An Application of Divergence Estimation to Projection Retrieval for Semi-supervised Classification and Clustering

Madalina Fiterau MFITERAU@CS.CMU.EDU
Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213 USA

Artur Dubrawski AWD@CS.CMU.EDU
Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213 USA

Abstract

Decision-support systems enable human users to handle tasks such as classification or clustering. Often, this expert intervention involves simply the validation of an outcome produced by an automated system. In such cases, the decision process must be made transparent and comprehensible. We formulate the projection retrieval problem as an optimization using a divergence-based objective specific to the learning task. The framework presented in the paper uses consistent divergence estimators to extract a set of low-dimensional projections which jointly form a solution to the learning task. The method works under the assumption that each projection addresses a different subspace of the feature space and that it is possible to learn a function selecting the appropriate projection for a given test point. Experiments show that the method recovers the underlying structure of the data and provides low-dimensional views that aid expert assessment.

1. Introduction

In the domain of operations research, systems are designed to provide decision support to human users. Typically, this involves automating tasks such as grouping or classification while offering the experts insight into how the learning task was solved and how the model is applied to new data. An ideal scenario for a multitude of practical applications is the following: a domain expert provides the system with training data for some learning task; the system learns a model for the task which uses only simple projections; a user provides queries (test points); for a given query point, the system selects the projection that is expected to be the best-performing for the query point; the system displays the outcome as well as a representation of how the task was performed within the selected projection.

The problem of recovering simple projections for the classification task has been formalized in (Fiterau & Dubrawski, 2012). The RECIP algorithm proposed by the authors uses point estimators for conditional entropy and recovers a set of low-dimensional projections which classify queries using non-parametric discriminators in an alternate fashion - each query point is classified based on one of the projections in the retrieved set depending on the characteristics of the point.

The current paper extends the concept to semi/unsupervised learning tasks by expressing the Informative Projection Retrieval (IPR) problem using a more generic formalization applicable to any learning task for which there exists a consistent loss function. The loss for the semi-supervised classification and clustering tasks we are exemplifying in this paper are based on local divergence estimators. We provide an evaluation of the different divergence-based objectives in terms of the recovery of underlying structure in data and by quantifying the perfomance of the resulting models for their respective tasks.

We also augment the system in the direction of providing the users with an overview of projection 'agreement'. Specifically, we compare each of the recovered projections with all existing projections of same dimensionality in the model in order to identify consensus, disagreement or alternative explanations. To this purpose we also use divergence-based metrics - their properties and significance are dicussed in Section 4.

Proceedings of the 30^{th} International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28.

2. Related Work

The use of dimensionality reduction techniques is a common preprocessing step in applications where the use of simplified classification models is preferable. Methods that learn linear combinations of features, such as Linear Discriminant Analysis, are not ideal for the task considered here, since we prefer to rely on the dimensions available in the original feature space. Feature selection methods, such as e.g. lasso, are suitable for identifying sets of relevant features, but do not consider interactions between them. Our work fits the areas of class dependent feature selection and context specific classification, highly connected to the concept of Transductive Learning (Gammerman et al., 1998). Other context-sensitive methods are Lazy and Data-Dependent Decision Trees, (Friedman et al., 1996) and (Marchand & Sokolova, 2005) respectively. (Ting et al., 2011) introduce the Feating submodel selection relies on simple attribute splits followed by fitting local predictors. (Obozinski et al., 2010) present a subspace selection method in the context of multitask learning. There are also numerous methods that clustering in low-dimensional subspaces (Agrawal et al., 1998; Kriegel et al., 2005). The work in (Bohm et al., 2004) uses the concept of local subspace preferences, which captures the directions of high point density. Divergence-based methods were previously used for clustering. The work by (Banerjee et al., 2005) use Bregman Divergence to unify centroid-based approaches. Clustering based on divergence metrics is also used in (Slonim et al., 2005; Liu et al., 2012). Additionally, divergences have been used for other machine learning tasks such as segmentation (Vemuri et al., 2011). Spectral clustering obtains low-dimensional projections, however, it relies on a similarity matrix (Dhillon et al., 2004).

Unlike most of those approaches, our method is designed to retrieve subsets of the feature space designed for use in a way that is complementary to the basic task at hand (classification or clustering) while providing query-specific information.

3. Informative Projection Retrieval

This section describes the formulation of the Informative Projection Retrieval (IPR) problem, then describes an algorithmic framework generalized from the RECIP procedure in (Fiterau & Dubrawski, 2012).

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 3.1 follow the setup for the RECIP procedure, the main improvement being the formalization of the problem for learning tasks other than classification and the capability to include learners of any given class while RECIP only considered nonparametric classifiers. Section 3.3 shows how divergence estimators fit into this framework.

3.1. Formalization of Projection Retrieval

Let us 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 or regression. The task solver for the learning task is selected from a solver class $\mathcal{T} = \{f : \mathcal{X} \to \mathcal{Y}\}$, were the risk for the solver class $\mathcal{T}$ is defined in terms of the loss ℓ as

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

We define the optimal solver for the task as

$$\tau^* \stackrel{def}{=} \arg\min_{\tau \in \mathcal{T}} \mathcal{R}(\tau, \mathcal{X})$$

We will use the notation $\tau_{\{X\}}$ to indicate the task solver from class $\mathcal{T}$ which is obtained by minimizing the empirical risk over the training set X.

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

We formalize the type of model that our IPR framework will construct. Class $\mathcal{M}$ contains models that have a set Π of projections of maximum dimension d, a set τ of task solvers and a selection function g:

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

The set $\mathbf{\Pi}$ contains all the axis-aligned projections. However, the subset $\Pi \subseteq \mathbf{\Pi}$ contained by $\mathcal{M}$ contains only projections that have at most d features. The parameter d is dictated by the application requirements - values of 2 or 3 are expected since they permit users to view the projections. The selection function g picks the adequate projection π and its corresponding task solver τ to handle a given query x.

Based on this model, we derive a composite solver which combines the benefits of the solvers operating on the low-dimensional projections. The loss of this solver can be expressed in terms of the component losses.

$$\begin{aligned} \tau_{\mathcal{M}}(x) &= \tau_i(\pi_i(x)) \quad \text{where } g(x) = i \\ \ell(x, \tau_{\mathcal{M}}) &= \ell(\pi_{g(x)}(x), \tau_{g(x)}) \end{aligned}$$

where $g(x)$ represents the index of the solver for point x is handled and $\pi_i(x)$ is the projection of x onto π_i. Optimizing over the *Informative Projection Model* class $\mathcal{M}$, the IPR problem for learning task $\mathcal{T}$ can be formulated as a minimization of the expected loss:

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

Since we are dealing with an unsupervized problem in terms of the selection function, there are limitations on its learnability. One example in which recovery is successful is a dataset containing regulatory features:

$$\forall x \exists x^j \text{ s.t. with } x^j \in A \ , \ \tau^*(x^1 \dots x^m) = \tau^*_A(x^{i_1} \dots x^{i_d})$$

In the example above, for a given point x, j is the regulatory feature. The interpretation is that for all points x whose j^{th} feature is in the set A, the targeted task can be optimally performed by the task solver τ^*_A by considering only features $\{i_1 \dots i_d\}$ of x. The task solver τ^*_A is only trained over samples for which $x^j \in A$.

3.2. Projection Recovery Framework (RIPR)

The starting point of the algorithm is writing the empirical version of (1) as a combinatorial problem over multiple projections. The algorithm is designed under the assumption of the existence of low-dimensional embeddings that enable capturing accurate models for the target task. In conformance with this assumption, every sample point x_i can be dealt with by just one projection π_j, in other words $g(x_i) = j$. We model this mapping as a binary matrix B:

$$B_{ij} = I[g(x_i) = j].$$

We write the minimizers of the risk and empirical risk:

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

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

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

$$\exists \hat{\ell} \text{ s.t.} \quad \forall x \in \mathcal{X} \quad \tau \in \mathcal{T} \quad \hat{\ell}(x, \tau) \overset{n \to \infty}{\to} \ell(x, \tau) \quad (2)$$

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

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

$$= \arg\min_{\mathcal{M}, |\Pi| < |\mathbf{\Pi}|} \sum_{i=1}^{n} \sum_{j=1}^{|\mathbf{\Pi}|} B_{ij} L_{ij} \quad , \quad L_{ij} = \hat{\ell}(\pi_j(x_i), \tau_i)$$

The loss estimators L_{ij} are computed for every point on every subspace of up to the user-specified size 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 of size up to d, $L^*_i \overset{def}{=} \min_j L_{ij}$, the operator $\odot$ isojections of size $\leq d$ 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 optimization procedure is described in detail in (Fiterau & Dubrawski, 2012), the key difference to its use here being that we are computing the loss matrix L differently. The technique resembles the adaptive lasso, which gradually reduces the number of non-zero columns in B until a small and stable set of projections is converged upon. As illustrated in 1, the procedure uses the multiplier δ to gradually bias projection selection towards projections that not only perform well but also suit a large number of data points.

Algorithm 1 RIPR Framework

$\delta = [1 \dots 1]$
repeat
  $B = \arg\min_B ||L^* - L \odot B||_2^2 +$
    $\lambda_1 \sum_{j=1}^{d^*} ||B_{\cdot,j}||_{\ell_1} + \lambda_2 |B\delta|_{\ell_1}$
      subject to
      $||B_{k,\cdot}||_{\ell_1} = 1 \qquad k = 1 \dots n$
  $\delta_j = ||B_{\cdot,j}||_{\ell_1} \qquad j = 1 \dots d^*$ (update muliplier)
  $\delta = (||\delta||_{\ell_1} - \delta)/||\delta||_{\ell_1}$
until δ converges
$\Pi = \{\pi_i; \quad |B_{\cdot,i}|_{\ell_1} > 0 \quad \forall i = 1 \dots d^*\}$
return Π

3.3. Divergence-based Objectives

Next, we show how to formulate IPR for different tasks. The aim is to find projections that are informative for a task given no knowledge of the actual class of solvers that will be be used. For instance, we might be given a high dimensional classification problem for which to find a set of low-dimensional projections without considering the classifier - linear, kernel-based or nonparametric - that will ultimately

be trained. Therefore, we incline to use nonparametric loss functions. The performance of the method will depend on the estimator's rate of convergence.

3.3.1. Classification

The IPR problem for classification is the topic of the previous work in (Fiterau & Dubrawski, 2012), we state some results obtained in the paper.

Proposition 3.1. *Given a variable $X \in \mathcal{X}$ and a binary variable Y, X sampled from the mixture model*

$$f(x) = p(y=0)f_0(x) + p(y=1)f_1(x) = p_0 f_0(x) + p_1 f_1(x)$$

$$H(Y|X) = -p_0 \log p_0 - p_1 \log p_1 - D_{KL}(f_0||f) - D_{KL}(f_1||f)$$

The conditional entropy over the points assigned to projection π_j is then shown to be estimated as follows:

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

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

Above, the notation $\pi(X)$ is used to represent the projection of vector X onto π. Also, we will use X_γ to represent the subset of the sample for which the label is γ. The notation $X \setminus x$ refers to the sample obtained when removing point x from X. The function $\nu_k(x, X)$ represents the k^{th} distance from point x to its k-nearest-neighbor from the sample X. $\tau^k_{\pi_j}$ is the k-nn classifier on projection π_j.

This result is obtained by using the Tsallis α-divergence estimator introduced in (Poczos & Schneider, 2011) and yields an estimator for the loss when the target task is binary classification. α is a constant set to a value close to 1 (such as 0.95) and $|\pi_j|$ is the dimensionality of the subspace π_j.

3.3.2. Semi-supervised Classification

We propose to extend the objective in 3.3.1 to semi-supervised classification. Assume we are given a semi-supervised learning 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 separator in a low-dimensional sub-space of the features such that the labeled samples are correctly classified and, at the same time, the unlabeled data allows substantial separation. That is, very few unlabeled data points remain between the clusters representing different classes.

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 points that are about equally-distanced from the point-clouds of each class - call this $\hat{\mathcal{R}}(X_u, \tau^k_\pi)$. We'll use the notation $\pi(X)$ to represent the projections of a set of points X.

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

It to be noted that, for all these learning tasks, the typical convergence issues encountered with nearest-neighbor estimators are bypassed because of the low-dimensionality of the projections.

3.3.3. Clustering

Unlike classification and regression, most types of clustering make it problematic to devise an objective that can be evaluated at every point, mainly because an overview of the data is needed for clustering, rather than local information. Distribution-based as well as centroid-based clustering fit a model on the entire set of points. This is an issue for the IPR problem because it is not known which data should be used for the set of submodels. To bypass this problem, we first learn the projections and the points corresponding to them using density-based clustering - which admits a local loss estimator. We then learn a clustering model (solver) on each projection using only the assigned points.

Density-based clustering uses areas of higher density than the remainder to group points. To achieve IPR for clustering, we consider the negative divergence, in the neighborhood of each sample, between the distribution from which the sample X is drawn and a uniform distribution on $\mathcal{X}$. Let be a sample of size n drawn uniformly from $\mathcal{X}$. Again, we use the nearest-neighbor estimator converging to the KL divergence. τ^{clu}_i is some clustering technique such as K-means.

$$\begin{aligned} \hat{\mathcal{R}}_{clu}(\pi_i(x), \tau^{clu}_i) &\rightarrow -KL(\pi_i(X)|||\pi_i(U)) \\ \hat{\ell}_{clu}(\pi_i(x), \tau^{clu}_i) &\approx \Big(\frac{d(\pi_i(x), \pi_i(X))}{d(\pi_i(x), U)}\Big)^{|\pi_i|(1-\alpha)} \end{aligned}$$

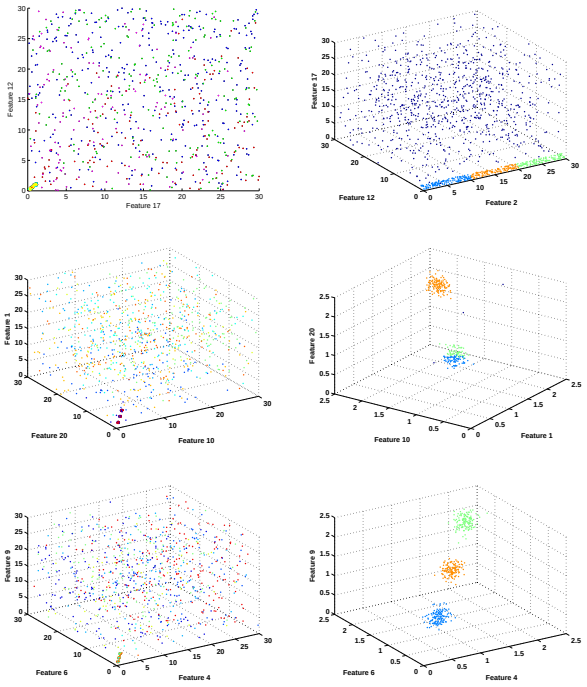
Much like the classification tasks, RIPR can be used with any clustering algorithm. We here focus here on K-means since it is a frequently-used method and, as a non-parametric method, it is more in tune with the non-parametric divergence estimators used to select

the projections. Running RIPR with other clustering algorithms is still work in progress.

3.4. Artificial Data Example

We now illustrate, on an artificial dataset, how RIPR clustering with K-means can improve over runnning K-means on all the features. The data we used has 20 features, and it contains three Gaussian clusters on each of its informative projections. The informative projections are $\{17, 12\}$, and $\{10, 20, 1\}$ and $\{4, 6, 9\}$. Figure 1 shows, on the left side, the clustering obtained by K-means, projected on the sets of informative features. On the right side, we plotted the projections as obtained by RIPR and the clustering of points in each of them. Every cluster is colored differently, with black points in RIPR plots representing the data not assigned to that projection. The number of clusters is picked by cross-validation for both k-means and RIPR.

Figure 1. Example Clustered Data for K-means (left) and RIPR (right). The plots of the left show the clustering obtained with K-means projected on the informative sets of features. The plots on the right show the clustering obtained by RIPR on the recovered sets of features.



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. This aspect is investigated in the next section.

4. Determining Projection Consensus

The set of projections recovered by the RIPR Framework is comprehensive in terms of dealing with the feature space. However, in certain cases, providing the user with multiple views of the data is useful. We propose an extension which analyzes, for each projection π_i in the selected set, its consensus with the other projections and identifies agreeing projections - which support the separation induced by π_i. Also, it picks out projections that lead to highly incompatible or contradictory decisions when compared to π_i.

We rely on information theoretical concept to compute projection agreement. We'll compute two metrics between π_i - a projection in the selected set - and π_j - any other axis-aligned. The first metric, called selection divergence, evaluates how closely packed the points classified with π_i when projected onto π_j:

$$SDiv_X(\pi_i, \pi_j) = H(I[g(X) = i]|\pi_j(X))$$

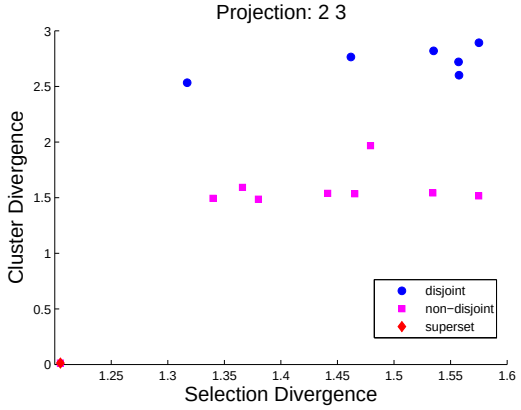
The notation $I[g(X) = i]$ is used to represent a labeling according to which of the samples have π_i assigned as their appropriate projection. To estimate this quantity, we can use the formulas presented in 3.3.1 for conditional entropy.

A second metric, called clustering divergence, evaluates how the points labeled by projection π_i are distributed in projection π_j. We use the notation $X_{[g(X)=i]}$ to refer to the points labeled by π_i and $\tau_i(\pi_i(X_{[g(X)=i]}))$ to refer to the vector of labels assigned by the task solver on π_i.

$$KDiv_X(\pi_i, \pi_j) = H(\tau_i(\pi_i(X_{[g(X)=i]}))|\pi_j(X_{[g(X)=i]}))$$

Figure 2 shows these metrics computed for a 2-dimensional projection π picked by RIPR compared to all other 2-dimensional projections in that dataset. $SDiv_X(\pi, \pi_j)$ is on the x-axis and $SDiv_K(\pi, \pi_j)$ is on the Y axis. It is apparent that the projections that have features in common with π (i.e. in agreement with π) are shown as closer to it on the plot.

Figure 2. Scatterplot of Projection Consensus metrics between the top projection mined by RIPR from an artificial dataset



5. Evaluation

This section illustrates the capabilities of the RIPR framework in recovering the underlying patterns in data and in training well-performing classification and clustering systems. Before explaining the results, we must make a note on computational complexity. While it is true that the method requires computing the loss for every combination of features, the point losses can be computed independently of everything else. Thus, the loss computation step is trivially parallel. In the adaptive lasso procedure, we can gradually discount non-informative projections, thus reducing the dimensionality of the optimization problem. Though so far we have tried the method on relatively small datasets, the improvements we mentioned allow the increase of the dimensionality by one order of magnitude.

We point out that, since RIPR is a generalization of the RECIP algorithm presented in (Fiterau & Dubrawski, 2012), RECIP is equivalent to RIPR when the task considered is classification and the task solvers are nearest-neighbor classifiers. We are not improving the performance for this case, so comparing against RECIP would not make much sense. However, we do present the notable benefits of RIPR over RECIP: its ability to solve other tasks than classification and to use existing classifiers/clustering algorithms.

5.1. Experiments on Artificial Data

Clustering has been heavily researched, resulting in a vast number of succesful techniques. We do not seek to supplant these, but rather to facilitate their use in a user-driven context. RIPR was developed such that we may apply any existing clustering or classification algorithm, maintaining its high performance while satisfying the stringent requirements of working with only 1,2 or 3-dimensional models. In this section, we show that, RIPR with K-means performs even better than the standard K-means by leveraging structure in data.

Table 1. Mean distances to cluster centroids for artificial data queries for RIPR and K-means. The RIPR0 column represents the distance to RIPR centroids on all dimensions. P represents the true number of informative projections, while R is the number of incoherent points.

P	R	Dist RIPR	Dist RIPR0	Dist K-means
2	0	50.94	9485.27	9424.34
2	100	387.19	10955.60	10824.42
2	200	625.50	12193.29	12077.29
2	400	1026.71	14689.88	14580.36
2	800	1848.42	19596.55	19519.62
3	0	81.42	21621.72	21324.13
3	100	573.62	23617.22	23236.56
3	200	914.30	25524.81	25174.39
3	400	1443.52	29177.94	28788.38
3	800	1443.52	29177.94	28788.38
5	0	252.36	59739.11	59061.49
5	100	990.57	63354.76	62101.09
5	200	1493.19	66310.79	65191.95
5	400	2326.89	72611.39	71433.86
5	800	2326.89	72611.39	71433.86

An immediate way to assess the performance of RIPR for clustering is by attempting to recover patterns from artificial data. We generated datasets with 20 features, P informative projections and k clusters on each projection. Each projection allows the clustering of a subset of the data (400 points). An example of how the data looks like was shown in Figure 1. Not all the points can be clustered - R of them are incoherent and do not follow the model, increasing the problem complexity. We trained RIPR and K-means models and evaluated their performance in grouping query points. K-means and RIPR parameters are obtained by cross-validation. The evaluation metric is the sum of distances from the test points to their assigned centroids. As shown in Table 1, the RIPR model distances are much smaller compared to the K-means ones (last column). It is debatable whether this is due to an improved clustering or simply because the dimensionality is reduced. For this reason, we also computed the distances from the test points to the centroids considering all dimensions, not just those used in the clustering. This value is presented in the RECIP0 column. Clearly, it is very difficult for this value to be less than the K-means sum of distances since K-means actively optimizes this metric over all the features. It is also a very pessimistic measure since the shape of RIPR models, considered in all dimensions, is cylindrical.

Table 2. RIPR clustering evaluation continued. This table compares the logarithm of the volume for the RIPR model and the K-means model. Also shown are the IP retrieval precision and recall.

P	R	LogVol RIPR	LogVol Kmeans	RIPR Prec.	RIPR Rec.
2	0	62.75	66.07	1.00	1.00
2	100	63.48	67.23	1.00	1.00
2	200	63.49	67.23	1.00	1.00
2	400	63.58	66.88	1.00	1.00
2	800	63.65	67.04	1.00	1.00
3	0	70.92	74.62	1.00	1.00
3	100	71.76	75.04	1.00	1.00
3	200	71.93	75.39	1.00	1.00
3	400	71.99	75.06	1.00	1.00
3	800	71.99	75.06	1.00	1.00
5	0	81.11	85.40	1.00	1.00
5	100	82.30	85.68	0.98	1.00
5	200	82.46	85.70	0.98	1.00
5	400	82.57	85.71	1.00	1.00
5	800	82.57	85.71	0.98	0.98

We also considered a different metric of success - the volume of the resulting clusters over all the features. This comparison is fair because the volumes have the same dimensionality. This is summarized in Table 2. It is apparent that RIPR obtains slightly compacter models than K-means, built based on only a fraction of the features. It is to be noted that the total number of centroids is roughly the same for K-means and RIPR, so the difference in volume is genuinely due to an improvement in clustering.

We were also interested in whether subspace recovery was possible, so we increased the number of incoherent points and informative projections until RIPR ceased to provide perfect precision and recall. For our datasets, RIPR had no problem recovering the correct models until the number of informative projections reached 5. Even past this point, we can still expect good retrieval rates, as aparent from Table 2. All results on artificial data are averages over 10 datasets coming from the same distribution.

Another aspect is the influence of the true number of clusters and informative patterns on the performance of RIPR as compared to that of K-means. Table 3 summarizes the results of these experiments. What we observed was that, with the existence of more clusters, both algorithms decreased the expected distance to the centroid. However, as expected, for RIPR models the decrease was drastic. RIPR does consistently better than k-means in terms of the volume, and, as expected, there is a general tendency to increase the cluster volume as more and more informative projections are added. For a fixed P, RIPR is slightly more sensitive in terms of the volume than k-means, which is a natural consequence of it being more in tune with the underlying model.

Table 3. Variation of distances and volumes with the variation of the number of IPs (P) and the number of clusters K in each projection.

P	K	Dist RIPR	Dist RIPR0	Dist KM	LVol RIPR	LVol KM
2	2	865	12,318	12,262	63.12	67.16
2	3	622	12,203	12,058	63.47	66.79
2	5	440	12,060	11,846	63.96	66.92
2	7	375	11,909	11,662	64.28	66.71
3	2	1,344	25,704	25,422	71.55	74.77
3	3	872	25,472	25,087	71.84	75.46
3	5	648	25,247	24,711	72.41	75.47
3	7	530	24,979	24,449	72.69	74.94
5	2	2,683	66,801	65,984	82.08	85.79
5	3	1,484	66,352	65,224	82.41	85.57
5	5	1,065	65,419	63,983	82.89	85.41
5	7	842	64,946	63,238	83.28	85.38
7	2	4,621	127,558	125,451	89.01	92.68
7	3	2,174	126,309	123,863	89.47	92.58
7	5	1,480	124,436	121,442	89.90	92.33
7	7	1,238	123,151	120,123	90.11	92.36

Table 4. Clustering results on real-world datasets

UCI	*Dist RIPR*	*Dist RIPR0*	*Dist KM*	*LVol RIPR*	*LogVol KM*
Seeds	16	174	107	7.68	9.70
Libras	9	620	265	-5.80	7.26
Boone	125	2,18e4	1,15e4	240.00	248.15
Cell	40,877	18,8e5	8,1e5	54.69	67.68
Conc.	1,370	68,865	55,594	49.24	52.75

5.2. Experiments on Real Data

We verified that our findings also hold for real-world data by running RIPR and K-means on datasets from the UCI repository. We notice that the volume obtained by RIPR clustering is consistently smaller and the expected distances are several orders of magnitude smaller than the ones obtained by K-means.

5.3. Examples of Recovered Subspaces

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 separated in the figure. The separation according to their aspect ratio is something that one might intuitively expect.

Figure 3. Clusters mined from the Seeds dataset

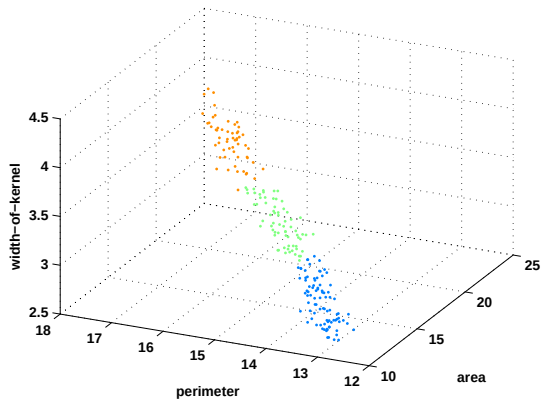


Figure 4 shows the two informative projections mined from the Concrete dataset. Here, different concrete mixtures are grouped by their content. While the first projection generates clusters according to the high/low contents of cement and high/low contents furnace residue, the second projection singles 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.

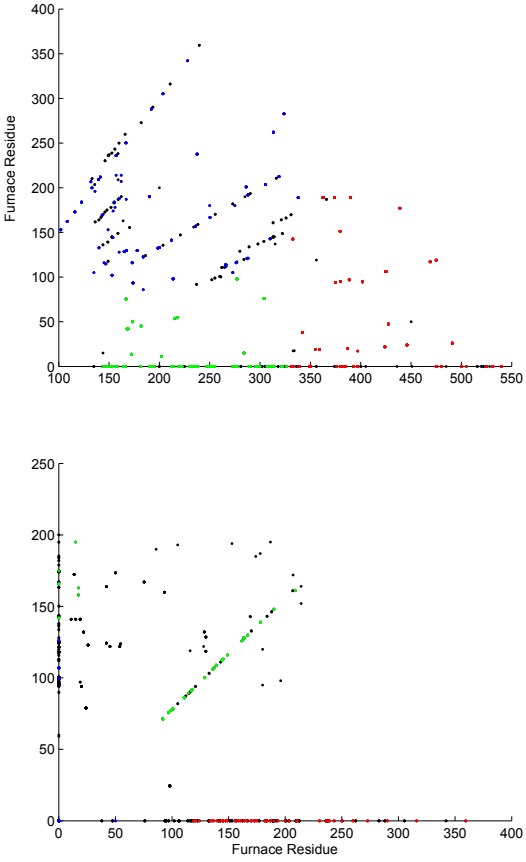
6. Conclusions

The current paper presents the problem of Informative Projection Retrieval and motivates its importance to applications which require user intervention. The solution we offer uses divergence estimation to obtain informative projections while enabling existing algorithms to provide models that are both intuitive and achieve good performance. Our experiments on artificial data show that our method, Regression-based IPR (RIPR), attains high values for precision and recall and mines compact clusters. The real-data examples show how our method maintains high performance while recovering compact and realistic models.

Acknowledgments

This material is based upon work supported by the NSF, under Grant No. IIS-0911032.

Figure 4. Clusters mined from the Concrete dataset



References

Agrawal, Rakesh, Gehrke, Johannes, Gunopulos, Dimitrios, and Raghavan, Prabhakar. Automatic subspace clustering of high dimensional data for data mining applications. *SIGMOD Rec.*, 27(2):94–105, June 1998. ISSN 0163-5808. doi: 10.1145/276305.276314. URL http://doi.acm.org/10.1145/276305.276314.

Banerjee, Arindam, Merugu, Srujana, Dhillon, Inderjit S., and Ghosh, Joydeep. Clustering with bregman divergences. *J. Mach. Learn. Res.*, 6:1705–1749, December 2005. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1046920.1194902.

Bohm, C., Railing, K., Kriegel, H.-P., and Kroger, P. Density connected clustering with local subspace preferences. In *Data Mining, 2004. ICDM '04. Fourth IEEE International Conference on*, pp. 27–34, 2004. doi: 10.1109/ICDM.2004.10087.

Dhillon, Inderjit S., Guan, Yuqiang, and Kulis, Brian. Kernel k-means: spectral clustering and normalized

cuts. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pp. 551–556, New York, NY, USA, 2004. ACM. ISBN 1-58113-888-1. doi: 10.1145/1014052.1014118. URL http://doi.acm.org/10.1145/1014052.1014118.

Fiterau, M. and Dubrawski, A. Projection retrieval for classification. In *Advances in Neural Information Processing Systems (NIPS 2012)*, volume 24, 2012.

Friedman, Jerome H., Kohavi, Ron, and Yun, Yeogirl. Lazy decision trees, 1996.

Gammerman, A., Vovk, V., and Vapnik, V. Learning by transduction. In *In Uncertainty in Artificial Intelligence*, pp. 148–155. Morgan Kaufmann, 1998.

Kriegel, Hans-Peter, Krger, Peer, Renz, Matthias, and Wurst, Sebastian. A generic framework for efficient subspace clustering of high-dimensional data. In *IN: PROC. ICDM*, pp. 250–257. IEEE Computer Society, 2005.

Liu, Meizhu, Vemuri, B.C., Amari, S.-I., and Nielsen, F. Shape retrieval using hierarchical total bregman soft clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(12):2407–2419, 2012. ISSN 0162-8828. doi: 10.1109/TPAMI.2012.44.

Marchand, Mario and Sokolova, Marina. Learning with decision lists of data-dependent features. *JOURNAL OF MACHINE LEARNING REASEARCH*, 6, 2005.

Obozinski, Guillaume, Taskar, Ben, and Jordan, Michael I. Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing*, 20(2):231–252, April 2010. ISSN 0960-3174. doi: 10.1007/s11222-008-9111-x. URL http://dx.doi.org/10.1007/s11222-008-9111-x.

Poczos, B. and Schneider, J. On the estimation of alpha-divergences. *AISTATS*, 2011.

Slonim, Noam, Atwal, Gurinder S., Tkačik, Gašper, and Bialek, William. Information-based clustering. *Proceedings of the National Academy of Sciences of the United States of America*, 102(51):18297–18302, December 2005. ISSN 1091-6490. doi: 10.1073/pnas.0507432102. URL http://dx.doi.org/10.1073/pnas.0507432102.

Ting, Kai, Wells, Jonathan, Tan, Swee, Teng, Shyh, and Webb, Geoffrey. Feature-subspace aggregating: ensembles for stable andunstable learners. *Machine Learning*, 82:375–397, 2011. ISSN 0885-6125. URL http://dx.doi.org/10.1007/s10994-010-5224-5. 10.1007/s10994-010-5224-5.

Vemuri, Baba C., Liu, Meizhu, ichi Amari, Shun, and Nielsen, Frank. Total bregman divergence and its applications to dti analysis. *IEEE Trans. Med. Imaging*, 30(2):475–483, 2011. doi: http://dx.doi.org/10.1109/TMI.2010.2086464.