Sparse Kernel Feature Analysis

Alex J. Smola
Alex.Smola@anu.edu.au
Department of Engineering and RSISE, ANU, Canberra
Olvi Mangasarian
olvi@cs.wisc.edu
Department of Computer Science, U Wisconsin, Madison
Bernhard Schölkopf
bsc@microsoft.com
Microsoft Research, Cambridge

THE AUSTRALIAN NATIONAL UNIVERSITY

Outline

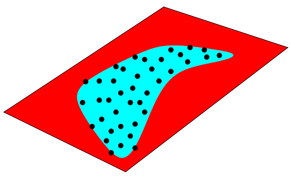
THE AUSTRALIAN NATIONAL UNIVERSITY

- **Feature Extraction**
  Properties and Applications
- **Nonlinear Methods**
  Feature Spaces
  Kernels and Regularization
- **Implementation**
  Rockafellar's Theorem, Maximum Search
  Divide and Conquer, Probabilistic Speedup
- **Experiments**

Methods in Unsupervised Learning

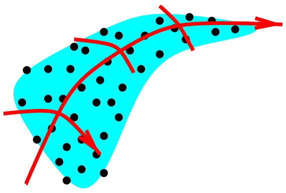
THE AUSTRALIAN NATIONAL UNIVERSITY

Data Description



Kohonen Map
Generative Topographical Map
Principal Curves
Regularized Principal Manifolds
Clustering

Feature Extraction



Principal Component Analysis
Projection Pursuit
Independent Component Analysis
Kernel-PCA
Kernel Feature Analysis

Feature Extractors

THE AUSTRALIAN NATIONAL UNIVERSITY

Data: Space $\mathfrak{X}$ of patterns with probability distribution $P(x)$ on $\mathfrak{X}$, in practice m-sample $\{x_1, \ldots, x_m\}$.

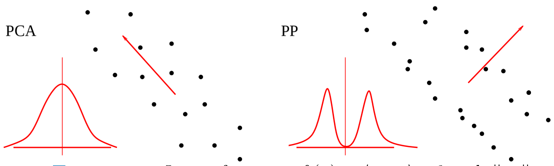
Aim: Find **feature extractors** $f_i : \mathfrak{X} \to \mathbb{R}$ such that

- $f_i(x)$ contains the **relevant information** about x
- the features $f_i(x)$ are **independent** or **orthogonal**
- there exists a **sufficient number** of features to use them **directly** in a linear classifier / regressor ...
- the feature extractors f_i are **simple** yet **meaningful**.

Applications: Classification, Compact Representation of Function Classes, Dimensionality Reduction (e.g. PCA), Data Mining (e.g. Projection Pursuit), Density Modelling, ...

PCA and Projection Pursuit

THE AUSTRALIAN NATIONAL UNIVERSITY



Feature Extractor: Linear functions $f_i(x) = \langle w_i, x\rangle + b$ with $\|w_i\| \le 1$.

Contrast Function $Q[f(x_1), \ldots, f(x_m)]$:

Principal Component Analysis: Maximize Variance of $f_i(x)$.

Projection Pursuit: Maximize Curtosis, Bimodality, ..., of $f_i(x)$.

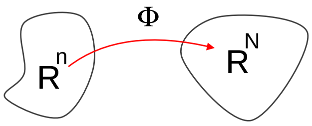
Optimization Problem:

$$\begin{aligned} &\text{maximize} \quad Q[f(x_1), \ldots, f(x_m)] \\ &\text{subject to} \quad f(x) = \langle w, x\rangle + b \text{ with } \|w\| \le 1 \end{aligned}$$

Nonlinear PCA and PP

THE AUSTRALIAN NATIONAL UNIVERSITY

Idea: Use a map Φ into some highdimensional **feature space** F and (hope that you can) extract more information via PCA there.



Example:

$$\Phi : (x_1, x_2) \to (x_1^2, \sqrt{2}x_1x_2, x_2^2)$$

Consequence: **Replace x by** $\Phi(x)$ wherever x appears and keep the rest

Optimization Problem:

$$\begin{aligned} &\text{maximize} \quad Q[f(x_1), \ldots, f(x_m)] \\ &\text{subject to} \quad f(x) = \langle w, \Phi(x)\rangle + b \text{ with } \|w\| \le 1 \end{aligned}$$

Kernels

THE AUSTRALIAN NATIONAL UNIVERSITY

Problem: F may be **very** highdimensional (10^{10} or much more). Hence we can not compute Φ explicitly.

Solution: (Aizerman, Braverman, Rozonoer, 1964, Boser, Guyon, Vapnik, 1992)

$$k(x_i, x_j) := (\Phi(x_i) \cdot \Phi(x_j))$$

Hence we do not have to compute $\Phi(x)$ explicitly.

Examples:

Polynomial Kernels

$$k(x, x') = (\langle x, x'\rangle + c)^d \text{ with } c \ge 0$$

Gaussian Kernels

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

Nonlinear Feature Extractors

THE AUSTRALIAN NATIONAL UNIVERSITY

Question: What does it **mean** to have a feature extractor of type

$$f(x) = \langle w, \Phi(x)\rangle \text{ with } \|w\|^2 \le 1$$

Function Spaces: Regularization Theory tells us that minimizing $\|w\|^2$ in Feature Space is equivalent to minimizing $\|Pf\|^2$ in Function Space.

$$f(x) \text{ with } \|Pf\|^2 = \|f\|^2 + \sum_{i=1}^{d} \|\partial_{x_i} f\|^2 \le c$$

Bayesian Interpretation: Using the connection between Regularization and Priors $p(f)$ over functions this means

$$f(x) \text{ with } p(f) \ge c : \text{ e.g. we could use a Gaussian Process prior}$$

General Rule:

$\boxed{\text{Simple functions with interesting properties on the data.}}$

Kernel PCA Implementation

Covariance Matrix: Assuming centering in feature space we obtain

$$C := \frac{1}{m}\sum_{i=1}^{m}\Phi(x_i)\Phi^\top(x_i)$$

Principal Components: We know that the principal components of C have to satisfy $v \in \operatorname{span}\{\Phi(x_1), \ldots \Phi(x_m)\}$, hence $v = \sum_{i=1}^m \alpha_i \Phi(x_i)$.

Solution: α can be found as solution of the eigenvalue problem

$$\frac{1}{m}K K\alpha = \lambda K\alpha \text{ where } K_{ij} := \langle\Phi(x_i), \Phi(x_j)\rangle = k(x_i, x_j).$$

Feature Extractor:

$$f_i(x) = \sum_{j=1}^{m}\alpha_{ij}k(x_j, x) + \mu_i \quad (\alpha_{i1}, \ldots \alpha_{im}) \text{ is Eigenvector of } K.$$

Problem: The kernel expansion is **dense** and $O(m^2)$ to compute.

(Sparse) Kernel Feature Analysis

Idea: Replace the $\|w\| \leq 1$ constraint by a **sparsity** constraint.

Optimization Problem:

$$\begin{aligned} &\text{maximize} \quad Q[f(x_1), \ldots, f(x_m)] \\ &\text{subject to} \quad f(x) = \langle w, \Phi(x)\rangle + b \text{ with } \Omega[f] \leq 1 \end{aligned}$$

Goal 1: **Sparse** and **simple** functions (Kernel-PCA needs m kernel functions no matter how many features computed).

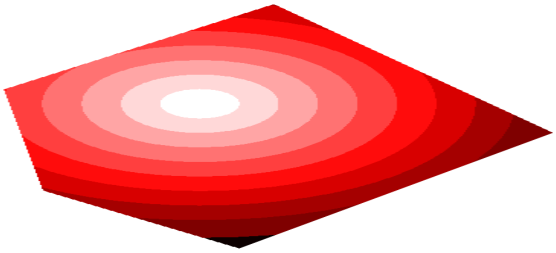
Goal 2: **Exactly Solvable Problem** (would be nice)

Goal 3: **Fast and Easy Algorithm**, i.e. $O(m)$ instead of $O(m^2)$ per feature as in Kernel-PCA

Goal 4: **Iterative Algorithm**, don't have to specify the number of features beforehand

Rockafellar's Theorem

Rockafellar, 1970: The problem of globally **maximizing a convex function** bounded above on a **polyhedral set** not containing straight lines that go to infinity in both directions has **a vertex solution**.



Visiting the Vertices

Idea: Use (distorted) ℓ_1 balls to restrict the set of functions, i.e.

$$f \in \left\{\langle w, \Phi(x)\rangle \,\middle|\, w = \sum_i \alpha_i\Phi(x_i) \text{ and } \sum_i |\alpha_i| \leq 1\right\}$$

Hence the **vertices** are the functions $\pm k(x_i, x)$ of which there are m.

Idea: Use any **convex contrast function** Q, e.g. Variance, Curtosis, etc.

Trick: The vertices of ℓ_1 balls are the patterns $\Phi(x_i)$.

Algorithm: We only have to perform a maximum search over all the basis functions $k(x_i, x)$, or rather the corresponding values $Q[k(x_i, x_1), \ldots k(x_i, x_m)]$ to solve the problem (Smola, Mangasarian, Schölkopf, 1999).

Iteration Strategies

Problem: Maximum search finds first feature $\bar{\imath}$ but what about the next?

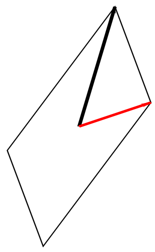
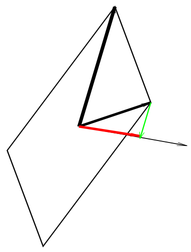
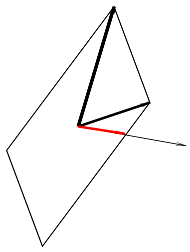
(Dumb) Idea : **Remove** $k(x_{\bar{\imath}}, \cdot)$ from the list of functions and **repeat**: boring features

(Dumb) Idea : **Cut** the ℓ_1 ball with the plane orthogonal to $k(x_{\bar{\imath}}, \cdot)$: computationally expensive, not sparse

Idea: **Project** the ℓ_1 ball on the plane orthogonal to $k(x_{\bar{\imath}}, \cdot)$: as cheap as first feature since $k(x_i, \cdot) \to k(x_i, \cdot) + \eta_i k(x_{\bar{\imath}}, \cdot)$.

Advantage: We only need **one more kernel function per new feature** since at stage i each vertex j is a linear combination of $k(x_j, \cdot)$ and the previous $i - 1$ (already used) kernel functions.

Iteration Strategies

Removal	Projection	Projection and Rescale
		
Problem: Similar data	**Advantage:** Fast and upper bound	**Advantage:** Fast and lower bound

A Probabilistic Enhancement

Problem: Maximum search is still too expensive: $O(m^2)$ since we have m possible functions and each one is $O(m)$ to compute.

Idea 1: **Approximate** max by considering only a **random subset** of n functions: this yields on average an $n/(n+1)$ quantile.

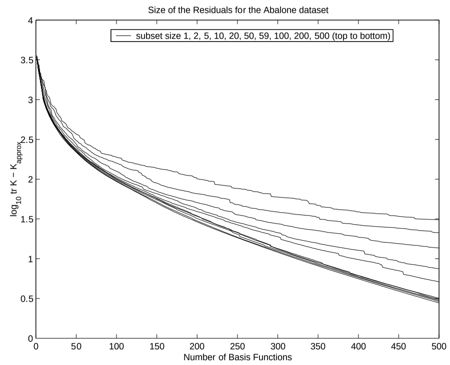
Result : This leads to an $O(nm)$ algorithm per feature.

Idea 2: Use a **divide and conquer** approach to optimize over **solutions of small subproblems** in a tree fashion.

Tool: **Hoeffding's bound** tells how many basis functions of the subproblem to consider: $\Pr\{S_m - E[S_m] \geq \varepsilon\} \leq e^{-\frac{m^2\varepsilon^2}{2\sum_{i=1}^m a^2}}$ where a bounds each r.v.

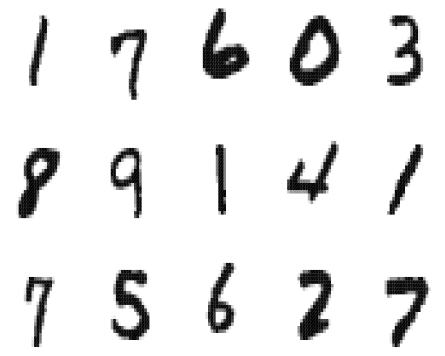
Result: This leads to an $O(m \log m)$ algorithm per feature on average.

Random Subset Selection



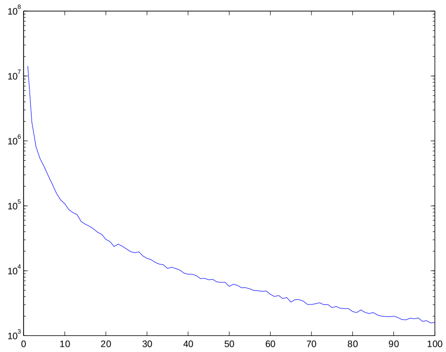
Handwritten Digits (NIST)

THE AUSTRALIAN NATIONAL UNIVERSITY



Remaining Variance (NIST)

THE AUSTRALIAN NATIONAL UNIVERSITY



Summary

THE AUSTRALIAN NATIONAL UNIVERSITY

- Feature Extraction approach to Unsupervised Learning
- Principal Component Analysis in Feature Space
- Kernel Feature Analysis (Contrast Function + Regularizer)
- Efficient Algorithm
- Modification available for Sparse Greedy Matrix Approximation

  For more information see `http://www.kernel-machines.org`

- Handout available at
  `http://www.kernel-machines.org/papers/gfkl_kfa_handout.ps.gz`