EPAT 2010

Kernel Methods

Algorithms

Marco Cuturi
Outline of the lectures
Outline

• Mathematical considerations (≤ 80′s)
  ○ Reproducing Kernel Hilbert Spaces
  ○ positive-definiteness, negative definiteness etc..
  ○ kernels, similarities and distances

• Defining kernels
  ○ Standard kernels (≤ 80′s)
  ○ Statistical modeling & kernels (> 1998)
  ○ Algebraic structures and kernels

• Kernel algorithms
  ○ supervised learning, SVM (≥ 1995)
  ○ representer theorem
  ○ unsupervised techniques, eigenfunctions of samples (≥ 1998)
  ○ density estimation and novelty detection (≥ 1999)
Kernel algorithms

algorithms which select functions with desirable properties in a RKHS
algorithms which only take as inputs Gram matrices $K$
Regression, Classification and other Supervised Tasks

- Two associated random variables
  - A random variable $x$, taking values in $\mathcal{X}$,
  - A random variable $y$, taking values in $\mathcal{Y}$.

- Two samples of $(x, y)$ i.i.d. distributed from their joint law
  - $\{(x_1, y_1), \cdots, (x_n, y_n)\}$, $n$ couples of $\mathcal{X} \times \mathcal{Y}$.

  Challenge: predict $y$ when given only $x$.

- In practice, find a function $\mathcal{X} \rightarrow \mathcal{Y}$ for which $f(x)$ is not too different from $y$ on average.
Binary Classification

• \( \mathcal{Y} = -1, 1 \).

• \( f \) needs to be a function that, given \( x \) predicts a label,
  \[
  f : \mathcal{X} \mapsto \{0, 1\}
  \]
  of course, many possible choices for \( f \)'s shape.

• We review here linear hyperplanes in \( \mathcal{X} = \mathbb{R}^d \) first.

• We represent it in \( \mathbb{R}^2 \) for simplicity.

Next slides will cover an important algorithm, the SVM algorithm

• this algorithm can be naturally expressed in terms of kernels. We review later other algorithms for which this is also the case.

  \textit{thanks to Jean-Philippe Vert for many of the following figures and slides.}
Linear classifier, some degrees of freedom
Linear classifier, some degrees of freedom
Linear classifier, some degrees of freedom
Linear classifier, some degrees of freedom
Linear classifier, some degrees of freedom
Linear classifier, some degrees of freedom
Linear classifier, some degrees of freedom
Which one is better?
A criterion to select a linear classifier: the margin
A criterion to select a linear classifier: the margin
A criterion to select a linear classifier: the margin
A criterion to select a linear classifier: the margin
A criterion to select a linear classifier: the margin
Largest Margin Linear Classifier
Support Vectors with Large Margin
In equations

- The **training set** is a finite set of \( n \) data/class pairs:

\[
\mathcal{T} = \{(x_1, y_1), \ldots, (x_N, y_N)\},
\]

where \( x_i \in \mathbb{R}^d \) and \( y_i \in \{-1, 1\} \).

- We assume (for the moment) that the data are **linearly separable**, i.e., that there exists \((w, b) \in \mathbb{R}^d \times \mathbb{R}\) such that:

\[
\begin{cases}
  w^T x_i + b > 0 & \text{if } y_i = 1, \\
  w^T x_i + b < 0 & \text{if } y_i = -1.
\end{cases}
\]
How to find the largest separating hyperplane?

For the linear classifier $f(x) = w^T x + b$ consider the *interstice* defined by the hyperplanes

- $f(x) = w^T x + b = +1$
- $f(x) = w^T x + b = -1$
The margin is $2/\|\mathbf{w}\|$.

- Indeed, the points $\mathbf{x}_1$ and $\mathbf{x}_2$ satisfy:

  \[
  \begin{align*}
  \mathbf{w}^T \mathbf{x}_1 + b &= 0, \\
  \mathbf{w}^T \mathbf{x}_2 + b &= 1.
  \end{align*}
  \]

- By subtracting we get $\mathbf{w}^T (\mathbf{x}_2 - \mathbf{x}_1) = 1$, and therefore:

  \[
  \gamma = 2 \|\mathbf{x}_2 - \mathbf{x}_1\| = \frac{2}{\|\mathbf{w}\|}.
  \]

  where $\gamma$ is the margin.
All training points should be on the appropriate side

- For positive examples ($y_i = 1$) this means:

$$w^T x_i + b \geq 1$$

- For negative examples ($y_i = -1$) this means:

$$w^T x_i + b \leq -1$$

- in both cases:

$$\forall i = 1, \ldots, n, \quad y_i (w^T x_i + b) \geq 1$$
Finding the optimal hyperplane

• Find \((w, b)\) which minimize:

\[ ||w||^2 \]

under the constraints:

\[ \forall i = 1, \ldots, n, \quad y_i (w^T x_i + b) - 1 \geq 0. \]

This is a classical quadratic program on \(\mathbb{R}^{d+1}\)

**linear constraints** - **quadratic objective**
In order to minimize:
\[
\frac{1}{2} \| w \|^2
\]
under the constraints:
\[
\forall i = 1, \ldots, n, \quad y_i (w^T x_i + b) - 1 \geq 0.
\]

introduce one dual variable \( \alpha_i \) for each constraint,

namely, for each training point. The Lagrangian is, for \( \alpha \geq 0 \),
\[
L(w, b, \alpha) = \frac{1}{2} \| w \|^2 - \sum_{i=1}^{n} \alpha_i (y_i (w^T x_i + b) - 1).
\]
The Lagrange dual function

\[ g(\alpha) = \inf_{w \in \mathbb{R}^d, b \in \mathbb{R}} \left\{ \frac{1}{2} \|w\|^2 - \sum_{i=1}^{n} \alpha_i \left( y_i (w^T x_i + b) - 1 \right) \right\} \]

is only defined when

\[ w = \sum_{i=1}^{n} \alpha_i y_i x_i, \quad \text{(derivating w.r.t w)} \quad (*) \]

\[ 0 = \sum_{i=1}^{n} \alpha_i y_i, \quad \text{(derivating w.r.t b)} \quad (**) \]

substituting (*) in \( g \), and using (**) as a constraint, we get the dual function \( g(\alpha) \).

- To compute the dual, just maximize \( g \) w.r.t. \( \alpha \).

- Strong duality holds. KKT gives us \( \alpha_i (y_i w^T x_i - 1) = 0 \), either \( \alpha_i = 0 \) or \( y_i w^T x_i = 1 \).

- \( \alpha_i \neq 0 \) only for points on the support hyperplanes \( \{(x, y) \mid y w^T x_i = 1\} \).
The dual problem is thus

maximize \( g(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \)

such that \( \alpha \succeq 0, \sum_{i=1}^{n} \alpha_i y_i = 0. \)

This is a quadratic program on \( \mathbb{R}^n \), with box constraints. \( \alpha^* \) can be found efficiently using dedicated optimization softwares.
Recovering the optimal hyperplane

- Once $\alpha^*$ is found, we recover $(w^T, b^*)$ corresponding to the optimal hyperplane.
- $w^T$ is given by $w^T = \sum_{i=1}^{n} \alpha_i x_i^T$,
- $b^*$ is given by the conditions on the support vectors $\alpha_i > 0$, $y_i (w^T x_i + b) = 1$,
  \[ b^* = -\frac{1}{2} \left( \min_{y_i=1, \alpha_i > 0} (w^T x_i) + \max_{y_i=-1, \alpha_i > 0} (w^T x_i) \right) \]
- the **decision function** is therefore:
  \[ f^*(x) = w^T x + b^* = \sum_{i=1}^{n} \alpha_i x_i^T x + b^*. \]
- Here the **dual** solution gives us directly the **primal** solution.
Interpretation: support vectors

\[ \alpha = 0 \]

\[ \alpha > 0 \]
What happens when the data is not linearly separable?
What happens when the data is not linearly separable?
What happens when the data is not linearly separable?
What happens when the data is not linearly separable?
Soft-margin SVM

- Find a trade-off between **large margin** and **few errors**.

- Mathematically:
  \[
  \min_f \left\{ \frac{1}{\text{margin}(f)} + C \times \text{errors}(f) \right\}
  \]

- \(C\) is a parameter
Soft-margin SVM formulation

- The **margin** of a labeled point \((x, y)\) is

\[
\text{margin}(x, y) = y \left( w^T x + b \right)
\]

- The **error** is
  - 0 if \(\text{margin}(x, y) > 1\),
  - \(1 - \text{margin}(x, y)\) otherwise.

- The soft margin SVM solves:

\[
\min_{w, b} \left\{ \|w\|^2 + C \sum_{i=1}^{n} \max\{0, 1 - y_i \left( w^T x_i + b \right) \} \right\}
\]

- \(c(u, y) = \max\{0, 1 - yu\}\) is known as the **hinge loss**.

- \(c(w^T x_i + b, y_i)\) associates a mistake cost to the decision \(w, b\) for example \(x_i\).
Dual formulation of soft-margin SVM

• The soft margin SVM program

\[
\min_{w,b} \{ \|w\|^2 + C \sum_{i=1}^{n} \max\{0, 1 - y_i (w^T x_i + b)\} \}
\]

can be rewritten as

\[
\begin{align*}
\text{minimize} & \quad \|w\|^2 + C \sum_{i=1}^{n} \xi_i \\
\text{such that} & \quad y_i (w^T x_i + b) \geq 1 - \xi_i
\end{align*}
\]

• In that case the dual function

\[
g(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j,
\]

which is finite under the constraints:

\[
\begin{cases}
0 \leq \alpha_i \leq C, & \text{for } i = 1, \ldots, n \\
\sum_{i=1}^{n} \alpha_i y_i = 0.
\end{cases}
\]
Interpretation: bounded and unbounded support vectors

\[ \alpha = 0 \]

\[ 0 < \alpha < C \]

\[ \alpha = C \]
Sometimes linear classifiers are not interesting
Let \( \phi(x) = (x_1^2, x_2^2)' \), \( w = (1, 1)' \) and \( b = 1 \). Then the decision function is:

\[
\begin{align*}
f(x) &= x_1^2 + x_2^2 - R^2 = \langle w, \phi(x) \rangle + b,
\end{align*}
\]
Kernel trick for SVM’s

- use a mapping $\phi$ from $\mathcal{X}$ to a feature space,
- which corresponds to the kernel $k$:

$$\forall x, x' \in \mathcal{X}, \quad k(x, x') = \langle \phi(x), \phi(x') \rangle$$

- Example: if $\phi(x) = \phi \left( \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right) = \begin{bmatrix} x_1^2 \\ x_2^2 \end{bmatrix}$, then

$$k(x, x') = \langle \phi(x), \phi(x') \rangle = (x_1)^2(x_1')^2 + (x_2)^2(x_2')^2.$$
Training a SVM in the feature space

Replace each $x^T x'$ in the SVM algorithm by $\langle \phi(x), \phi(x') \rangle = k(x, x')$

- The dual problem is to maximize

$$g(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j k(x_i, x_j),$$

under the constraints:

$$\begin{cases} 0 \leq \alpha_i \leq C, & \text{for } i = 1, \ldots, n \\ \sum_{i=1}^{n} \alpha_i y_i = 0. \end{cases}$$

- The decision function becomes:

$$f(x) = \langle w, \phi(x) \rangle + b^* = \sum_{i=1}^{n} \alpha_i k(x_i, x) + b^*.$$  \hspace{1cm} (1)
The kernel trick

- The explicit computation of $\phi(x)$ is not necessary. The kernel $k(x, x')$ is enough.

- The SVM optimization for $\alpha$ works **implicitly** in the feature space.

- The SVM is a kernel algorithm: only need to input $K$ and $y$:

  \[
  \text{maximize} \quad g(\alpha) = \alpha^T \mathbf{1} - \frac{1}{2} \alpha^T (y^T K y) \alpha \\
  \text{such that} \quad 0 \leq \alpha_i \leq C, \quad \text{for } i = 1, \ldots, n \\
  \sum_{i=1}^{n} \alpha_i y_i = 0.
  \]

- in the end the solution $f(x) = \sum_{i=1}^{n} \alpha_i k(x_i, \cdot) + b$. 

EPAT’10 - M. Cuturi
Kernel example: polynomial kernel

- For $x = (x_1, x_2)^\top \in \mathbb{R}^2$, let $\phi(x) = (x_1^2, \sqrt{2}x_1x_2, x_2^2) \in \mathbb{R}^3$:

$$
K(x, x') = x_1'^2x_1^2 + 2x_1x_2x_1'x_2' + x_2'^2x_2^2 \\
= \{x_1x'_1 + x_2x'_2\}^2 \\
= \{x^T x'\}^2.
$$
Empirical Risk Minimization

- Starting with \( \{(x_1, y_1), \cdots , (x_n, y_n)\} \), \( n \) couples of \( X \times Y \),
- A functional class \( \mathcal{F} \),
- A cost function \( c : Y \times Y, c \geq 0 \), which penalizes discrepancies (distances? squared-distance?)
- find the function which minimizes

\[
\hat{f} \in \arg\min_{f \in \mathcal{F}} \hat{R}(f) = \frac{1}{n} \sum_{i=1}^{n} c(f(x_i), y_i)
\]

and use this \( f \) as a decision function.

- As usual in minimizations, we love:
  - Convex problems, unique minimizers
  - Stable solutions numerically.
Linear least squares

- When $\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y} = \mathbb{R}$,
- $\mathcal{F} = \{x \mapsto \beta^T x + b, \beta \in \mathbb{R}^d, b \in \mathbb{R}\}$, $c(y_1, y_2) = \|y_1 - y_2\|^2$,
- The problem is known as regression with the least squares criterion.
- In this case, the minimizer

$$\arg\min_{f \in \mathcal{F}} \hat{R}(f) = \frac{1}{n} \sum_{i=1}^{n} c(f(x_i), y_i)$$

is unique (assuming $n > d$), and is equal to

$$\begin{bmatrix} b \\ \beta \end{bmatrix} = (XX^T)^{-1} X \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

where $X = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \cdots & \vdots \\ x_1 & x_2 & \cdots & x_n \\ \vdots & \vdots & \cdots & \vdots \end{bmatrix}$
Minimizers on general functional classes

- In this case a few factors contribute to the uniqueness:
  - convexity of $c$,
  - the feasible set, $\mathcal{F}$ is sufficiently small to show no-degeneracy.
- Imagine we use instead a RKHS for $\mathcal{F}$.
- Usually two sources of problems:
  - selecting functions in (infinite dimensional) RKHS can be ill-posed:
    \[ \text{card}\{\arg\min_{f \in \mathcal{H}} \hat{R}(f)\} \text{ could be } \infty \]
  - within these solutions, some are more desirable than others. In particular, better select smoother functions.
Minimizers in RKHS

• Main message: we do not want to deal with problems of optimization in infinite dimensional Hilbert spaces using finite numbers of constraints.

• Two major intuitions:

  Bias the selection towards functions of low norm $\|f\|_H$

  • the norm quantifies the roughness of the function.
  • if possible, better choose a smooth function for a decision function.
Minimizers in RKHS

Bias the selection towards functions we know in $\mathcal{H}$, namely $\mathcal{H}_n$

• When the criterion only depends on the values of $f$ on a sample $\{x_1, \cdots, x_n\} \in \mathcal{X}$, as in $\hat{R}$, under certain conditions,

$$\arg\min \hat{R} \subset \mathcal{H}_n \overset{\text{def}}{=} \text{span}\{k(x_i, \cdot)_{i=1,\ldots,n}\}.$$ 

• As a consequence, $f$ can be selected within the optimum set

$$\arg\min_{f \in \mathcal{H}_n} \hat{R}(f),$$ 

$\mathcal{H}_n$ is a finite dimensional subspace of $\mathcal{H}$. Always easier to handle mathematically.
**Theorem 1.** Let \( \{x_i\}_{1 \leq i \leq n} \) be points in \( \mathcal{X} \) and let \( \Psi : \mathbb{R}^{n+1} \rightarrow \mathbb{R} \) be any function that is strictly increasing with respect to its last argument. Then any solution to the problem

\[
\min_{f \in \mathcal{H}} \Psi(f(x_1), \cdots, f(x_n), \|f\|_{\mathcal{H}_k})
\]

is in \( \mathcal{H}_n \).

**Proof.** Let \( f = f_n + f^\perp \), where \( f_n \in \mathcal{H}_n, f^\perp \in \mathcal{H}^\perp_n \).

- We have that \( f(x_i) = f_n(x_i) \) since

\[
f(x_i) = \langle f, k(x_i, \cdot) \rangle = \langle f, k(x_i, \cdot) \rangle = \langle f_n, k(x_i, \cdot) \rangle + \langle f^\perp, k(x_i, \cdot) \rangle = f_n(x_i).
\]

Hence for any function \( f \in \mathcal{H} \), \( \Psi(f_n) < \Psi(f) \) hence any optimal \( f^* \) must be such that \( f^* \in \mathcal{H}_n \).
Empirical Risk Minimization

- We can now write for a strictly convex loss \( c \),

\[
\hat{f} = \arg \min_{f \in \mathcal{H}_n} \hat{R}_\lambda(f) = \frac{1}{n} \sum_{i=1}^{n} c(f(x_i), y_i) + \lambda \| f \|^2_{\mathcal{H}}
\]

and this \( \hat{f} \) is unique

- \( \lambda > 0 \) balances the tradeoff between
  - a good fit for the data at hand
  - a smoothness as measured by \( \| f \| \).

- This formulation can be generalized to any measure of smoothness \( J \) on \( \mathcal{F} \),

\[
R^\lambda_c(f) \overset{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} c(f(x_i), y_i) + \lambda J(f).
\]
A few examples

- $\mathcal{X}$ is Euclidian, $\mathcal{Y} = \mathbb{R}$, $\mathcal{F} = \mathcal{X}^*$, the dual of $\mathcal{X}$ and $c(f(x), y) = (y - f(x))^2$, minimizing $R^\lambda_c$ is known as
  - least-square regression when $\lambda = 0$;
  - ridge regression when $\lambda > 0$ and $J$ is the Euclidian 2-norm;
  - the lasso when $\lambda > 0$ and $J$ is the 1-norm.

- $\mathcal{X} = [0, 1]$, $\mathcal{Y} = \mathbb{R}$, $\mathcal{F}$ is the space of $m$-times differentiable functions on $[0, 1]$ and $J = \int_{[0, 1]} (f^{(m)}(t))^2 \, dt$, we obtain regression by natural splines of order $m$. 
A few examples

- $\mathcal{X}$ is a set endowed with a kernel $k$ and $\mathcal{Y} = \{-1, 1\}$, $\mathcal{F} = \mathcal{H}$, $J = \| \cdot \|_{\mathcal{H}}$ and
  - the hinge loss $c(f(x), y) = (1 - yf(x))^+$ → SVM
  - $c(f(x), y) = (y - f(x))^2$ → LS-SVM,
  - $c(f(x), y) = \ln(1 + e^{-yf(x)})$ → kernel logistic regression.

- When $\mathcal{X}$ is an arbitrary set endowed with a kernel $k$ and $\mathcal{Y} = \mathbb{R}$, $\mathcal{F} = \mathcal{H}$, $J = \| \cdot \|_{\mathcal{H}}$ and $c(f(x), y) = (|y - f(x)| - \varepsilon)^+$, the $\varepsilon$-insensitive loss function, the solution to this program is known as support vector regression.
Unsupervised Techniques

Principal Component Analysis in $\mathbb{R}^d$.

- Start from a sample $X = \{x_1, \cdots, x_n\}$.
- Look for directions $v_1, \cdots, v_d$ of $\mathbb{R}^d$ such that for $1 \leq j \leq d$,

$$v_j = \arg\max_{v \in \mathbb{R}^d, \|v\|=1, v \perp \{v_1, \cdots, v_{j-1}\}} \text{var}_X[v^T x],$$

- For $f : \mathbb{R}^d \to \mathbb{R}$, $\text{var}_X[f]$ is the empirical variance w.r.t. sample $X$, that is

$$\text{var}_X[f] = E_X(f(x) - E_X[f(x)])^2 = \frac{1}{n} \sum_{i=1}^{n} \left( f(x_i) - \frac{1}{n} \sum_{i=1}^{n} f(x_i) \right)^2.$$

- The $r$ first eigenvectors $v_1, \cdots, v_r$ are the principal components.
Unsupervised Techniques

Canonical Correlation Analysis in $\mathbb{R}^{d,d'}$.

- **Two associated samples** $X$ paired with $Y = \{y_i\}_{1 \leq i \leq n}$ in $\mathbb{R}^{d'}$.
- Assume that the pairs $(x_i, y_i)$ are drawn from a i.i.d law.
- CCA looks for relationships between $X$ and $Y$ by looking for linear projections of the samples $X$ and $Y$,

  $$\alpha^T x_i \text{ and } \beta^T y_j,$$

  such that $\text{corr}(\alpha^T x_i, \beta^T y_i)$ is high.

  $$(\alpha, \beta) = \arg\max_{\xi \in \mathbb{R}^d, \zeta \in \mathbb{R}^{d'}} \text{corr}[\alpha^T, \beta^T]_{X,Y}$$
  $$= \arg\max_{\xi \in \mathbb{R}^d, \zeta \in \mathbb{R}^{d'}} \frac{\text{cov}_{X,Y}[\alpha^T, \beta^T]}{\sqrt{\text{var}_X[\alpha^T] \text{var}_Y[\beta^T]}}$$
where for two real valued functions $f : \mathcal{X} \to \mathbb{R}$ and $g : \mathcal{Y} \to \mathbb{R}$ we write

$$\text{var}_X[f] = E_X(f(x) - E_X[f(x)])^2 = \frac{1}{n} \sum_{i=1}^{n} \left( f(x_i) - \frac{1}{n} \sum_{j=1}^{n} f(x_j) \right)^2,$$

$$\text{var}_Y[g] = E_X(g(y) - E_Y[g(y)])^2 = \frac{1}{n} \sum_{i=1}^{n} \left( g(y_i) - \frac{1}{n} \sum_{j=1}^{n} g(y_j) \right)^2,$$

$$\text{cov}_{X,Y}[f, g] = E_{X,Y}[(f(x) - E_X[f(x)])(g(y) - E_Y[g(y)])]$$

$$= \frac{1}{n} \sum_{i=1}^{n} \left( f(x_i) - \frac{1}{n} \sum_{j=1}^{n} f(x_j) \right) \left( g(y_i) - \frac{1}{n} \sum_{j=1}^{n} g(y_j) \right)$$
Unsupervised Techniques

Both non-convex optimizations look for vectors in \( \mathbb{R}^d \), that is linear projections which summarize the data.

- Although non-convex, the optima can be computed through eigenvalue decompositions of matrices.
- Courant-Weyl-Fisher minimax principle for Rayleigh quotients.

Yet, these tools have limitations: linearity.

Kernel methods allow us to study nonlinear eigenfunctions and CCA-projections.
Consider $X$ as spanning $\mathcal{H}_n$ the two previous optimizations become

$$f_j = \underset{f \in \mathcal{H}_X, \|f\|_{\mathcal{H}_X}=1, f \perp \{f_1, \ldots, f_{j-1}\}}{\text{argmax}} \text{var}_{X}[\langle f, k_X(x, \cdot) \rangle_{\mathcal{H}}],$$

for $1 \leq j \leq n$.

Using the $n \times n$ kernel matrix $K_X$, more precisely its centered counterpart

$$\bar{K}_X = (I_n - \frac{1}{n}1_{n,n})K_X(I_n - \frac{1}{n}1_{n,n}).$$

The eigenfunctions $f_i$ are recovered through the eigenvalue/eigenvector pairs $(e_i, d_i)$ of $\bar{K}_X$,

$$\bar{K}_X = EDE^T$$

where $D = \text{diag}(d)$ and $E$ is an orthogonal matrix. Writing $U = ED^{-1/2}$ we have that

$$f_j(\cdot) = \sum_{i=1}^{n} U_{i,j} k(x_i, \cdot)$$

with $\text{var}_{X}[f_j(x)] = \frac{d_j}{n}$. 
A direct adaptation of the CCA criterion to infinite dimensional RKHS,

\[(f, g) = \arg\max_{f \in \mathcal{H}_X, g \in \mathcal{H}_Y} \frac{\text{cov}_{X,Y} [\langle f, k_X(x, \cdot) \rangle_{\mathcal{H}_X}, \langle g, k_Y(y, \cdot) \rangle_{\mathcal{H}_X}]}{\sqrt{\text{var}_X [\langle f, k_X(x, \cdot) \rangle_{\mathcal{H}_X}] \text{var}_Y [\langle g, k_Y(y, \cdot) \rangle_{\mathcal{H}_Y}]}},\]

This does not work numerically on finite samples. Denominator goes to zero.

In [FBG07], it is shown that using

\[(f, g) = \arg\max_{f \in \mathcal{X}, g \in \mathcal{Y}} \frac{\text{corr}_{X,Y} [f, g]}{\sqrt{(\text{var}_X [f] + \lambda \|f\|^2)(\text{var}_Y [g] + \lambda \|g\|^2)}},\]

and letting \(\lambda \to 0\) as \(n \to \infty\) works.
kernel-Canonical Correlation Analysis [Aka01,BJ02]

- The finite sample estimates $f^n$ and $g^n$ can be recovered as

$$
\begin{align*}
    f^n(\cdot) &= \sum_{i=1}^{n} \xi_i \varphi_i(\cdot), \\
    g^n(\cdot) &= \sum_{i=1}^{n} \zeta_i \psi_i(\cdot)
\end{align*}
$$

where $\xi$ and $\zeta$ are the solutions of

$$
(\xi, \zeta) = \arg\max_{\xi, \zeta \in \mathbb{R}^n} \xi^T \bar{K}_Y \bar{K}_X \xi
$$

$$
\xi^T (\bar{K}_X^2 + n\lambda \bar{K}_X) \xi = \zeta^T (\bar{K}_Y^2 + n\lambda \bar{K}_Y) \zeta = 1
$$

and

$$
\varphi_i(\cdot) = k_X(x_i, \cdot) - \frac{1}{n} \sum_{j=1}^{n} k_X(x_i, \cdot), \quad \psi_i(\cdot) = k_Y(y_i, \cdot) - \frac{1}{n} \sum_{j=1}^{n} k_Y(y_i, \cdot),
$$

are the centered projections of $(x_i)$ and $(y_j)$ in $\mathcal{H}_X$ and $\mathcal{H}_Y$