An Approximate Version of Kernel PCA

Shawn Martin (smartin@sandia.gov)
Sandia National Laboratories, Albuquerque, NM

Abstract

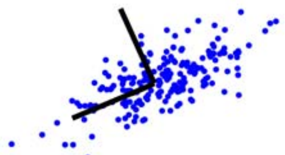
We propose an analog of kernel Principal Component Analysis (kernel PCA). Our algorithm is based on an approximation of PCA which uses Gram-Schmidt orthonormalization. We combine this approximation with Support Vector Machine kernels to obtain a nonlinear generalization of PCA. By using our approximation to PCA we are able to provide a more easily computed (in the case of many data points) and readily interpretable version of kernel PCA. After demonstrating our algorithm on some examples, we explore its use in applications to fluid flow and microarray data.

Principal Component Analysis (PCA)

Principal Component Analysis (PCA) can be described as a procedure for successively capturing the maximal variance in a dataset. If $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ is a dataset of rank m then the PCA basis can be computed as

$$\begin{aligned}
\mathbf{u}_1 &= \arg\max_{\mathbf{u}} \textstyle\sum_{i=1}^{n} (\mathbf{u}, \mathbf{x}_i)^2 \\
\mathbf{u}_2 &= \arg\max_{\mathbf{u}} \textstyle\sum_{i=1}^{n} (\mathbf{u}, \mathbf{x}_i - (\mathbf{x}_i, \mathbf{u}_1)\mathbf{u}_1)^2 \\
&\vdots \\
\mathbf{u}_m &= \arg\max_{\mathbf{u}} \textstyle\sum_{i=1}^{n} (\mathbf{u}, \mathbf{x}_i - \sum_{j=1}^{m-1} (\mathbf{x}_i, \mathbf{u}_j)\mathbf{u}_j)^2.
\end{aligned}$$

where $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m\}$ are also required to be orthonormal, and $(\mathbf{x}, \mathbf{y})$ represents the inner product between $\mathbf{x}$ and $\mathbf{y}$.



The black lines show the PCA basis vectors.

Gram-Schmidt Orthonormalization

Gram-Schmidt orthonormalization is a procedure for transforming a set of linearly independent vectors $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$ into an orthonormal basis $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m\}$. This basis is constructed iteratively by projection, where

$$\begin{gathered}
\mathbf{u}_1 = \tfrac{\mathbf{x}_1}{\|\mathbf{x}_1\|},\ \mathbf{p}_1 = (\mathbf{x}_2, \mathbf{u}_1)\mathbf{u}_1 \\
\mathbf{u}_2 = \tfrac{\mathbf{x}_2 - \mathbf{p}_1}{\|\mathbf{x}_2 - \mathbf{p}_1\|},\ \mathbf{p}_2 = (\mathbf{x}_3, \mathbf{u}_2)\mathbf{u}_2 + (\mathbf{x}_3, \mathbf{u}_1)\mathbf{u}_1 \\
\vdots \\
\mathbf{u}_m = \tfrac{\mathbf{x}_m - \mathbf{p}_{m-1}}{\|\mathbf{x}_m - \mathbf{p}_{m-1}\|}.
\end{gathered}$$

Gram-Schmidt can also be re-written in terms of inner products only, as

$$\begin{aligned}
(\mathbf{x}_1, \mathbf{u}_1) &= \|\mathbf{x}_1\| = \tfrac{(\mathbf{x}_1, \mathbf{x}_1)}{(\mathbf{x}_1, \mathbf{u}_1)} \\
(\mathbf{x}_2, \mathbf{u}_1) &= \tfrac{(\mathbf{x}_2, \mathbf{x}_1)}{\|\mathbf{x}_1\|} = \tfrac{(\mathbf{x}_2, \mathbf{x}_1)}{(\mathbf{x}_1, \mathbf{u}_1)} \\
(\mathbf{x}_2, \mathbf{u}_2)^2 &= \|\mathbf{x}_2 - \mathbf{p}_1\|^2 = \|\mathbf{x}_2\|^2 - \|\mathbf{p}_1\|^2 \\
(\mathbf{x}_2, \mathbf{u}_2) &= \tfrac{1}{(\mathbf{x}_2, \mathbf{u}_2)} \left[(\mathbf{x}_2, \mathbf{x}_2) - (\mathbf{x}_2, \mathbf{u}_1)^2\right] \\
(\mathbf{x}_3, \mathbf{u}_1) &= \tfrac{(\mathbf{x}_3, \mathbf{x}_1)}{(\mathbf{x}_1, \mathbf{u}_1)} \\
(\mathbf{x}_3, \mathbf{u}_2) &= \tfrac{1}{(\mathbf{x}_2, \mathbf{u}_2)} \left[(\mathbf{x}_3, \mathbf{x}_2) - (\mathbf{x}_2, \mathbf{u}_1)(\mathbf{x}_3, \mathbf{u}_1)\right]
\end{aligned}$$

$$\vdots$$

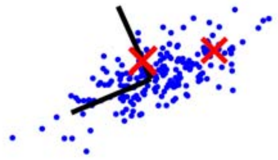
$$(\mathbf{x}_i, \mathbf{u}_j) = \frac{1}{(\mathbf{x}_j, \mathbf{u}_j)} \left[(\mathbf{x}_i, \mathbf{x}_j) - \sum_{k=1}^{j-1} (\mathbf{x}_j, \mathbf{u}_k)(\mathbf{x}_i, \mathbf{u}_k)\right].$$

An Approximate Version of PCA (APCA)

We can use the inner product version of Gram-Schmidt in combination with PCA to select a linearly independent subset $\{\mathbf{x}_{i1}, \mathbf{x}_{i2}, \ldots, \mathbf{x}_{im}\}$ of $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ with properties similar to those of PCA

$$\begin{aligned}
\mathbf{x}_{i_1} &= \arg\max_{\mathbf{x}_i} \tfrac{1}{\|\mathbf{x}_i\|^2} \textstyle\sum_{j=1}^{n} (\mathbf{x}_i, \mathbf{x}_j)^2 \\
\mathbf{x}_{i_2} &= \arg\max_{\mathbf{x}_i} \\
&\quad \tfrac{1}{\|\mathbf{x}_i\|^2} \textstyle\sum_{j=1}^{n} \left[(\mathbf{x}_i, \mathbf{x}_j) - (\mathbf{x}_i, \mathbf{u}_1)(\mathbf{x}_j, \mathbf{u}_1)\right]^2 \\
&\vdots \\
\mathbf{x}_{i_m} &= \arg\max_{\mathbf{x}_i} \\
&\quad \tfrac{1}{\|\mathbf{x}_i\|^2} \sum_{j=1}^{n} \left[(\mathbf{x}_i, \mathbf{x}_j) - \sum_{l=1}^{m-1} (\mathbf{x}_i, \mathbf{u}_l)(\mathbf{x}_j, \mathbf{u}_l)\right]^2.
\end{aligned}$$

The vectors $\{\mathbf{x}_{i1}, \mathbf{x}_{i2}, \ldots, \mathbf{x}_{im}\}$ give the Approximate PCA (APCA) basis vectors.



The black lines show the PCA basis vectors and the red X's show the APCA basis vectors (actual data points).

The primary advantage of APCA over PCA is that the APCA basis vectors are easier to interpret than the PCA basis vectors, due to the fact that the APCA basis vectors are in fact instances of the original dataset.

Support Vector Machine (SVM) Kernels

A Support Vector Machine (SVM) kernel is a function $k : \circ^{d} \times \circ^{d} \to \circ$ with an associated map $\Phi : \circ^{d} \to F$ such that

$$k(\mathbf{x}, \mathbf{y}) = (\Phi(\mathbf{x}), \Phi(\mathbf{y})),$$

where F is an inner product space. Kernels act to introduce nonlinear interactions in the original space $\circ^{d}$ while maintaining linear relations in the higher dimensional space F.

Examples of kernel functions include

$$\begin{aligned}
&k(\mathbf{x}, \mathbf{y}) = (\mathbf{x}, \mathbf{y}) && \text{(linear)} \\
&k(\mathbf{x}, \mathbf{y}) = ((\mathbf{x}, \mathbf{y}) + a)^b, b \in \square && \text{(polynomial)} \\
&k(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\| / 2\sigma^2), \sigma \neq 0 && \text{(Gaussian).}
\end{aligned}$$

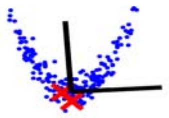
SVM kernel functions can be used to make any linear algorithm expressed in terms of inner products into a non-linear algorithm.

An Approximate Version of Kernel PCA (AKPCA)

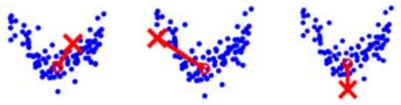
We replace the inner products in APCA with kernel functions to obtain a nonlinear approximate version of PCA, which we call Approximate kernel PCA, or AKPCA.

The primary advantage of AKPCA over APCA is that we can now examine datasets with nonlinear relationships. Since AKPCA is based on APCA, we inherit the advantage of interpretability. AKPCA basis vectors are still vectors that have occurred in the original dataset.

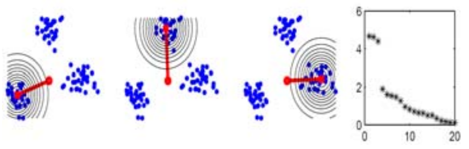
Examples



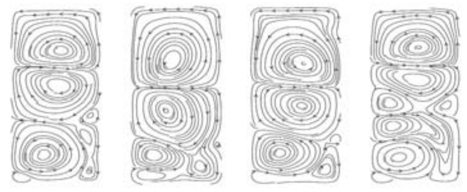
PCA and APCA on a parabolic dataset (PCA basis vectors are black line, APCA basis vectors are red X's). Neither are able to explain the parabola.



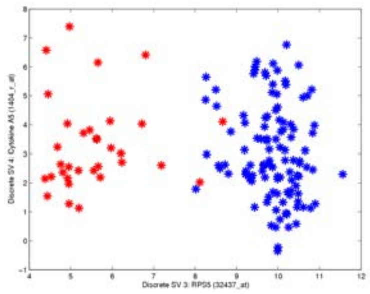
AKPCA is able to decipher parabolic nature of dataset. The first two basis vectors give arms of parabola, and the third basis vector gives noise (in the *y* direction).



AKPCA is able to identify three clusters using an RBF kernel. From left to right the first three basis vectors give the cluster centers and the plot on the far right shows the relative importance of all basis vectors identified (only the first three are important).



Taylor-Couette flow occurs in fluid trapped between two independently rotated cylinders. APCA was used to determine the most frequently occurring flow patterns. The 1st to 4th basis vectors are shown from left to right.



APCA was used with a Leukemia microarray dataset to identify a group of anomalous patients, shown here in red, as well as the gene most relevant to distinguishing the cluster (*x*-axis, or RPS5).

Conclusions

AKPCA provides nonlinear capabilities for data analysis and is easier to interpret than standard PCA and kernel PCA. It is also easier to use on very large datasets because it is iterative and can be made parallel.

Sandia is a multiprogram laboratory operated by Sandia Corporation, a LockheedMartin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.