Knowledge-Based Systems 72 (2014) 37–47

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Feature selection for noisy variation patterns using kernel principal component analysis

Anshuman Sahu [a,*], Daniel W. Apley [b], George C. Runger [a]

[a] School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ 85287, USA
[b] Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL 60208, USA

ARTICLE INFO

Article history:
Received 12 February 2014
Received in revised form 5 August 2014
Accepted 29 August 2014
Available online 16 September 2014

Keywords:
Nonlinear PCA
Kernel feature space
Preimages
Variation patterns
Feature ensembles

ABSTRACT

Kernel Principal Component Analysis (KPCA) is a technique widely used to understand and visualize non-linear variation patterns by inverse mapping the projected data from a high-dimensional feature space back to the original input space. Variation patterns often occur in a small number of relevant features out of the overall set of features that are recorded in the data. It is, therefore, crucial to discern this set of relevant features that define the pattern. Here we propose a feature selection procedure that augments KPCA to obtain importance estimates of the features given the noisy training data. Our feature selection strategy involves projecting the data points onto sparse random vectors for calculating the kernel matrix. We then match pairs of such projections, and determine the preimages of the data with and without a feature, thereby trying to identify the importance of that feature. Thus, preimages' differences within pairs are used to identify the relevant features. An advantage of our method is it can be used with any suitable KPCA algorithm. Moreover, the computations can be parallelized easily leading to significant speedup. We demonstrate our method on several simulated and real data sets, and compare the results to alternative approaches in the literature.

1. Introduction

Advances in signal acquisition and computational processing coupled with cheap storage have resulted in massive multivariate data being collected in today's processes like semiconductor manufacturing, automobile-body assemblies, inspection systems, etc. The data can be in form of spatial profiles, time series or images where the measurements are recorded over several features. These features are affected by different sources of variation which result in variation patterns in the data. The goal, therefore, is to identify these sources of variation based on the process data collected. Moreover, the variation pattern may be present in only a small subset of the process variables that are collected. Finding this relevant subset of features is, therefore, critical to understand the process, and is the focus of our work presented in this paper.

Principal Component Analysis (PCA) is a common technique to identify variation pattern in data by projecting along the directions of maximum variability in the data. However, PCA can only identify linear relationships among features in the data. Kernel Principal Component Analysis (KPCA) extends PCA to the case where data contain non-linear patterns as shown by Schölkopf et al. [1]. KPCA identifies non-linear patterns in data by mapping the data from input space to a high-dimensional (possibly infinite) feature space, and performing PCA in the feature space. This is achieved by employing the kernel trick [2]. Thus, only calculations in terms of dot products in the input space are required, without an explicit mapping to the feature space. KPCA is widely used for nonlinear process monitoring [3–5], fault detection and diagnosis [6–9], and anomaly detection [10,11].

To visualize the variation pattern in input space, an inverse transform is used to map the denoised data from feature space back to the input space. The exact preimage of a denoised point in feature space might not exist, so that a number of algorithms for estimating approximate preimages have been proposed [12–15]. Also, [16,17] considered meta-methods to improve the preimage results by averaging from ensembles.

Our task now is to identify the relevant subset of the original set of features over which the pattern exists (a feature selection task). The difficulty is to handle the non-linear relationships between features in input space. Because the feature space in KPCA already provides an avenue to consider higher-order interactions between features, it is more appealing to apply a feature selection procedure

* Corresponding author.
E-mail addresses: anshuman.sahu@asu.edu (A. Sahu), apley@northwestern.edu (D.W. Apley), george.runger@asu.edu (G.C. Runger).

http://dx.doi.org/10.1016/j.knosys.2014.08.027
0950-7051/

in feature space itself. However, it is not always possible to obtain the feature representation in feature space (for example, in the case of a Gaussian kernel) because the data are not explicitly mapped. Therefore, the challenge here is to perform feature selection in the feature space.

Some work has considered feature selection in feature space for supervised learning. A weighted feature approach was provided by Allen [18] where weights are assigned to features while computing the kernel. This feature weighting is incorporated into the loss function corresponding to classification or regression problem and a lasso penalty is put on the weights. The features corresponding to non-zero weights obtained after minimizing the objective (loss function with penalty) are considered the important ones. Similarly, recent work [19,20] also employed feature weighting for the cases of Support Vector Machine (SVM) classification and regression, respectively. For both the cases, an anisotropic Gaussian kernel was used to supply weights to features. Specifically, Maldonado et al. [19] provided an iterative algorithm for solving the feature selection problem by embedding the feature weighting in the dual formulation of SVM problem. The algorithm begins with an initial set of weights. At each iteration, it solves the SVM problem for the given set of feature weights, updates the weights using the gradient of the objective function, and removes the features that are below a certain given threshold. This procedure is repeated till convergence. Finally, the features obtained with non-zero weights are considered important.

Since KPCA is unsupervised, we next consider feature selection in feature space for unsupervised learning. One common aspect of all these algorithms, similar to their counterparts in supervised setting, is they involve some kind of feature weighting mechanism, and the relevant features are obtained by regularizing (shrinking) the weights of irrelevant features using some criteria. A method for feature selection in Local Learning-Based Clustering [21] was proposed by Zeng and ming Cheung [22]. The feature selection is achieved by regularizing the weights assigned to features. A method to measure variable importance in KPCA was suggested by Muniz et al. [23]. They computed the kernel between two data points as weighted sum of individual kernels where each individual kernel is computed on a single feature of each of the two data points, and the weights assigned to each kernel serve as a measure of importance of the feature involved in computing the kernel. They formulated a loss function where a lasso penalty was imposed on the weights to determine the non-zero weights (and the corresponding relevant features). In addition to feature selection in feature space for unsupervised learning, there exist several other feature selection procedures for unsupervised learning that operate in the input space. Laplacian Score (LS) was proposed by He et al. [24] for each feature to estimate its ability to preserve local structure. The authors construct a nearest neighbor graph, and identify the important features as those which maintain this graph structure. Multi-Cluster Feature Selection (MCFS) proposed by Cai et al. [25] used spectral analysis to select the features that preserve the multi-cluster structure of the data. The authors compute the nearest neighbors graph, define weights on edges in the graph, construct the graph Laplacian, and solve the generalized eigen-problem [26] to obtain the top K eigenvectors. For each eigenvector, the contribution of each feature is found by solving a L1-regularized regression. Each feature now has K contribution values, and the maximum of it is assigned as the MCFS score of the feature. The features with higher MCFS scores are important. Unsupervised Discriminative Feature Selection (UDFS) proposed by Yang et al. [27] aims to select the most discriminative features which preserve the local structure of the data (via manifold) while simultaneously accounting for feature correlation. The authors assume the existence of a linear classifier that classifies each data point to a class. They propose learning the classifier that maximizes their local discriminative score. To this end, they propose a regularized optimization problem by inducing $\ell_{2,1}$ norm on the coefficients of the classifier. Note that each coefficient of the linear classifier corresponds to a feature in the dataset. They also propose an iterative algorithm to solve this optimization problem. The top features are determined based on sorting the ℓ_2 norm of the coefficient vectors over all iterations in descending order.

The approaches provided in the literature focus on the case when noise-free training data are available. However, this is not the case in areas like manufacturing variation analysis. In practice, the data are corrupted with noise and has a lot of irrelevant features. Thus, we work with a noisy data set from which we need to find the relevant subset of the features over which the patterns in the data exist. To this end, we propose our novel approach.

As pointed out previously, an innovative way to do feature selection in high-dimensional feature space is to assign weights to features in input space. By using such an approach, we can compute the kernel using all the features instead of iteratively computing it using a subset of features at a time. The goal next is to identify the weights (by some regularization criterion) so that the non-zero weights correspond to the relevant features. We propose an alternative approach for this feature weighting mechanism. Instead of trying to determine the feature weights through a regularization approach, we multiply the features by sparse random vectors whose entries are independent and identically distributed drawn from a distribution (such as Gaussian). After projecting data points onto random subsets of features, we measure feature importance from differences in preimages, where preimages are computed with and without a feature. Therefore, more important features are expected to result in greater differences. The process is repeated iteratively with different sparse random vectors and the differences are averaged to estimate the final feature importance. Our approach above provides robustness to irrelevant features in the data by being able to project only on a small random subset of features at a time, and calculating the final mapped data matrix in input space from an ensemble of feature subsets. Another advantage of our approach is it works with any KPCA preimage algorithm.

We organize the remaining part of our paper as follows. Section 2 provides a brief description of different methods used to visualize the variation patterns in KPCA. For our feature selection method, we can consider any one of them as the base algorithm. Section 3 presents a mathematical description of our methodology. Section 4 shows the results of implementing our algorithm on several simulated and real datasets. Finally Section 5 provides conclusions.

2. Background on preimages in KPCA

KPCA is equivalent to PCA in feature space [1]. Let $\mathbf{X}$ denote the data set with N instances and F features where the instances are denoted by $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$. Similar to PCA, we want to find the eigenvalues and eigenvectors of the covariance matrix $\mathbf{C}$ in feature space. If the corresponding set of points mapped in the feature space $\varphi(\mathbf{x}_i),\ i = 1, 2, \ldots, N$ are assumed to be centered, $\mathbf{C}$ can be calculated by

$$\mathbf{C} = \frac{1}{N}\sum_{i=1}^{N}\varphi(\mathbf{x}_i)\varphi(\mathbf{x}_i)' \tag{1}$$

The eigenvalues λ and eigenvectors $\mathbf{v}$ of matrix $\mathbf{C}$ are given by

$$\mathbf{C}\mathbf{v} = \lambda\mathbf{v} \tag{2}$$

It can be shown that an eigenvector corresponding to non-zero eigenvalue of $\mathbf{C}$ can be written as a linear combination of $\varphi(\mathbf{x}_1), \ldots, \varphi(\mathbf{x}_N)$. Using this simplification reduces the original