Prototypal Analysis and Prototypal Regression

Chenyue Wu CHENYUE@CIMS.NYU.EDU
Esteban G. Tabak TABAK@CIMS.NYU.EDU
Courant Institute of Mathematical Sciences
New York University
New York, NY 10012, USA

Editor:

Abstract

Prototypal analysis is introduced to overcome two shortcoming of archetypal analysis: its sensitivity two outliers and its non-locality, which reduces its applicability as a learning tool. Same as archetypal analysis, prototypal analysis finds prototypes through convex combination of the data points and approximates the data through convex combination of the archetypes, but it adds a penalty for using prototypes distant from the data points for their reconstruction. Prototypal analysis can be extended via kernel embedding to probability distributions, since the convexity of the prototypes makes them interpretable as mixtures. Finally, prototypal regression is developed, a robust supervised procedure which allows the use of distributions as either features or labels.

Keywords: Archetypal Analysis, Prototype Analysis, Distribution Regression, Reproducing Kernel Hilbert Space, Kernel Embedding

1. Introduction

Archetypal analysis, an unsupervised learning method introduced by Cutler and Breiman (1994), approximates a set of data points by convex combinations of archetypes, which are themselves convex combinations of the original data. At the cost of introducing convexity constraints into the optimization, archetypal analysis achieves interpretability, as a convex combination can be thought of as a weighted sum of its components –not so a general linear combination, where components can be subtracted as well as added. This extra computational cost can be handled efficiently, as several studies have shown (Bauckhage and Thurau, 2009; Mørup and Hansen, 2012; Chen et al., 2014).

Archetypal analysis has been applied in physics (Stone and Cutler, 1996; Stone, 2002; Chan et al., 2003), biology (Huggins et al., 2007; Römer et al., 2012; Thøgersen et al., 2013), psychology (Thurau and Drachen, 2011; Drachen et al., 2012, 2016; Sifa and Bauckhage, 2013), marketing (Li et al., 2003; DEsposito et al., 2006), performance analysis (Porzio et al., 2006, 2008; Eugster, 2012; Seiler and Wohlrabe, 2013) and computer vision (Marinetti et al., 2006; Thurau and Bauckhage, 2009; Cheema et al., 2011; Asbach et al., 2013; Xiong et al., 2013).

Despite the many positive features of archetypal analysis, one can point out two significant drawbacks. One is its sensitivity to outliers: since the data is approximated by its projection on the convex hull of the archetypes, adding a point outside of the boundary of the data impacts the archetypes to a large degree. Another drawback of the methodology is its non-locality: data points are approximated as convex combinations of archetypes that may be very far away. For many learning tools, such as regression, such representation is of little use.

This paper introduces prototypal analysis as a robust alternative to archetypal analysis without these drawbacks. Prototypal analysis preserves interpretability, as it finds prototypes via convex combinations of the data and reconstructs the data as convex combinations of the prototypes. The difference between archetypal and prototypal analysis is that the former allows arbitrary convex combination of archetypes for representing the data, while the later penalizes the use of prototypes far away from a data point to represent it. Technically, this is achieved by adding a L_1 penalty term on the reconstructing coefficients for each point, with weights that depend on the distance between the point and the prototype under consideration. As a consequence, a point far away from the majority of the data would contribute little to the reconstruction and will not be chosen as a prototype.

The locality of the reconstruction by prototypes makes them useful for key learning tasks such as regression. Given training data on predictors and responses, regression concerns inferring the response for new instances of the predictors. We introduce prototypal regression as a new regression method with the advantage of interpretability and robustness. Prototypal regression uses convex combinations to extract prototypes from both the predictors and the response. The regression relationship is built with pairs of one prototype from the predictor and one prototype from the response, i.e. prototypal regression maps each prototype from the predictor to one prototype from the response and extends to all values of the predictors via local convex combinations. Here convexity is the source of interpretability and, combined with locality, of robustness, as an outlier will only affect the predictions in its immediate neighborhood.

Kernel methods and reproducing kernel Hilbert space (RKHS) are widely used in machine learning to extend algorithms where only inner products among data points are required (Schölkopf and Smola, 2002; Shawe-Taylor and Cristianini, 2004; Hofmann et al., 2008). This is the case of archetypal analysis, which can therefore be extended via kernels (Mørup and Hansen, 2012). Examples of application can be found in time series clustering (Bauckhage and Manshaei, 2014), behavior analysis (Sifa et al., 2014) and image processing (Zhao et al., 2015; Zhao and Zhao, 2016). Prototypal analysis and prototypal regression can be kernelized as well, enabling in particular the use of probability distributions as either features or outputs, in lieu of the more conventional discrete or real-valued scalars and vectors. This extension is particularly well suited for archetypal and prototypal analysis, as their underlying convex combinations correspond to mixtures of distributions. We adopt kernel embedding (also known as kernel mean embedding) to extend archetypal analysis, prototypal analysis and prototypal regression to handle distributional data. Kernel embedding maps probability distributions or their samples into a RKHS. Using the inner products of the RKHS, one can find archetypes and prototypes of distributions and also perform regression in this infinite dimensional setting. More generally, kernel embedding enables prototypal analysis to deal with a blend of categorical, numerical and distributional data.

In prior work, Muandet et al. (2012) extends support vector machine to support measure machine for classification of distributions using the kernel embedding induced inner product. Szabó et al. (2015, 2016) performs a similar extension for kernel ridge regression.

Póczos et al. (2013) regresses numbers from distributions through a kernel-kernel estimator, which involves one kernel for density estimation and another for kernel smoothing, using the distance between the distributions to weight the response variables. Oliva et al. (2013) introduces a distribution to distribution regression model via orthogonal series density estimation on the response distributions and kernel density estimation on the predictor distributions and the new input.

The rest of this paper is organized as follows: section 2 briefly reviews archetypal analysis and empirically shows that it is not robust to outliers and that, as it concentrates on the boundary of the data, it does not resolve the underlying space well. Section 3 introduces prototypal analysis as a robust unsupervised method to find prototypes and build data-driven barycentric coordinates system without these two drawbacks. Section 4 introduces simple and multiple prototypal regression –the latter applicable to features of different nature that cannot naturally be regarded as components of a vector. Section 5 extends archetypal and prototypal analysis and prototypal regression via kernels and applies it to the analysis of distributional data.

2. Archetypal Analysis

Archetypal analysis approximates data points by convex combination of "archetypes", which are themselves convex combinations of the data points (see Cutler and Breiman, 1994). Given a data set $\{\mathbf{x}_i\}_{i=1}^n$, one seeks archetypes of the form

$$\mathbf{u}_j = \sum_{i=1}^n b_{ij}\mathbf{x}_i, \quad \sum_{i=1}^n b_{ij} = 1, \quad b_{ij} \geq 0, \quad j \in [1, k] \tag{1}$$

and approximates each data point through

$$\mathbf{x}_i \approx \sum_{j=1}^k a_{ji}\mathbf{u}_j, \quad \sum_{j=1}^k a_{ji} = 1, \quad a_{ji} \geq 0, \quad i \in [1, n], \tag{2}$$

by solving the following optimization problem:

$$\min_{\substack{a_{ji}\geq 0, b_{lj}\geq 0 \\ \sum_{j=1}^k a_{ji}=1 \\ \sum_{l=1}^n b_{lj}=1}} \sum_{i=1}^n \left\| \mathbf{x}_i - \sum_{j=1}^k a_{ji} \sum_{l=1}^n b_{lj}\mathbf{x}_l \right\|^2. \tag{3}$$

As archetypal analysis minimizes the distance between the data and the convex hull of the archetypes, it tends to choose as archetypes extreme points among the data in order to enlarge this convex hull. In particular, when the data includes outliers, these are typically chosen as archetypes, as illustrated in Figure 1. As the number k of archetypes grows, they sit on the boundary of the convex hull of the data, not resolving its interior, as shown in Figure 2. Also, when k is sufficiently large (typically when $k > d + 1$, where d is the dimension of the space $\mathbf{x}$ of features), the a_{ji} are not uniquely defined.

Algorithm 1 Archetypal Analysis

```
Input: Data $\{x_i\}_{i=1}^n$, $k$: number of archetypes.
Output: Archetypes $\{u_j\}_{j=1}^k$ and approximation $\{\hat{x}_i\}_{i=1}^n$ to data by their convex combination.
$(a_{ji}), (b_{lj}) \leftarrow \mathop{\arg\min}_{\substack{a_{ji}\geq 0, b_{lj}\geq 0 \\ a_{1i}+\cdots+a_{ki}=1 \\ b_{1j}+\cdots+b_{nj}=1}} \sum_{i=1}^{n} \left\| \mathbf{x}_i - \sum_{j=1}^{k} a_{ji} \sum_{l=1}^{n} b_{lj}\mathbf{x}_l \right\|^2$
for $j = 1, \cdots, k$ do
    $u_j \leftarrow b_{1j}\mathbf{x}_1 + \cdots + b_{nj}\mathbf{x}_n$
end for
for $i = 1, \cdots, n$ do
    $\hat{x}_i \leftarrow a_{1i}\mathbf{u}_1 + \cdots + a_{ki}\mathbf{u}_k$
end for
return $\{u_j\}_{j=1}^k$, $\{\hat{x}_i\}_{i=1}^n$
```

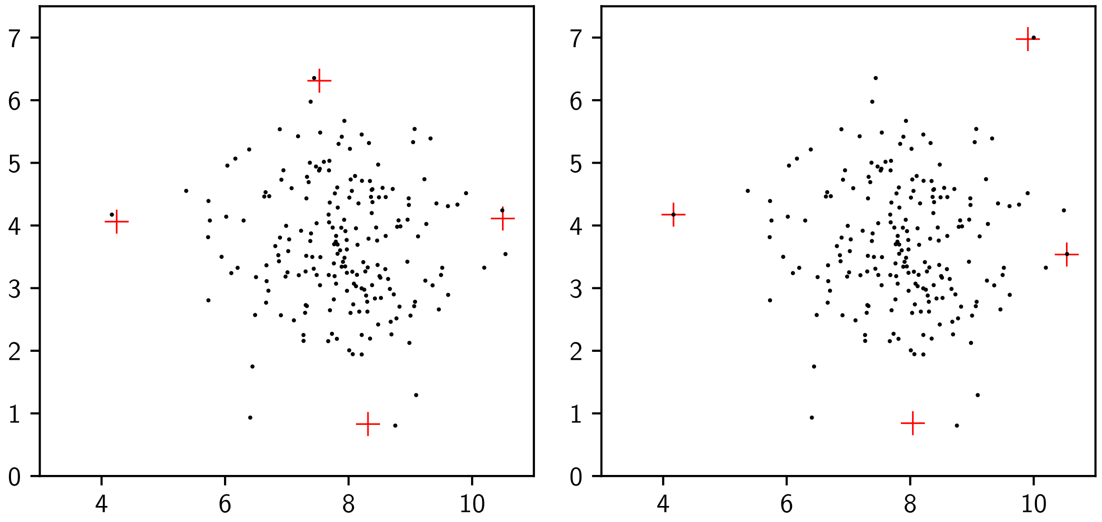

Figure 1: Archetypal analysis on two dimensional data with 4 archetypes. Data of the right figure contains one more outlier than the left figure. The archetypes are visualized using the '+' sign. Adding one outlier fundamentally changes the location of the archetypes. In addition, the reconstruction of many data-points in terms of the archetypes is not unique.

3. Prototypal Analysis

Like archetypal analysis, prototypal analysis finds prototypes $\{\mathbf{u}_j\}_{j=1}^k$as convex combinations of the data points $\{\mathbf{x}_i\}_{i=1}^n$, and approximates the latter as convex combinations of the former, as in (1, 2). The difference lies in that, when reconstructing each data point, prototypal analysis is biased toward using prototypes near that point. To this end, it adds

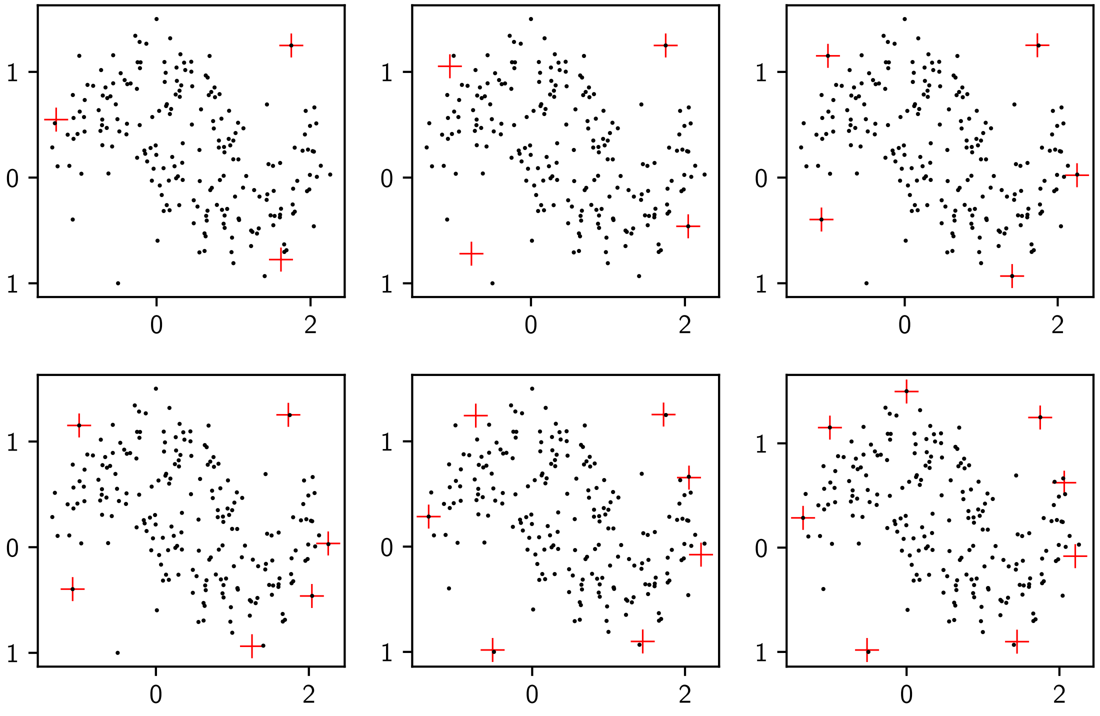

Figure 2: Archetypal analysis on two dimensional data with 3, 4, 5, 6, 7 and 8 archetypes. The archetypes are visualized using the '+' sign. As the number of archetypes grows, they cover just the perimeter of the convex hull of the data.

a penalty term on the distance between points and prototypes, replacing the objective function in (3) by

$$\min_{\substack{a_{ji}\geq 0, b_{lj}\geq 0\\ \sum_{j=1}^{k} a_{ji}=1\\ \sum_{l=1}^{n} b_{lj}=1}} \sum_{i=1}^{n} \left\| \mathbf{x}_i - \sum_{j=1}^{k} a_{ji} \sum_{l=1}^{n} b_{lj}\mathbf{x}_l \right\|^2 + \lambda \sum_{i=1}^{n}\sum_{j=1}^{k} a_{ji} \left\| \mathbf{x}_i - \sum_{l=1}^{n} b_{lj}\mathbf{x}_l \right\|^2, \tag{4}$$

where $\lambda \geq 0$ is a tuning parameter. In the penalty term, a_{ji}, the weight of the j-th archetype in the reconstruction of $\mathbf{x}_i$, is multiplied by $\|\mathbf{x}_i - \sum_{l=1}^{n} b_{lj}\mathbf{x}_l\|^2$, the square of distance between data point $\mathbf{x}_i$ and the j-th prototype $\mathbf{u}_j = \sum_{l=1}^{n} b_{lj}\mathbf{x}_l$. Hence the closer $\mathbf{x}_i$ is to the j-th prototype, the more weight this prototype will be assigned in the reconstruction. Compared with archetypal analysis, which tends to use extreme points as archetypes, prototypal analysis has prototypes that resemble the original data. Hence it is less sensitive to outliers. Figure 3 shows the prototypes corresponding to the same data of Figure 1. In this case, adding one outlier does not change the archetypes significantly. In the computational procedure we use to minimize (4), we alternate between minimizing over the a and b, which is also the procedure of choice in archetypal analysis (Cutler and Breiman, 1994).

Algorithm 2 Prototypal Analysis

```
Input: Data $\{x_i\}_{i=1}^n$, number of prototypes $k$, penalty coefficient $\lambda$.
Output: Prototypes $\{u_j\}_{j=1}^k$ and reconstruction of data by archetypes $\{\hat{x}_i\}_{i=1}^n$.
(a_{ji}), (b_{lj}) ← $\underset{\substack{a_{ji}\geq 0, b_{lj}\geq 0\\ a_{1i}+\cdots+a_{ki}=1\\ b_{1j}+\cdots+b_{nj}=1}}{\arg\min} \sum_{i=1}^n \left\| \mathbf{x}_i - \sum_{j=1}^k a_{ji} \sum_{l=1}^n b_{lj}\mathbf{x}_l \right\|^2 + \lambda \sum_{i=1}^n \sum_{j=1}^k a_{ji} \left\| \mathbf{x}_i - \sum_{l=1}^n b_{lj}\mathbf{x}_l \right\|^2$
for $j = 1, \cdots, k$ do
    $u_j \leftarrow b_{1j}\mathbf{x}_1 + \cdots + b_{nj}\mathbf{x}_n$
end for
for $i = 1, \cdots, n$ do
    $\hat{x}_i \leftarrow a_{1i}\mathbf{u}_1 + \cdots + a_{ki}\mathbf{u}_k$
end for
return $\{u_j\}_{j=1}^k$, $\{\hat{x}_i\}_{i=1}^n$
```

Prototypal analysis can be viewed as a mixture of archetypal analysis and k-means clustering. When λ goes to infinity, only the penalty term remains in prototypal analysis, and the problem reduces to

$$\min_{\substack{a_{ji}\geq 0, b_{lj}\geq 0\\ \sum_{j=1}^k a_{ji}=1\\ \sum_{l=1}^n b_{lj}=1}} \sum_{i=1}^n \sum_{j=1}^k a_{ji} \left\| \mathbf{x}_i - \sum_{l=1}^n b_{lj}\mathbf{x}_l \right\|^2, \tag{5}$$

which is equivalent to K-means clustering, with the prototypes $u_j = \sum_{l=1}^n b_{lj}\mathbf{x}_l$ playing the role of barycenters. To see this equivalence, notice two facts about the solution to (5):

1. For each observation $\mathbf{x}_i$, the only nonzero a_{ji} corresponds to the closest u_j, for which $a_{ji} = 1$.

2. For each prototype u_j, the only nonzero b_{lj} correspond to those l such that u_j is the closest prototype to $\mathbf{x}_i$. Moreover, these b_{lj} all have the same value, as the barycenter of a set of points is the minimizer of the sum of the square distances to them.

4. Prototypal Regression

Given a set of predictor-response pairs $(\mathbf{x}_i, \mathbf{y}_i)$, regression is the task estimating the response $\mathbf{y}_0$ corresponding to a new value $\mathbf{x}_0$ of the predictor. Performing prototypal analysis on the $\{\mathbf{x}_i\}$ yields the prototypes $\{u_j\}$ and a rule that approximates $\mathbf{x}_0$ as a convex combination of a local subset of the $\{u_j\}$. Hence introducing prototypes $\{v_j\}$ in $\mathbf{y}$-space that approximate the images of the $\{u_j\}$, one can estimate $\mathbf{y}_0$ as the corresponding convex combination of the $\{v_j\}$.

4.1 Simple Prototypal Regression

Simple prototypal regression estimates the response $\mathbf{y}$ from a single predictor $\mathbf{x}$, where both predictor and response can be vectorial, using prototypes of both $\mathbf{x}$ and $\mathbf{y}$. The prototypes

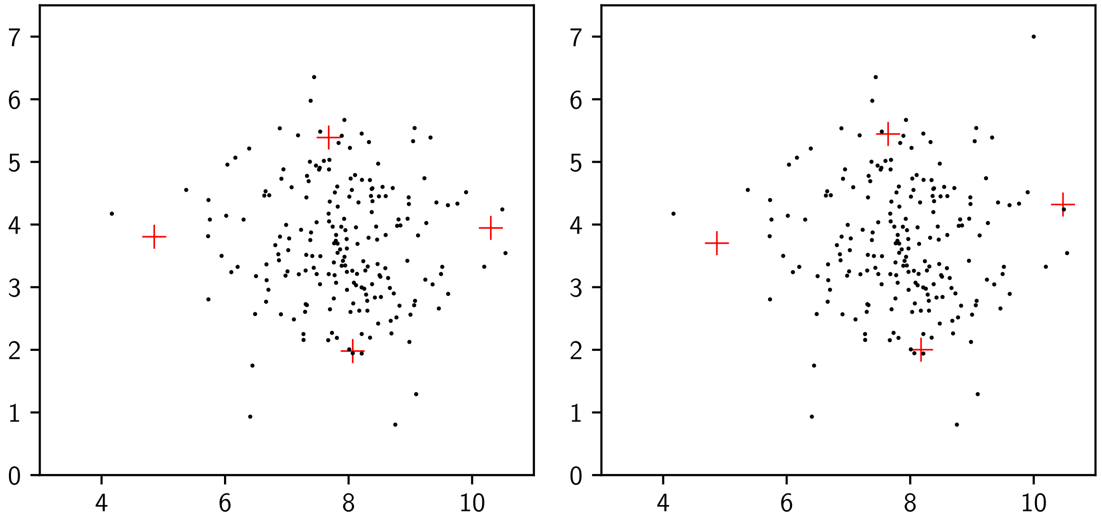

Figure 3: Prototypal analysis on two dimensional data with 4 prototypes and penalty 0.05. The data of the right figure contains one more outlier than the left figure, but this affects the location of the prototypes only minimally. The prototypes are visualized using '+' signs.

of $\mathbf{x}$ come directly from prototypal analysis, i.e. solving (4), while the choice of prototypes of $\mathbf{y}$ takes the regression into account. Denoting by $\mathbf{u}_j$ the prototypes of $\mathbf{x}$ and by $\mathbf{v}_j$ the prototypes of $\mathbf{y}$, the prototype pair $(\mathbf{u}_j, \mathbf{v}_j)$ defines the regression function $\hat{\mathbf{f}}$ via

$$\hat{\mathbf{f}}(\mathbf{x}_0) = a_{10}\mathbf{v}_1 + \cdots + a_{k0}\mathbf{v}_k. \tag{6}$$

where a_{j0} is the barycentric coordinates of $\mathbf{x}_0$ in prototypal analysis:

$$\min_{\substack{a_{j0}\geq 0\\ \sum_{j=1}^k a_{j0}=1}} \left\| \mathbf{x}_0 - \sum_{j=1}^k a_{j0}\mathbf{u}_j \right\|^2 + \lambda \sum_{j=1}^k a_{j0} \left\|\mathbf{x}_0 - \mathbf{u}_j\right\|^2. \tag{7}$$

Given the weights $\{a_{ji}\}$ for reconstructing $\mathbf{x}_i$ in terms of the $\{\mathbf{u}_j\}$, the prototypes $\mathbf{v}_j$ are obtained by minimizing the squared errors of (6) on $(\mathbf{x}_i, \mathbf{y}_i)$, i.e.

$$\mathbf{v}_j = \sum_{i=1}^n c_{ij}\mathbf{y}_i, \quad c = \mathop{\arg\min}_{\substack{c_{lj}\geq 0\\ \sum_{i=1}^n c_{ij}=1}} \sum_{i=1}^n \left\| \mathbf{y}_i - \sum_{j=1}^k a_{ji} \sum_{l=1}^n c_{lj}\mathbf{y}_l \right\|^2. \tag{8}$$

Figure 5 illustrates simple prototypal regression, kernel regression, regression tree and k nearest-neighbor regression on a one-dimensional synthetic data set.

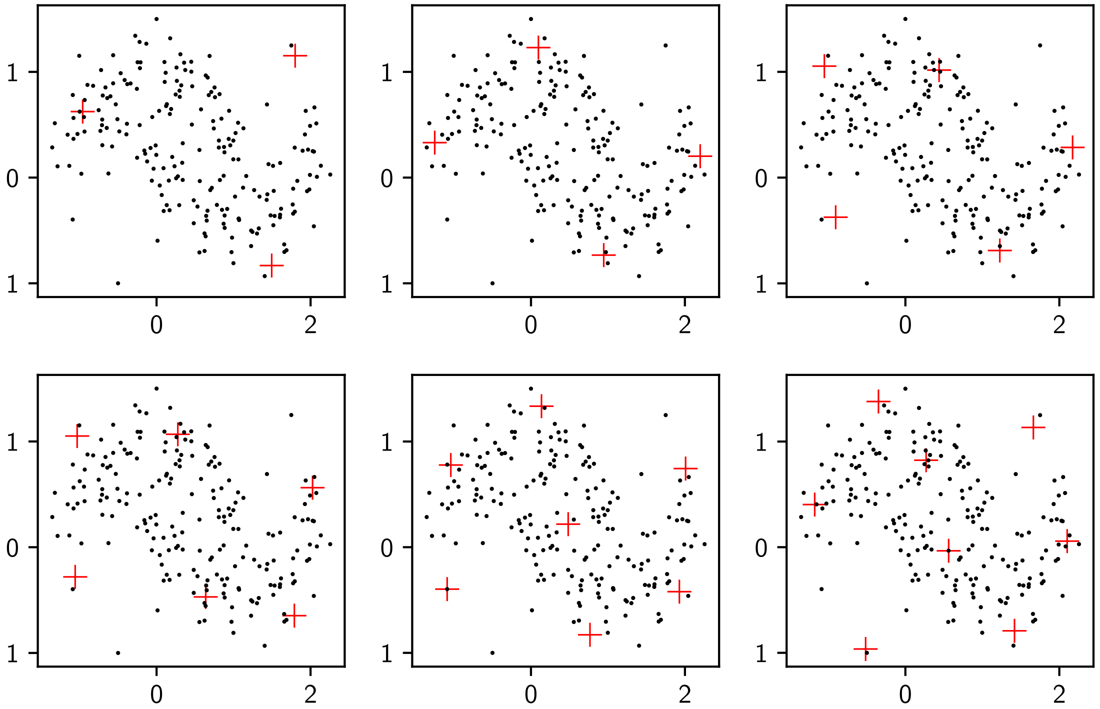

Figure 4: Prototypal analysis on two dimensional data with penalty 0.05. The number k of prototypes is set to 3, 4, 5, 6, 7 and 8. The prototypes are visualized using '+' signs. Unlike archetypes, as the number of prototypes grows, they populate all data-rich areas.

4.2 Multiple Prototypal Regression

Multiple prototypal regression estimates the response $\mathbf{y}$ using m predictors $\{\mathbf{x}^{(l)}\}_{l=1}^{m}$ (again, both the response and each of the predictors can be vectorial.) As in simple prototypal regression, it finds prototypes for $\mathbf{x}^{(l)}$ and $\mathbf{y}$ and builds the regression function on prototypes.

The prototypes of $\mathbf{x}^{(l)}$ still come from direct prototypal analysis, i.e. solving (4) for each $\{\mathbf{x}_i^{(l)}\}_{i=1}^{n}$. Each predictor has k_l prototypes and penalty coefficient λ_l, these need not be the same across predictors. When finding prototypes for $\mathbf{y}$, we weight the prototypes of each $\mathbf{x}^{(l)}$ by an importance coefficient. Denoting by $\mathbf{u}_j^{(l)}$ the prototypes of $\mathbf{x}^{(l)}$ and by $\mathbf{v}_j^{(l)}$ the prototypes of $\mathbf{y}$ corresponding to the l-th predictor, the regression function $\hat{\mathbf{f}}$ in multiple prototypal regression is given by

$$\hat{\mathbf{f}}(\mathbf{x}_0) = \sum_{l=1}^{m} \tau_l \sum_{j=1}^{k_l} a_{j0}^{(l)} \mathbf{v}_j^{(l)}, \tag{9}$$

where $a_j^{(l)}$ are the barycentric coordinates of $\mathbf{x}^{(l)}$ in prototypal analysis as in (7).

The importance coefficients τ_l in (9) are non-negative and add up to one. Both the importance coefficients and the prototypes of $\mathbf{y}$ are obtained by minimizing the squared

Algorithm 3 Simple Prototypal Regression - Fitting

```
Input: Predictor data $\{\mathbf{x}_i\}_{i=1}^n$, response data $\{\mathbf{y}_i\}_{i=1}^n$, number of prototypes $k$, penalty coefficient $\lambda$.
Output: Prototypes $\{\mathbf{u}_j\}_{j=1}^k$ and $\{\mathbf{v}_j\}_{j=1}^k$ for predictor and response respectively.
(a_{ji}), (b_{lj}) ← $\underset{\substack{a_{ji}\geq 0, b_{lj}\geq 0\\ a_{1i}+\cdots+a_{ki}=1\\ b_{1j}+\cdots+b_{nj}=1}}{\arg\min} \sum_{i=1}^{n} \left\| \mathbf{x}_i - \sum_{j=1}^{k} a_{ji} \sum_{l=1}^{n} b_{lj}\mathbf{x}_l \right\|^2 + \lambda \sum_{i=1}^{n}\sum_{j=1}^{k} a_{ji} \left\| \mathbf{x}_i - \sum_{l=1}^{n} b_{lj}\mathbf{x}_l \right\|^2$
for $j = 1, \cdots, k$ do
    $\mathbf{u}_j \leftarrow b_{1j}\mathbf{x}_1 + \cdots + b_{nj}\mathbf{x}_n$
end for
$(c_{lj}) \leftarrow \underset{\substack{c_{lj}\geq 0\\ c_{1j}+\cdots+c_{nj}=1}}{\arg\min} \sum_{i=1}^{n} \left\| \mathbf{y}_i - \sum_{j=1}^{k} a_{ji} \sum_{l=1}^{n} c_{lj}\mathbf{y}_l \right\|^2$
for $j = 1, \cdots, k$ do
    $\mathbf{v}_j \leftarrow c_{1j}\mathbf{y}_1 + \cdots + c_{nj}\mathbf{y}_n$
end for
return $\{\mathbf{u}_j\}_{j=1}^k$, $\{\mathbf{v}_j\}_{j=1}^k$
```

Algorithm 4 Simple Prototypal Regression - Prediction

```
Input: Value $\mathbf{x}_0$ of the predictor, prototypes $\{\mathbf{u}_j\}_{j=1}^k$ and $\{\mathbf{v}_j\}_{j=1}^k$ for predictor and response respectively, penalty coefficient $\lambda$.
Output: Predicted $\hat{\mathbf{y}}_0$.
$(a_j) \leftarrow \underset{\substack{a_j\geq 0\\ a_1+\cdots+a_k=1}}{\arg\min} \left\| \mathbf{x}_0 - \sum_{j=1}^{k} a_j\mathbf{u}_j \right\|^2 + \lambda \sum_{j=1}^{k} a_j \left\| \mathbf{x}_0 - \mathbf{u}_j \right\|^2$
$\hat{\mathbf{y}}_0 \leftarrow a_1\mathbf{v}_1 + \cdots + a_n\mathbf{v}_n$
return $\hat{\mathbf{y}}_0$
```

errors of (9) on the data: denoting by $a_{ji}^{(l)}$ the weight of $\mathbf{u}_j^{(l)}$ for reconstructing $\mathbf{x}_i^{(l)}$,

$$\mathbf{v}_j^{(l)} = \sum_{i=1}^{n} c_{ij}^{(l)}\mathbf{y}_i, \quad c, \tau = \underset{\substack{c_{hj}^{(l)}, \tau_l \geq 0\\ c_{1j}^{(l)}+\cdots+c_{nj}^{(l)}=1\\ \tau_1+\cdots+\tau_m=1}}{\arg\min} \sum_{i=1}^{n} \left\| \mathbf{y}_i - \sum_{l=1}^{m} \tau_l \sum_{j=1}^{k_l} a_{ji}^{(l)} \sum_{h=1}^{n} c_{hj}^{(l)}\mathbf{y}_h \right\|^2. \tag{10}$$

Here the optimization is carried out through the alternate minimization over the b and τ.

4.3 Applications

4.3.1 Iris Flowers

We apply multiple prototypal regression to the data set for classification of Iris into species introduced by Fisher (1936). This includes three Iris species with four features for each flower: sepal length, sepal width, petal length and petal width. In this example, we treat

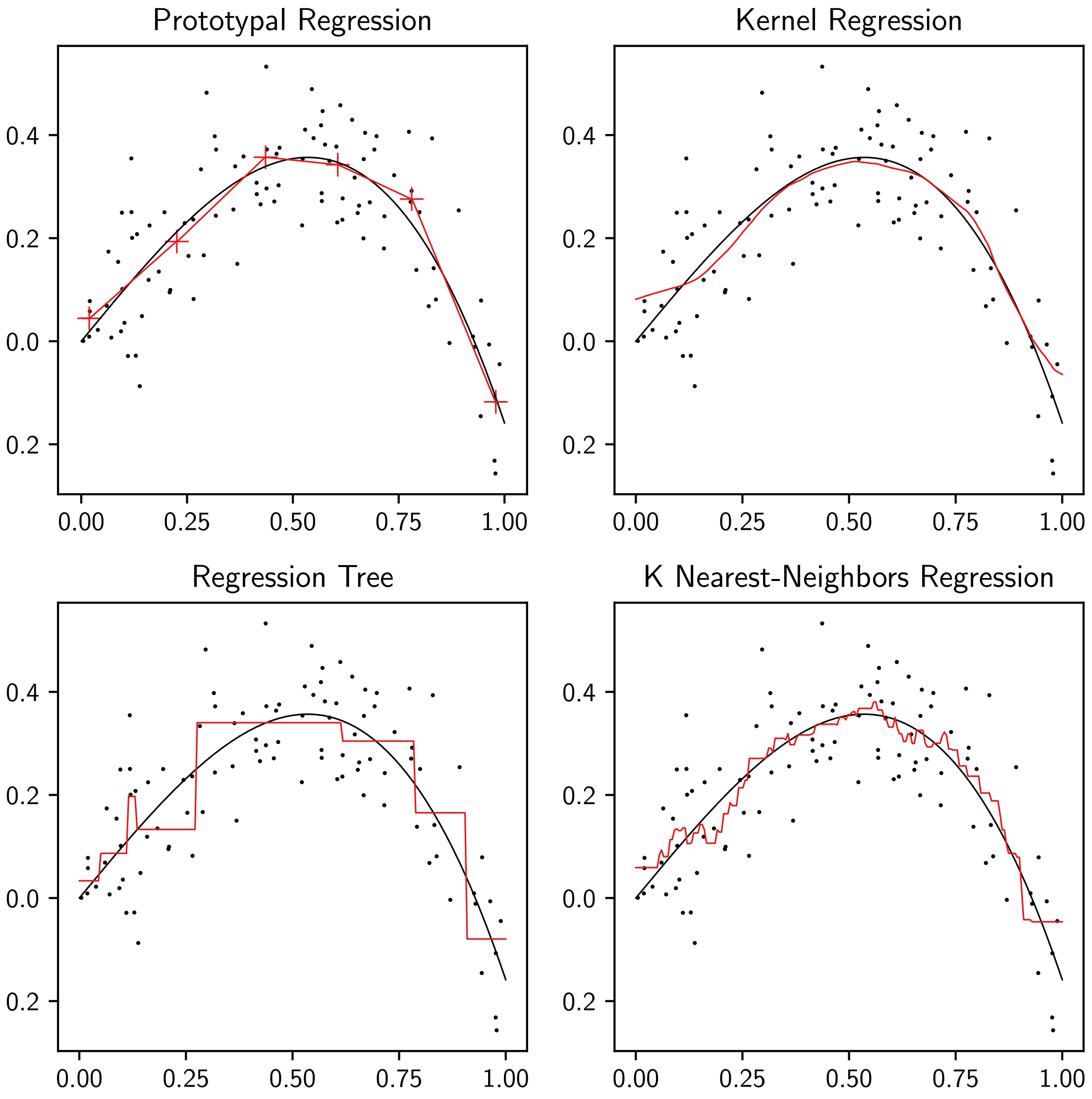

Figure 5: 100 pairs x_i, y_i are sampled from a Gaussian conditional distribution with conditional mean $\bar{y} = \sin(x) - x^3$ (the black curve), $x \sim U[0,1]$, $y = \bar{y} + \epsilon, \epsilon \sim \mathcal{N}(0, 0.1)$. The red curves arise from regression. Top left panel: prototypal regression with 6 prototypes and penalty 0.01 (The prototypes of x_i and y_i are visualized using '+' signs.) Top right panel: kernel regression with Epanechnikov kernel with (half) window width $\lambda = 0.15$. Lower left panel: regression tree. Lower right panel: 10-nearest-neighbor regression.

the sepal and petal dimensions as two two-dimensional predictors and one-hot encode the three species as $(1, 0, 0)$, $(0, 1, 0)$ and $(0, 0, 1)$. Multiple prototypal regression predicts a probability vector given the sepal and petal features. The species with highest probability is then adopted as predicted label.

Algorithm 5 Multiple Prototypal Regression - Fitting

```
Input: Predictor data $\{\mathbf{x}_i^{(1)}\}_{i=1}^n, \cdots, \{\mathbf{x}_i^{(m)}\}_{i=1}^n$, response data $\{\mathbf{y}_i\}_{i=1}^n$, number of prototypes $k_1, \cdots, k_m$, penalty coefficient $\lambda_1, \cdots, \lambda_m$.
Output: Prototypes $\{\mathbf{u}_j^{(1)}\}_{j=1}^{k_1}, \cdots, \{\mathbf{u}_j^{(m)}\}_{j=1}^{k_m}$ for predictors and $\{\mathbf{v}_j^{(1)}\}_{j=1}^{k_1}, \cdots, \{\mathbf{v}_j^{(m)}\}_{j=1}^{k_m}$ for response, importance coefficients $\tau_1, \cdots, \tau_m$.
for $l = 1, \cdots, m$ do
    $(a_{ji}^{(l)}), (b_{hj}^{(l)}) \leftarrow \underset{\substack{a_{ji}^{(l)} \geq 0, b_{lj}^{(l)} \geq 0 \\ a_{1i}^{(l)} + \cdots + a_{ki}^{(l)} = 1 \\ b_{1j}^{(l)} + \cdots + b_{nj}^{(l)} = 1}}{\arg\min} \sum_{i=1}^n \left\| \mathbf{x}_i^{(l)} - \sum_{j=1}^{k_l} a_{ji}^{(l)} \sum_{h=1}^n b_{hj}^{(l)} \mathbf{x}_h^{(l)} \right\|^2 + \lambda_l \sum_{i=1}^n \sum_{j=1}^k a_{ji}^{(l)} \left\| \mathbf{x}_i^{(l)} - \sum_{h=1}^n b_{hj}^{(l)} \mathbf{x}_h^{(l)} \right\|^2$
    for $j = 1, \cdots, k_l$ do
        $\mathbf{u}_j^{(l)} \leftarrow b_{1j}^{(l)} \mathbf{x}_1^{(l)} + \cdots + b_{nj}^{(l)} \mathbf{x}_n^{(l)}$
    end for
end for
$(c_{hj}^{(l)}), (\tau_l) \leftarrow \underset{\substack{b_{hj}^{(l)}, \tau_l \geq 0 \\ c_{1j}^{(l)} + \cdots + c_{nj}^{(l)} = 1 \\ \tau_1 + \cdots + \tau_m = 1}}{\arg\min} \sum_{i=1}^n \left\| \mathbf{y}_i - \sum_{l=1}^m \tau_l \sum_{j=1}^{k_l} a_{ji}^{(l)} \sum_{h=1}^n c_{hj}^{(l)} \mathbf{y}_h \right\|^2$
for $l = 1, \cdots, m$ do
    for $j = 1, \cdots, k$ do
        $\mathbf{v}_j^{(l)} \leftarrow c_{1j}^{(l)} \mathbf{y}_1 + \cdots + c_{nj}^{(l)} \mathbf{y}_n$
    end for
end for
return $\{\mathbf{u}_j^{(1)}\}_{j=1}^{k_1}, \cdots, \{\mathbf{u}_j^{(m)}\}_{j=1}^{k_m}, \{\mathbf{v}_j^{(1)}\}_{j=1}^{k_1}, \cdots, \{\mathbf{v}_j^{(m)}\}_{j=1}^{k_m}, \tau_1, \cdots, \tau_m$
```

There are 150 samples in the Iris data set with 50 samples for each species. Using stratified sampling, we randomly split the samples into a training set of 105 samples and a test set of 45 samples. By grid search with cross validation on the training data, we pick the number of prototypes to be 11 and the penalty coefficient to be 0.1 for both features. The accuracy scores on the training and testing sets are shown in Table 1.

The Iris data set and the prototypes of the sepal and petal dimensions are shown in Figure 6. Figure 6 suggests the petal dimensions are more informative than the sepal's for the classification task. This agrees with the importance coefficients of prototypal regression, which are 3×10^{-7} and 0.9999997 for the sepal and petal dimensions respectively. Figure 7 shows the responses of this classification problem and the prototypes of the responses corresponding to the petal dimensions.

Algorithm 6 Multiple Prototypal Regression - Prediction

```
Input: Values $\mathbf{x}_0 = \left(\mathbf{x}_0^{(1)}, \cdots, \mathbf{x}_0^{(m)}\right)$ of the predictors, prototypes $\{\mathbf{u}_j^{(1)}\}_{j=1}^{k_1}, \cdots, \{\mathbf{u}_j^{(m)}\}_{j=1}^{k_m}$ for predictors and $\{\mathbf{v}_j^{(1)}\}_{j=1}^{k_1}, \cdots, \{\mathbf{v}_j^{(m)}\}_{j=1}^{k_m}$ for response, importance coefficients $\tau_1, \cdots, \tau_m$, penalty coefficients $\lambda_1, \cdots, \lambda_m$.
Output: Predicted $\hat{\mathbf{y}}_0$.
for $l = 1, \cdots, m$ do
    $(a_j^{(l)}) \leftarrow \underset{\substack{a_j^{(l)} \geq 0 \\ a_1^{(l)} + \cdots + a_k^{(l)} = 1}}{\arg\min} \left\| \mathbf{x}_0^{(l)} - \sum_{j=1}^{k_l} a_j^{(l)} \mathbf{u}_j^{(l)} \right\|^2 + \lambda_l \sum_{j=1}^{k_l} a_j^{(l)} \left\| \mathbf{x}_0^{(l)} - \mathbf{u}_j^{(l)} \right\|^2$
end for
$\hat{\mathbf{y}}_0 \leftarrow \sum_{l=1}^{m} \tau_l \sum_{j=1}^{k_l} a_j^{(l)} \mathbf{v}_j^{(l)}$
return $\hat{\mathbf{y}}_0$
```

	training score	test score
prototypal regression	0.96	1.00

Table 1: Accuracy score on Iris flowers data set.

5. Kernels and Extension to Probability Distributions

5.1 Prototypal Learning with Kernels

Archetypal analysis, prototypal analysis and prototypal regression involve the data only through the pairwise inner products

$$\langle \mathbf{x}_i, \mathbf{x}_j \rangle \ , \quad \langle \mathbf{y}_i, \mathbf{y}_j \rangle), \tag{11}$$

as follows from expanding the squared norms in formulations (3), (4), (8) and (10). Hence we can extend all three to reproducing kernel Hilbert spaces. Choosing a symmetric and positive semidefinite kernel function K, the map from $\mathbf{x}_i$ to $h(\mathbf{x}_i) = K(\cdot, \mathbf{x}_i)$ yields the inner product

$$\langle h(\mathbf{x}_i), h(\mathbf{x}_j) \rangle = K(\mathbf{x}_i, \mathbf{x}_j), \tag{12}$$

which replaced in (12) in (3), (4), (8) or (10), extends archetypal analysis, prototypal analysis and prototypal regression to a (potentially infinite-dimensional) reproducing kernel Hilbert space.

5.2 Prototypal Learning on Distributions through Kernel Embedding

Probability distributions or samples thereof can also be mapped to a reproducing kernel Hilbert space via kernel embedding (see Berlinet and Thomas-Agnan, 2004; Gretton et al., 2006; Smola et al., 2007; Sriperumbudur et al., 2010; Sejdinovic et al., 2012; Muandet et al., 2017). With a symmetric, positive semidefinite kernel function $K(\cdot, \cdot)$ on $\mathcal{X} \times \mathcal{X}$, the kernel

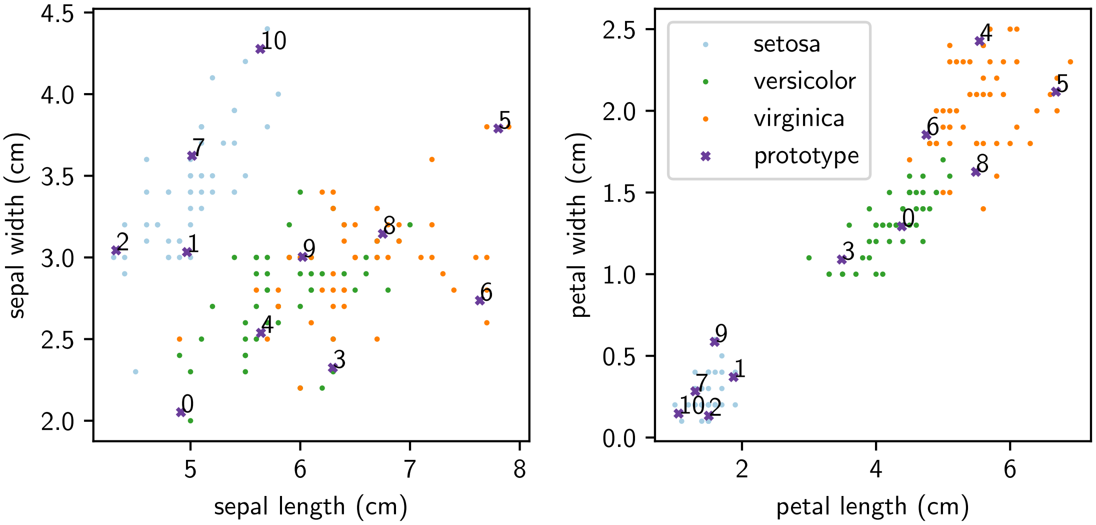

Figure 6: Sepal dimensions and petal dimensions of Iris flowers and their prototypes.

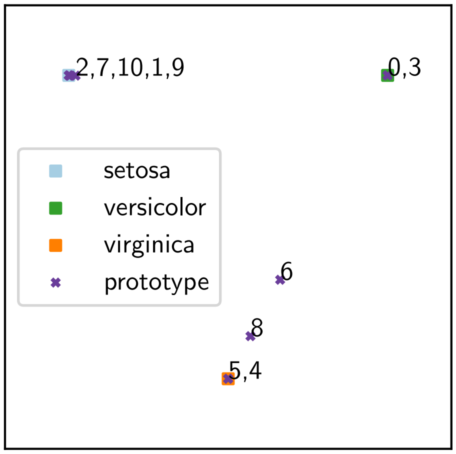

Figure 7: Species of Iris flowers and prototypes corresponding to petal dimensions. This plot of the three-dimensional object (P_1, P_2, P_3) is represented here in barycentric coordinates, where the three vertices of the triangle correspond to the three species.

embedding g maps a probability measure $\mu(\cdot)$ on $\mathcal{X}$ to a reproducing kernel Hilbert space through

$$\mu(\cdot) \mapsto g(\mu(\cdot)) = \int_{\mathcal{X}} K(\cdot, x) d\mu(x), \tag{13}$$

with induced inner product given by

$$\langle g(\mu_1(\cdot)), g(\mu_2(\cdot))\rangle = \int_{\mathcal{X}\times\mathcal{X}} K(x_1,x_2) d\mu_1(x_1) d\mu_2(x_2). \tag{14}$$

Kernel embedding does not necessarily yield an injective map; Sriperumbudur et al. (2010) give several criteria for whether a kernel induces an injective embedding for distributions on $\mathbb{R}^d$ and $\mathbb{T}^d$. Some commonly used kernels on $\mathbb{R}^d$ for injective kernel embeddings are listed in Table 2. The Gaussian, Laplacian and B_{2n+1}-spline kernels are shown to induce injective embeddings in Sriperumbudur et al. (2010). The energy distance kernel induces an embedding well-defined on distributions with finite first moment. The energy distance D_{ED} (Székely and Rizzo, 2013; Rizzo and Székely, 2016):

$$\begin{aligned} D^2_{\mathrm{ED}}(\mu_1(\cdot),\mu_2(\cdot)) =& 2\int_{\mathcal{X}\times\mathcal{X}} \left\| x_1 - x_2 \right\| d\mu_1(x_1) d\mu_2(x_2) - \int_{\mathcal{X}\times\mathcal{X}} \left\| x_1 - x_2 \right\| d\mu_1(x_1) d\mu_1(x_2) \\ & - \int_{\mathcal{X}\times\mathcal{X}} \left\| x_1 - x_2 \right\| d\mu_2(x_1) d\mu_2(x_2) \end{aligned}$$

is proved in Klebanov (2002) to yield a metric, implying that the energy distance kernel induces an injective embedding.

Replacing the integrals in (13) and (14) by the corresponding empirical means gives the kernel embedding and induced inner product for samples of distributions. Given samples $\{\mathbf{x}_i\}_{i=1}^n$ of μ, the kernel embedding for the empirical distribution $\hat{\mu}$ is

$$\hat{\mu}(\cdot) \mapsto g(\hat{\mu}(\cdot)) = \sum_{i=1}^{n} K(\cdot, x_i), \tag{15}$$

and given samples $\{\mathbf{x}_i^{(1)}\}_{i=1}^{n_1}$, $\{\mathbf{x}_i^{(2)}\}_{i=1}^{n_2}$ of μ_1 and μ_2, the induced inner product of the empirical distributions $\hat{\mu_1}$ and $\hat{\mu_2}$ is

$$\langle g(\hat{\mu}_1(\cdot)), g(\hat{\mu}_2(\cdot))\rangle = \sum_{i_1=1}^{n_1}\sum_{i_2=1}^{n_2} K(x_1,x_2). \tag{16}$$

In general, the time complexity of evaluating the inner product is $O(n_1 n_2)$. For the Gaussian kernel, the time complexity for the inner product can be reduced to $O(n_1+n_2)$ via the fast Gauss transform (Greengard and Strain, 1991) or improved fast Gauss transform (Yang et al., 2003). For the energy distance kernel on sorted samples of one-dimensional distributions, the time complexity of evaluating the inner product is $O(n_1+n_2)$, as shown in Appendix A.

We can extend archetypal analysis, prototypal analysis and prototypal regression to distributions with the inner products induced by kernel embedding. In archetypal/prototypal analysis, the archetypes/prototypes are mixtures of the input distributions and their mixtures are used to reconstruct the input distributions. In prototypal regression, we can have distributions as predictors, responses or both. In multiple prototypal regression, we can blend numerical, categorical and distributional predictors.

kernel	$K(x,y)$
Gaussian	$e^{-\sigma\|x-y\|^2}$
Laplacian	$e^{-\sigma\|x-y\|_1}$
B_{2n+1}-spline	$\prod_{i=1}^{d} B_{2n+1}(x_i - y_i)$
energy distance	$\|x\| + \|y\| - \|x-y\|$

Table 2: Some commonly used kernels on $\mathbb{R}^d$ for injective kernel embeddings. For B_{2n+1}-spline, $B_{2n+1}(x) = *_1^{(2n+2)} \mathbf{1}_{[-\frac{1}{2},\frac{1}{2}]}(x)$, where the symbol $*_1^{(2n+2)}$ represents the $(2n+2)$-fold convolution.

5.3 Applications

5.3.1 Smartphone-based Human Activities Recognition Data Set

The smartphone-based human activities recognition data set in Anguita et al. (2013) and Reyes-Ortiz et al. (2016) contains activity data collected by smartphone's inertial sensors. In their experiments, 30 volunteers conducted 6 activities: walking, walking upstairs, walking downstairs, sitting, standing and laying while wearing a wrist-mounted smartphone. The data set contains raw and processed data. The raw data are the triaxial signals from the accelerometer and the gyroscope of smartphones at a constant rate of 50Hz for each activity. The processed data include statistics, such as the mean, standard deviation and auto correlation of the raw signals, and other data, such as the magnitude and the fast Fourier transform of the raw signals.

Anguita et al. (2013) and Reyes-Ortiz et al. (2016) use the processed data to classify the activities. We use the raw data instead, i.e. the triaxial signals from the accelerometer and gyroscope. Each trial in the raw data set contains two three-dimensional time series of the accelerometer and the gyroscope respectively and a label of the activity. We divide the data set into a training data set of 772 trials and a test data set of 84 trials. Multiple prototypal regression is applied for this classification task. The samples of triaxial signals from the accelerometer and the gyroscope are the two predictors in multiple prototypal regression and energy distance kernel is used for kernel embedding. The labels are binarized via one-hot encoding. The number of prototypes is set to be 70 and the penalty coefficient is set to be 1. We achieve a 97.62% accuracy on the testing subset. The confusion matrix for the test data is shown in Table 3, the importance coefficients are listed in Table 4.

5.3.2 EPA Outdoor Air Quality Data Set

The EPA Outdoor Air Quality Data (US Environmental Protection Agency, 2017) collects pollutant and meteorological data at outdoor monitors across the United States, Puerto Rico, and the U. S. Virgin Islands. This data set contains hourly data of criteria gases (Ozone, SO_2, CO and NO_2), toxics and precursors (HAPs, VOCs, NONOxNOy and lead), particulates (PM2.5 FRM/FEM Mass, PM2.5 non FRM/FEM Mass, PM10 Mass and PM2.5 Speciation) and meteorological data (winds, temperature, barometric pressure, relative humidity and dew point).

	walk	upstairs	downstairs	sit	stand	lay
walk	12	0	0	0	0	0
upstairs	1	17	0	0	0	0
downstairs	0	0	18	0	0	0
sit	0	0	0	11	1	0
stand	0	0	0	0	12	0
lay	0	0	0	0	0	12

Table 3: Confusion matrix of multiply prototypal regression on smartphone-based human activities recognition data set. The rows are the actual classes and the columns are the predicted classes.

	accelerometer	gyroscope
importance coefficients	0.44	0.56

Table 4: Importance coefficients of multiply prototypal regression on smartphone-based human activities recognition data set.

We use multiple prototypal regression to estimate the distributions of the nitrogen dioxide (NO_2) density from the geophysical locations (the latitude and longitude of the stations) and the distributions of the meteorological data. The meteorological data that we use are the one-dimensional distribution of wind speed, the one-dimensional distribution of wind direction and one-dimensional distribution of outdoor temperature. The training data set contains the data collected in the year 2016 at 200 stations and the test data set contains the data collected in the same year 2016 at 23 other stations. We use the energy distance kernel for embedding. The number of prototypes is set to 40 and the penalty coefficient to 0.1. The importance coefficients are listed in Table 5 and the out-of-sample predictions are illustrated in Figure 8.

	location	temperature	wind direction	wind speed
importance coefficients	0.23	0.40	0.13	0.24

Table 5: Importance coefficients of multiply prototypal regression on EPA outdoor air quality data set.

6. Conclusions

We have proposed and developed prototypal analysis and regression, two robust extensions of archetypal analysis. In addition, we have shown how these methodologies can be extended via kernel embedding to handle learning problems where the data points are probability distributions known through samples. Here the interpretability associated with the convex

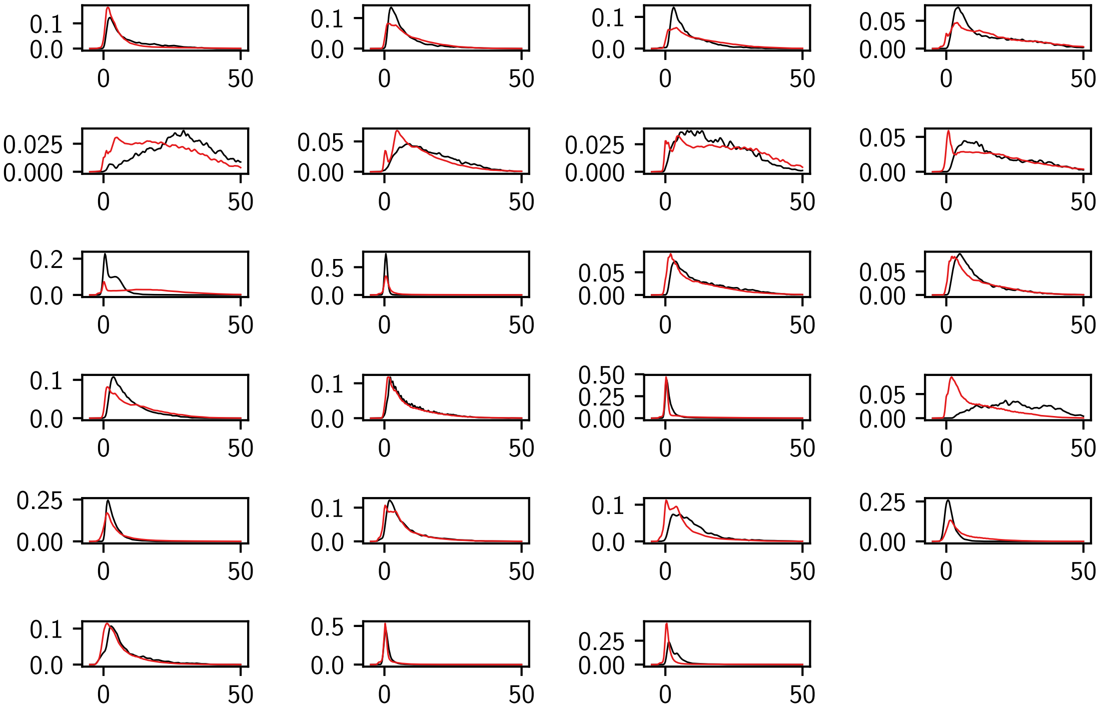

Figure 8: Out-of-sample prediction of NO_2 density distribution. The black curves are the true NO_2 distributions at each station and the red curves are the predicted NO_2 distributions by multiple prototypal regression.

combinations involved is clearest, as these combinations can be interpreted as mixtures of distributions.

Prototypal analysis adds to the objective function of archetypal analysis a term that penalizes the use of distant prototypes for the reconstruction of data points. It can be regarded of as an interpolation between archetypal analysis –corresponding to a zero value of the penalization parameter λ– and k-means, which arises as $\lambda \to \infty$. This adds robustness to outliers and a sense of locality, which becomes particularly useful when the methodology is used for regression.

We illustrate through real-life examples the applicability of the procedure, particularly to scenarios that blend numerical and distributional features or that have probability distributions as labels to predict.

Acknowledgments

The work of E. G. Tabak was partially supported by grants from the Office of Naval Research and the Mathematical Division of the National Science Foundation

Appendix A. Energy Distance Kernel of One-Dimensional Distributions

The energy distance kernel on distributions μ, ν can be estimated using their samples $\{x_i\}_{i=1}^{n_x}, \{y_j\}_{i=1}^{n_y}$ through the empirical mean:

$$k_{\text{ED}}(\mu,\nu) \approx \frac{1}{n_x}\sum_{i=1}^{n_x}\|x_i\| + \frac{1}{n_y}\sum_{j=1}^{n_y}\|y_j\| - \frac{1}{n_x n_y}\sum_{i=1}^{n_x}\sum_{j=1}^{n_y}\|x_i - y_j\|. \tag{17}$$

The time complexity of (17) is $O(n_x n_y)$.

For one-dimensional distributions, the time complexity of (17) can be reduced to the linear $O(n_x + n_y)$ when the samples $\{x_i\}_{i=1}^{n_x}, \{y_j\}_{i=1}^{n_y}$ are sorted, as illustrated in Algorithm 7. The intuition behind is that each term in $\sum_{i=1}^{n_x}\sum_{j=1}^{n_y}\|x_i - y_j\|$ can be expanded into

$$\|x_i - y_j\| = \mathbf{1}_{x_i>y_j}(x_i - y_j) - \mathbf{1}_{x_i\le y_j}(x_i - y_j) = (\mathbf{1}_{x_i>y_j} - \mathbf{1}_{x_i\le y_j})x_i + (\mathbf{1}_{x_i\le y_j} - \mathbf{1}_{x_i>y_j})y_j,$$

yielding

$$\sum_{i=1}^{n_x}\sum_{j=1}^{n_y}\|x_i - y_j\| = \sum_{i=1}^{n_x}\left[\sum_{j=1}^{n_y}(\mathbf{1}_{x_i>y_j} - \mathbf{1}_{x_i\le y_j})\right]x_i + \sum_{j=1}^{n_y}\left[\sum_{i=1}^{n_x}(\mathbf{1}_{x_i\le y_j} - \mathbf{1}_{x_i>y_j})\right]y_j. \tag{18}$$

Equation (18) implies that we only need to count how many y_j's are smaller than each x_i and how many x_i's are smaller than each y_j. If the samples are sorted, this counting can be done in linear time.

References

Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge Luis Reyes-Ortiz. A public domain dataset for human activity recognition using smartphones. In *ESANN*, 2013.

M Asbach, Dirk Mauruschat, and Burkhard Plinke. Understanding multi-spectral images of wood particles with matrix factorization. *Optical Characterization of Materials (OCM) Karlsruhe Institute for Technology*, pages 191–201, 2013.

Christian Bauckhage and Kasra Manshaei. Kernel archetypal analysis for clustering web search frequency time series. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 1544–1549. IEEE, 2014.

Christian Bauckhage and Christian Thurau. Making archetypal analysis practical. In *DAGM-Symposium*, pages 272–281. Springer, 2009.

Alain Berlinet and Christine Thomas-Agnan. *Reproducing kernel Hilbert spaces in probability and statistics*. Kluwer, 2004.

Ben HP Chan, Daniel A Mitchell, and Lawrence E Cram. Archetypal analysis of galaxy spectra. *Monthly Notices of the Royal Astronomical Society*, 338(3):790–795, 2003.

Algorithm 7 Energy Distance Kernel of 1D distributions

```
Input: Sorted samples $\{x_i\}, \{y_j\}$ of 1D distributions $\mu, \nu$.
Output: Empirical estimation of energy distance kernel $k_{\mathrm{ED}}(\mu, \nu)$
$\mathrm{sum}_x \leftarrow 0, \mathrm{sum}_y \leftarrow 0, i \leftarrow 1, j \leftarrow 1$
while $i \leq n_x$ and $j \leq n_y$ do
    if $x_i \leq y_j$ then
        $\mathrm{sum}_x \leftarrow \mathrm{sum}_x + \{(j-1) - [n_y - (j-1)]\}x_i$
        $i \leftarrow i + 1$
    else
        $\mathrm{sum}_y \leftarrow \mathrm{sum}_y + \{(i-1) - [n_x - (i-1)]\}y_j$
        $j \leftarrow j + 1$
    end if
end while
if $i > n_x$ then
    $\mathrm{sum}_y \leftarrow \mathrm{sum}_y + n_x \sum_{k=j}^{n_y} y_k$
else
    $\mathrm{sum}_x \leftarrow \mathrm{sum}_x + n_y \sum_{k=i}^{n_x} x_k$
end if
$k_{\mathrm{ED}}(\mu, \nu) \leftarrow \left(\sum_{k=1}^{n_x} x_k\right)/n_x + \left(\sum_{k=1}^{n_y} y_k\right)/n_y - (\mathrm{sum}_x + \mathrm{sum}_y)/(n_x n_y)$
return $k_{\mathrm{ED}}(\mu, \nu)$
```

Shahzad Cheema, Abdalrahman Eweiwi, Christian Thurau, and Christian Bauckhage. Action recognition by learning discriminative key poses. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1302–1309. IEEE, 2011.

Yuansi Chen, Julien Mairal, and Zaid Harchaoui. Fast and robust archetypal analysis for representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1478–1485, 2014.

Adele Cutler and Leo Breiman. Archetypal analysis. *Technometrics*, 36(4):338–347, 1994.

Anders Drachen, Rafet Sifa, Christian Bauckhage, and Christian Thurau. Guns, swords and data: Clustering of player behavior in computer games in the wild. In *Computational Intelligence and Games (CIG), 2012 IEEE Conference on*, pages 163–170. IEEE, 2012.

Anders Drachen, James Green, Chester Gray, Elie Harik, Patty Lu, Rafet Sifa, and Diego Klabjan. Guns and guardians: Comparative cluster analysis and behavioral profiling in destiny. In *Computational Intelligence and Games (CIG), 2016 IEEE Conference on*, pages 1–8. IEEE, 2016.

Maria Rosaria DEsposito, Francesco Palumbo, and Giancarlo Ragozini. Archetypal analysis for interval data in marketing research. *Ital. J. Appl. Stat*, 18:343–358, 2006.

Manuel JA Eugster. Performance profiles based on archetypal athletes. *International Journal of Performance Analysis in Sport*, 12(1):166–187, 2012.

Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of human genetics*, 7(2):179–188, 1936.

Leslie Greengard and John Strain. The fast gauss transform. *SIAM Journal on Scientific and Statistical Computing*, 12(1):79–94, 1991.

Arthur Gretton, Karsten M Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex J Smola. A kernel method for the two-sample-problem. In *Advances in neural information processing systems*, pages 513–520, 2006.

Thomas Hofmann, Bernhard Schölkopf, and Alexander J Smola. Kernel methods in machine learning. *The annals of statistics*, pages 1171–1220, 2008.

Peter Huggins, Lior Pachter, and Bernd Sturmfels. Toward the human genotope. *Bulletin of mathematical biology*, 69(8):2723–2735, 2007.

Lev B Klebanov. A class of probability metrics and its statistical applications. In *Statistical Data Analysis Based on the L1-Norm and Related Methods*, pages 241–252. Springer, 2002.

Shan Li, PZ Wang, JJ Louviere, and Richard Carson. Archetypal analysis: A new way to segment markets based on extreme individuals. In *Australian and New Zealand Marketing Academy Conference*. ANZMAC, 2003.

S Marinetti, L Finesso, and E Marsilio. Matrix factorization methods: Application to thermal ndt/e. *NDT & E International*, 39(8):611–616, 2006.

Morten Mørup and Lars Kai Hansen. Archetypal analysis for machine learning and data mining. *Neurocomputing*, 80:54–63, 2012.

Krikamol Muandet, Kenji Fukumizu, Francesco Dinuzzo, and Bernhard Schölkopf. Learning from distributions via support measure machines. In *Advances in neural information processing systems*, pages 10–18, 2012.

Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, Bernhard Schölkopf, et al. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends® in Machine Learning*, 10(1-2):1–141, 2017.

Junier B Oliva, Barnabás Póczos, and Jeff G Schneider. Distribution to distribution regression. In *ICML (3)*, pages 1049–1057, 2013.

Barnabás Póczos, Alessandro Rinaldo, Aarti Singh, and Larry Wasserman. Distribution-free distribution regression. *arXiv preprint arXiv:1302.0082*, 2013.

Giovanni Porzio, Giancarlo Ragozini, and Domenico Vistocco. Archetypal analysis for data driven benchmarking. *Data Analysis, Classification and the Forward Search*, pages 309–318, 2006.

Giovanni C Porzio, Giancarlo Ragozini, and Domenico Vistocco. On the use of archetypes as benchmarks. *Applied Stochastic Models in Business and Industry*, 24(5):419–437, 2008.

Jorge-L Reyes-Ortiz, Luca Oneto, Albert Sama, Xavier Parra, and Davide Anguita. Transition-aware human activity recognition using smartphones. *Neurocomputing*, 171: 754–767, 2016.

Maria L Rizzo and Gábor J Székely. Energy distance. *Wiley Interdisciplinary Reviews: Computational Statistics*, 8(1):27–38, 2016.

Christoph Römer, Mirwaes Wahabzada, Agim Ballvora, Francisco Pinto, Micol Rossini, Cinzia Panigada, Jan Behmann, Jens Léon, Christian Thurau, Christian Bauckhage, et al. Early drought stress detection in cereals: simplex volume maximisation for hyperspectral image analysis. *Functional Plant Biology*, 39(11):878–890, 2012.

Bernhard Schölkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond.* MIT press, 2002.

Christian Seiler and Klaus Wohlrabe. Archetypal scientists. *Journal of Informetrics*, 7(2): 345–356, 2013.

D. Sejdinovic, A. Gretton, B. Sriperumbudur, and K. Fukumizu. Hypothesis testing using pairwise distances and associated kernels. In *Proceedings of the 29th International Conference on Machine Learning*, pages 1111–1118, New York, NY, USA, 2012. Omnipress.

John Shawe-Taylor and Nello Cristianini. *Kernel methods for pattern analysis.* Cambridge university press, 2004.

Rafet Sifa and Christian Bauckhage. Archetypical motion: Supervised game behavior learning with archetypal analysis. In *Computational Intelligence in Games (CIG), 2013 IEEE Conference on*, pages 1–8. IEEE, 2013.

Rafet Sifa, Christian Bauckhage, and Anders Drachen. The playtime principle: Large-scale cross-games interest modeling. In *Computational Intelligence and Games (CIG), 2014 IEEE Conference on*, pages 1–8. IEEE, 2014.

Alex Smola, Arthur Gretton, Le Song, and Bernhard Schölkopf. A hilbert space embedding for distributions. In *International Conference on Algorithmic Learning Theory*, pages 13–31. Springer, 2007.

Bharath K Sriperumbudur, Arthur Gretton, Kenji Fukumizu, Bernhard Schölkopf, and Gert RG Lanckriet. Hilbert space embeddings and metrics on probability measures. *Journal of Machine Learning Research*, 11(Apr):1517–1561, 2010.

Emily Stone. Exploring archetypal dynamics of pattern formation in cellular flames. *Physica D: Nonlinear Phenomena*, 161(3):163–186, 2002.

Emily Stone and Adele Cutler. Archetypal analysis of spatio-temporal dynamics. *Physica D: Nonlinear Phenomena*, 90(3):209–224, 1996.

Zoltán Szabó, Arthur Gretton, Barnabás Póczos, and Bharath Sriperumbudur. Two-stage sampled learning theory on distributions. In *Artificial Intelligence and Statistics*, pages 948–957, 2015.

Zoltán Szabó, Bharath Sriperumbudur, Barnabás Póczos, and Arthur Gretton. Learning theory for distribution regression. *Journal of Machine Learning Research*, 17(152):1–40, 2016.

Gábor J Székely and Maria L Rizzo. Energy statistics: A class of statistics based on distances. *Journal of statistical planning and inference*, 143(8):1249–1272, 2013.

Juliane Charlotte Thøgersen, Morten Mørup, Søren Damkiær, Søren Molin, and Lars Jelsbak. Archetypal analysis of diverse pseudomonas aeruginosa transcriptomes reveals adaptation in cystic fibrosis airways. *BMC bioinformatics*, 14(1):279, 2013.

C Thurau and A Drachen. Introducing archetypal analysis for player classification in games. In *2nd International Workshop on Evaluating Player Experience in Games (epex 2011)*, 2011.

Christian Thurau and Christian Bauckhage. Archetypal images in large photo collections. In *Semantic Computing, 2009. ICSC'09. IEEE International Conference on*, pages 129–136. IEEE, 2009.

US Environmental Protection Agency. Air quality system data mart [internet database]. Available via `https://www.epa.gov/airdata`, Accessed June 23, 2017.

Yuanjun Xiong, Wei Liu, Deli Zhao, and Xiaoou Tang. Face recognition via archetype hull ranking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 585–592, 2013.

Changjiang Yang, Ramani Duraiswami, Nail A Gumerov, and Larry Davis. Improved fast gauss transform and efficient kernel density estimation. In *null*, page 464. IEEE, 2003.

Genping Zhao and Chunhui Zhao. Bilateral filtering abundance features for multilayer unmixing. In *Geoscience and Remote Sensing Symposium (IGARSS), 2016 IEEE International*, pages 6557–6560. IEEE, 2016.

Genping Zhao, Xiuping Jia, and Chunhui Zhao. Multiple endmembers based unmixing using archetypal analysis. In *Geoscience and Remote Sensing Symposium (IGARSS), 2015 IEEE International*, pages 5039–5042. IEEE, 2015.