Informative Projection Recovery for Classification, Clustering and Regression

Madalina Fiterau and Artur Dubrawski
Carnegie Mellon University, Pittsburgh, PA 15213
Email: {mfiterau,awd}@cs.cmu.edu

***Abstract*—Data driven decision support systems often benefit from human participation to validate outcomes produced by automated procedures. Perceived utility hinges on the system's ability to learn transparent, comprehensible models from data. We introduce and formalize Informative Projection Recovery: the problem of extracting a set of low-dimensional projections of data which jointly form an accurate solution to a given learning task. We approach this problem with RIPR: a regression-based algorithm that identifies informative projections by optimizing over a matrix of point-wise loss estimators. It generalizes from our previous algorithm, offering solutions to classification, clustering, and regression tasks. Experiments show that RIPR can discover and leverage structures of informative projections in data, if they exist, while yielding accurate and compact models. It is particularly useful in applications involving multivariate numeric data in which expert assessment of the results is of the essence.**

I. Problem Statement

Intelligent decision support systems are rarely fully automated. Data limitations, absence of contextual information, as well as the need for accountability, often require human involvement. The stringency of the requirement usually escalates with the stakes of decisions being made. Notable examples include medical diagnosis or nuclear threat detection, but the benefits of explainable analytics are universal. To meet these requirements, the output of a regression, clustering, or a classification system must therefore be presented in a form that is comprehensible and intuitive to humans, while offering the users insight into how the learning task was accomplished. A desirable solution consists of a small number of low-dimensional (not higher than 3-D) projections of data, selected from among the original dimensions, that jointly provide good acccuracy while exposing the processes of inference and prediction to visual inspection by humans.

We formulate Informative Projection Recovery (IPR) as the problem of identifying small groups of features which encapsulate enough information to allow learning of well-performing models. Each such feature group – equivalent to a low-dimensional axis-aligned projection – handles a different subset of data with a specific model. The resulting set of projections, with their corresponding models, jointly form a solution to the IPR problem. We have previously proposed such a solution tailored to non-parametric classification. Our RECIP algorithm [10] employs point estimators for conditional entropy to recover a set of low-dimensional projections that classify queries using non-parametric discriminators in an alternate fashion – each query is classified using one specific projection from the retrieved set.

In this paper, we substantially extend the Informative Projection Recovery (IPR) problem using a formalization applicable to any learning task for which a consistent estimator of the loss function exists. To solve the generalized IPR problem, we introduce the Regression-based Informative Projection Recovery (RIPR) algorithm. It is applicable to a broad variety of machine learning tasks such as semi-supervised classification, clustering, or regression, as well as to various generic machine learning algorithms that can be tailored to fit the problem framework. RIPR is useful when (1) There exist low-dimensional embeddings of data for which accurate models for the target tasks can be learned; (2) It is feasible to identify a low-dimensional model that can correctly process given queries. We formulate loss functions that can be used to implement IPR solutions for common learning problems, and we introduce additive estimators for them. We empirically show that RIPR can succeed in recovering the underlying structures. For synthetic data, it yields a very good recall of known informative projections. For real-world data, it reveals groups of features confirmed to be relevant by domain experts. We observe that low-dimensional RIPR can perform at least as well as models using learners from the same class, trained using all features in the data.

II. Related Work

Dimensionality reduction is a common preprocessing step in applications where simplified models are preferable. Methods that learn linear combinations of features, such as LDA, are not quite appropriate for the task considered here, since for compehensibility we prefer to rely on the dimensions available natively in the original feature space. Feature selection methods, such as the lasso, are suitable for identifying informative features, but they do not identify specific data points for which they are relevant. Our work specifically fits the goals of query dependent feature selection and context specific classification.

One relevant feature selection method [16] learns disjunctions and conjunctions of features leading to models of slightly greater complexity than what we desire. Ting et al. [21] introduce Feating, where the submodel selection relies on simple attribute splits followed by fitting local predictors. Although the model has some similarities to what we propose, such as reliance on a decision structure to pick the classification model, the algorithm itself is substantially different. Obozinski et al. [17] present a subspace selection method in the context of multitask learning. Gu et al. [12] propose a joint method for feature selection and subspace learning, however, their classification model is not query specific. Multiple algorithms that transform complex or unintelligible models to user-friendly equivalents have been proposed [5], [8], [14], [9]. Algorithms specifically designed to yield understandable models are a precious few. A rule learning method is described

in [18], even though the resulting rules can make visualization difficult. Itemset mining [15] is attribute-focused, but not specifically designed for classification. Numerous methods perform clustering in low-dimensional subspaces [1], [13]. Local subspace preferences [4] capture the directions of high data density. Clustering has employed information theoretical approaches [20] and divergence-based methods, e.g. Bregman Divergence, to unify centroid-based approaches [2]. Spectral clustering obtains low-dimensional projections, however, it relies on a similarity matrix [7]. Unlike most of those approaches, our method is designed to retrieve subsets of the feature space for use in a complementary manner to provide query-specific solutions. RIPR brings the following improvements over other dimensionality reduction techniques, including our RECIP [10]: (1) It is suitable for a wide variety of learning tasks; (2) It is designed to optimize various types of machine learning solvers; and (3) It handles missing data.

III. Informative Projection Recovery

This section formalizes the IPR problem and describes an algorithmic framework generalized from the RECIP procedure in [10]. The algorithm solves IPR when the learning task can be expressed in terms of a loss function and there exists a consistent point-estimator for the risk. The derivations in Section III-A follow the setup of RECIP, the main improvement being the formalization of the problem for learning tasks other than classification and the capability to include learners of arbitrary class – instead of just nonparametric classifiers.

A. Formalization of Projection Recovery

Assume we are given a dataset $X = \{x_1 \dots x_n\} \in \mathcal{X}^n$ where each sample $x_i \in \mathcal{X} \subseteq \mathbb{R}^m$ and a learning task on the space $\mathcal{X}$ with output in a space $\mathcal{Y}$ such as classification, clustering or regression. The learner for the task is selected from a class $\mathcal{T} = \{f : \mathcal{X} \to \mathcal{Y}\}$, where the risk for the class $\mathcal{T}$ is defined in terms of the loss ℓ as

$$\mathcal{R}(\tau, \mathcal{X}) = \mathbb{E}_{\mathcal{X}} \ell(x, \tau) \quad \forall \tau \in \mathcal{T}.$$

The optimal learner for the task is $\tau^* \stackrel{def}{=} \arg\min_{\tau \in \mathcal{T}} \mathcal{R}(\tau, \mathcal{X})$. We indicate by $\tau_{\{X\}}$ the learner from class $\mathcal{T}$ obtained by minimizing the empirical risk over the training set X.

$$\tau_{\{X\}} \stackrel{def}{=} \arg\min_{\tau \in \mathcal{T}} \hat{\mathcal{R}}(\mathcal{T}, X) = \arg\min_{\tau \in \mathcal{T}} \frac{1}{n} \sum_{i=1}^{n} \ell(x_i, \tau)$$

The class $\mathcal{M}$ of models constructed by our IPR framework is formalized as having a set Π of projections with dimension at most d, a set τ of learners and a selection function g:

$$\begin{aligned} \mathcal{M} = \{ & \Pi = \{\pi;\ \pi \in \mathbf{\Pi}, |\pi| \le d\}, \\ & \tau = \{\tau;\ \tau_i \in \mathcal{T}, \tau_i : \pi_i(\mathcal{X}) \to \mathcal{Y} \quad \forall i = 1 \dots |\Pi|\}, \\ & g \in \{f : \mathcal{X} \to \{1 \dots |\Pi|\}\}\ \}\ . \end{aligned}$$

$\mathbf{\Pi}$ contains all axis-aligned projections; the subset $\Pi \subseteq \mathbf{\Pi}$ in $\mathcal{M}$ contains only projections with at most d features. The value d is application-specific; usually 2 or 3, to permit users to view the projections. Function g selects the adequate projection π and its corresponding learner τ to handle a given query x.

Based on this model, we derive a composite learner which combines the learners operating on the individual low-dimensional projections. The loss of this learner can be expressed in terms of the component losses: $\tau_{\mathcal{M}}(x) = \tau_j(\pi_j(x))$, $\ell(x, \tau_{\mathcal{M}}) = \ell(\pi_j(x), \tau_j)$, where $g(x) = j$ represents the index of the learner which handles data point x and $\pi_j(x)$ is the projection of x onto π_j. Optimizing over the model class $\mathcal{M}$, the IPR problem for learning task $\mathcal{T}$ can be formulated as a minimization of the expected loss:

$$M^* = \arg\min_{\mathcal{M}} \mathbb{E}_{\mathcal{X}} \ell(\pi_{g(x)}(x), \tau_{g(x)}) \tag{1}$$

Since we are dealing with an unsupervised problem in terms of the selection function (it is unknown which submodel should be applied for which point), there are limitations on its learnability. Recovery is possible in the following example: For all data x whose j^{th} feature is in the set A, the targeted task can be optimally performed by the learner τ^*_A. It uses features $\{i_1 \dots i_d\}$ of x. The approximation of τ^*_A should only be trained over samples for which $x^j \in A$.
$\exists j, A$ s.t. $\forall x$ with $x^j \in A$, $\tau^*(x^1 \dots x^m) = \tau^*_A(x^{i_1} \dots x^{i_d})$

B. Regression-based Informative Projection Recovery (RIPR)

The crux of the algorithm is writing the empirical version of (1) as a combinatorial problem over multiple projections. The algorithm is designed assuming there exist low-dimensional embeddings that enable capturing accurate models for the target task. Thus, every sample data x_i can be dealt with by just one projection π_j – recall that $g(x_i) = j$. We model this as a binary matrix B: $B_{ij} = I[g(x_i) = j]$.
The minimizers of the risk and empirical risk are:

$$M^* = \arg\min_{\mathcal{M}} \mathbb{E}_{\mathcal{X}} \sum_{j=1}^{|\Pi|} I[g(x) = j] \ell(\pi_j(x), \tau_j)$$

$$\hat{M}^* = \arg\min_{\mathcal{M}} \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{|\Pi|} I[g(x_i) = j] \ell(\pi_j(x_i), \tau_j) \tag{2}$$

Assume now that we can consistently estimate the loss of a task learner τ at each available sample, that is

$$\exists \hat{\ell} \text{ s.t.} \quad \forall x \in \mathcal{X}, \tau \in \mathcal{T} \quad plim_{n \to \infty} \hat{\ell}(x, \tau) = \ell(x, \tau) \tag{3}$$

Plugging (3) into (2) yields the final form used to obtain the estimated model:

$$\begin{aligned} \hat{M} &= \arg\min_{\mathcal{M}} \sum_{i=1}^{n} \sum_{j=1}^{|\Pi|} I[g(x_i) = j] \hat{\ell}(\pi_j(x_i), \tau_i) \\ &= \arg\min_{\mathcal{M}, |\Pi| < |\mathbf{\Pi}|} \sum_{i=1}^{n} \sum_{j=1}^{|\mathbf{\Pi}|} B_{ij} L_{ij} \quad , \quad L_{ij} = \hat{\ell}(\pi_j(x_i), \tau_i) \end{aligned}$$

The loss estimators L_{ij} are computed for every data point on every subspace of up to the user-specified dimensionality d. B is learned through a regularized regression procedure that penalizes the number of projections Π used in the model. This translates to an ℓ_0 penalty on the number of non-zero columns in B, relaxed to ℓ_1. The ℓ_0 penalty is written as $I[B_{\cdot,j} \neq 0]$, while its relaxation is $||B||_{1,1}$.

$$\hat{B} = \arg\min_{B} ||L^* - L \odot B||_2^2 + \lambda \sum_{j=1}^{d^*} I[B_{\cdot,j} \neq 0]$$

where d^* is the number of projections, $L^*_i \stackrel{def}{=} \min_j L_{ij}$ and the operator $\odot$ is defined as

$$\odot : \mathbb{R}^{n,d^*} \times \mathbb{R}^{n,d^*} \to \mathbb{R}^n, \quad (L \odot B)_i = \sum_{j=1}^{d^*} L_{ij} B_{ij}$$

The basic optimization procedure is detailed in [10], the key difference here is in the computation of the loss matrix L. The technique resembles the adaptive lasso. It gradually reduces the number of non-zero columns in B until convergence

Algorithm 1 RIPR framework.

$\delta = [1 \dots 1]$
repeat
  $B = \arg\min_B ||L^* - L \odot B||_2^2 + \lambda |B\delta|_{\ell_1}$
    subject to $||B_{k,.}||_{\ell_1} = 1 \qquad k = 1 \dots n$
  $\delta_j = ||B_{.,j}||_{\ell_1} \qquad j = 1 \dots d^*$ (update multiplier)
  $\delta = (||\delta||_{\ell_1} - \delta)/||\delta||_{\ell_1}$
until δ converges
return $\Pi = \{\pi_i; \quad |B_{.,i}|_{\ell_1} > 0 \quad \forall i = 1 \dots d^*\}$

to a stable set of projections. As illustrated in Algorithm 1, the procedure uses the multiplier δ to gradually bias selection towards projections that not only perform well but also suit a large number of data points.

C. Customizing RIPR for Different Learning Tasks

Next, we show how to formulate IPR for different learning tasks. When the aim is to find informative projections without knowing the class of learners to be used, we employ nonparametric estimators of loss. The performance of the algorithm will depend on their rates of convergence.

1) Classification: We have addressed the IPR problem for classification in [10]. To run the RECIP algorithm using the RIPR framework, the risk is the conditional entropy of the label given the features. The conditional entropy over data assigned to projection π_j is shown to be estimated as follows:

$$\hat{H}(Y|\pi(X); \{x|g(x) = j\}) \propto \frac{1}{n} \sum_{i=1}^{n} I[g(x_i) = j] \hat{\ell}(x_i, \tau_{\pi_j}^k)$$

$$\hat{\ell}(x_i, \tau_{\pi_j}^k) = \Big(\frac{(n-1)\nu_k(\pi_j(x_i), \pi_j(X_{y(x_i)} \setminus x_i))}{n\nu_k(\pi_j(x_i), \pi_j(X_{\neg y(x_i)}))} \Big)^{(1-\alpha)|\pi_j|}$$

$\pi(X)$ is the projection of vector X onto π; X_γ is the subset of the sample for which the label is γ; $X \setminus x$ is the sample obtained when removing point x from X; $\nu_k(x, X)$ represents the k^{th} distance from point x to its k-nearest-neighbor from the sample X; $\tau_{\pi_j}^k$ is the k-NN classifier on projection π_j.

This result is obtained by using the Tsallis α-divergence estimator [19] and yields a loss estimator for binary classification. α is a constant close to 1 (e.g., 0.95) and $|\pi_j|$ is the dimensionality of the subspace π_j.

2) Semi-supervised Classification: RIPR allows an extension to semi-supervised classification. Consider a problem with labeled samples X_+ and X_- and unlabeled samples X_u, where each sample belongs to $\mathbb{R}^m$. The objective is to find a discriminator in a low-dimensional sub-space of features that correctly classifies the labeled samples and simultaneously allows substantial separation for unlabeled data, i.e., very few unlabeled data points remain between the clusters of data from different classes. We choose a loss function that penalizes unlabeled data according to how ambivalent they are to the label assigned. This is equivalent to considering all possible label assignments and assuming the most 'confident' one – the label with the lowest loss – for unlabeled data. The estimator for labeled data is the same as for supervised classification. The score for a projection is computed by using the same estimator for KL divergence between class distributions, to which we add a metric for unlabeled data which penalizes samples that are about equidistant from the point-clouds of each class: $\hat{\mathcal{R}}(X_u, \tau_\pi^k)$. We use the notation $\pi(X)$ to represent the projections of a set of data points X:

$$\begin{aligned}\hat{\mathcal{R}}(X, \tau_\pi^k) = &\sum_{x \in X_+} \Big(\frac{\nu_{k+1}(\pi(x), \pi(X_+))}{\nu_k(\pi(x), \pi(X_-))} \Big)^{(1-\alpha)|\pi|} \\ &+ \sum_{x \in X_-} \Big(\frac{\nu_{k+1}(\pi(x), \pi(X_-))}{\nu_k(\pi(x), \pi(X_+))} \Big)^{(1-\alpha)|\pi|} \\ &+ \sum_{x \in X_u} \min \Big(\frac{\nu_k(\pi(x), \pi(X_-))}{\nu_k(\pi(x), \pi(X_+))}, \frac{\nu_k(\pi(x), \pi(X_+))}{\nu_k(\pi(x), \pi(X_-))} \Big)^{(1-\alpha)|\pi|}\end{aligned}$$

In these learning tasks, typical convergence issues encountered with nearest-neighbor estimators can often be remedied thanks to low dimensionality of the projections.

3) Clustering: It is not always straightforward to devise additive point estimators of loss for clustering since some methods rely on global as well as local information. Distribution-based and centroid-based clustering fit models on the entire sets of data. This is an issue for the IPR problem because it is not known upfront how data should be assigned to the submodels. To go around this, we first learn a RIPR model for density-based clustering, and then cluster each projection using only data assignment provided by it. Of course, that is not required if density-based clustering is the method of choice. To solve IPR for density-based clustering, we consider the negative divergence, in the neighborhood of each sample, between the distribution from which the sample X is drawn and the uniform distribution on $\mathcal{X}$. Let U be the size n sample drawn uniformly from $\mathcal{X}$. Again, we use the nearest-neighbor estimator converging to the KL divergence. τ_i^{clu} is some clustering technique such as k-means.

$$\hat{\mathcal{R}}_{clu}(\pi_i(x), \tau_i^{clu}) \to -KL(\pi_i(X) || \pi_i(U))$$

$$\hat{\ell}_{clu}(\pi_i(x), \tau_i^{clu}) \approx \Big(\frac{d(\pi_i(x), \pi_i(X))}{d(\pi_i(x), U)} \Big)^{|\pi_i|(1-\alpha)}$$

We now illustrate how RIPR clustering with k-means can improve over applying k-means to the entire set of features. Synthetic data used has 20 numeric features, and contains three Gaussian clusters on each of its informative projections. The informative projections comprise the following sets of feature indices: $\{17, 12\}$, $\{10, 20, 1\}$ and $\{4, 6, 9\}$. Clusterings obtained by k-means shown in those projections are depicted in the left part of Figure 1. The right part of it shows reslts obtained with RIPR. Every cluster is colored differently, with black representing data not assigned to that projection. The number of clusters is selected with cross-validation for both k-means and RIPR. The clustering obtained with k-means on all dimensions looks very noisy when projected on the actual informative features. The explanation is that the clustering might look correct in the 20-dimensional space, but when projected, it no longer makes sense. On the other hand, RIPR recovers the underlying model enabling the correct identification of the clusters. Naturally, recovery is only possible as long as the number of incoherent data points (that do not respect the low-dimensional model) stays below a certain level.

4) Regression: Our intent for RIPR is to enable projection retrieval independently of the type of a regressor used, so the natural choice for a loss metric is a non-parametric estimator. We consider k-NN regression - computing the value at a query point by averaging the values at the k-nearest neighbors of the query. To factor in spatial placement, we weigh the values by their inverse distance from query, then estimate predicted value

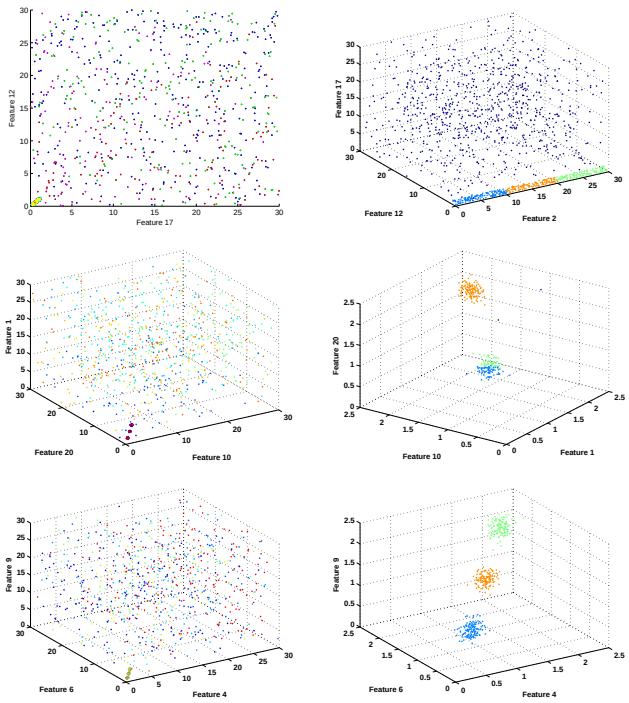

Fig. 1: Projections of k-means clusters on the informative features and RIPR low-dimensional clusters induced from synthetic data.

as normalized weighted average of the neighbor values.

$$\hat{\ell}_{reg}(\pi_i(x), \tau_i(\pi_i(x))) = (\hat{\tau}(\pi_i(x)) - y)^2 \qquad \hat{\ell}_{reg} \to 0$$

$$\hat{\tau}_i(\pi_i(x)) = \frac{\sum_{i=1}^{k} w_{(i)} y_{(i)}}{\sum_{i=1}^{k} w_{(i)}}, \qquad \text{where } w_{(i)} = \frac{1}{||x - x_{(i)}||_2}$$

Concerning the selection function, we identify two possible approaches. The first is to label each training data point according to the projections in the set used to solve it, then train a classifier using these labels. The second is to simply estimate, based on the regressor accuracy at neighboring data, the probability that the regressor is appropriate for this data point. We opt for the latter because it avoids the issues with an additional training step and it is consistent with the regressors themselves in the usage of neighborhood information.

$$\hat{g}(x) = \underset{j \in \{1 \ldots |\Pi|\}}{\arg\min} \frac{\sum_{i=1}^{k} w_{(i)} B_{(i)j}}{\sum_{i=1}^{k} w_{(i)}}, \qquad w_{(i)} = \frac{1}{||x - x_{(i)}||_2}$$

Interestingly, because of the consistency properties of the nearest-neighbor methods [6], the composite regressor is also consistent under the assumption of existence of embedding.

D. Computational Complexity

RIPR requires estimating the loss for every data point, for every combination of features. To find the k[th] nearest neighbors of every sample point using a using a k-d tree [11] – which costs $O(dn \log n)$ to build – for every projection of up to size d costs $O(\log n)$. Thus, for all $d^* = O(m^d)$ projections, the total time required to compute the loss matrix is $O(d^*(d+1)n \log n)$, or, in terms of the feature size m, $O(dm^d n \log n)$.

For the complexity of Algorithm 1, we use the bounds in [3]. The optimization is over a matrix of size $N = d^* n$. Computing the values and derivatives of the objective and the constraints requires $M = O(d^* n)$ operations. The upper and lower bound on the number of operations needed to obtain a solution ϵ away from the optimum are $O(NM)ln(\frac{1}{\epsilon})$ and $O(N(N^3 + M))ln(\frac{1}{\epsilon})$ respectively. Thus, the worst case runtime for the optimization is $O\left(m^{4d} n^4\right) \ln\left(\frac{1}{\epsilon}\right)$. Although the complexity increases exponentially with d, for the applications we consider d is typically 2, resulting in a runtime of $O\left(m^8 n^4\right) \ln\left(\frac{1}{\epsilon}\right) + O\left(m^2 n \log n\right)$.

In the adaptive lasso procedure, we can discount projections that are not informative for any of the sample data points so the dimensionality of the optimization problem is reduced from $n \times d^*$ to $n \times \min(d^*, n)$. When $m^d > n$, the runtime depends largely on n (4), which is beneficial for datasets that are underdetermined (small sample size but large number of features) – a frequent case in e.g. computational biology.

$$O\left(n^8\right) \ln\left(\frac{1}{\epsilon}\right) + O\left(dn^2 \log n\right) \tag{4}$$

IV. Experimental Results

This section illustrates the capabilities of the RIPR framework in recovering the underlying patterns in data and in training well-performing classification and regression systems. The lasso appears to be a natural contender for our method. However, it only retrieves features rather than projections. In [10], we have attempted to adapt the lasso to the projection recovery problem, but even the improved version of the lasso performed poorly for the recovery of query-specific models.

A. Semi-supervised Classification

To evaluate RIPR semi-supervised classification, we use the same type of synthetic data as in [10], but we obscure some labels before training to see if the projection recovery performance is maintained. The synthetic data for this section contains $P = 2$ informative projections and $M = 10$ features. Every projection has $N = 1,000$ data points which it can classify. There are also R noisy data points that cannot be classified by any projection; this parameter varies between experiments. Also variable is the proportion of unlabeled data. We start with fully labeled data, then for every u points in the training set we obscure one label, so for smaller u, the larger proportion of unlabeled data, and the harder the task.

TABLE I: Accuracy of semi-supervised RIPR on synthetic data compared to a k-NN model on all features and projection recovery.

	no u	***u=7***	***u=5***	***u=3***	***no u***	***u=7***	***u=5***	***u=3***
R	**Accuracy RIPR SSC**				**Accuracy k-NN**			
0	0.928	0.931	0.918	0.928	0.722	0.713	0.714	0.707
30	0.923	0.919	0.931	0.928	0.726	0.724	0.717	0.714
50	0.904	0.896	0.898	0.886	0.726	0.701	0.701	0.699
100	0.893	0.882	0.878	0.877	0.717	0.711	0.698	0.715
1000	0.688	0.687	0.693	0.705	0.627	0.621	0.612	0.607

Table I summarizes the accuracy of RIPR for semi-supervised classification using k-NN models on each of the projections. We call this method Ripped k-NN. We have included the performance of a k-NN model trained using all features. As expected, RIPR outperforms the high-dimensional model. Even though noise impacts RIPR performance, our technique performs better than k-NN even for $R = 1,000$. This improvement is not limited to k-NN classifiers: Section IV-C shows similar results when comparing SVM regressors to their Ripped version. RIPR achieves very good precision and recall for all values of R, despite the noise and unlabeled data.

B. Clustering

RIPR can be wrapped around virtually any existing clustering, regression, or classification algorithm, maintaining their high performance while satisfying the requirement of working with only a few dimensions of data at a time. Below we show that RIPR combined with k-means – which we informally call Ripped k-means – performs better than the standard k-means by leveraging the low-dimensional structure in data.

We trained RIPR and k-means models and evaluated their performance on datasets from the UCI repository. Meta-parameters for both methods were optimized via cross-validation. The data was scaled to $[0, 1)$ before clustering. We used distortion as the evaluation metric as it is native to k-means. We opt against using Rand index since in its standard form it requires the actual labels that are unavailable in most real-world clustering data sets. As shown in Table II, the distortion results for the RIPR model are better than for plain k-means. The resulting cluster dimensionalities vary as well, which is why we also considered another metric of success: the volume of the resulting clusters measured in full feature space. This comparison is fair because the volumes are computed in the same dimensionality. For k-means, we approximated the volume of each cluster by its enclosing hyper-ellipsoid. For RIPR, the approximation for each cluster used its enclosing cylinder, the base of which was the ellipsoid corresponding to the actual identified low-dimensional cluster. This comparison is also provided in Table II. It is apparent that RIPR obtains slightly more compact models than k-means, but has the advantage that only a fraction of the features are used by it. The total number of centroids is roughly the same for k-means and RIPR, so the difference in volume is genuinely due to the improvement fidelity of clustering.

TABLE II: Results of clustering of real-world datasets.

UCI	*Avg Dist RIPR*	*Avg Dist k-means*	*LogVol RIPR*	*LogVol k-means*
Seeds	16	107	7.68	9.70
Libras	9	265	-5.80	7.26
Boone	125	1.15e6	240.00	248.15
Cell	40,877	8.18e6	54.69	67.68
Concrete	1,370	55,594	49.24	52.75

C. Regression

As with clustering, RIPR regression is meant to complement existing regression algorithms. We exemplify by enhancing SVM and comparing it with the standard SVM. The synthetic data we use contains 20 features generated uniformly with Gaussian noise. The first feature and q pairs of other features (j_1, j_2) determine the regression function as follows:

$$f(x) = \sum_{j=1}^{q} I[j <= x_1 < j+1] f_j(x_{j_1}, x_{j_2}) + \epsilon \qquad \forall j \in 1 \dots q$$

Table III shows that 'Ripped Kernel SVM' achieves better accuracy that Kernel SVM trained on all features. The explanation is that RIPR actively identifies and ignores noisy features and useless data while learning each submodel. Additionally, we tested whether the underlying projections are correctly recovered by computing precision and recall metrics. Recall is always high, while precision is high as long as the projections do not overlap significantly in the feature space. It is because partially-informative projections can also be recovered if feature overlaps exist. This behavior can be controlled by adjusting the extent of regularization.

TABLE III: RIPR SVM and standard SVM compared on synthetic data

IP #	***2***	***3***	***5***	***7***	***10***	***2***	***3***	***5***	***7***	***10***
	MSE RIPR					***MSE SVM***				
0	0.05	0.27	0.05	0.02	0.23	0.27	1.16	0.11	0.1	0.43
100	0.42	1.26	0.34	1.45	0.52	0.8	1.02	0.6	2.99	0.94
200	0.5	0.86	0.8	0.33	0.99	0.97	1.27	0.29	0.68	1.44
400	0.63	1.47	1.34	1.61	0.11	0.4	1.26	1.64	1.71	0.08
800	0.69	0.38	1.12	0.68	1.1	0.52	0.06	0.91	0.9	1.16
	RIPR Precision for IPR					***RIPR Recall for IPR***				
0	1	1	0.4	0.43	0.3	0.67	1	0.67	1	1
100	1	0.67	0.6	0.43	0.2	0.67	0.67	1	1	0.67
200	1	1	0.6	0.43	0.3	0.67	1	1	1	1
400	1	1	0.6	0.43	0.1	0.67	1	1	1	0.33
800	1	0.67	0.4	0.29	0.3	0.67	0.67	0.67	0.67	1

D. Case Study 1: Artifact Detection from Partially-Observed Vital Signals for Monitoring Intensive Care Unit Patients

An additional extension of RIPR (vs. our previous algorithm RECIP) is its tolerance to missing data. For a data point x, the values of the loss estimators are set to ∞ for all projections that involve missing values for x. This ensures that data tends to be explained using projections that have a full description for it, while projections with some missingness are not prefereable though not ignored. This extends RIPR's range of applications, which include a medical informatics task.

Recovery of meaningful, exmplainable models is fundamental for the clinical decision-making process. We work with a cardio-respiratory monitoring system designed to process multiple vital signs indicative of the current health status of a patient. The system issues an alert whenever some form of instability requires attention. In practice, a substantial fraction of these alerts are not due to real emergencies (true alerts), but instead are triggered by malfunctions or inaccuracies of the sensing equipment (artifacts). Each system-generated alert is associated with a vital sign that initiated it: either heart rate (HR), respiratory rate (RR), blood pressure (BP), or peripheral arterial oxygen saturation (SpO_2). Here, we show as an example the analysis of respiratory rate alerts, i.e. we consider episodes when this vital sign was the first to exceed its control limits, triggering an alert. A modest subset of data was manually reviewed and labeled by clinicians, and true alerts were distinguished from apparent artifacts. Our aim was to learn an artifact-identification model and to apply it to data not yet labeled. The objective was to identify artifact alerts that can be dismissed on-the-fly to reduce the impact of alert fatigue among medical personnel and to enable improvements of the quality of care. We extracted multiple temporal features for each vital sign independently over duration of each alert and a window of 4 minutes preceding its onset. These features included metrics of data density, as well as common moving-window statistics computed for each of the vital timeseries.

Figure 2 shows the RIPR semi-supervised classification model obtained for the RR artifact detection. The features used are the data densities for HR, RR and SpO_2 and the minimum value of RR over a time window of observation. These retrieved models are consistent with the intuition of seasoned clinicians. The accuracy of the model is 97.8%, precision and recall for genuine alert recovery are 97.9% and 99.1% respectively, cross-validated. Some instances were classified by the system as artifacts while domain experts initially considered them to be true alerts. Yet, on a closer visual inspection made possible by the low-dimensional RIPR projections, they exhibited artifact-like characteristics. Further analysis showed that the expert-assigned labels were incorrect.

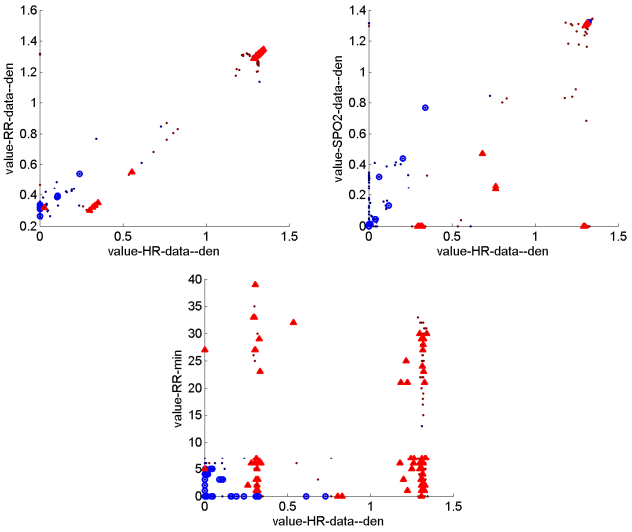

Fig. 2: RIPR for RR alerts. Artifacts: Blue. True instabilities: Red.

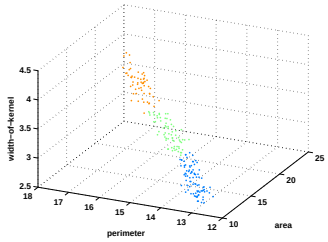

Fig. 3: Clusters mined from the Seeds dataset

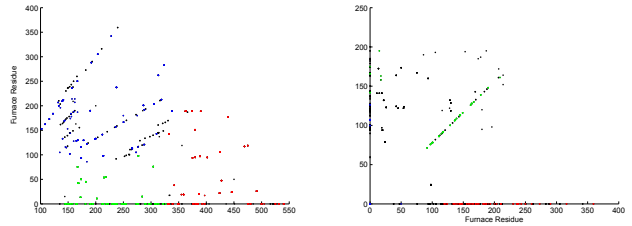

Fig. 4: Clusters induced from the Concrete dataset.

E. Case Study 2: Clustering of UCI Data

We ran RIPR clustering with k-means submodels on two datasets from the UCI repository to demonstrate how patterns in data can be mined with our approach. Figure 3 shows the model recovered from the Seeds dataset. The clustering that RIPR constructs uses the size and shape of seeds to achieve their placement into three categories, clearly visually separated in the figure. The separation according to their aspect ratio is something that one might intuitively expect. Figure 4 shows the two informative projections mined from the Concrete dataset. Here, different concrete mixtures are grouped by their content. While the first projection generates clusters according to the high/low contents of cement and high/low contents furnace residue, the second projection singles out the mixtures that have (1) No fly ash, (2) No furnace residue or (3) Equal amounts of each. The clusters seem to capture what an experimenter might manually label.

V. Conclusions

We formulated the problem of Informative Projection Recovery, and motivated its importance to applications which involve user intervention. We proposed a solution which embeds existing machine learning algorithms to yield models that are intuitive and achieve good performance. Our experiments with synthetic data show that our approach is capable of achieving high precision and recall metrics, induces compact clusters, and yields reliable predictive models. It outperforms standard counterparts when the underlying data has low-dimensional structures. Real-world data examples illustrate how our method enables tangible improvements of practical utility and user acceptance for machine learning based decision support systems.

Acknowledgment

This work has been partially funded by the National Science Foundation (awards 0911032, 1320347) and the National Institutes of Health (R01NR013912).

References

[1] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. *SIGMOD Record*, 27(2):94–105, June 1998.

[2] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh. Clustering with Bregman divergences. *J. Mach. Learn. Res.*, 6:1705–1749, Dec. 2005.

[3] A. Ben-Tal and A. Nemirovski. *Lectures on Modern Convex Optimization*. Society for Industrial and Applied Mathematics (SIAM), 2001.

[4] C. Böhm, K. Kailing, H.-P. Kriegel, and P. Kröger. Density connected clustering with local subspace preferences. In *ICDM,* p. 27–34, 2004.

[5] M. W. Craven and J. W. Shavlik. Extracting tree-structured representations of trained networks. In *NIPS*, pages 24–30, 1995.

[6] L. Devroye, L. Györfi, and G. Lugosi. *A probabilistic theory of pattern recognition*. Springer, 1996.

[7] I. S. Dhillon, Y. Guan, and B. Kulis. Kernel k-means: spectral clustering and normalized cuts. In *Proc. 10th KDD*, pages 551–556, 2004.

[8] P. Domingos. Knowledge discovery via multiple models. *Intelligent Data Analysis*, 2:187–202, 1998.

[9] E. M. Dos Santos, R. Sabourin, and P. Maupin. A dynamic overproduce-and-choose strategy for the selection of classifier ensembles. *Pattern Recognition*, 41:2993–3009, October 2008.

[10] M. Fiterau and A. Dubrawski. Projection retrieval for classification. In *Proc. NIPS*, volume 24, pages 3032–3040, 2012.

[11] J. H. Friedman, J. L. Bentley, and R. A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, 3(3):209–226, Sept. 1977.

[12] Q. Gu, Z. Li, and J. Han. Joint feature selection and subspace learning. In *Int'l Joint Conf. on Artificial Intelligence*, pages 1294–1299, 2011.

[13] H.-P. Kriegel, P. Kröger, M. Renz, and S. Wurst. A generic framework for efficient subspace clustering of high-dimensional data. In *5th IEEE International Conference on Data Mining*, pages 250–257, 2005.

[14] B. Liu, M. Hu, and W. Hsu. Intuitive representation of decision trees using general rules and exceptions. In *17th AAAI*, pages 615–620, 2000.

[15] M. Mampaey, N. Tatti, and J. Vreeken. Tell me what I need to know: Succinctly summarizing data with itemsets. In *KDD,* p. 573–581, 2011.

[16] M. Marchand and M. Sokolova. Learning with decision lists of data-dependent features. *J. Machine Learning Research*, 6:427–451, 2005.

[17] G. Obozinski, B. Taskar, and M. I. Jordan. Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing*, 20(2):231–252, Apr. 2010.

[18] M. J. Pazzani, S. Mani, and W. R. Shankle. Beyond concise and colorful: Learning intelligible rules. In *3rd KDD*, pages 235–238, 1997.

[19] B. Poczós and J. G. Schneider. On the estimation of alpha-divergences. *J. Mach. Learn. Res. – Proc. Track (AISTATS)*, 15:609–617, June 2011.

[20] N. Slonim, G. S. Atwal, G. Tkačik, and W. Bialek. Information-based clustering. *Proc. Nat. Acad. Sci. USA*, 102(51):18297–18302, Dec. 2005.

[21] K. Ting, J. Wells, S. Tan, S. Teng, and G. Webb. Feature-subspace aggregating: ensembles for stable and unstable learners. *Machine Learning*, 82:375–397, 2011.