Discovering Compact and Informative Structures through Data Partitioning

Madalina Fiterau
mfiterau@cs.cmu.edu

27 October 2014

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Artur Dubrawski, chair, Carnegie Mellon University
Geoff Gordon , Carnegie Mellon University
Alex Smola, Carnegie Mellon University
Andreas Krause, ETH Zürich

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Keywords: informative projection recovery, cost-based feature selection, ensemble methods, data partitioning, active learning, clinical data analysis, artifact adjudication, nuclear threat detection

This research was done in collaboration with:
Karen Chen, CMU, Auton Lab
Gilles Clermont, University of Pittsburgh
Artur Dubrawski, CMU, Auton Lab
Nick Gisolfi, CMU, Robotics
Geoff Gordon, CMU, MLD
Mathieu Guillaume-Bert, CMU, Auton Lab
Marilyn Hravnak, University of Pittsburgh
Michael R. Pinsky, University of Pittsburgh
Alex Smola, CMU, MLD
Donghan Wang, CMU, Auton Lab

Abstract

In many practical scenarios, prediction for high-dimensional observations can be accurately performed using only a fraction of the existing features. However, the set of relevant predictive features, known as the sparsity pattern, varies across data. For instance, features that are informative for a subset of observations might be useless for the rest. In fact, in such cases, the dataset can be seen as an aggregation of samples belonging to several low-dimensional sub-models, potentially due to different generative processes. My thesis introduces several techniques for identifying sparse predictive structures and the areas of the feature space where these structures are effective. This information allows the training of models which perform better than those obtained through traditional feature selection.

We 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. Our solution to this problem is a regression-based algorithm that identifies informative projections by optimizing over a matrix of point-wise loss estimators. It generalizes to a number of machine learning problems, offering solutions to classification, clustering and regression tasks. Experiments show that our method can discover and leverage low-dimensional structure, yielding accurate and compact models. Our method is particularly useful in applications involving multivariate numeric data in which expert assessment of the results is of the essence. Additionally, we developed an active learning framework which works with the obtained compact models in finding unlabeled data deemed to be worth expert evaluation. For this purpose, we enhance standard active selection criteria using the information encapsulated by the trained model. The advantage of our approach is that the labeling effort is expended mainly on samples which benefit models from the hypothesis class we are considering. Additionally, the domain experts benefit from the availability of informative axis aligned projections at the time of labeling. Experiments show that this results in an improved learning rate over standard selection criteria, both for synthetic data and real-world data from the clinical domain, while the comprehensible view of the data supports the labeling process and helps preempt labeling errors.

The focus of forthcoming research is cost-sensitive feature selection, in the context of data partitioning. We consider the process used to generate the features, as well as their reliability and interdependence to reduce the overall cost enquired by prediction. Typically, our applications rely on a core set of features obtained through expensive measurements, enhanced using transformations derived from one or several core features. Our preliminary results show that leveraging this structure results in more powerful classifiers without an increase in the cost of feature acquisition. The crux of our proposed technique attempt to leverage the submodular cost and the redundancy of the features by generating penalties according to the structure of the dependency graph. We will then develop online, adaptive policy-learning optimization procedures for feature selection with submodular cost constraints. We will first consider the batch mode setting and learn a model that maps samples to the appropriate feature subset, achievable by maximizing a submodular objective. The aim is to then efficiently update this mapping as more data becomes available; the main challenge is the trade-off between flexibility and robustness. The result we aim for is a framework which dynamically changes the features used in the classification process, useful in the case when there is a constraint on the number of readings which can be performed.

Acknowledgments

This material is based upon work supported by the National Science Foundation, under Grants No. IIS-0911032, IIS-1320347 and the National Institutes of Health R01NR013912.

Contents

Chapter 1

Introduction to Sparse Predictive Structures with Data Partitioning

1.1 Motivation and application requirements

Feature selection is an essential part of model learning for high-dimensional data, especially when few samples are available. Standard approaches to feature selection do not always yield concise models which accurately reflect the underlying structure of the data, mainly because they target the selection of a globally-useful set of features without accounting for the characteristics of individual samples. At the other end of the spectrum, recent advances into query-specific models with feature selection such as localized feature selection and locally-linear embeddings leverage neighborhood information in order to generate a plethora of models, each tailored to a diminutive portion of the feature space.

There are cases in which neither of these two extremes provides a satisfactory solution. On one hand, shoe-horning the entire dataset into the same low-dimensional model through techniques such as the lasso runs the risk of bringing unnecessary features into the prediction process for some of the samples, which could hurt accuracy. On the other hand, local models are prone to overfitting, have limited applicability and risk introducing needless complexity. All the while, neither captures a compact but comprehensive picture of the dataset, as sought by domain experts. My thesis explores the idea of building small ensembles of low-dimensional components (sub-models) which are applicable to significant subsets of data.

To exemplify, consider a medical application where existing vital sign readings, signals derived from them and a number of other contextual features are used to predict a potentially multivariate output signal such as diagnostics or health-status change alerts. The input space is extensive, containing, at the very least, the readings computed within a window of a few minutes with their corresponding statistics. Each event of interest needs to be manually labeled by clinicians, which requires considerable time and effort, yielding a short supply of labeled data. Given the high feature-to-sample ratio (the problem could even be underdetermined), feature selection is necessary. However, we expect that patients suffer from different underlying conditions and have different characteristics, which is why having several sparse models which are used alternatively, rather than a single generic one, makes more sense. Standard feature selection could pinpoint that blood sugar level is relevant to predicting heart failures. In contrast, a small ensemble model can also identify the conditions under which the feature affects the prediction. For instance, we might find that blood sugar level is only a factor in heart failure prediction when an affine combination of the blood pressure, heart rate and risk of diabetes is above a certain threshold.

As an added incentive, small ensembles of low-dimensional models are also amenable to visualization. This is particularly appealing for applications where human operators have to gain an understanding of the data, and/or quickly validate the system-made predictions. An example of such an application is the detection of nuclear threats at border control points based on vehicle characteristics and measured characteristics of emitted radiation. The automated threat detection system assigns a threat/non-threat label to each vehicle, but it is ultimately up to the border control agents to permit/deny entry or submit the vehicle to further verification. Establishing confidence in the system's decision, if possible, is an important aspect of this application, and can be achieved by providing a visual representation of the classification process. To our ensemble-building methods, this translates as an upper

bound on the dimensionality of the components.

Our proposed family of methods works under the assumption that groups of samples can be classified with different small subsets of features. The aim is to uncover the informative sparsity patterns across the feature space, provided that the changes in feature relevance can also be characterized through sparse functions. We propose to achieve this by training ensembles of low-dimensional components such that every sample can be handled using one of these sub-models or using a sparse mixture of them. We assume no prior knowledge of the sample groups, which could overlap. The assignment of samples to sub-models and the dimensionality reduction for the learners on the sub-models are performed jointly, avoiding the pitfalls of EM-like approaches.

1.2 Scope and novelty of Proposed Approach

To address the demand for concise, interpretable and visualizable models, we develop a framework which recovers compact ensembles, consisting of solvers (which can be regressors or classifiers) trained on what we call 'informative projections' [18, 19]. An Informative Projection is a low-dimensional transformation of the features, where the learning task can be accurately and reliably solved for a group of samples. We obtain these models through convex procedures, avoiding the issues typically encountered with mixture formulations by estimating the performance of low-dimensional solvers on the training data. The low-dimensional projections responsible for each part of the feature space are selected through an optimization which factors in the appropriate sparsity, smoothness and cost constraints over the parameters. Conceptually, we are combining the flexibility of hierarchical latent variable models [8] and sparse mixture models [27] with the convex formulations and the theoretical guarantees inherent to sparse structured learning [1, 28].

One of the novel aspects of our approach to building compact ensembles is the computation of a matrix which estimates the performance of the low-dimensional solvers at each sample, typically using some non-parametric divergence-based estimator. Once this loss matrix is obtained, it is used in a convex program which optimizes the empirical risk given the established model class. The procedure is detailed in the following chapter. A prerequisite for this type of method is that the learning task admits risk-consistent loss estimators. The only other established methods which learn models resembling those we seek involve non-convex learning procedures to obtain sparse mixtures, such as the method introduced by Larsson and Ugander [31] for MAP inference with a sparsity-inducing generalization of the Dirichlet prior.

Since the overall objective is to obtain a compact representation of the data, the size of the ensemble should be constrained. Determining the number of sub-models intrinsic to a dataset is a key model-selection challenge, which we address through regularization by adding component-wise sparsity penalties. To further compress the model, each component in the ensemble will be low-dimensional, with sparsity being the most favorable option. Regularization is also used to reduce component dimensionality, with the caveat that, in some scenarios, additional restrictions will be imposed. For instance, if human-interpretable visualization is desired, each component would only use up to three features. The components learned with our method will differ significantly either in terms of their sparsity patterns or their parameters, with the discrepancies increasing as the number of sub-models becomes more limited. The range varies between ensembles with few, very different components and larger ensembles where some characteristics (features) can be shared across the components.

During the ensemble learning process, samples are assigned to the components as the sub-models are being built. Each sample can be allocated to one sub-model, thus achieving a partitioning of the feature space, or to a very small number of them, similar to sparse mixture models. Conceptually, the partitioning variant makes it easier for human users to understand the trained model and to follow the handling of test queries. However, enforcing a hard division of samples across sub-models could be contrary to the realities of the data. We explore and compare these two design options, choosing the appropriate one depending on the application and dataset characteristics.

1.3 Challenges in learning data partitioning ensembles

One of the main computational issues characteristic to this type of model is the 'chicken and egg' problem associated with assigning training samples to sub-models. This happens because the sub-models themselves are built based on their assigned samples. While traditional methods would rely on expectation-maximization, our methods avert this

complication through the use of non-parametric estimators that assess the benefit of different candidate models in the neighborhood of each sample of the dataset. A consensus across the samples is then reached concerning which set of models is most useful overall. This technique is inherently robust in that, in the neighborhood of any sample point, the model is less sensitive to changes elsewhere in the feature space. We are currently investigating ways to approach our learning task directly as a function factorization problem, where one factor represents the sample-component assignment and the second is the solution given by the sub-model. The process of finding the models we are targeting raises some issues, a notable one being identifiability. Namely, there could be several very different, albeit accurate, alternatives which solve the learning problem under the settings described above. While our methods work by formulating an objective function and selecting the best performing one, we also take steps to ensure robustness of the selected model and derive necessary conditions for identifiability. A related issue is the use of regularization and the trade-offs between ensemble size and component complexity, which we investigate (so far, only empirically) in order to determine how to best set the parameters (and constraints) to obtain optimal performance for a given dataset.

Our method of learning ensembles of compact solvers improve on existing non-specialized models, at least for data which complies with the assumption that any given query can be handled using only a subset of the initial features. We primarily target classification, although the basic concepts also apply to regression and clustering. We showed experimentally and are we looking to prove that the models obtain faster learning rates, in terms of sample size, than (1) non-specialized models with solvers from the same hypothesis space using all the features and (2) non-specialized models with solvers from the same hypothesis space using the same number of features as the ensemble. In the latter case, we also expect to obtain higher limiting accuracy (3).[1]

1.4 Related Work

Extensive research in dimensionality reduction has resulted in a number of techniques which we use in the development and analysis of our algorithms. The problem we address is related to structured sparse learning [29] and compressed sensing [9]. Our method has an advantage over them as it partitions the data, as opposed to building a universal model. Specifically, the analysis of our methods relies on existing theoretical results in structured sparsity [24, 36, 40, 46, 47], as well as the optimization methods that make this type of learning possible [2, 3, 35]. Also, low-dimensional ensemble components can be learned under the assumption that subsets of the given samples can be written as sparse signals in some basis and thus admit a compressed representation (in the form of basis/matching pursuit), which can be determined through existing techniques [5, 21].

We also note some conceptual similarities to hierarchical latent variable models [8, 33] and sparse mixture models [16, 43] – the notion of several underlying processes that determine the output signal. However, our methodology remains very different from standard algorithms on these topics, as we avoid non-convexity by directly operating on the feature space, without the use of intermediaries such as latent variables or mixture components.

Currently, our approaches use axis-aligned subspaces (through lasso penalties) or linear combinations of features (via compressed sensing), but if these fail to deliver the required compact ensemble, we will approach the problem from a nonlinear perspective [32, 41]. Given the multi-model characteristics of the data we target, we use techniques which explicitly learn several manifolds before training the set of solvers [45] or, alternatively, employ multiple kernel learning [23]. Either way, these techniques assume that all data falls under the same model and extra mechanisms are required to assign groups of samples to manifolds/kernels.

Currently, there exist several ensemble-based methods to which we can relate our work [12, 14, 25, 44]. Most of these are, however, purely empirical and not accommodating of theoretical analysis. Our approach not only provides a model which is more representative of the underlying processes and more communicative to the domain experts, but it does so in a manner that makes it possible to obtain theoretical guarantees.

[1] Points (2) and (3) are straightforward to show since, for partitions, the ensemble is a more generic class, implying that it will fit the data better, but will take longer to train. The elimination of spurious features reduces the amount of needed training samples.

Chapter 2

Informative Projection Recovery and its use in Machine Learning Tasks

2.1 Projection retrieval for classification

Intelligent decision support systems often require human involvement because of data limitations, such as the absence of contextual information, as well as due to the need for accountability. 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 3D) projections of data, selected from among the original dimensions, that jointly provide good accuracy 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, jointly with their corresponding models, form a solution to the IPR problem. We have previously proposed such a solution tailored to non-parametric classification. Our RIPR algorithm [17] 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.

Solving the IPR problem is relevant in many practical applications. For instance, consider a nuclear threat detection system installed at a border check point. Vehicles crossing the border are scanned with sensors so that a large array of measurements of radioactivity and secondary contextual information is being collected. These observations are fed into a classification system that determines whether the scanned vehicle may carry a threat. Given the potentially devastating consequences of a false negative, a border control agent is requested to validate the prediction and decide whether to submit the vehicle for a costly further inspection. With the positive classification rate of the system under strict bounds because of limitations in the control process, the risk of false negatives is increased. Despite its crucial role, human intervention should only be withheld for cases in which there is reason to doubt the validity of classification. In order for a user to attest the validity of the decision, the user must have a good understanding of the classification process, which happens more readily when the classifier only uses the original dataset features rather than combinations of them and when the discrimination models are low-dimensional.

2.1.1 Formalization of informative projection recovery

In this context, we aim to learn a set of classifiers in low-dimensional subspaces and a decision function which selects the subspace under which a test point is to be classified. Assume we are given a dataset $\{(x_1, y_1) \dots (x_n, y_n)\} \in$

$\mathcal{X}^n \times \{0,1\}^n$ and a class of discriminators $\mathcal{H}$. The model will contain a set Π of subspaces of $\mathcal{X}$; $\Pi \subseteq \mathbf{\Pi}$, where $\mathbf{\Pi}$ is the set of all axis-aligned subspaces of the original feature space, the power set of the features. To each projection $\pi_i \in \Pi$ corresponds one discriminator from the given hypothesis space $h_i \in \mathcal{H}$. It will also contain a selection function $g : \mathcal{X} \rightarrow \Pi \times \mathcal{H}$, which yields, for a query point x, the projection/discriminator pair with which this point will be classified. The notation $\pi(x)$ refers to the projection of the point x onto the subspace π while $h(\pi(x))$ represents the predicted label for x. Formally, we describe the model class as

$$\begin{aligned} \mathcal{M}_d &= \{\Pi = \{\pi : \pi \in \mathbf{\Pi}, dim(\pi) \leq d\}, \\ &\quad H = \{h_i : h_i \in \mathcal{H}, h : \pi_i \rightarrow \mathcal{Y}, \forall i = 1 \ldots |\Pi|\}, \\ &\quad g \in \{f : \mathcal{X} \rightarrow \{1 \ldots |\mathbf{\Pi}|\}\} \quad . \end{aligned}$$

where $dim(\pi)$ presents the dimensionality of the subspace determined by the projection π. Note that only projections up to size d will be considered, where d is a parameter specific to the application. The set H contains one discriminator from the hypothesis class $\mathcal{H}$ for each projection.

Intuitively, the aim is to minimize the expected classification error over $\mathcal{M}_d$, however, a notable modification is that the projection and, implicitly, the discriminator, are chosen according to the data point that needs to be classified. Given a query x in the space $\mathcal{X}$, $g(x)$ will yield the subspace $\pi_{g(x)}$ onto which the query is projected and the discriminator $h_{g(x)}$ for it. Distinct test points can be handled using different combinations of subspaces and discriminators. We consider models that minimize 0/1 loss. Hence, the PRC problem can be stated as follows:

$$M^* = \arg\min_{M \in \mathcal{M}_d} \mathbb{E}_{\mathcal{X},\mathcal{Y}}\Big[y \neq h_{g(x)}(\pi_{g(x)}(x))\Big]$$

There are limitations on the type of selection function g that can be learned. A simple example for which g can be recovered is a set of signal readings x for which, if one of the readings x_i exceeds a threshold t_i, the label can be predicted just based on x_i. A more complex one is a dataset containing regulatory variables, that is, for x_i in the interval $[a_k, b_k]$ the label only depends on $(x_k^1 \ldots x_k^{n_k})$ - datasets that fall into the latter category fulfill what we call the Subspace-Separability Assumption.

2.1.2 Recovering informative projections with the RIPR algorithm

To solve IPR, we need means by which to ascertain which projections are useful in terms of discriminating data from the two classes. Since our model allows the use of distinct projections depending on the query point, it is expected that each projection would potentially benefit different areas of the feature space. $\mathcal{A}(\pi)$ refers to the area of the feature space where the projection π is selected.

$$\mathcal{A}(\pi) \quad = \quad \{x \in \mathcal{X} : \pi_{g(x)} = \pi\}$$

The objective becomes

$$\min_{M \in \mathcal{M}_d} E_{\mathcal{X},\mathcal{Y}}\Big[y \neq h_{g(x)}(\pi_{g(x)}(x))\Big] \quad = \quad \min_{M \in \mathcal{M}_d} \sum_{\pi \in \Pi} p(\mathcal{A}(\pi)) \mathbb{E}_{x \in \mathcal{A}(\pi)}\Big[y \neq h_{g(x)}(\pi_{g(x)}(x))\Big] \quad .$$

The expected classification error over $\mathcal{A}(\pi)$ is linked to the conditional entropy of $Y|X$. Fano's inequality provides a lower bound on the error while Feder and Merhav [15] derive a tight upper bound on the minimal error probability in terms of the entropy. This means that conditional entropy characterizes the potential of a subset of the feature space to separate data, which is more generic than simply quantifying classification accuracy for a specific discriminator.

In view of this connection between classification accuracy and entropy, we adapt the objective to

$$\min_{M \in \mathcal{M}_d} \sum_{\pi \in \Pi} p(\mathcal{A}(\pi)) H(Y|\pi(X); X \in \mathcal{A}(\pi)) \tag{2.1}$$

The method we propose optimizes an empirical analog of (2.1) which we develop below and for which we will need the following result.

Proposition 2.1.1. *Given a continuous variable $X \in \mathcal{X}$ and a binary variable Y, where X is sampled from the mixture model.*

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

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

Next, we will use the nonparametric estimator presented in [37] for Tsallis α-divergence. Given samples $u_i \sim U$, with $i = 1, n$ and $v_j \sim V$ with $j = 1, m$, the divergence is estimated as follows:

$$\hat{T}_\alpha(u||v) = \frac{1}{1-\alpha}\Big[\frac{1}{n}\sum_{i=1}^{n}\Big(\frac{(n-1)\nu_k(u_i, U \setminus u_i)^d}{m\nu_k(u_i, V)^d}\Big)^{1-\alpha} B_{k,\alpha} - 1\Big]\,, \tag{2.2}$$

where d is the dimensionality of the variables U and V and $\nu_k(z, Z)$ represents the distance from z to its k^{th} nearest neighbor of the set of points Z. For $\alpha \approx 1$ and $n \to \infty$, $\hat{T}_\alpha(u||v) \approx D_{KL}(u||v)$.

Local estimators of entropy

We will now plug (2.2) in the formula obtained by Proposition 2.1.1 to estimate the quantity (2.1). We use the notation X_0 to represent the n_0 samples from X which have the labels Y equal to 0, and X_1 to represent the n_1 samples from X which have the labels set to 1. Also, $X_{y(x)}$ represents the set of samples that have labels equal to the label of x and $X_{\neg y(x)}$ the data that have labels opposite to the label of x.

$$\hat{H}(Y|X; X \in \mathcal{A}) \;=\; -C(p_0) - C(p_1) - \hat{T}(f_0^x||f^x) - \hat{T}(f_1^x||f^x) \qquad \alpha \approx 1$$

$$\begin{aligned}
\hat{H}(Y|X; X \in \mathcal{A}) \;\propto\; & \frac{1}{n_0}\sum_{i=1}^{n_0} I[x_i \in \mathcal{A}]\Big(\frac{(n_0-1)\nu_k(x_i, X_0 \setminus x_i)^d}{n\nu_k(x_i, X \setminus x_i)^d}\Big)^{1-\alpha} \\
+ \; & \frac{1}{n_1}\sum_{i=1}^{n_1} I[x_i \in \mathcal{A}]\Big(\frac{(n_1-1)\nu_k(x_i, X_1 \setminus x_i)^d}{n\nu_k(x_i, X \setminus x_i)^d}\Big)^{1-\alpha} \\
\propto \; & \frac{1}{n_0}\sum_{i=1}^{n_0} I[x_i \in \mathcal{A}]\Big(\frac{(n_0-1)\nu_k(x_i, X_0 \setminus x_i)^d}{n\nu_k(x_i, X_1 \setminus x_i)^d}\Big)^{1-\alpha} \\
+ \; & \frac{1}{n_1}\sum_{i=1}^{n_1} I[x_i \in \mathcal{A}]\Big(\frac{(n_1-1)\nu_k(x_i, X_1 \setminus x_i)^d}{n\nu_k(x_i, X_0 \setminus x_i)^d}\Big)^{1-\alpha} \\
\propto \; & \frac{1}{n}\sum_{i=1}^{n} I[x_i \in \mathcal{A}]\Big(\frac{(n-1)\nu_k(x_i, X_{y(x_i)} \setminus x_i)^d}{n\nu_k(x_i, X_{\neg y(x_i)} \setminus x_i)^d}\Big)^{1-\alpha}
\end{aligned}$$

The estimator for the entropy of the data that is classified with projection π is as follows:

$$\hat{H}(Y|\pi(X); X \in \mathcal{A}(\pi)) \;\propto\; \frac{1}{n}\sum_{i=1}^{n} I[x_i \in \mathcal{A}(\pi)]\Big(\frac{(n-1)\nu_k(\pi(x_i), \pi(X_{y(x_i)}) \setminus \pi(x_i))^d}{n\nu_k(\pi(x_i), \pi(X_{\neg y(x_i)} \setminus x_i))^d}\Big)^{1-\alpha} \tag{2.3}$$

From 2.3 and using the fact that $I[x_i \in \mathcal{A}(\pi)] = I[\pi_{g(x_i)} = \pi]$ for which we use the notation $I[g(x_i) \to \pi]$, we estimate the objective as

$$\min_{M \in \mathcal{M}_d} \sum_{\pi \in \Pi} \frac{1}{n}\sum_{i=1}^{n} I[g(x_i) \to \pi]\Big(\frac{(n-1)\nu_k(\pi(x_i), \pi(X_{y(x_i)}) \setminus \pi(x_i))^d}{n\nu_k(\pi(x_i), \pi(X_{\neg y(x_i)} \setminus x_i))^d}\Big)^{1-\alpha} \tag{2.4}$$

Therefore, the contribution of each data point to the objective corresponds to a distance ratio on the projection π^* where the class of the point is obtained with the highest confidence (data is separable in the neighborhood of the

point). We start by computing the distance-based metric of each point on each projection of size up to d - there are d^* such projections.

This procedure yields an extended set of features L, which we name local entropy estimates:

$$L_{ij} \quad = \quad \Big(\frac{\nu_k(\pi(x_i), \pi(X_{y(x_i)}) \setminus \pi(x_i))}{\nu_k(\pi(x_i), \pi(X_{\neg y(x_i)} \setminus x_i))}\Big)^{d(1-\alpha)} \qquad \alpha \approx 1 \quad j \in \{1 \dots d^*\} \tag{2.5}$$

For each training point, we compute the best distance ratio amid all the projections, which is simply $L_i^* = \min_{j \in [d^*]} L_{ij}$.

The objective can be then further rewritten as a function of the entropy estimates:

$$\min_{M \in \mathcal{M}_d} \sum_{i=1}^{n} \sum_{\pi_j \in \Pi} I[g(x_i) \to \pi_j] L_{ij} \tag{2.6}$$

From the definition of L^*, it is also clear that

$$\min_{M \in \mathcal{M}_d} \sum_{i=1}^{n} \sum_{\pi_j \in \Pi} I[g(x_i) \to \pi_j] L_{ij} \quad \geq \quad \sum_{i=1}^{n} L_i^* \,. \tag{2.7}$$

2.1.3 Projection selection as a combinatorial problem

Considering form (2.6) of the objective, and given that the estimates Z_{ij} are constants, depending only on the training set, the projection retrieval problem is reduced to finding g for all training points, which will implicitly select the subset of projections to be contained by the model. Naturally, one might assume the best-performing classification model is the one containing all the axis-aligned subspaces. This model achieves the lower bound (2.7) for the training set. However, the larger the set of projections, the more values the function g takes, and thus the problem of selecting the correct projection becomes more difficult. It becomes apparent that the number of projections should be somehow restricted to allow generalization. Assuming a hard threshold of at most t projections, the optimization (2.6) becomes an entry selection problem over matrix Z where one value must be picked from each row under a limitation on the number of columns that can be used. This problem cannot be solved exactly in polynomial time. Instead, it can be formulated as an optimization problem under ℓ_1 constraints.

Projection retrieval through regularized regression

To transform the projection retrieval to a regression problem we consider T, the minimum obtainable value of the entropy estimator for each point, as the output which the method needs to predict. Each row i of the parameter matrix B represents the degrees to which the entropy estimates on each projection contribute to the entropy estimator of point x_i. Thus, the sum over each row of B is 1, and the regularization penalty applies to the number of non-zero columns in B.

$$\min_{B} ||T - \sum_{i=1}^{|PI|} Z \odot B||_2^2 + \lambda \sum_{i=1}^{d^*} [B_i \neq 0] \tag{2.8}$$

$$\text{subject to} \tag{2.9}$$

$$|B_k|_{\ell_1} = 1 \quad k = \overline{1,n} \tag{2.10}$$

The problem with this optimization is that it is not convex. A typical walk-around of this issue is to use the convex relaxation for $B_i \neq 0$, that is ℓ_1 norm. This would transform the penalized term to $\sum_{i=1}^{d^*} |B_i|_{\ell_1}$. However, $\sum_{i=1}^{d^*} |B_i|_{\ell_1} = \sum_{k=1}^{n} |B_k|_{\ell_1} = n$, so this penalty really has no effect. An alternative mechanism to bias the non-zero elements in B towards a small number of columns is to add a penalty term in the form of $B\delta$, where δ is a d^*-size column vector with each element representing the penalty for a column in B. With no prior information about which subspaces are more informative, δ starts as an all-1 vector. An initial value for B is obtained through the optimization (2.8). Since our goal is to handle data using a small number of projections, δ is then updated such

that its value is lower for the denser columns in B. The matrix B itself is updated, and this 2-step process continues until convergence of δ. Once δ converges, the projections corresponding to the non-zero columns of B are added to the model. The procedure is shown in Algorithm 2.1.1.

Algorithm 2.1.1 Framework for Informative Projection Recovery

$\delta = [1 \dots 1]$
repeat
  $B = \arg\min_B ||L^* - L \odot B||_2^2 + \lambda_1 \sum_{j=1}^{d^*} ||B_{.,j}||_{\ell_1} + \lambda_2 |B\delta|_{\ell_1}$
    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 $\Pi = \{\pi_i; \quad |B_{.,i}|_{\ell_1} > 0 \quad \forall i = 1 \dots d^*\}$

2.1.4 Lasso for projection selection

We will compare our algorithm to lasso regularization that ranks the projections in terms of their potential for data separability. We write this as an ℓ_1-penalized optimization on the extended feature set Z, with the objective $T: \quad \min_\beta |T - Z\beta|_2 + \lambda|\beta|_{\ell_1}$. The lasso penalty to the coefficient vector encourages sparsity. For a high enough λ, the sparsity pattern in β is indicative of the usefulness of the projections. In practice, we will use a robust version of this optimization.

The selection function

Once the projections are selected, the second stage of the algorithm deals with assigning the projection with which to classify a particular query point. An immediate way of selecting the correct projection starts by computing the local entropy estimator for each subspace with each class assignment. Then, we may select the label/subspace combination that minimizes the empirical entropy.

$$(i^*, \theta^*) = \arg\min_{i,\theta} \left(\frac{\nu_k(\pi_i(x), \pi_i(X_\theta))}{\nu_k(\pi_i(x), \pi_i(X_{\neg\theta}))} \right)^{dim(\pi_i)(1-\alpha)} \qquad i = 1 \dots d^* \quad , \quad \alpha \approx 1 \tag{2.11}$$

2.2 Extension of the IPR problem to other machine learning tasks

We now 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.

2.2.1 Generalized projection recovery problem

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:

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

$\mathbf{\Pi}$ 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.

Figure 2.1 shows the procedure of labeling a test sample given a RIPR model with $r \stackrel{def}{=} |\Pi|$ projections. The framework accepts a query point x, selects the low-dimensional subspace of the features $\pi_{g(x)}$ on which to project the point, then applies the task solver $\tau_{g(x)}$ of the subspace. Finally, the classification outcome is shown in the context of the low-dimensional projection, highlighting the projection $\pi_{g(x)}(x)$ of the test point as well as its neighbors.

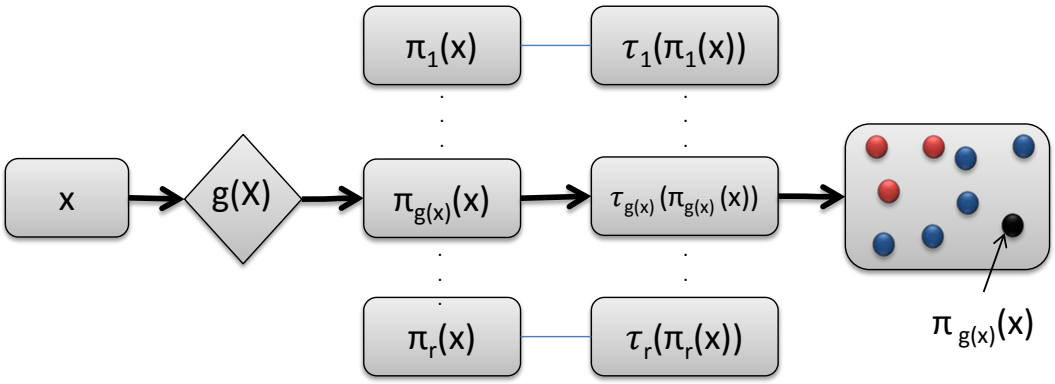

Figure 2.1: The sample labeling procedure.

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: $\tau_{\mathcal{M}}(x) = \tau_j(\pi_j(x)), \quad \ell(x, \tau_{\mathcal{M}}) = \ell(\pi_j(x), \tau_j)$, 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:

$$M^* = \arg\min_{\mathcal{M}} \mathbb{E}_{\mathcal{X}} \ell(\pi_{g(x)}(x), \tau_{g(x)}) \tag{2.12}$$

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

The minimizers of the risk and empirical risk are:

$$M^* = \arg\min_{\mathcal{M}} \mathbb{E}_{\mathcal{X}} \sum_{j=1}^{|\Pi|} I[g(x) = j] \ell(\pi_j(x), \tau_j)$$
$$\hat{M}^* = \arg\min_{\mathcal{M}} \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{|\Pi|} I[g(x_i) = j] \ell(\pi_j(x_i), \tau_j) \tag{2.13}$$

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

$$\exists \hat{\ell} \text{ s.t.} \quad \forall x \in \mathcal{X}, \tau \in \mathcal{T} \quad plim_{n\to\infty} \hat{\ell}(x, \tau) = \ell(x, \tau) \tag{2.14}$$

Plugging (2.14) into (2.13) 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 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 remains the same one shown in Algorithm 2.1.1 for all learning tasks, 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 to a stable set of projections. As illustrated in Algorithm 2.1.1, 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.

2.2.2 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.

Semi-supervised classification

While the case of classification has been handled in the previous section, RIPR does allow 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.

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 sub-models. 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.

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

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 2.2. The right part of it shows results 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.

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

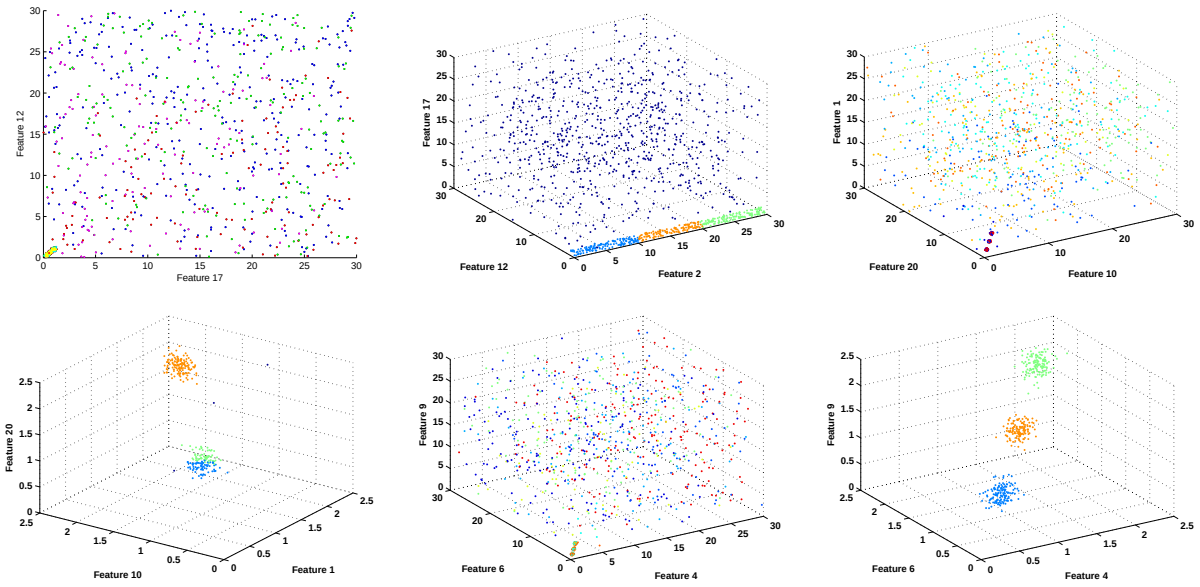

Figure 2.2: Projections of k-means clusters on the informative features and RIPR low-dimensional clusters induced from synthetic data.. Each cluster determined by the algorithm is shown in a different color.

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 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 [11], the composite regressor is also consistent under the assumption of existence of embedding.

2.2.3 Computational Complexity

RIPR requires estimating loss for every data point, for every combination of features, and requires finding the k[th] nearest neighbors. Using k-d trees [20] for every projection of size d, the time required to build the tree is $O(dn \log n)$ and the time needed to find the neighbors of one sample point is $O(\log n)$. Thus, for all $d^* = O(m^d)$ projections of up to size d, 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 2.1.1, we use the bounds in [4]. 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 $O\Big(m^{4d}n^4\Big)\ln\Big(\frac{1}{\epsilon}\Big)$. Although the complexity increases exponentially with d, for the applications we consider d is typically 2, resulting in a runtime of $O\Big(m^8n^4\Big)\ln\Big(\frac{1}{\epsilon}\Big)+O\Big(m^2n\log n\Big)$.

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 (2.15), which is beneficial for datasets that are underdetermined (small sample size but large number of features) – a frequent case in e.g. computational biology.

$$O\Big(n^8\Big)\ln\Big(\frac{1}{\epsilon}\Big)+O\Big(dn^2\log n\Big) \tag{2.15}$$

2.3 Experimental results for informative projection recovery

Table 2.1: Accuracy of K-NN and RECIP

Dataset	***KNN***	***RECIP***
Breast Cancer Wis	**0.8415**	0.8275
Breast Tissue	**1.0000**	**1.0000**
Cell	0.7072	**0.7640**
MiniBOONE*	**0.7896**	0.7396
Spam	**0.7680**	**0.7680**
Vowel	**0.9839**	**0.9839**

Table 2.1 shows the standard K-NN and RIPR+K-NN performance on UCI datasets. We also test the methods on a Cell dataset containing a set of measurements such as the area and perimeter of the cell and a label which specifies whether the cell has been subjected to treatment or not. In the Vowel dataset, a nearest-neighbor approach works exceptionally well, even beating random forests (0.94 accuracy), which is an indication that all features are jointly relevant. For some d lower than the number of features, RECIP picks projections of only one feature, but if there is no such limitation, RECIP picks the space of all the features as informative.

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 2.2, the distortion results for the RIPR model are better than for plain k-means.

Table 2.2: 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

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 2.2. 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.

Chapter 3

Discovering Informative Projections in an Active Learning Setting

We adapt standard active learning sample selection heuristics to work directly with the RIPR models and introduce new heuristics that find unlabeled data worth expert evaluation based on their appearance in low-dimensional subspaces. We also modify the RIPR optimization to find contradictory patterns in data, which is useful in the active learning context when the intent is to prompt the domain experts into disambiguating samples which are difficult to classify automatically. This method is part of the annotation system which doctors used to label a subset of alerts as real or artifactual.

3.1 Overview of active learning with dimensionality reduction

We introduce an approach which recovers informative projections in the more challenging *active learning* setting. Our framework selects samples to be labeled based on the relevant dimensions of the current classification model, trained on previously annotated data. The effort is thus shifted to labeling samples that *specifically target performance improvement* for the class of low-dimensional models we are considering. An important outcome is that *high accuracy is achieved faster* than with standard sampling techniques, reducing the data annotation effort exerted by domain experts. An added benefit is that the *compact models are available to experts during labeling*, in addition to the full-featured data. The informative projections[1] highlight structure that experts should be aware of during the labeling process, which helps prevent user errors, as illustrated in a case study. Moreover, our active learning framework selects the most controversial, most informative and/or most uncertain data yet unlabeled (depending on the selected sampling technique), presenting it to the human experts in an intuitive and comprehensible manner, typically using 2 or 3-dimensional projections, which further simplifies the annotation process.

We have previously formulated *Informative Projection Retrieval* (IPR) as the problem of finding query-specific models using ensembles of classifiers trained on small subsets of features. The *Regression for Informative Projection Retrieval* (RIPR) algorithm [19] provides a solution to this problem in the form of compact models consisting of low-dimensional projections. We will call them *RIPR models*. This chapter presents a framework, called *ActiveRIPR*, which enables active selection of yet unlabeled data which specifically targets the construction of accurate RIPR models. For this purpose, we adapt established active learning query criteria to the IPR task. Our contributions are: (i) we solve the Informative Projection Retrieval problem in the active learning setting; (ii) we compare various querying strategies under different noise models; (iii) we apply ActiveRIPR to alert adjudication leading to considerable reduction of labeling effort.

Active learning is an intensely-studied branch of machine learning, with many successful sampling methods currently available [42]. Adding to established methods such as uncertainty sampling, information gain and query by committee, are recent developments such as the Kernel Query by Committee [22], sampling based on mutual

[1] In this paper, we focus exclusively on axis aligned projections (sets of features), since domain experts have no difficulty interpreting them.

information [26] and the use of importance weighting in a scheme which works with general loss functions to correct sampling bias [6]. Our sample selection criteria take into account the utility of the samples for each of the projections in our ensemble. Previous work considering ensembles include the approach of Körner and Wrobel [30], who compare different approaches that use ensemble disagreement adapted to the problem of multiclass learning and show that margins are the best performing for the purpose. Donmez et al. [13] consider the existence of an ensemble of labeling sources and investigate how to jointly learn their accuracy and obtain the most informative labels while minimizing labeling effort. Examples of structured prediction being enhanced by active learning include the work by Culotta and McCallum [10], introducing a selective sampling framework which considers not just the number of samples requested for active learning in structured prediction, but also the effort required in labeling them. Liang et al. [34] also investigate the interplay between structured learning and model enhancement using contextual features, using unlabeled data to shift predictive power between models. The algorithm they present interleaves labeling features and samples, which improves the active learning performance. Bilgic proposes dynamic dimensionality reduction for active learning [7], a method which, during the query selection process, performs PCA on the data, selecting the features with the largest eigenvalues and performing L_2 regularization on them. There are some notable differences to their approach, the most important of which is that, in their setup, the allowed number of features is increased as more samples become available. The method of Raghavan et al. [38] directly incorporates human feedback in the feature selection procedure through feature weighting, while Rashidi and Cook [39] introduce a method that reduces the effort needed for labeling by requesting, in each iteration, labels for all samples matching a rule.

Our main improvement over related work is that our framework is designed to train accurate intelligible models which domain experts can use during the labeling process. ActiveRIPR not only queries the samples which improve model accuracy, but also considers human involvement and targets compact, user-friendly models, such that, at every step in the active learning procedure, the experts can consult the current informative model. Access to this visualizable model can make expert adjudication faster and more reliable. Also, clinicians can observe the classification model in action and be better prepared to decide whether it is mature enough for deployment.

3.2 Active informative projection recovery framework

Active learning iteratively selects samples for labeling until the model meets some accuracy criteria. Assume now that, at iteration k, the samples X_ℓ^k are labeled as Y_ℓ^k and the samples X_u^k are available for labeling. Also let the RIPR model built so far be M^k, with its components Π^k, τ^k and g^k. The problem of selecting samples for IPR is reduced to finding a scoring function $s : \mathcal{M} \times \mathcal{X} \to \mathbb{R}$, used to select the next sample to be labeled:

$$x^{k+1} = \arg\min_{x \in X_u^k} s(M^k, x)$$

The expected error of a model $M^k = \{\Pi^k, \tau^k, g^k\}$ is

$$Err(M^k) = \mathbb{E}_{x \in \mathcal{X}}[I(\tau^k_{g^k(x)}(\pi^k_{g^k(x)}(x)) \neq y)]$$

We use the notation M_s^k to refer to a model obtained after k iterations of labeling, using the scoring function s. If the labeled samples are picked adequately, the training error will decrease (or at least not increase) with each iteration: $Err(M_s^{k+1}) \leq Err(M_s^k)$. Given the maximum acceptable error ϵ, and a set $\mathcal{S}$ of scoring functions, selecting the optimal strategy can be expressed as follows:

$$s^* = \arg\min_{s \in \mathcal{S}} \min_k \{k \text{ s.t. } Err(M_s^k) \leq \epsilon\} \tag{3.1}$$

ActiveRIPR starts by requesting the labels of a set of r_0 randomly selected samples. It then builds a RIPR model from these samples. Using a function which scores yet-unlabeled data considering the current model, ActiveRIPR selects the next set of samples to be labeled. The next section describes several such scoring functions. New models are trained as additional samples are added to the pool. While it is possible to efficiently update the current model

using the new samples, we currently re-train from scratch, both for simplicity and to avoid any possibility of bias. The Active RIPR procedure is shown in Algorithm 3.2.1. X_u are the unlabeled samples, X_ℓ are the samples for which labels have been requested and Y_ℓ are their provided labels. X_t and Y_t represent a separate set of samples used for testing. M_s^k is the model trained at iteration k, based on samples queried using scoring function s. Err_s^k is the error of model M_s^k.

Algorithm 3.2.1 Active RIPR with scoring function s

X_u (unlabeled samples), X_t (test samples), Y_t (test labels), $k = 0$ (iterations)
X_ℓ = SelectRandom(X_u)
Y_ℓ = LabelSamples(X_ℓ)
repeat
  $k = k + 1$
  M_s^k = TrainRiprModel(X_ℓ, Y_ℓ, X_u)
  Err_s^k = EvaluateRiprModel(M_s^k, X_t, Y_t)
  $x_i = \arg\min_{x_i \in X_u} s(M_s^k, x_i)$
  y_i = LabelSample(x_i)
  $X_u = X_u \setminus \{x_i\}$, $X_\ell = X_\ell \cup \{x_i\}$, $Y_\ell = Y_\ell \cup \{y_i\}$
until $Err_s^k \leq \epsilon$ or $|X_u| = 0$
return M_s^k

3.3 Active Sample Selection

Extensive research in the domain of active learning has led to a variety of algorithms which determine which points should be labeled next. We do not seek to supplant these, but rather adapt a subset of them to work with the class of model we target. The intuition is that, for data where most of the features are spurious, adapting the scoring function to consider only the significant features for each sample has the potential to improve the learning rate.

Uncertainty Sampling

This score is used to pick the unlabeled data for which the label is the most uncertain, typically this translates to selecting the samples with the highest conditional entropy of the output given the features. Under the RIPR assumption, the label of a sample depends only on the projection to which the point is assigned. Using a RIPR model $M^k = \{\Pi^k, \tau^k\}$, the corresponding projection for a sample x and its label $\hat{y}(x)$ are determined as follows:

$$g^k(x) := \underset{(\pi,\tau)\in(\Pi^k,\tau^k)}{\arg\min} \hat{h}(\tau(\pi(x))|\pi(x))$$
$$\hat{y}(x) := \tau^k_{g^k(x)}(x),$$

where $\hat{h}$ denotes the conditional entropy estimator for a label given a subset of the features and $\hat{y}(x)$ is the prediction made for a sample x. The score for ActiveRIPR using uncertainty sampling simply considers the lowest conditional entropy on the projections of the model M^k_{uncrt}:

$$s_{uncrt}(x) = \min_{\pi\in\Pi^k_{uncrt},\tau\in\tau^k} \hat{h}(\tau(\pi(x))|\pi(x)) \tag{3.2}$$

Query by Committee

Query by committee selects the samples on which the classifiers in an ensemble disagree. For a RIPR model M^k_{qbc}, this is simply obtained by comparing the labels assigned by each of the classifiers in τ^k_{qbc}.

$$s_{qbc}(x) = \max_{\tau_i,\tau_j\in\tau^k_{qbc}} I(\tau_i(\pi_i(x)) \neq \tau_j(\pi_j(x))) \tag{3.3}$$

Information Gain

The information gain criterion sorts unlabeled data according to the expected reduction in conditional entropy upon labeling each point. We use the notation $\hat{H}^k_{X_0,Y_0}(X_1)$ to represent the estimated conditional entropy of the samples X_1 given the samples X_0 and their labels Y_0. Assuming that, at iteration k, ActiveRIPR based on Information Gain has selected samples $X^k_{\ell,ig}$ while samples $X^k_{u,ig}$ are available for labeling, the information gain score can be expressed as follows:

$$\begin{aligned}\forall x \in X^k_{u,ig}, \quad s_{ig}(x) = &\hat{H}^k_{X_\ell,Y_\ell}(X^k_{u,ig}) \\ &- p(y=0)\hat{H}^k_{X_\ell\cup\{x\},Y_\ell\cup\{0\}}(X^k_{u,ig}) \\ &- p(y=1)\hat{H}^k_{X_\ell\cup\{x\},Y_\ell\cup\{1\}}(X^k_{u,ig})\end{aligned}$$

Low Conditional Entropy

Selecting samples with high uncertainty makes sense when there are aspects of the model not yet discovered – in the case of RIPR models, there might be projections that are informative, but are only relevant for a small subset of the data. However, once the informative projections have been discovered, selecting samples with high uncertainty often leads to the selection of purely noisy samples. In this case, selecting the data for which the classification is the most confident improves the model, as it is more likely that these points satisfy the model assumptions and can be used in the classification of their neighboring samples. This claim is verified experimentally, and the score for this query selection criteria is simply the opposite of the uncertainty sampling score:

$$s_{mc}(x) = 1 - \min_{\pi\in\Pi^k_{mc},\tau\in\tau^k_{mc}} \hat{h}(\tau(\pi(x))|\pi(x))$$

3.4 Annotation framework for the classification of clinical alerts in vital sign monitoring systems

Recovery of meaningful, explainable models is fundamental for the clinical decision-making process. We work with cardio-respiratory monitoring systems designed to process multiple vital signs indicative of the current health status of a critical care patient. The Step Down Unit (SDU) patients are connected to monitors, which continuously track the variability of multiple vital signs over time. The system issues an alert whenever some form of instability requires attention, that is, when any of the vitals exceeds pre-set control limits. Typically, such deviations indicate serious decline in patient health status. In practice, a substantial fraction of the issued alerts are not due to real emergencies (true alerts), but instead are triggered by malfunctions such as probe dislocation or inaccuracies of the sensing equipment (artifacts). Each system-generated alert is associated with the vital sign that initiated it: heart rate (HR), respiratory rate (RR), blood pressure (BP), or peripheral arterial oxygen saturation (SpO_2).

In order to reduce alarm fatigue in clinical staff, the ideal monitoring system would dismiss artifactual alerts on-the-fly and allow interpretable validation of true alerts by human experts when they are issued. As expected, the preparation of a high-quality and comprehensive sample of data needed to train an effective artifact adjudication system could be a tedious process in which important parts of the feature space are easy to neglect. This strenuous effort is often compounded by the sheer complexity of the involved feature space. Without a framework similar to the one presented here, precious expert time would be spent primarily navigating the dimensions of the data to establish grounds for labeling specific instances. We propose to not only select the minimal set of unlabeled data for human adjudication, but to also concurrently determine and present the informative small projections of this otherwise high-dimensional data.

We use ActiveRIPR to predict oxygen saturation alerts, treating the existing labeled data as the pool of samples available for active learning. There are 50 features in total. Roughly 10% of the data has been manually labeled and the aim is to use that subset to determine which of of the unlabeled samples are worth the experts' attention. We performed 10-fold cross validation, training the ActiveRIPR model on 90% of the labeled samples and using the remainder to calculate the learning curve. Table 3.1 shows the number of samples required to reach an accuracy

of 0.85 (a value deemed acceptable by clinicians) and 0.88 (the maximum achievable accuracy). Information Gain performs considerably better than the rest and uncertainty sampling, despite having performed poorly in simulations, is also competitive. The results indicate that an accuracy of 0.88 can be achieved by labeling less than 25% of the total samples using the InfoGain scoring function.

Table 3.1 summarizes the proportion of samples needed by ActiveRIPR and ActiveRIPRssc to achieve 0.85 or 0.88 accuracy on the hold out test data of the oxygen saturation alert dataset.

Table 3.1: Percentage of samples needed by ActiveRIPR and ActiveRIPRssc to achieve accuracies of 0.85 and 0.88 in oxygen saturation alert adjudication.

	ActiveRIPR		ActiveRIPRssc	
Target Accuracy	**0.85**	**0.88**	**0.85**	**0.88**
Score Function				
Uncertainty	18.33	18.33	36.67	50.00
QbC	46.67	46.67	86.67	86.67
InfoGain	21.67	25.00	25.00	51.67
CondEntropy	43.33	46.67	48.33	63.33

Given the success of ActiveRIPR using the InfoGain selection criterion for the oxygen saturation alert adjudication, we proceeded to apply it to detecting blood pressure alerts. This time, we compared it against other classification methods using uncertainty sampling. This type of sampling differs from the uncertainty score used by ActiveRIPR in that it considers the entire feature space as opposed to only low-dimensional projections when making the selection. Also, the classifiers are trained on all features as opposed to only a subset, so it is expected that they would perform well. Random Forests and KernelSVM are some of the well-performing classifiers, which we selected because we aim to assess how accurate the system can be when there are no restrictions on model dimensionality.

Table 3.4 presents the mean leave-one-out accuracy of after 20, 50 and 75 labels. ActiveRIPR's performance approaches that of Random Forests and, at times, outperforms KernelSVM, while maintaining compactness of representation and performing drastic feature reduction. The RIPR models used, at any time, at most two 3-dimensional projections, so 6 features in total. ActiveRIPR wins by a sizeable margin over K-NN which we tested because of its potential for interpretability.

Samples	K-nn	K-SVM	RF	ActiveRIPR
20	0.61	0.64	**0.65**	**0.65**
50	0.58	0.66	**0.71**	0.70
75	0.6	0.63	0.71	**0.75**

Table 3.2: Active learning for blood pressure alerts

Chapter 4

Proposed: Extensions to Informative Projection Recovery

4.1 Informative projections for multiple labels and tasks (future work)

We propose to generalize of RIPR to multitask learning, needed to classify nuclear threats or clinical alerts into sub-categories. Not only are we grouping features and samples, but also features/samples/tasks. The loss matrix becomes a loss tensor and the assignment procedure is an optimization, with the appropriate constraints, over the loss tensor. We are currently looking into modifying RIPR to perform multi-model low-dimensional canonical-correlation analysis, the outcome of which would be a set of canonical parameter pairs.

4.2 Learning informative projections for timeseries (future work)

We propose to extend the concept of informative projections to time series data. In this context, the time-varying models can be learned by imposing smoothness constraints over parameters at consecutive timestamps through penalties such as the fused lasso. Aside from the ensemble coherence constraints needed across samples, which ensure the use of only a small number of feature combinations, we will need to impose transition constraints which will prevent samples to be subject to model switching not encountered elsewhere in the data. Trends in the data, as well as the actual feature values, will have to be considered. A usage example is instability prediction due to blood loss under the assumption that the mode of response to a health crisis is patient-dependent.

Chapter 5

Proposed: Low-dimensional Model Learning for Feature Hierarchies

We improve budget-constrained feature selection by leveraging the structure of the feature dependency graph and information about the cost required to compute each feature. We consider the process used to generate the features, as well as their cost, reliability and interdependence. Typically, our applications rely on a core set of features obtained through expensive measurements, enhanced using transformations derived (cheaply) from one or several core features. Also, some measurements can be obtained through more than one procedure. This structure, which is not considered in our previous work, could make our classifiers more powerful for the same total cost. Our proposed method works by generating, based on the feature dependencies, a regularizer which ensures that, once the cost for a feature is paid, all the features it depends on add no extra penalty. Thus, we leverage the submodular cost and the redundancy of the features by generating penalties according to the structure of the dependency graph. This improves accuracy compared to a model obtained using the lasso at no increase in cost.

5.1 Cost sensitive feature selection

We are given a dataset $(X \in \mathbb{R}^{n\times m}, Y \in \mathbb{R}^n)$ with features $A = \{a_1 \dots a_m\}$, a cost function $c : A \to \mathbb{R}$ and information about feature dependencies in the form of the directed graph (A, D), where $(a_i, a_j) \in D$ iff feature j depends on feature i. Learning the set of parameters $w \in \mathbb{R}^m$ involves minimizing a convex loss function f with a regularizer g which penalizes according to the feature cost.

$$w^* = \arg\min_w \sum_{i=1}^{n} f(w, x_i, y_i) + g(w) \tag{5.1}$$

A standard way of using the cost in performing feature selection is the weighted lasso $g_{\ell_1}(w) = \sum_{i=1}^{m} c(a_i)|w_i|$. The issue with this procedure is that it considers only the total cost for each feature ignoring the manner in which the cost decomposes across the dependency graph, which results in a potentially suboptimal selection of the sparsity pattern for a fixed cost in terms of accuracy, since some features that are virtually free are ignored.

5.2 Exploiting the feature dependency graphs through ℓ_1 and ℓ_2 penalties

Our procedure links each feature to their children in a dependency graph through ℓ_2 norms instead of penalizing them separately. Define the index set of children of a feature a_i as

$$\phi(a_i) = \{1 \le j \le m | (a_i, a_j) \in D\}. \tag{5.2}$$

The modified regularizer becomes $g_{c,D}(w) = \sum_{i=1}^{m} c(a_i)||w_{i,\phi(i)}||_2$.

For features that have no children, the term simply equals the ℓ_1 norm. For the rest, however, the ℓ_2 penalty decreases the weight magnitude, but only actually encourages sparsity on the parent feature to be 0 when the weights of all child features are 0. In this case, the ℓ_2 norm simply becomes an ℓ_1 norm and the feature is penalized as in the standard lasso case.

In some applications, the information can be relayed through several different sources, resulting in highly correlated – or even identical – features in the dataset. An example when this situation may occur is health monitoring. For many vital signs, there exist multiple means of obtaining measurements: invasive, non-invasive and computed indirectly from other vitals. Such correlated features are also present in data which holds responses to queries sent to several servers. Although features in the same series are all informative, it is clear that only one of them is needed at a time, including in the construction of child features. This leads to an 'OR' constraint – the presence of one of the features is necessary and sufficient to derive child features.

We enforce this constraint through a penalty which distributes the weight across the redundant features. Assume that $a_i^1 \dots a_i^r$ is a series of features, either of which can be used to obtain a_i. The parameter w_i corresponding to a_i decomposes into the auxiliary components $w_i^1 \dots w_i^r$, only one of which is non-zero. Let $\phi(i)$ denote any child features of a_i. The additional penalty for w_i is

$$g_{OR}(w_i) = c(a_i)||w_{i,\phi(i)}||_2 + \sum_{j=1}^{r}\sum_{k\neq j}^{r} c(a_i^j)||\bar{w}_i^j, w_i^k||_2\,, \qquad \text{where } \bar{w}_i^j = \max\left(\frac{1}{w_i^j + 0.5} - 0.5, 0\right), \tag{5.3}$$

with the following constraint added to the optimization procedure: $\sum_{j=1}^{r} w_i^j = w_i$.

5.3 Preliminary results for feature selection in vital sign monitoring

We applied our method to a classification problem involving clinical data obtained from a cardio-respiratory monitoring system. The system is designed to process multiple vital signs indicative of the current health status of a critical care patient and issue an alert whenever some form of instability requires medical 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 the vital sign that initiated it: heart rate (HR), respiratory rate (RR), blood pressure (BP), or peripheral arterial oxygen saturation (SpO_2). We extracted multiple temporal features independently for each vital sign over the duration of each alert and a window of 4 minutes preceding its onset. The 150 interdependent features included metrics of data density, as well as common moving-window statistics computed for each of the vital timeseries. Here, the cost of all base features is a unit, and one cost unit is added for each additional operation which needs to be performed to obtained derived features. The dataset has a total of 812 samples (alerts). Our type of regularization increases performance for the same cost when compared to the lasso.

Table 5.1: Comparison of our procedure against the lasso on the clinical data.

Cost	MSE (CFS)	MSE (lasso)		Cost	MSE (CFS)	MSE (lasso)
0	0.777094	0.777094		4	**0.244362**	0.250995
1	**0.343564**	0.435285		6	**0.244267**	0.250995
2	**0.245647**	0.250995		12	0.243772	0.243772

Chapter 6

Proposed: Online Cost-constrained Subset Selection Policies

This research direction will focus on online, adaptive policy-learning optimization procedures for feature selection with submodular cost constraints. We first consider the batch mode setting and learn the group of features which yields the best classification performance while satisfying the cost constraints. Next, we consider the case when the features needed depend on context, which requires a mapping from each sample to the appropriate feature subset. The aim is to then efficiently update this mapping as more data becomes available, thus moving towards an online policy learning algorithm. The result will be a framework which dynamically changes the features used in the classification process. This method can be used, for instance, for medical applications where there is a constraint on the number of readings which can be performed and the system needs to be able to adapt to patient characteristics.

6.1 Learning a classifier with submodular constraints on feature cost

Let $\{(x_1, y_1) \dots (x_n, y_n)\} \in \mathcal{X}^n\{0,1\}^n$ be a dataset with $\mathcal{X} \subseteq \mathbb{R}^m$. The set of available features is $A \stackrel{def}{=} \{a_1, \dots, a_m\}$. Each set of features is associated a cost determined by $c : 2^A \to \mathbb{R}$, where c is submodular. For a feature set $s \subseteq A$, we define $X^s \in \mathbb{R}^{n\times}$ as the projection of the samples on the subspace determined by the features in s. Given the hypothesis classes $\mathcal{H}^s = \{h : x^s \to \{0,1\}\}$ to be used in the classification task, the objective is to find the set of features that minimizes the empirical error, while not exceeding budget constraints expressed as uppers bounds on the costs of feature sets - there are r such constraints.

In the batch [1], single-model [2] setting, the problem translates to a maximization of a submodular function with submodular cost constraints, as follows:

$$\begin{aligned}(s^*, h^*) = \arg\min_{s\in 2^A} \min_{h\in H^s} ||\hat{h}(X^s) - Y||_2 \\ \text{subject to } c(s_i) \quad \leq B_i \qquad \forall i \in \{1, \dots, r\}\end{aligned} \tag{6.1}$$

For the cases when the objective is convex or admits a convex relaxation, the intended approach is to modularize the cost constraints, making them more restrictive, then solve the problem. The solution satisfies the initial constraints, but, if it is suboptimal, it will also activate one of the modularized constraints. In this case, the constraints must be re-modularized to allow improvement of the solution.

[1] all training samples are available
[2] all samples use the same set of features

6.2 Instance-based feature selection with submodular constraints

In the batch, context-dependent (multi-model) [3] setting, the objective becomes non-submodular. However, using loss estimators, the problem can be reformulated as a convex procedure over a binary assignment matrix and solved by combining the iterated-modularization above with the RIPR approach.

$$\begin{aligned} s^* = & \underset{\{s^1,\ldots,s^p\}\subseteq 2^A}{\arg\min} \; \underset{\{h^1\in\mathcal{H}^{s_1},\ldots,h^p\mathcal{H}^{s_p}\}}{\min} \sum_{i=1}^{n} ||h_{g(x_i)}(x_i^{s^g(x_i)}) - y_i||_2 \\ & \text{subject to } c(s_i) \quad \leq B_i \qquad \forall i \in \{1,\ldots,r\} \end{aligned} \tag{6.2}$$

where g is the selection function mapping a sample to the appropriate set of features needed to classify it.

In the batch, adaptive setting, each sample uses a different set of features, so we would have to learn a mapping from the sample space to a binary feature selection vector, resulting in a sparse multilabel prediction problem.

[3] each sample uses one of a set of low-dimensional projections

Chapter 7

Timeline

Below is a tentative timeline for the completion of the thesis.

Contribution	*Status*	*Estimated completion*	*Publications*
Informative Projection Recovery	completed	Spring 2013	[17] [19]
Active IPR Framework	completed	Spring 2014	
Low-dimensional Model Learning for Feature Hierarchies	in progress	Winter 2015	
Online Cost-constrained Subset Selection Policies	future work	Spring 2015	
Efficient RIPR implementation and extensions	in progress	Summer 2015	

Bibliography

[1] Francis Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. Structured sparsity through convex optimization. *Statistical Science*, 27(4):450–468, 11 2012. doi: 10.1214/12-STS394. URL http://dx.doi.org/10.1214/12-STS394. 1.2

[2] Francis Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. Structured sparsity through convex optimization. *Statistical Science*, 27(4):450–468, 2012. 1.4

[3] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009. 1.4

[4] Aharon Ben-Tal and Arkadi Nemirovski. *Lectures on Modern Convex Optimization*. Society for Industrial and Applied Mathematics (SIAM), 2001. 2.2.3

[5] Radu Berinde, Piotr Indyk, and Milan Ruzic. Practical near-optimal sparse recovery in the l1 norm. In *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*, pages 198–205. IEEE, 2008. 1.4

[6] Alina Beygelzimer, Sanjoy Dasgupta, and John Langford. Importance weighted active learning. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, pages 49–56. ACM, 2009. 3.1

[7] Mustafa Bilgic. Combining active learning and dynamic dimensionality reduction. In *SDM*, pages 696–707, 2012. 3.1

[8] Christopher M Bishop and Michael E Tipping. A hierarchical latent variable model for data visualization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(3):281–293, 1998. 1.2, 1.4

[9] Emmanuel J Candes, Justin K Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on pure and applied mathematics*, 59(8):1207–1223, 2006. 1.4

[10] Aron Culotta and Andrew McCallum. Reducing labeling effort for structured prediction tasks. In *AAAI/IAAI*, pages 746–751, 2005. 3.1

[11] Luc Devroye, László Györfi, and Gábor Lugosi. *A probabilistic theory of pattern recognition*. Springer, 1996. 2.2.2

[12] Pedro Domingos. Knowledge discovery via multiple models. *Intelligent Data Analysis*, 2:187–202, 1998. 1.4

[13] Pinar Donmez, Jaime G. Carbonell, and Jeff Schneider. Efficiently learning the accuracy of labeling sources for selective sampling. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 259–268, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-495-9. doi: 10.1145/1557019.1557053. URL http://doi.acm.org/10.1145/1557019.1557053. 3.1

[14] Eulanda M. Dos Santos, Robert Sabourin, and Patrick Maupin. A dynamic overproduce-and-choose strategy for the selection of classifier ensembles. *Pattern Recogn.*, 41:2993–3009, October 2008. ISSN 0031-3203. doi: 10.1016/j.patcog.2008.03.027. URL http://dl.acm.org/citation.cfm?id=1385702.1385963. 1.4

[15] Meir Feder and Neri Merhav. Relations between entropy and error probability. *IEEE Transactions on Information Theory*, 40(1):259–266, January 1994. doi: 10.1109/18.272494. URL http://dx.doi.org/10.

1109/18.272494. 2.1.2

[16] Mario A. T. Figueiredo and Anil K. Jain. Unsupervised learning of finite mixture models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(3):381–396, 2002. 1.4

[17] Madalina Fiterau and Artur Dubrawski. Projection retrieval for classification. In *Advances in Neural Information Processing Systems 25 (NIPS)*, pages 3032–3040, 2012. 2.1, 7, 7

[18] Madalina Fiterau and Artur Dubrawski. Projection retrieval for classification. In *Advances in Neural Information Processing Systems 25*, pages 3032–3040, 2012. 1.2

[19] Madalina Fiterau and Artur Dubrawski. Informative projection recovery for classification, clustering and regression. In *International Conference on Machine Learning and Applications*, volume 12, 2013. 1.2, 3.1, 7

[20] Jerome H. Friedman, Jon Louis Bentley, and Raphael Ari Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, 3(3):209–226, September 1977. ISSN 0098-3500. doi: 10.1145/355744.355745. URL http://doi.acm.org/10.1145/355744.355745. 2.2.3

[21] Rahul Garg and Rohit Khandekar. Gradient descent with sparsification: an iterative algorithm for sparse recovery with restricted isometry property. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 337–344. ACM, 2009. 1.4

[22] Ran Gilad-Bachrach, Amir Navot, and Naftali Tishby. Query by committee made real. In *Advances in Neural Information Processing Systems 18 (NIPS)*, 2005. 3.1

[23] Mehmet Gönen and Ethem Alpaydın. Multiple kernel learning algorithms. *The Journal of Machine Learning Research*, 999999:2211–2268, 2011. 1.4

[24] Pinghua Gong, Jieping Ye, and Changshui Zhang. Multi-stage multi-task feature learning. *Journal of Machine Learning Research*, 14:2979–3010, 2013. 1.4

[25] Quanquan Gu, Zhenhui Li, and Jiawei Han. Joint feature selection and subspace learning, 2011. URL https://www.aaai.org/ocs/index.php/IJCAI/IJCAI11/paper/view/2910. 1.4

[26] Yuhong Guo and Russell Greiner. Optimistic active-learning using mutual information. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 823–829, 2007. 3.1

[27] Xiaofei He, Deng Cai, Yuanlong Shao, Hujun Bao, and Jiawei Han. Laplacian regularized gaussian mixture model for data clustering. *IEEE Transactions on Knowledge and Data Engineering*, 2010. 1.2

[28] Junzhou Huang, Tong Zhang, and Dimitris Metaxas. Learning with structured sparsity. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 417–424, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-516-1. doi: 10.1145/1553374.1553429. URL http://doi.acm.org/10.1145/1553374.1553429. 1.2

[29] Junzhou Huang, Tong Zhang, and Dimitris Metaxas. Learning with structured sparsity. *The Journal of Machine Learning Research*, 12:3371–3412, 2011. 1.4

[30] Christine Körner and Stefan Wrobel. Multi-class ensemble-based active learning. In Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou, editors, *Proceedings of the 17th European Conference on Machine Learning (ECML)*, volume 4212, pages 687–694. Springer, 2006. 3.1

[31] Martin O Larsson and Johan Ugander. A concave regularization technique for sparse mixture models. In *Advances in Neural Information Processing Systems*, pages 1890–1898, 2011. 1.2

[32] Martin HC Law and Anil K Jain. Incremental nonlinear dimensionality reduction by manifold learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(3):377–391, 2006. 1.4

[33] Neil D. Lawrence. Hierarchical gaussian process latent variable models. In *In International Conference in Machine Learning*, 2007. 1.4

[34] Percy Liang, Hal Daumé III, and Dan Klein. Structure compilation: trading structure for features. In *Pro-*

ceedings of the 25th International Conference on Machine Learning (ICML), pages 592–599. ACM, 2008. 3.1

[35] Jun Liu, Lei Yuan, and Jieping Ye. An efficient algorithm for a class of fused lasso problems. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 323–332. ACM, 2010. 1.4

[36] Karim Lounici, Massimiliano Pontil, Alexandre B Tsybakov, and Sara van de Geer. Taking advantage of sparsity in multi-task learning. *arXiv preprint arXiv:0903.1468*, 2009. 1.4

[37] Barnabás Poczós and Jeff G. Schneider. On the estimation of alpha-divergences. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 609–617, 2011. 2.1.2

[38] Hema Raghavan, Omid Madani, and Rosie Jones. Active learning with feedback on features and instances. *Journal of Machine Learning Research*, 7:1655–1686, 2006. 3.1

[39] Parisa Rashidi and Diane J. Cook. Ask me better questions: active learning queries based on rule induction. In Chid Apt, Joydeep Ghosh, and Padhraic Smyth, editors, *KDD*, pages 904–912. ACM, 2011. ISBN 978-1-4503-0813-7. 3.1

[40] Alessandro Rinaldo. Properties and refinements of the fused lasso. *The Annals of Statistics*, 37(5B):2922–2952, 2009. 1.4

[41] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *Artificial Neural NetworksICANN'97*, pages 583–588. Springer, 1997. 1.4

[42] Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009. 3.1

[43] Nicolas Städler, Peter Bühlmann, and Sara Van De Geer. l 1-penalization for mixture regression models. *Test*, 19(2):209–256, 2010. 1.4

[44] Kai Ming Ting, Jonathan R. Wells, Swee Chuan Tan, Shyh Wei Teng, and Geoffrey I. Webb. Feature-subspace aggregating: ensembles for stable and unstable learners. *Machine Learning*, 82(3):375–397, 2011. ISSN 0885-6125. URL `http://dx.doi.org/10.1007/s10994-010-5224-5`. 1.4

[45] Yong Wang, Yuan Jiang, Yi Wu, and Zhi-Hua Zhou. Spectral clustering on multiple manifolds. *Neural Networks, IEEE Transactions on*, 22(7):1149–1161, 2011. 1.4

[46] Lei Yuan, Jun Liu, and Jieping Ye. Efficient methods for overlapping group lasso. 2013. 1.4

[47] Tong Zhang. Analysis of multi-stage convex relaxation for sparse regularization. *The Journal of Machine Learning Research*, 11:1081–1107, 2010. 1.4

Appendix

RIPR results on artificial data for supervised classification

Table 7.2 shows the classification accuracy for the standard RECIP method obtained for synthetic data. As expected, the observed performance is initially high when there are few known informative projections in data and it decreases noise and ambiguity of the injected patterns increase.

Most types of ensemble learners would use a voting scheme to arrive at final classification of a testing sample, rather than use a model selection scheme. For this reason, we have also compared predictive accuracy revealed by RECIP against a method based on majority voting among multiple candidate subspaces. Table **??** shows that the accuracy of this technique is lower than the accuracy of RECIP, regardless of whether the informative projections are recovered by the algorithm or assumed to be known a priori. This confirms the intuition that a selection-based approach can be more effective than voting for data which satisfies the subspace separability assumption.

For reference, we have also classified the synthetic data using K-Nearest-Neighbors algorithm using all available features at once. The results of that experiment are shown in Table **??**. Since RECIP uses neighbor information, K-NN is conceptually the closest among the popular alternatives. Compared to RECIP, K-NN performs worse when there are fewer synthetic patterns injected in data to form informative projections. It is because some features used then by K-NN are noisy. As more features become informative, the K-NN accuracy improves. This example shows the benefit of a selective approach to feature space and using a subset of the most explanatory projections to support not only explanatory analyses but also classification tasks in such circumstances.

Table 7.1: Projection Recovery for Artificial Datasets with $1 \ldots 7$ informative features and noise level $0 \ldots 0.2$ in terms of mean and variance of *Precision* and *Recall*. Mean/var obtained for each setting by repeating the experiment with datasets with different informative projections.

PRECISION										
	Mean					**Variance**				
	0	***0.02***	***0.05***	***0.1***	***0.2***	***0***	***0.02***	***0.05***	***0.1***	***0.2***
1	1	1	1	0.9286	0.9286	0	0	0	0.0306	0.0306
2	1	1	1	1	1	0	0	0	0	0
3	1	1	1	1	1	0	0	0	0	0
4	1	1	1	1	1	0	0	0	0	0
5	1	1	1	1	1	0	0	0	0	0
6	1	1	1	1	1	0	0	0	0	0
7	1	1	1	1	1	0	0	0	0	0
RECALL										
	Mean					**Variance**				
	0	***0.02***	***0.05***	***0.1***	***0.2***	***0***	***0.02***	***0.05***	***0.1***	***0.2***
1	1	1	1	1	1	0	0	0	0	0
2	1	1	1	1	1	0	0	0	0	0
3	1	1	0.9524	0.9524	1	0	0	0.0136	0.0136	0
4	0.9643	0.9643	0.9643	0.9643	0.9286	0.0077	0.0077	0.0077	0.0077	0.0128
5	0.7714	0.7429	0.8286	0.8571	0.7714	0.0163	0.0196	0.0049	0.0082	0.0278
6	0.6429	0.6905	0.6905	0.6905	0.6905	0.0113	0.0113	0.0272	0.0113	0.0113
7	0.6327	0.5918	0.5918	0.5714	0.551	0.0225	0.02	0.0258	0.0233	0.02

Table 7.2: RECIP Classification Accuracy on Artificial Data

CLASSIFICATION ACCURACY										
	Mean					Variance				
	0	***0.02***	***0.05***	***0.1***	***0.2***	***0***	***0.02***	***0.05***	***0.1***	***0.2***
1	0.9751	0.9731	0.9686	0.9543	0.9420	0.0000	0.0000	0.0000	0.0008	0.0007
2	0.9333	0.9297	0.9227	0.9067	0.8946	0.0001	0.0001	0.0001	0.0001	0.0001
3	0.9053	0.8967	0.8764	0.8640	0.8618	0.0004	0.0005	0.0016	0.0028	0.0007
4	0.8725	0.8685	0.8589	0.8454	0.8187	0.0020	0.0020	0.0019	0.0025	0.0032
5	0.8113	0.8009	0.8105	0.8105	0.7782	0.0042	0.0044	0.0033	0.0036	0.0044
6	0.7655	0.7739	0.7669	0.7632	0.7511	0.0025	0.0021	0.0026	0.0025	0.0027
7	0.7534	0.7399	0.7347	0.7278	0.7205	0.0034	0.0040	0.0042	0.0042	0.0045
CLASSIFICATION ACCURACY - KNOWN PROJECTIONS										
	Mean					Variance				
	0	***0.02***	***0.05***	***0.1***	***0.2***	***0***	***0.02***	***0.05***	***0.1***	***0.2***
1	0.9751	0.9731	0.9686	0.9637	0.9514	0.0000	0.0000	0.0000	0.0001	0.0000
2	0.9333	0.9297	0.9227	0.9067	0.8946	0.0001	0.0001	0.0001	0.0001	0.0001
3	0.9053	0.8967	0.8914	0.8777	0.8618	0.0004	0.0005	0.0005	0.0007	0.0007
4	0.8820	0.8781	0.8657	0.8541	0.8331	0.0011	0.0011	0.0014	0.0014	0.0020
5	0.8714	0.8641	0.8523	0.8429	0.8209	0.0015	0.0015	0.0018	0.0019	0.0023
6	0.8566	0.8497	0.8377	0.8285	0.8074	0.0014	0.0015	0.0016	0.0023	0.0021
7	0.8429	0.8371	0.8256	0.8122	0.7988	0.0015	0.0018	0.0018	0.0021	0.0020

RIPR results on artificial data for semi-supervised classification

To evaluate RIPR semi-supervised classification, we use the same type of synthetic data as in [17], 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 7.3: 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 7.3 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: Similar results are obtained 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.

RIPR results on artificial data for 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:

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

Table 7.4 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 7.4: 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

RIPR case studies on real data

Artifact Detection from Partially-Observed Vital Signals of Intensive Care Unit Patients

A feature of the RIPR algorithm 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 new capability expands practical applicability of RIPR. The set of relevant examples includes a medical informatics application.

Recovery of meaningful, explainable 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.

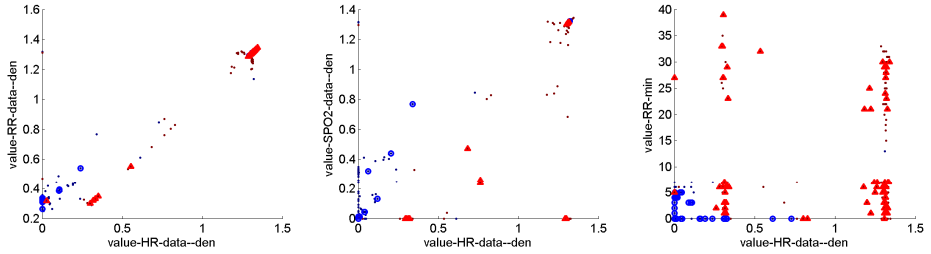

Figure 7.1: RIPR for Respiratory Rate alerts. Artifacts: Blue circles. True instability: Red triangles.

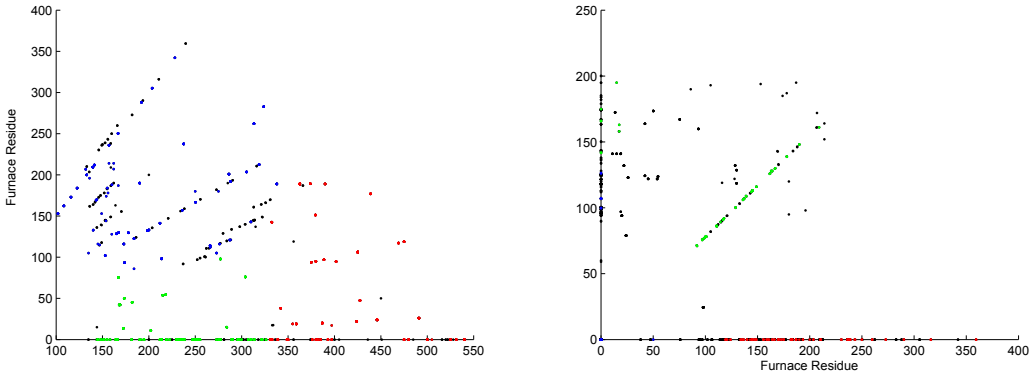

Figure 7.3: Clusters induced from the Concrete dataset.

Figure 7.1 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, all computed with leave-one-out cross-validation. 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 were found to exhibit artifact-like characteristics. Further validation shown these instances to be labeled incorrectly in the original data.

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 7.2 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.

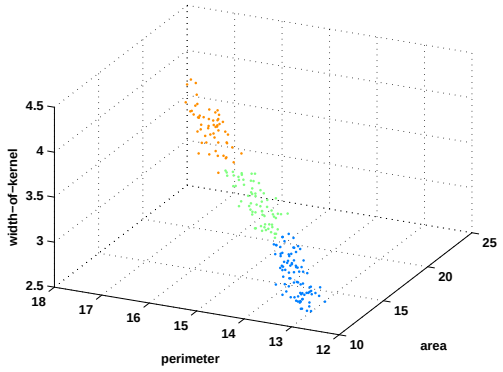

Figure 7.2: Clusters from the Seeds dataset

Figure 7.3 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.

Comparison of ActiveRIPR scoring functions on artificial data

At each iteration, a batch of 30 training data points is selected for expert labeling. We track the accuracy of the models at each iteration using hold-out test data. For this setup, model-conforming points will improve the model accuracy when selected for labeling. Each can be classified correctly using one of the informative projections, and thus the placement of the low losses they incur pinpoints the appropriate set of projections. On the other hand, non-conforming (noisy) data do not follow this pattern and tend to confuse the model as their labels are random. In view of this, we consider a baseline strategy of requesting labels for conforming points first. Clearly, for non-artificial data we would not be able to apply this since we would have no prior knowledge of which data have noisy labels, but this baseline is an indicator of the upper limits of performance. When noise is distributed uniformly, this strategy is optimal, since all the samples that are labeled can actually be useful to the model.

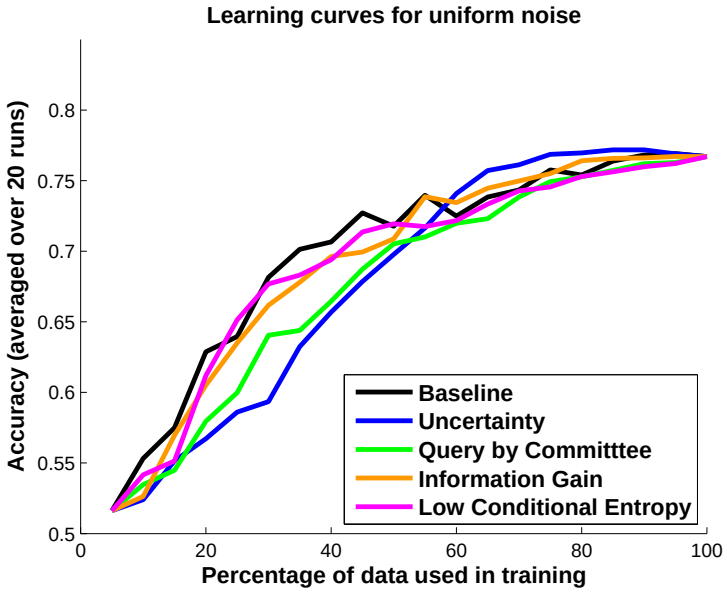

Figure 7.4: ActiveRIPR on artificial data with uniform noise.

Figure 7.4 shows learning curves for ActiveRIPR using different scoring functions when the non-conforming points in the artificial data are distributed uniformly. All methods rebuild RIPR models from scratch after each batch of data is labeled. We do this to mitigate any model bias from previous iterations. The results confirm our intuition about the noisy samples: we can see that as long as model-conforming data is available for labeling, the baseline performs, overall, slightly better than the rest, while its performance saturates once only noisy data is available. Sampling by low conditional entropy and information gain perform well. Uncertainty sampling seems to pick out the non-conforming samples, unhelpful to the models.

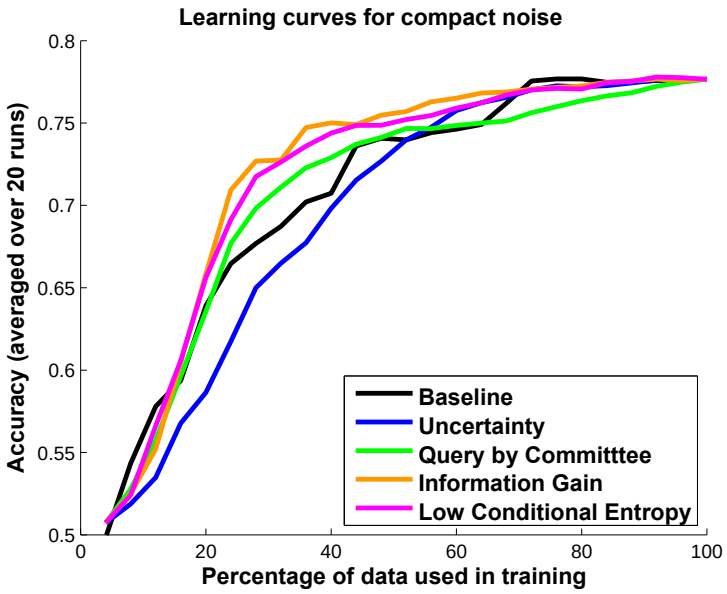

Figure 7.5: ActiveRIPR on artificial data with compact noise.

It is apparent that little improvement can be brought to this type of data if the noise is distributed randomly. In fact, random sample selection does not perform significantly worse than either of the sampling methods used by ActiveRIPR. We now turn our attention to the case when the noise is distributed in more compact areas of the feature space. This time, the scoring functions we previously introduced prove useful, as shown in Figure 7.5. We keep the same baseline as in the previous experiment: the model-conforming samples are to be selected first. However, for compact noise, this strategy is no longer optimal as the model-conforming samples differ in their proximity to, or overlap with, the noisy part of the feature space. Although using model-conforming

data helps briefly at first, the baseline is soon outperformed by information gain and low conditional entropy-based selection. On the other hand, uncertainty sampling performs poorly and query by committee is also not competitive.

These results are averaged over twenty executions of the algorithm with the same generated data, but with different starting samples. It turns out that the steepness of the curve differs considerably depending on the starting sample. For this reason, it is difficult to establish confidence bounds on accuracy with respect to data permutations. As a reminder, our algorithm is deterministic once the initial sample is set as this best answers the needs of our application. Nevertheless, we can determine whether the relative performance of the scoring techniques is consistent. We perform paired t-tests for pair-wise comparison between alternative methods, after each batch of training data. Thus, at each active learning iteration, for any scoring function s_0 and any of its competitors $s_{\bar{0}}$, we obtain a p-value indicating the significance of the win/loss of s_0 over $s_{\bar{0}}$.

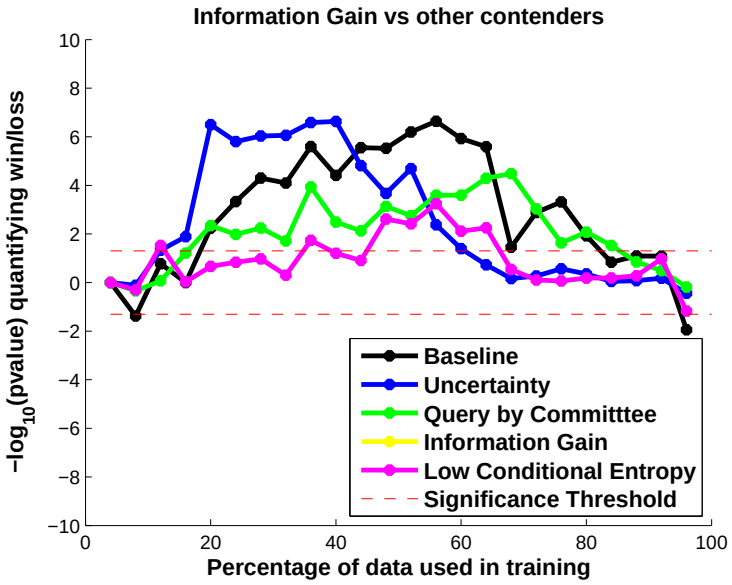

Figure 7.6: ActiveRIPR comparison significance for Information Gain scoring against other contenders. Significant wins/losses are above/below the red dash coresponding to a p-value of 0.05. Artificial data with compact noise.

For each considered method, we plot the negative decimal logarithm of the p-value in a *win/loss graph*, such as the one shown in Figure 7.6 for the InfoGain scoring method. Each line corresponds to a set of p-values obtained when comparing InfoGain to another contender. The p-value in the case of a win – i.e. when InfoGain outperforms the contender – is placed in the positive interval of the y-axis. On the other hand, if the method loses, the p-value is reversed. The two dashed lines distinguish significant wins/losses from insignificant ones. The top dashed line corresponds to $y = -\log_{10}(0.05)$, whereas the bottom one corresponds to $y = \log_{10}(0.05)$. Thus, anything above the top line is a significant win, anything between the dashed lines is not significant, and anything below the bottom line shows a significant loss. Also, we are mainly interested in significant results in the first and the middle part of the x-axis. In the first few iterations, we do not expect considerable difference between scoring functions since the initial sample is the same. With more iterations, the well performing scoring methods may achieve significant wins. Finally, when all useful data is labeled, all methods begin to converge to the same accuracy, typically with no significant wins/losses. The plot for InfoGain scoring in Figure 7.6, for compact noise artificial data, shows that InfoGain obtains significant wins over all other methods. For conditional entropy-based scoring this only starts after 50% of the data has been labeled. For all other scoring functions, this begins to happen after only 20% of the data has been requested for labeling.

Figure 7.7 displays the win/loss graphs for the other scoring functions. We may conclude that uncertainty scoring loses consistently against other methods until after 60% of the data has been labeled, that query-by-committee has no significant wins and that the baseline is outperformed by both InfoGain and LowCondEntropy. In fact, LowCondEntropy seems the second-best perfomer after InfoGain in terms of significant wins, while being the cheapest to compute.

The *computational efficiency* of InfoGain with ActiveRIPR has a linear (not a high-order) dependency on the InfoGain selection because the training and sample selection are performed sequentially. Moreover, only features selected by RIPR are used by the InfoGain selection, making the procedure *less time-consuming than computing information gain over the full-dimensional space*.

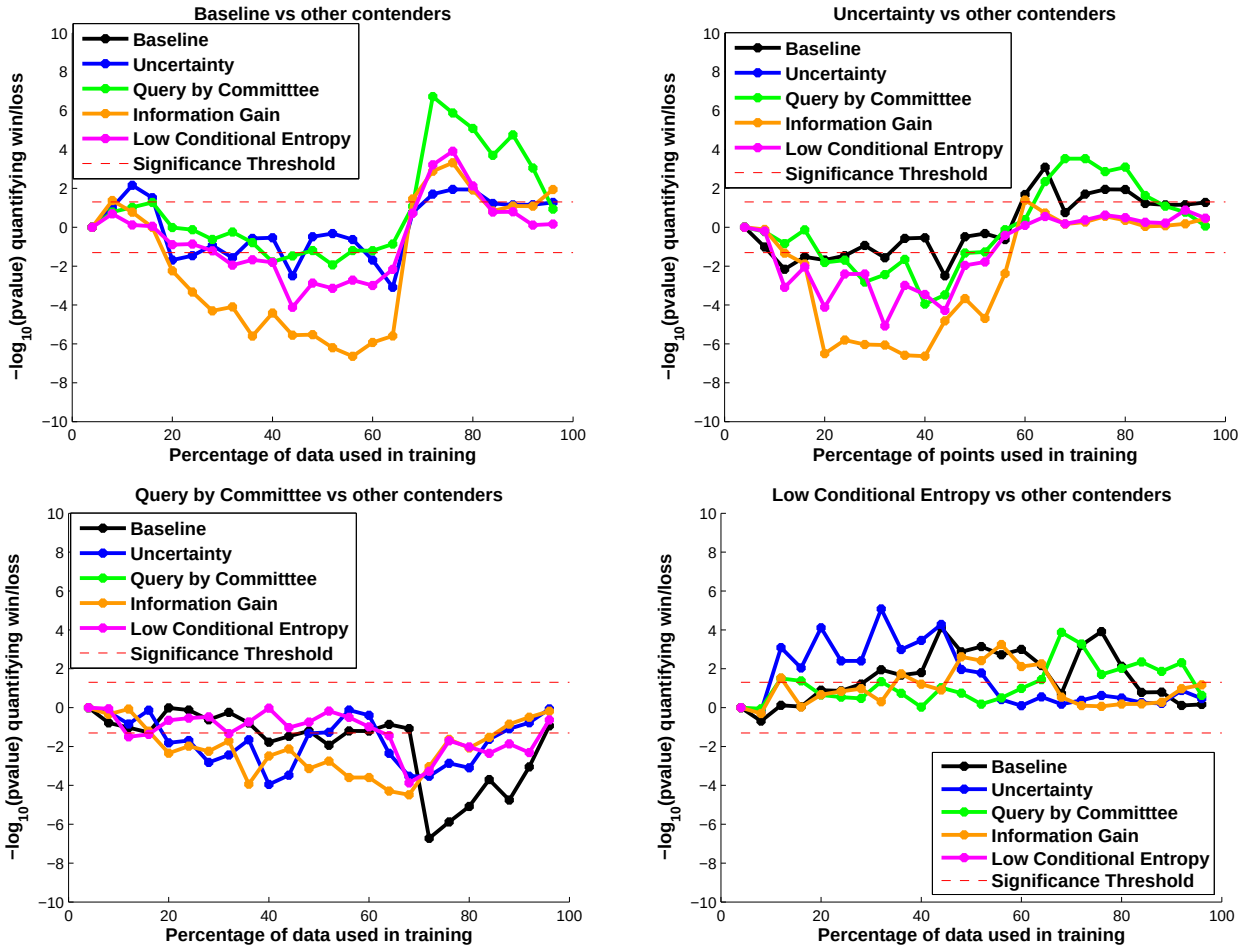

Figure 7.7: ActiveRIPR comparison significance for the baseline (top left), uncertainty (top right), query-by-committee (bottom left) and conditional entropy (bottom right) scoring against their respective contenders. Significant wins/losses are above/below the red dash coresponding to a p-value of 0.05. Artificial data with compact noise.