Data Visualization via Kernel Machines

- Introduction
- Kernel Machines in the Framework of an RKHS
- Kernel Principal Component Analysis
- Computation of KPCA
- Kernel Canonical Correlation Analysis
- Kernel Cluster Analysis

Introduction – The Challenge of Modern Data

The Need for New Analytic Tools

- **Massive and Complex Datasets:** The rapid evolution of information technology has resulted in the accumulation of enormous amounts of data from diverse sources. This influx has created a significant demand for innovative analytic tools capable of handling complex datasets that traditional statistical methods cannot effectively tackle.
- **Obstacles in Visualization:** High dimensionality remains a persistent obstacle to successful data visualization. Exploring hidden structures in complicated datasets is challenging, as standard parametric models are often inadequate for these complexities.
- **Limitations of Traditional Methods:** Traditional nonparametric methods can become unstable or computationally expensive due to the **curse of dimensionality**. Developing scalable and robust nonparametric approaches is therefore essential for modern data science.

Introduction – The Power of Kernel Methods

Leveraging Machine Learning Success

- **Flexible Nonlinear Analysis:** Kernel methods enable flexible and versatile nonlinear analysis by operating in very high-dimensional — often infinite-dimensional — **Reproducing Kernel Hilbert Spaces (RKHS)**.
- **Rich Mathematical Structure:** The RKHS framework provides strong mathematical foundations, including geometric and probabilistic interpretations, enabling meaningful statistical inference for complex data distributions.
- **Computational Efficiency:** Kernel machines are well-suited for massive computation. Unlike classical approaches that struggle with scale, kernel methods are designed to efficiently handle large datasets in visualization and learning tasks.

Introduction – The Kernel Transformation Process

Mapping to Hilbert Space

- **From Euclidean to Feature Space:** Traditional statistical procedures operate directly in Euclidean space $\mathbb{R}^p$. Kernel methods first map data into a high-dimensional Hilbert space using a kernel function $K(x_i, x_j)$.
- **Applying Classical Procedures:** After transformation, classical statistical methods are applied to the kernel-transformed representation, combining traditional statistical strength with high-dimensional flexibility.
- **New Metrics for Similarity:** Kernel transformations define new notions of similarity and distance between data points:

$$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$$

This allows hidden nonlinear relationships to be uncovered in feature space.

Introduction – Nonlinear Dimension Reduction

Visualizing Complicated Structures

- **Preparing Data for Analysis:** Once raw data is represented in kernel form, standard statistical and computational tools can be applied to explore nonlinear data structures.
- **Kernel Principal Component Analysis (KPCA):** KPCA performs nonlinear dimension reduction by projecting data in feature space onto principal components. It reveals structures that linear PCA cannot detect.
- **Overcoming Computational Difficulties:** Kernel-based approaches allow visualization of highly nonlinear structures in massive datasets without constructing explicit complex models, thereby reducing computational burden.

Kernel Machines in the Framework of an RKHS

Basic Definitions – Foundations of the Framework

- **Sample Space and Positive Definite Kernels:** Let $X \subset \mathbb{R}^p$ be the sample space. A real symmetric function

$$\kappa : X \times X \to \mathbb{R}$$

is called **positive definite** if for any $n \in \mathbb{N}$, any $x_1, \dots, x_n \in X$, and any real numbers $c_1, \dots, c_n$,

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j \kappa(x_i, x_j) \geq 0.$$

- **Defining the RKHS:** A Reproducing Kernel Hilbert Space (RKHS) is a Hilbert space $\mathcal{H}_\kappa$ of real-valued functions on X where all evaluation functionals are bounded linear functionals. Convergence in $\| \cdot \|_{\mathcal{H}_\kappa}$ implies pointwise convergence.
- **Uniqueness:** For every positive definite kernel κ, there exists a unique RKHS denoted $\mathcal{H}_\kappa$. This one-to-one relationship forms the mathematical backbone of kernel machines.

Kernel Machines in the Framework of an RKHS

The Reproducing Property – The Core Mechanism

- **The Reproducing Property:** The defining characteristic of an RKHS is

$$\langle f(\cdot), \kappa(x, \cdot) \rangle_{\mathcal{H}_\kappa} = f(x), \quad \forall f \in \mathcal{H}_\kappa, \; x \in X.$$

- **Correspondence:** There exists a one-to-one correspondence between an RKHS and its kernel. Conversely, every RKHS has a unique positive definite kernel satisfying the reproducing property.
- **Discrete Spectrum:** Under suitable integrability conditions, a positive definite kernel admits a countable spectral decomposition:

$$\kappa(x, u) = \sum_{m=1}^{\infty} \lambda_m \psi_m(x) \psi_m(u),$$

where λ_m are eigenvalues and ψ_m are eigenfunctions.

Kernel Machines in the Framework of an RKHS

Feature Maps – Embedding Data into Feature Space

- **Mapping Strategy:** Data in Euclidean space X is mapped into a high (possibly infinite) dimensional Hilbert space to enable powerful statistical analysis.
- **Two Isomorphic Feature Maps:**
  1. **Spectrum-based Map:**
  $$\Phi(x) = \left(\sqrt{\lambda_1}\psi_1(x), \sqrt{\lambda_2}\psi_2(x), \ldots\right)$$
  2. **Aronszajn Map:**
  $$\gamma(x) = \kappa(x, \cdot)$$
  which directly embeds x as a function in $\mathscr{H}_{\kappa}$.
- **Equivalence:** These two representations are isometrically isomorphic. In practice, the Aronszajn map is preferred for KPCA and KCCA visualization.

Kernel Machines in the Framework of an RKHS

The Kernel Trick – Computing Without Explicit Mapping

- **Inner Products in Feature Space:** If $Z = \Phi(X)$, then

  $$\Phi(x) \cdot \Phi(u) = \kappa(x, u).$$

  This identity is known as the **kernel trick**.
- **Implicit Computation:** Linear algorithms can operate in the high-dimensional feature space without explicitly computing $\Phi(x)$. Only kernel evaluations $\kappa(x_i, x_j)$ are required.
- **Nonlinear Variants:** Replacing dot products with kernel evaluations transforms linear algorithms into nonlinear methods in the original space.

Kernel Machines in the Framework of an RKHS

Kernelization – Hybrid Statistical Models

- **Parallel Solutions:** Classical statistical procedures (e.g., PCA, CCA) can be implemented in $\mathscr{H}_\kappa$ using the kernel representation.
- **Nonparametric Nature:** From the original space perspective, kernelization acts as a nonparametric approach using kernel mixtures, while maintaining computational advantages similar to parametric models.
- **Implementation:** Kernel algorithms are essentially classical linear algorithms applied to the kernel feature space. Practically, only the kernel matrix

  $$K_{ij} = \kappa(x_i, x_j)$$

  needs to be constructed before running standard software routines.

Kernel Machines in the Framework of an RKHS

Reduced Kernel Method – Handling Large Datasets

- **Computational Costs:** Constructing the full $n \times n$ kernel matrix becomes computationally expensive for large datasets due to high CPU and memory requirements. Certain algorithms (e.g., SVMs) may have cubic complexity:

$$\mathcal{O}(n^3).$$

- **The Reduced Kernel Solution:** Instead of using the full kernel matrix, a small subset of columns is randomly selected to construct a thin rectangular matrix. This approach uses partial kernel bases while retaining all data points for fitting.
- **Efficiency and Accuracy:** The reduced kernel method significantly lowers computational load and memory usage while maintaining strong predictive performance. It also regularizes model complexity by limiting the number of basis functions.

Kernel Principal Component Analysis (KPCA)

Concept – Extending PCA to Nonlinear Data

- **Classical PCA Limitations:** PCA seeks linear subspaces that maximize variance. However, it can only detect linear relationships in the data.
- **The KPCA Innovation:** KPCA performs PCA in the high-dimensional feature space $Z = \Phi(X)$, enabling detection of nonlinear structures and higher-order correlations.
- **Implicit Transformation:** The mapping $\Phi(x)$ is not computed explicitly. All computations are performed using kernel evaluations:

$$\Phi(x_i) \cdot \Phi(x_j) = \kappa(x_i, x_j).$$

Thus, nonlinear PCA is achieved via the kernel trick.

Computation of KPCA

Classical PCA Review – The Linear Foundation

- **Maximizing Variance:** Find a unit vector w that maximizes projected variance:

$$\max_{w} \; w^\top \Sigma w, \quad \text{subject to } w^\top w = 1.$$

- **Lagrangian Formulation:**

$$\mathscr{L}(w, \alpha) = w^\top \Sigma w - \alpha (w^\top w - 1).$$

  The solution satisfies:

$$\Sigma w = \alpha w,$$

  i.e., w is an eigenvector of Σ.

- **Sequential Extraction:** Subsequent components are eigenvectors orthogonal to previous ones, corresponding to descending eigenvalues.

Computation of KPCA

Kernel Covariance – Moving to the RKHS

- **Covariance Operator:** Let $\gamma_1, \ldots, \gamma_n \in \mathscr{H}_\kappa$ be mapped data points. The sample covariance operator is:

  $$C_n = \frac{1}{n}\sum_{j=1}^{n}(\gamma_j - \bar{\gamma}) \otimes (\gamma_j - \bar{\gamma}).$$

- **Eigencomponent Search:** Find $h \in \mathscr{H}_\kappa$ that maximizes:

  $$\langle h, C_n h \rangle, \quad \text{subject to } \|h\|_{\mathscr{H}_\kappa} = 1.$$

- **Solution Form:** The optimal solution has the representation:

  $$h = \sum_{j=1}^{n} \beta_j \gamma_j,$$

  reducing the infinite-dimensional problem to finite coefficients β_j.

Computation of KPCA

Optimization Problem – Kernel Eigenvalue Equation

- **Reformulation:** The optimization can be expressed using the kernel matrix K:

  $$K_{ij} = \kappa(x_i, x_j).$$

  We maximize a quadratic form in β involving K^2.

- **Generalized Eigenvalue Problem:** After centering, the optimization leads to:

  $$\left(K - \frac{1}{n}\mathbf{1}\mathbf{1}^\top K\right)^2 \beta = n\alpha K\beta.$$

- **Finding Components:** Eigenvectors β corresponding to the largest eigenvalues define kernel principal components, enabling nonlinear dimension reduction.

Computation of KPCA

Projections – Visualizing the Results

- **Projecting Data:** For a data point x with feature image $\gamma(x)$, the projection onto the k-th eigencomponent h_k is:

$$\langle \gamma(x), h_k \rangle = \sum_{j=1}^{n} \beta_{kj} \kappa(x_j, x).$$

- **Dimension Reduction:** Projection onto the subspace spanned by the leading r eigencomponents produces a vector in $\mathbb{R}^r$:

$$(\langle \gamma(x), h_1 \rangle, \dots, \langle \gamma(x), h_r \rangle).$$

  These become the nonlinear coordinates of x.

- **Data Visualization:** Plotting the leading kernel principal components allows visualization of complex nonlinear structures hidden in the original space.

KPCA Example – ”Two Moons” Dataset

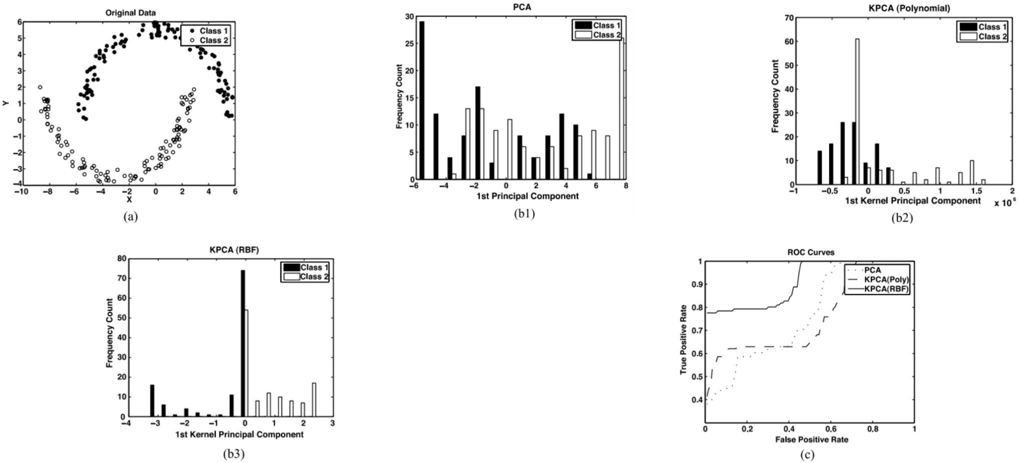
Linear vs. Nonlinear Separation

- **The Dataset:** The synthetic ”two moons” dataset contains two interlocking classes in 2D space. They are visually separable but not linearly separable.
- **PCA Failure:** Classical PCA cannot separate the classes effectively. The first principal component shows strong overlap between groups.
- **KPCA Success:** Using a Gaussian (RBF) kernel:

$$\kappa(x, u) = \exp\left(-\frac{\|x - u\|^2}{2\sigma^2}\right),$$

KPCA achieves clear separation. ROC AUC improves from 0.77 (PCA) to 0.91 (KPCA-RBF).

KPCA Example – ”Two Moons” Dataset



(a)

(b1)

(b2)

(b3)

(c)

Figure: Comparison of PCA and KPCA (Polynomial, RBF) on the two-moons dataset with ROC performance

KPCA Example – Pima Diabetes & Image Segmentation

Real-World Applications

- **Pima Diabetes Dataset:** KPCA with Gaussian kernels reveals nonlinear structures not detected by PCA. Different scale parameters σ provide multiple nonlinear perspectives.
- **Image Segmentation Dataset:** For classes such as ”brickface” and ”path,” KPCA (RBF kernel) produces clearer separation compared to linear PCA.
- **Value of Nonlinearity:** KPCA extracts nonlinear information with only modest additional computational effort beyond PCA.

KPCA Example – Pima Diabetes & Image Segmentation

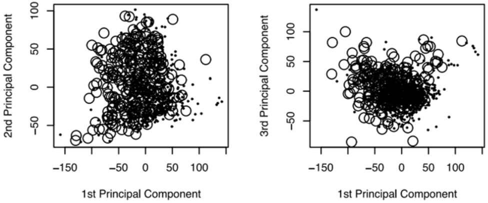


Figure 1: Results from PCA based on original input variables

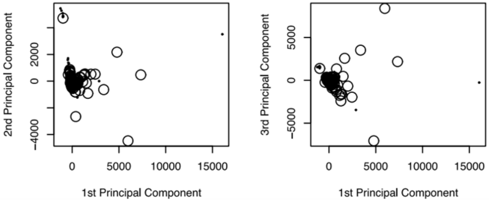
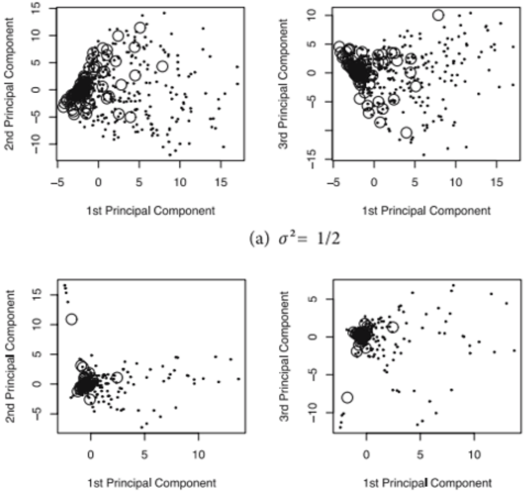


Figure 2: Results from KPCA with the polynomial kernel of degree 3 and scale 1

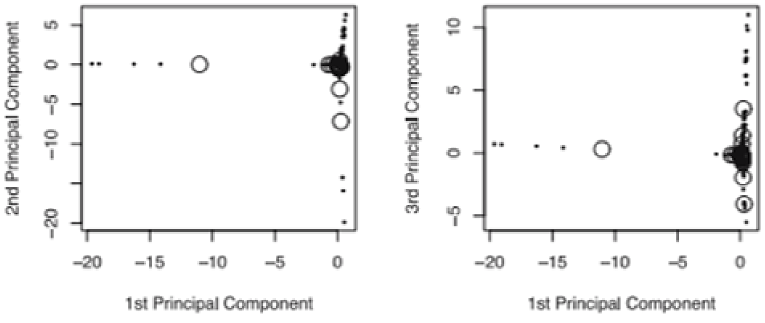
KPCA Example – Pima Diabetes & Image Segmentation



(a) $\sigma^2 = 1/2$

(b) $\sigma^2 = 1/6$

KPCA Example – Pima Diabetes & Image Segmentation



(c) $\sigma^2 = 1/10$

Figure: Results from KPCA with Gaussian kernels

Kernel Canonical Correlation Analysis (KCCA)

Overview – Analyzing Relations Between Variable Sets

- **Canonical Correlation Analysis (CCA):** CCA studies relationships between two variable sets:

  $$X^{(1)} \quad \text{and} \quad X^{(2)}.$$

  It finds linear transformations maximizing cross-correlation.
- **Linear Limitation:** Classical CCA can only capture linear relationships between the two sets.
- **The Hybrid Approach (KCCA):** KCCA integrates CCA with kernel methods, enabling discovery of nonlinear relations between variable groups.

KCCA – Methodology

Implementing Kernel Canonical Correlation Analysis

- **Kernel Transformation:** Apply two kernels:

  $$\kappa_1 \text{ for } X^{(1)}, \quad \kappa_2 \text{ for } X^{(2)}.$$

  This yields kernel matrices K_1 and K_2, forming:

  $$K = [K_1 \; K_2].$$

- **Applying Classical CCA:** CCA is performed on the kernel-transformed data, leveraging standard eigenvalue-based algorithms.
- **Regularization:** Regularization is required to solve the spectral problem. The reduced kernel method effectively controls model complexity for large datasets.

KCCA – Example: Handwritten Digits

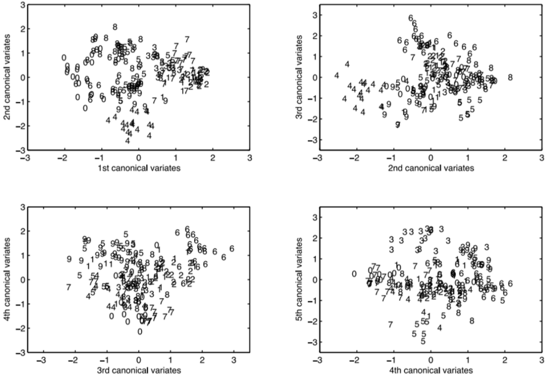
Nonlinear Discriminant Analysis

- **Experimental Setup:** KCCA was applied to the ”pen-based recognition of handwritten digits” dataset. Input measurements (16 dimensions) formed $X^{(1)}$, while digit labels (0–9) formed $X^{(2)}$.
- **Kernel Setup:** Gaussian kernel for input data:

$$\kappa(x, u) = \exp\left(-\frac{\|x - u\|^2}{2\sigma^2}\right).$$

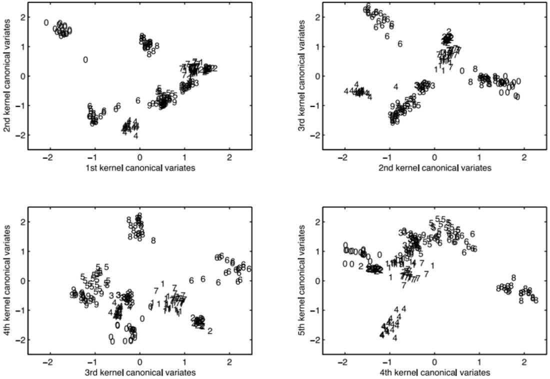
  Dummy variables (linear kernel) for labels. Reduced kernel size = 300.
- **Results:** CCA-derived variates showed weak separation. KCCA-derived variates produced clear clustering of digit groups in projected space.

KCCA – Example: Handwritten Digits



Scatter plots of pen digits over CCA-derived variates

KCCA – Example: Handwritten Digits



Scatter plots of pen digits over KCCA-derived variates

Kernel Cluster Analysis

Introduction – Unsupervised Learning with Kernels

- **Cluster Analysis Basics:** Cluster analysis groups similar unlabeled data points. Traditional *k*-means:
  - Requires predefined *k*
  - Sensitive to initialization
- **Support Vector Clustering (SVC):** Uses kernel methods to create flexible cluster boundaries via nonlinear mapping into feature space.
- **Geometry of SVC:** Find the smallest enclosing sphere in feature space instead of centroid-based partitioning.

Kernel Cluster Analysis

SVC Mechanism – The Optimization Problem

- **Enclosing Sphere:** Minimize radius R such that all mapped points satisfy:

$$\|\Phi(x_i) - a\|^2 \leq R^2.$$

- **Dual Formulation:** Eliminating primal variables yields a dual optimization problem in coefficients β.
- **Cluster Boundaries:** The enclosing sphere in feature space maps to probability contours in the original space, allowing complex and non-convex cluster shapes.

Kernel Cluster Analysis

Support Vectors & Parameters

- **Support Vectors (SVs):** Points lying on the boundary of the enclosing sphere.
- **Bounded Support Vectors (BSVs):** Points outside the boundary (treated as outliers).
- **Soft Margin:** Slack variables allow outliers:

  $$\min R^2 + C \sum \xi_i$$

  Parameter C controls sphere size vs. tolerance.
- **Role of Kernel Width (q):** Gaussian kernel width controls resolution. Varying q produces hierarchical clustering structures.

Kernel Cluster Analysis

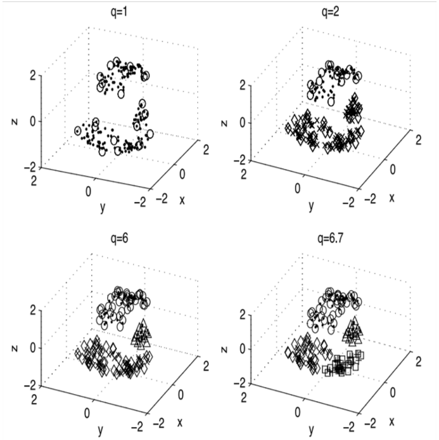
Cluster Assignment – From Contours to Groups

- **Adjacency Matrix:** Define

$$A_{jj'} = \begin{cases} 1 & \text{if } j \text{ and } j' \text{ belong to same cluster} \\ 0 & \text{otherwise} \end{cases}$$

- **Geometric Check:** Check whether the line segment between two mapped points stays within the enclosing sphere.
- **Graph Components:** Clusters are defined as connected components of the graph induced by A. This method is robust and independent of cluster shape.

Kernel Cluster Analysis



Scatter plots of pen digits over KCCA-derived variates

Conclusion

Summary of Kernel Machine Visualization

- **A Unified Framework:** Kernel machines extend classical linear methods (PCA, CCA, clustering) to nonlinear high-dimensional settings via RKHS mapping.
- **Computational Feasibility:** The reduced kernel method enables practical implementation on massive datasets.
- **Enhanced Visualization:** Examples such as ”two moons” and handwritten digits demonstrate that kernelized methods reveal structures invisible to linear techniques.