizes an l_1 relaxation for formulating a convex problem instead of (2). The LASSO [4], for instance, solves a dual problem of the form

$$\min_{\hat{\beta}} \| Y - V\hat{\beta} \|_2 \ \text{ s.t. } \ \| \hat{\beta} \|_1 \leq t \tag{3}$$

This formulation promotes sparse solutions of which the sparseness degree (i.e., $\#\{\text{supp}(\hat{\beta})\}$) depends on the tuning parameter t.

3.1 Compressed Learning

A remarkable result that has emerged from the new theory of compressed sensing (CS) [1–3] shows that under certain conditions on the sensing matrix V, the l_1-relaxation yields the exact solution for the noiseless case in the sense that it coincides with the solution of the NP-hard problem mentioned above. In the noisy case, the same conditions ensure accurate recovery up to a certain level. We will wait a few sections before elaborating on such conditions and focus now on the formulation of the CS problem.

The noisy CS problem replaces the l_0 norm in (2) with the l_1 norm

$$\min_{\hat{\beta}} \| \hat{\beta} \|_1 \ \text{ s.t. } \ \| Y - V\hat{\beta} \|_2 \leq \epsilon \tag{4}$$

Following the underlying arguments of CS theory, Candes and Tao [5] suggest an estimator to the CS problem known as the Dantzig Selector (DS). The DS is essentially aimed at solving

$$\min_{\hat{\beta} \in \mathbb{R}^n} \| \hat{\beta} \|_1 \ \text{ s.t. } \ \| V^T(Y - V\hat{\beta}) \|_\infty \leq (1 + t^{-1})\sqrt{2 \log n}\sigma \tag{5}$$

where $\sigma I_{m \times m}$ denotes the standard deviation matrix of the noise ζ in (1). The DS share some similarities with the LASSO where the later is expressed in its dual form as in (4). Both estimators promote sparse solutions to the regression problem (1).

In some cases the obtained solution is not sparse but rather is compressible. A compressible vector consists of significant amount of negligible entries. The formal definition follows below.

Definition 2. *A compressible vector $x \in \mathbb{R}^n$ obeys*

$$\#\{x_i \mid |x_i| > \epsilon,\ i = 1, \ldots, n\} << n \tag{6}$$

for some sufficiently small $\epsilon > 0$.

In a recent work [6] it has been shown how a compressible solution to a stochastic CS (SCS) problem can be recursively obtained using the well-known Kalman filter (KF). The class of algorithms, which was termed CSKF in [6, 7], is aimed at solving a generalized problem of the form

$$\min_{\hat{\beta}} E_{\beta|Y}\left[\| \beta - \hat{\beta} \|_2\right] \text{ s.t. } \| \hat{\beta} \|_1 \leq t \tag{7}$$

where $E_{A|B}[\cdot]$ denotes the conditional expectation operator. In contrary to the previous formulations, here β is assumed to be a random vector. If it is further assumed that this vector has a stationary distribution $p(\beta)$, then the above problem can be written in a similar fashion to the LASSO objective in (3)

$$\min_{\hat{\beta}} E_{\beta|Y}\left[\| Y - V\hat{\beta} \|_2\right] \text{ s.t. } \| \hat{\beta} \|_1 \leq t \tag{8}$$

It is worthwhile mentioning that the CSKF method is not directly tuned by the parameter t. The technique used to regulate the compressibility degree of the obtained solution is based on replacing the l_1 constraint with the pseudo-measurement [6]

$$\| \hat{\beta} \|_1 - e = 0 \tag{9}$$

where $e \sim \mathcal{N}(0, R_e)$. Thus, instead of tuning a deterministic threshold parameter we set the variance R_e of the random variable e.

In virtue of its KF mechanism, the CSKF can process each sample of V individually. A single iteration of this algorithm is given in the pseudo-code in Algorithm 1.

4 Compressed Random Field Classifiers

Lets get back to our classification problem in which the response space consists of only two outcomes. All the above mentioned algorithms can be used to learn a sparse or compressible model $\hat{\beta}$ which then can be used to predict the response Y' for the unlabeled test data V' by

$$Y' = \text{sign}(V'\hat{\beta}) \tag{12}$$

Other classification methods construct a typical model for each class. The predicted class is then taken as the one of which the model best "explains" the test data. The base classifiers that are derived in this work learn a sparse or compressible random field model for each class. Such models have the advantage of avoiding overfitting by eliminating insignificant statistical relations between

Algorithm 1 CSKF

1: Set P_0 as the prior covariance of β. Set R as the covariance of the noise ζ. Let $\hat{\beta}_0 = 0$. Let also V_k be the k-th sample in V.

$$K_k = P_k V_k^T \left(V_k P_k V_k^T + R \right)^{-1} \tag{10a}$$

$$\hat{\beta}_{k+1} = \hat{\beta}_k + K_k \left(y_k - V_k \hat{\beta}_k \right) \tag{10b}$$

$$P_{k+1} = (I - K_k V_k) P_k (I - K_k V_k)^T + K_k R K_k^T \tag{10c}$$

2: *CS stage:* Let $P^1 = P_{k+1}$ and $\hat{\beta}^1 = \hat{\beta}_{k+1}$.
3: **for** $\tau = 1, 2, \ldots, N_\tau - 1$ iterations **do**
4:

$$H_\tau = \left[\text{sign}(\hat{\beta}_1^\tau), \ldots, \text{sign}(\hat{\beta}_n^\tau) \right] \tag{11a}$$

$$K^\tau = P^\tau H_\tau^T \left(H_\tau P^\tau H_\tau^T + R_e \right)^{-1} \tag{11b}$$

$$\hat{\beta}^{\tau+1} = (I - K^\tau H_\tau) \hat{\beta}^\tau \tag{11c}$$

$$P^{\tau+1} = (I - K^\tau H_\tau) P^\tau \tag{11d}$$

5: **end for**
6: Set $P_{k+1} = P^{N_\tau}$ and $\hat{\beta}_{k+1} = \hat{\beta}^{N_\tau}$.

features.

The Random Field Model.
Let $G = (E^m, V^m)$ be a finite graph with a vertex set V^m and an edge set E^m. The sample space Ω consists of all possible assignments of the vertices in V^m. A random field on G is a probability distribution on Ω. The random field is Markov whenever each vertex assumes a value depending exclusively on its immediate neighbors, or in terms of conditional probabilities

$$p\left(V^m(i) = v^m(i) \mid V^m(j) = v^m(j),\ i \neq j\right) = p\left(V^m(i) = v^m(i) \mid A(i),\ (A(i), V^m(i)) \in E^m\right) \tag{13}$$

where $A(i)$ denotes the Markov blanket of the i-th vertex $V^m(i)$.

At this point we assume that our random field model obeys a Gibbs distribution. This in turn allows us to specify linear Gaussian connections of the form

$$V^m(i) = H(i)\beta(i) + \zeta(i), \quad \zeta(i) \sim \mathcal{N}(\mu_i, \sigma_i^2 I) \tag{14}$$

where $H(i) = [V^m(j),\ j \neq i]$ is a matrix composed out of the entire vertex set excluding the i-th one, and $\beta(i)$ is a parameter vector associated with the i-th vertex. An alternative formulation of (14) embeds a bias term within $\beta(i)$ and assumes a zero-mean noise

$$V^m(i) = [H(i) \quad \mathbf{1}]\beta(i) + \zeta(i), \quad \zeta(i) \sim \mathcal{N}(0, \sigma_i^2 I) \tag{15}$$

where $\mathbf{1}$ is a vector of which the entries are all 1's. It can be easily verified that the conditional probabilities associated with (15) are given by

$$p(V^m(i) \mid H(i), \beta(i), \sigma_i) \propto \exp\left(-\frac{1}{2\sigma_i^2} \parallel V^m(i) - [H(i) \quad \mathbf{1}]\beta(i)) \parallel_2^2\right) \tag{16}$$

Learning Over The Feature Space.
Let V^m be a set of n_f features from V. The random field structure associated with a given class θ in Y can be then learned by locally solving either (14) or (15) for every feature [4] using any of the methods mentioned in Section 3. The obtained parameters $\beta^\theta(i)$, $i = 1, \ldots, n_f$ associated with a class θ can be then used for approximating the corresponding noise variances σ_i^θ

$$\left(\sigma_i^\theta\right)^2 = (k_\theta - 1)^{-1} \sum_{j=1}^{k_\theta} \left[V_j^m(i) - [H_j(i) \quad 1]\beta^\theta(i)\right]^2 \tag{17}$$

where the subscript j denotes the j-th sample, and k_θ denotes the total number of samples for the class θ.

Classification Rule.
Having the random field parameters for both classes $\theta = \pm 1$, the predicted class of each and every sample in the new feature space V' is chosen as the one which maximizes the posterior probability

$$y'_j = \arg \max_{\theta=\{+1,-1\}} \log p\left((V')_j^m \mid \left\{\beta^\theta(i), \sigma_i^\theta\right\}_{i=1}^{n_f}\right) \tag{18}$$

where the subscript j denotes the j-th sample. In practice, the exact posterior may not be easy to compute. This, however, can be alleviated by computing the pseudo-likelihood over the entire network. An approximate solution to (18) is then given as

$$y'_j = \arg \max_{\theta=\{+1,-1\}} \sum_{i=1}^{n_f} \log p\left((V')_j^m(i) \mid H(i), \beta^\theta(i), \sigma_i^\theta\right) \tag{19}$$

where the conditionals are given in (16).

4.1 Relation to Other Methods

When the random field models are Markov, and the variables are assumed to be Gaussian, one can use existing approaches to learning sparse Gaussian MRFs [8,9]. Since the structure of the Gaussian MRF is encoded by the zero-pattern of the corresponding inverse-covariance matrix C, the problem is therefore reduced to recovering a sparse inverse covariance matrix from data. Note that simply inverting the empirical covariance matrix, i.e. obtaining a maximum-likelihood

[4] Here, a feature consists of all samples that are associated with a specific class.

estimate of C, does not typically produce any elements that are exactly zeros; thus an explicit sparsity-enforcing constraint is required, and a popular approach is to use l_1 regularization. Particularly, we will follow the approach of [9] that solves

$$\max_{C \succ 0} \ln \det(C) - \mathrm{tr}(SC) - \lambda ||C||_1 \tag{20}$$

where S is the empirical covariance matrix, and where $\det(A)$ and $\mathrm{tr}(A)$ denote the determinant and the trace (sum of the diagonal elements) of matrix A, respectively. The advantage of the above approach is that the problem in eq. 20 is convex, its optimal solution is unique [9] and can be found efficiently using recently proposed methods such as, for example, COVSEL of [9] or *glasso* [10]. The regularization parameter λ controls the sparsity of the solution.

5 Isometric Data Transformations

The theory of CS shows that the solutions of the noiseless convex problem (the deterministic variant of (4))

$$\min_{\hat{\beta}} \parallel \hat{\beta} \parallel_1 \text{ s.t. } Y = V\hat{\beta} \tag{21}$$

and the original NP-hard problem, in which the l_1 norm in (21) is substituted by the l_0 norm, coincides under the restriction that the sensing matrix V obeys a so-called restricted isometry property (RIP) at a certain level. In detail, the RIP is defined as

$$(1 - \delta_s) \parallel x \parallel_2^2 \leq \parallel Vx \parallel_2^2 \leq (1 + \delta_s) \parallel x \parallel_2^2 \tag{22}$$

for some $\delta_s \in (0, 1)$ and any x that is s-sparse at most. In other words, every subset of V of dimension $m \times s$ acts as nearly orthonormal system. The RIP constant δ_s gives an indication of the actual proximity of any subset to orthogonality. In the noisy case (4) the RIP constant sets an upper bound on the norm estimation error $\parallel \beta - \hat{\beta} \parallel_2$ where β is the actual sparse solution. The reader is referred to [1,3] for an extensive discussion about the RIP and its role in CS.

5.1 Main Result

The classification method suggested in the previous section locally solves a regression problem for each feature in V^m (see Section 4.2). This is accomplished by applying a CS-based method or some other l_1-regularization technique using either (14) or (15). Following the above argument, it is preferable to have an RIP-satisfying sensing (i.e., data) matrix locally at each node. However, this cannot be guaranteed for the original data set V whatsoever. Bearing this in mind, we provide a detailed description of a technique for producing an RIP-satisfying data matrix out of the original one while preserving distance ratios in the transformed space. Before proceeding, however, we introduce the notion of block-sparseness which is used in the ensuing.

Definition 3. *A vector $x \in \mathbb{R}^{dm}$ with $d, m \in \mathbb{N}$ is m-block-sparse if its non-zero entries are concentrated in blocks of dimension m. That is, if*

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$$

where $x_i \in \mathbb{R}^m$, $i = 1, \ldots, d$, and

$$\#\{x_i \mid \ x_i \neq 0, \ i = 1, \ldots, d\} << d$$

then x is said to be m-block-sparse.

Theorem 1 (Isometric Transformation). *Suppose that $H \in \mathbb{R}^{m \times dm}$ for some $d, m \in \mathbb{N}$, and let*

$$T = \operatorname{diag}(H_1^{-1} P_1, \ldots, H_d^{-1} P_d), \quad \operatorname{Ker}(T) = \emptyset \tag{23}$$

where $H_i \in \mathbb{R}^{m \times m}$ and $P_i \in \mathbb{R}^{m \times m}$, $i = 1, \ldots, d$ are the partitions of H and some RIP-satisfying matrix $P \in \mathbb{R}^{m \times dm}$, respectively. Then there exists an orthogonal transformation $\hat{T} \in \mathbb{R}^{dm \times dm}$ and scalars $\alpha > 0$ and $\delta \in (0, 1)$ for which

$$(1 - \delta) \parallel x \parallel_2^2 \leq \parallel \alpha H \hat{T} x \parallel_2^2 \leq (1 + \delta) \parallel x \parallel_2^2$$

for m-block-sparse x. In particular

$$\hat{T} = \operatorname{diag}(C_1 D_1^T, \ldots, C_d D_d^T)$$

where $C_i \Lambda_i D_i^T$, $\Lambda_i = \operatorname{diag}(\lambda_i^1, \ldots, \lambda_i^m)$, $\lambda_i^1 \geq \lambda_i^2 \geq \cdots \geq \lambda_i^m$ is the singular values decomposition (SVD) of $H_i^{-1} P_i$. Letting $\lambda_{\max} = \arg\max_{i \in [1,d]} \lambda_i^1$ and $\lambda_{\min} = \arg\min_{i \in [1,d]} \lambda_i^n$, the scaling factor can be approximated by

$$\alpha \approx 2 \left(\lambda_{\min}^{-1} + \lambda_{\max}^{-1} \right)^{-1}$$

and the RIP constant is given as

$$\delta = \frac{(1 + \delta_m)\lambda_{\min}^{-1} - (1 - \delta_m)\lambda_{\max}^{-1}}{(1 + \delta_m)\lambda_{\min}^{-1} + (1 - \delta_m)\lambda_{\max}^{-1}} < 1$$

where $\delta_m \in (0, 1)$ is the RIP constant associated with P.

Proof. Notice that by definition (23), $P = HT$ obeys the RIP

$$(1 - \delta_m) \parallel z \parallel_2^2 \leq \parallel HTz \parallel_2^2 \leq (1 + \delta_m) \parallel z \parallel_2^2 \tag{24}$$

for some m-sparse z. Now, let us set

$$z = D \Lambda^{-1} D^T x \tag{25}$$

where $C\Lambda D^T$ is the SVD of T. Notice that Λ^{-1} exists owing to $\mathrm{Ker}(T) = \emptyset$.

The transformation of the parameter space in (25) changes the sparsity basis of x. In general, we cannot expect that both vectors, the original and transformed, will have the same sparseness degree. However, the following subtle observation changes the whole picture. It can be easily verified that the block structure of T in (23) induces a similar structure of $D\Lambda^{-1}D^T$ in (25). This further implies that if x is m-block-sparse then so z.

The definition of z implies

$$\| z \|_2^2 = x^T D\Lambda^{-1}D^T D\Lambda^{-1}D^T x = \| \Lambda^{-1}D^T x \|_2^2 \tag{26}$$

Substituting (26) and the SVD of T into (24) yields

$$(1-\delta_m) \| \Lambda^{-1}D^T x \|_2^2 \leq \| HCD^T x \|_2^2 \leq (1+\delta_m) \| \Lambda^{-1}D^T x \|_2^2 \tag{27}$$

Without any loss of generality let us assume at this point that $\| x \|_2 = 1$. Hence,

$$\lambda_{\max}^{-1}(\Lambda) = \lambda_{\min}(\Lambda^{-1}D^T) \leq \| \Lambda^{-1}D^T x \|_2^2 \leq \lambda_{\max}(\Lambda^{-1}D^T) = \lambda_{\min}^{-1}(\Lambda) \tag{28}$$

Using the above in (27) while multiplying by some $\alpha > 0$ yields

$$\alpha(1-\delta_m)\lambda_{\max}^{-1} \leq \| \alpha HCD^T x \|_2^2 \leq \alpha(1+\delta_m)\lambda_{\min}^{-1} \tag{29}$$

Finally setting

$$\alpha(1-\delta_m)\lambda_{\max}^{-1} = 1-\delta \tag{30a}$$

$$\alpha(1+\delta_m)\lambda_{\min}^{-1} = 1+\delta \tag{30b}$$

and solving for α and δ yields the theorem.

5.2 Block Sparseness Equivalence and Column Ordering

As can be easily recognized from (25) and the definition of T in (23), the isometric transformation assumes a block sparse form of the projected linear space. This fact raises a question. How can we impose the block sparse form on β, our original parameter space? A straightforward approach for alleviating this problem is to reorder the columns of V or more precisely of V^m (see Section 4.2) so as to group significant features in blocks. This in turn increases the chance of having a block sparse solution to the CS problem

$$\min_{\beta^\theta(i)} \| \beta^\theta(i) \|_1 \quad \text{s.t.} \quad \| V^m(i) - H(i)\beta^\theta(i) \|_2 \leq \epsilon \tag{31}$$

which forms the heart of the random field classifiers of Section 4.

Reordering of columns can be carried out using either a ranking method or a feature selection technique (e.g., correlation, t-test, LDA and Fisher linear discriminant). The columns will be then reordered according to their measure of significance. Notice, however, that the ordering of columns within the $m \times m$ blocks of V does not really affect the block sparseness degree of β.

In this work we have used either t-test or (slightly modified) correlation method, described below; the columns are sorted in either descending or ascending order according to their associated rank.

A Linearized Correlation Coefficient The idea that motivates the following derivation is to somehow take into account the unlabeled testing samples in the computation of the correlation coefficient. The sample correlation of the i-th feature is given by

$$\rho_i = \frac{m\sum_{j=1}^{m} v_j(i)y_j - \sum_{j=1}^{m} v_j(i)\sum_{j=1}^{m} y_j}{(m\sum_j v_j(i)^2 - (\sum_j v_j(i))^2)^{1/2}(m\sum_j y_j^2 - (\sum_j y_j)^2)^{1/2}} \tag{32}$$

In practice, however, ρ_i cannot be computed over the entire set of samples owing to the fact that the testing set (which in our case consists of 2 samples) is unlabeled, i.e., the corresponding values in the response vector are unspecified. To alleviate this, we treat Y as a real-valued vector for obtaining a first-order expansion of ρ_i,

$$\hat{\rho}_i = \rho_i(y_m = 0) \pm \left.\frac{\partial \rho_i}{\partial y_m}\right|_{y_m=0} \tag{33}$$

where it is assumed, without any loss of generality, that the m-th sample is used for testing. Since the m-th sample is unlabeled the sign in (33) is undetermined. Nevertheless, the ranking is carried out by taking the absolute value of ρ_i which in turn sets an upper bound on $\hat{\rho}_i$

$$|\hat{\rho}_i| \leq \left|\rho_i(y_m = 0)\right| + \left|\frac{\partial \rho_i}{\partial y_m}\right|_{y_m=0} \tag{34}$$

that can be used instead of $|\hat{\rho}_i|$. This approach can be easily extended to more than one testing sample. In addition, we suggest to add a tuning constant $a \in (0, 1]$ for tightening the bound. Thus,

$$\tilde{\rho}_i = \left|\rho_i(y_{m-l+1} = 0, \ldots, y_m = 0)\right| + a \sum_{j=m-l+1}^{m} \left|\frac{\partial \rho_i}{\partial y_j}\right|_{y_j=0} \tag{35}$$

for l testing samples, $y_{m-l+1}, \ldots, y_m$. In our case, assuming that the training data set is balanced (i.e., $\sum_{j=1}^{m} y_j = 0$) the derivative in (35) is obtained as

$$\left.\frac{\partial \rho_i}{\partial y_j}\right|_{y_j=0} = \frac{m v_j(i)}{\left(m^2 \sum_{j=1}^{m} v_j(i)^2 \sum_{j=1}^{m} y_j^2 - m\left(\sum_{j=1}^{m} v_j(i)\right)^2 \sum_{j=1}^{m} y_j^2\right)^{1/2}} \tag{36}$$

In this work we have set $a = 0.5$ which seemed to yield improved accuracy in all cases.

5.3 Practical Implementation: Random Projections

The isometric transformation relies on the existence of some RIP-satisfying matrix of the same dimension as the original data set. Constructing such a matrix is generally a non-trivial task. Nevertheless, it is well known fact that some random matrices obey the RIP with high probability [1, 3].

Consider a matrix $P \in \mathbb{R}^{m \times n}$ of which the entries are independent identically distributed (iid) samples from $\mathcal{N}(0, m^{-1})$. Then if s, the maximal sparseness degree of the underlying parameter vector, satisfies

$$s = \mathcal{O}\left(m / \log(n/m)\right) \tag{37}$$

the matrix P obeys the RIP with probability exceeding $1 - \mathcal{O}\left(\exp(-\gamma n)\right)$ for some $\gamma > 0$ [1]. Similar result exists for a binary measurement matrix of which the entries are sampled according to

$$\Pr\left(P_{ij} = \pm 1/\sqrt{m}\right) = 0.5 \tag{38}$$

In the case of high dimensional feature space it seems that the random approach is the only one that can guarantee the RIP to some extent. Taking random P, however, imposes a conceptual problem. Thus, we can expect that there might be realizations of P that render the new data set less informative thereby deteriorating the classification accuracy. In order to avoid such instances we propose an additional stage in which a proper realization of P would be chosen by cross-validating over a transformed development set. This technique is demonstrated in the numerical study section in the ensuing.

5.4 Transductive Approach

In practice the isometric transformation can be applied to a feature space that is augmented by the test data set V'

$$\bar{V} = \begin{bmatrix} V \\ V' \end{bmatrix} \in \mathbb{R}^{(m+l) \times n} \tag{39}$$

This technique, which is similar to the transductive learning, and can be also thought of as a form of semi-supervised learning, yields a transformation that depends on both the training and the unlabeled testing data sets. This approach, which is used in the numerical study part of this work, has shown to significantly improve the classification accuracy.

Summary of our Approach.

Given an arbitrary (augmented) data matrix $H \in \mathbb{R}^{(m+l) \times n}$ where $n = d(m+l)$ and $d, m, l \in \mathbb{N}$, the transformation is carried out as follows.

1. Generate a realization of an isometric (Gaussian or Binary) random matrix P of the same dimensions as H.
2. Reorder the columns of H according to some ranking or feature selection technique.
3. Partition $H = [H_1, \ldots, H_d]$, $P = [P_1, \ldots, P_d]$ where $H_i, P_i \in \mathbb{R}^{(m+l) \times (m+l)}$.
4. Compute the transformation $\hat{T}_i = C_i D_i^T$ for all $i = 1, \ldots, d$ where $C_i \Lambda_i D_i^T$ is the SVD of $H_i^{-1} P_i$.
5. Compute the scaling factor α.
6. The transformed data set is $\bar{H} = \alpha \left[H_1 \hat{T}_1, \ldots, H_d \hat{T}_d \right]$.

6 Empirical Evaluation

The concepts of previous sections are demonstrated in fMRI classification. In particular, the effect of the isometric transformation on the classification accuracy of various compressed random field classifiers is studied. The random field classifiers that are considered herein are: 1) RF-CSKF that utilizes the CSKF for locally learning the random field structure as described in Section 4.2, 2) RF-Dantzig that is based on a similar approach where the learning is carried out using the Dantzig selector, and 3) Sparse-MRF that learns the covariance structure while assuming a Markov property of the random field model. We also asses the effects of the transformation on the LASSO-based classifier.

All classifiers were coded in Matlab's environment. The Dantzig selector implementation uses the built-in function 'linprog' that is based on a linear interior point solver. Sparse MRF uses the COVSEL package[5] for learning sparse inverse-covariance matrices that define a sparse Gaussian MRF for each class; the most likely class label is then selected based on the likelihood predicted by each sparse MRF model. LASSO classifier uses the MATLAB implementation [11] of LARS algorithm [12] to solve the linear regression problem with the class label treated as a real-valued response variable; in order to obtain binary prediction on a test sample, we simply threshold the output of LASSO model (i.e., predict +1 if the output is positive and -1 otherwise).

6.1 fMRI Data Sets

The fMRI data sets are those that were used in [13]. The detailed description of these sets can be found at the StarPlus web-site at
http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-81/www/. The data consists of a series of trials in which the subject is being shown either a picture $(+1)$ or a sentence (-1). The brain activity is monitored over a time interval of 9 seconds during which a fMRI scan is performed every 1 second. We have used this set-up for producing two data sets for each subject. The first set, which we termed 'sliced', consists of the 1st scan in each trial whereas the second one involves the average of 6 fMRI scans (from 1 to 6). The resulting data sets consist of nearly 2000 features, and 40 relevant samples (see the website for more detail).

6.2 Experimental Setup

We use a 2-out cross-validation scheme for testing the underlying classifiers. This procedure involves 20 trials in which 2 samples (one of each class) are taken as a testing set while the remaining samples are used for training. The classifiers are applied using 12 data sets, these account for 6 sets (sliced and averaged for each of the 3 subjects) and their transformed versions. In all tests, the random field-based methods utilize a total of 100 nodes taken as those with the highest

[5] Available at http://www.princeton.edu/ aspremon/CovSelCode.htm.

ranking based on either t-test or cross-correlation. In case of Sparse-MRF on the original data, we select from 30 to 100 variables using a particular ranking method before learning Sparse-MRF; we use same ranking method to generate the transformed data, as well as to select a subset of the transformed data to learn a Sparse-MRF on (the reason for variable selection here is the efficiency concerns related to learning large sparse inverse covariance matrices). Finally, the LASSO classifier approach runs on all features, and selects a desired number of variables (that is specified as an input parameter to LARS procedure) automatically.

For clarity we detail the testing procedure below.

1. Reorder the columns of the augmented data matrix $\bar{V}$ (see (39)) according to some ranking or feature selection technique using the training set only (see section 5.2).
2. Obtain a transformed augmented data matrix using some realization of the random matrix P.
3. Split the transformed data into testing and training sets.
4. Perform feature selection/ranking using the training set.
5. Learn the random field structure based on the training set.
6. Classify the testing set.

In all tests a binary random matrix P (see Section 5.3) was used for producing the transformed set. The realization of P is chosen as the one that yields the best classification accuracy when applying the above procedure on a predetermined development set which in our case was composed out of 20 samples from the training set.

6.3 Results

The classification accuracy of the various methods are shown for the sliced and the averaged data sets in Tables 1 and 2, respectively. The boldface font is used to highlight cases when transformation improves the prediction accuracy. The fact that becomes clear from both these tables is that the isometric transformation significantly improves the classification accuracy of the CS-based random field classifiers the RF-CSKF and RF-Dantzig. In the least informative case, when using the sliced data sets, the accuracy increases by more than 30% for both these methods. For the averaged data sets, however, a relatively modest improvement of around 5% is gained for the first two subjects. The transformed data set of the remaining subject, 04820, in this set shows, once again, an improvement of nearly 30%.

As for the Sparse-MRF, the advantage of using the transformation is prominent for the sliced sets in Table 1. It seems that these sets, which tend to be less informative than the averaged ones, pose difficulties to the Sparse-MRF. Nevertheless, the transformation frequently helps to increase the prediction accuracy in this case, sometimes dramatically (e.g., by more than 25% in case of subject 04820). However, the transformation seems to be less useful for Sparse-MRF on averaged datasets (Table 2). It still helps sometimes to improve the

Method	04847		05680		04820	
	Original	**Transf.**	Original	**Transf.**	Original	**Transf.**
RF-CSKF	0.50	**0.92**	0.52	**0.87**	0.55	**0.82**
RF-Dantzig	0.50	**0.87**	0.55	**0.87**	0.50	**0.75**
Sparse-MRF(30 vars, ttest)	0.75	**0.78**	0.75	**0.87**	0.48	**0.75**
Sparse-MRF(50 vars, ttest)	0.77	0.75	0.80	0.80	0.52	**0.82**
Sparse-MRF (100 vars, ttest)	0.70	**0.78**	0.73	**0.83**	0.70	**0.82**
Sparse-MRF(30 vars, lin.corr)	0.70	**0.80**	0.70	**0.85**	0.48	**0.75**
Sparse-MRF(50 vars, lin.cor)	0.78	**0.83**	0.80	0.77	0.55	**0.75**
Sparse-MRF (100 vars, lin.corr)	0.68	**0.80**	0.75	**0.77**	0.70	**0.80**
LASSO(100 vars, ttest)	0.85	**0.90**	0.90	**0.90**	0.45	**0.65**
LASSO(100 vars, lin.corr)	0.85	0.70	0.90	**0.90**	0.45	**0.70**

Table 1. Classification accuracy on Sliced data sets.

Method	04847		05680		04820	
	Original	**Transf.**	Original	**Transf.**	Original	**Transf.**
RF-CSKF	0.87	**0.92**	0.85	**0.90**	0.65	**0.92**
RF-Dantzig	0.80	**0.87**	0.85	**0.90**	0.65	**0.92**
Sparse-MRF(30 vars, ttest)	0.85	**0.88**	0.80	**0.83**	0.75	**0.80**
Sparse-MRF(50 vars, ttest)	0.87	0.80	0.90	0.83	0.82	0.78
Sparse-MRF (100 vars, ttest)	0.95	0.80	0.95	0.78	0.90	0.78
Sparse-MRF(30 vars, lin.corr)	0.85	0.85	0.80	0.80	0.77	**0.78**
Sparse-MRF(50 vars, lin.corr)	0.87	0.83	0.83	0.83	0.77	**0.78**
Sparse-MRF (100 vars, lin.corr)	0.85	0.85	0.83	0.83	0.77	**0.78**
LASSO(100 vars, ttest)	0.95	**0.95**	1.00	0.95	0.75	**0.90**
LASSO(100 vars, lin.corr)	0.95	**1.00**	1.00	0.95	0.75	**0.95**

Table 2. Classification accuracy on Averaged data sets.

accuracy by 3-5%, especially when the number of variables is small; most of the time it matches already quite high accuracy of Sparse-MRF on the original averaged data, but in some cases it could actually hurt the performance, especially when the original accuracy is already very high (e.g., 95%). In general, we expect the transformation to help Sparse-MRFs when their performance has a room for improvement, but possibly avoid it when the performance is already nearly-perfect.

Similar observations can be made for LASSO: transformation improves LASSO's performance on challenging sliced data, but on the easier averaged data the performance remains roughly unchanged, except for the case of subject 04820 (column 3 in Table 2) where transformation appears to help dramatically.

We have assessed the effect of the linearized correlation ranking over the prediction accuracy of the RF-CSKF classifier. Thus, the accuracies that were obtained when using a standard correlation coefficient for column reordering of the sliced data sets are 0.85 for subject 04847, 0.82 for subject 05680, and 0.75

for subject 04820. The corresponding accuracies that were obtained when using the linearized correlation method are shown in Table 1.

Figure 1 demonstrates the role of the random RIP matrix P. Shown in this figure are histograms of classification accuracies over 20 runs each of which is based on a distinct realization of P. Here we have used the sliced data set for subject 04820. It can be clearly shown that the classification accuracy greatly depends on the realization used. Therefore, a selection mechanism for picking a "good" realization (e.g., cross-validating over a development set) is crucial. It is worthwhile noting that the mean prediction accuracy in this example is still higher than the obtained one when using the non transformed data set.

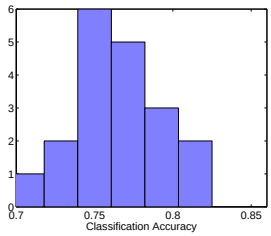

Fig. 1. The effect of the random projection on the classification accuracy. Showing the accuracy for sliced data set 04820 using 20 different realization of P.

7 Summary and Discussion

The isometric transformation, which is essentially random by nature, significantly improves the performance of the random field compressed sensing-based classifiers presented in this work. In the case where the least informative sliced data sets are used the transformation improves the prediction accuracy, though by a much modest percentage, of the Sparse-MRF and LASSO classifiers. However, it is shown that for the averaged data sets which yields already high predictive accuracies the transformation may not always help. The transformation is proved to enforce the RIP on an arbitrary data matrix while assuming block-sparse structure of the parameter space. This property explains the improvement gained by the compressed sensing-based classifiers. The transductive approach which consists of applying the transformation to the augmented data matrix that is composed of both the training and (unlabeled) test samples may be regarded as a form of semi-supervised learning (or adaptation).

It is shown that the realization of the random matrix P used for producing the transformed data set greatly effects the classification accuracy. It is therefore suggested that a "good" realization would be selected based on some preprocessing such as the cross-validation method used in this work. One possible reason

for the relatively wide span of accuracies produced by randomly picking P is related to the RIP constant δ_s of the chosen realization. It is well known fact that evaluating this constant for a given matrix is generally intractable. This problem is a part of the authors future research plans that is aimed at devising a computational method for picking the best realization out of a few possible outcomes.

References

1. Candes, E. J., "Compressive Sampling," European Mathematical Society, Proceedings of the International Congress of Mathematicians, Madrid, Spain, 2006.
2. Donoho, D., "Compressed Sensing," *IEEE Transactions on Information Theory*, Vol. 52, 2006, pp. 1289–1306.
3. Candes, E. J., Romberg, J., and Tao, T., "Robust Uncertainty Principles: Exact Signal Reconstruction from Highly Incomplete Frequency Information," *IEEE Transactions on Information Theory*, Vol. 52, 2006, pp. 489–509.
4. Tibshirani, R., "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol. 58, No. 1, 1996, pp. 267–288.
5. Candes, E. and Tao, T., "The dantzig selector: statistical estimation when p is much larger than n," *Annals of Statistics*, Vol. 35, 2007, pp. 2313–2351.
6. Carmi, A., Gurfil, P., and Kanevsky, D., "Methods for Sparse Signal Recovery Using Kalman Filtering with Embedded Pseudo-Measurement Norms and Quasi-Norms," *Submitted to IEEE Transactions on Signal Processing*, 2009.
7. Carmi, A., Gurfil, P., and Kanevsky, D., "A Simple Method for Sparse Signal Recovery from Noisy Observations Using Kalman Filtering," *IBM Tech. Report RC24709*, 2008.
8. Meinshausen, N. and Buhlmann, P., "High dimensional graphs and variable selection with the Lasso," *Annals of Statistics*, Vol. 34(3), 2006, pp. 1436–1462.
9. O.Banerjee, El Ghaoui, L., and d'Aspremont, A., "Model Selection Through Sparse Maximum Likelihood Estimation for Multivariate Gaussian or Binary Data," *Journal of Machine Learning Research*, Vol. 9, March 2008, pp. 485–516.
10. Friedman, J., Hastie, T., and Tibshirani, R., "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, 2007.
11. Sjöstrand, K., "Matlab implementation of LASSO, LARS, the elastic net and SPCA," jun 2005, Version 2.0.
12. Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R., "Least angle regression," *Ann. Statist.*, Vol. 32, No. 1, 2004, pp. 407–499.
13. Mitchell, T., Hutchinson, R., Niculescu, R., Pereira, F., Wang, X., Just, M., and Newman, S., "Learning to Decode Cognitive States from Brain Images," *Machine Learning*, Vol. 57, 2004, pp. 145–175.