Political Methodology III: Model Based Inference

Justin Grimmer

Professor
Department of Political Science
Stanford University

May 29th, 2019

Support Vector Machines

Observation i is an $J \times 1$ vector

$$\boldsymbol{x}_i = (x_{1i}, x_{2i}, \ldots, x_{Ji})$$

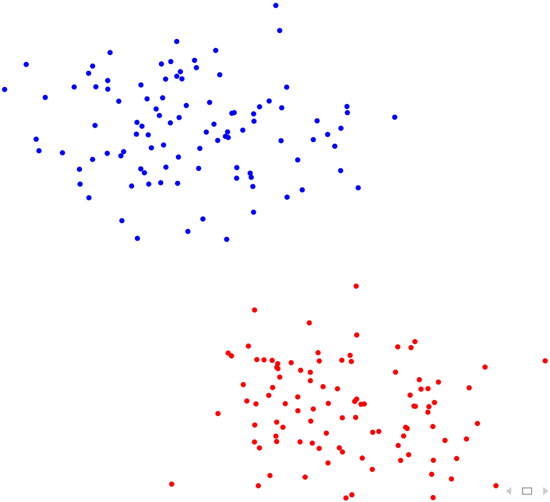
Suppose we have two classes, C_1, C_2.

$$\begin{aligned} Y_i &= 1 \text{ if } i \text{ is in class 1} \\ Y_i &= -1 \text{ if } i \text{ is in class 2} \end{aligned}$$

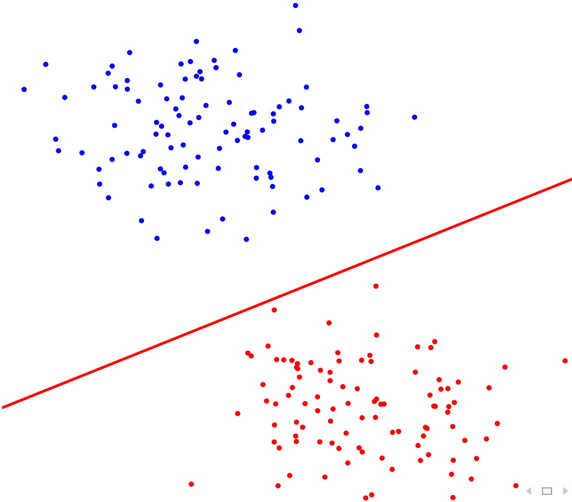
Suppose they are separable:

- Draw a line between groups
- Goal: identify the line in the middle
- Maximum margin

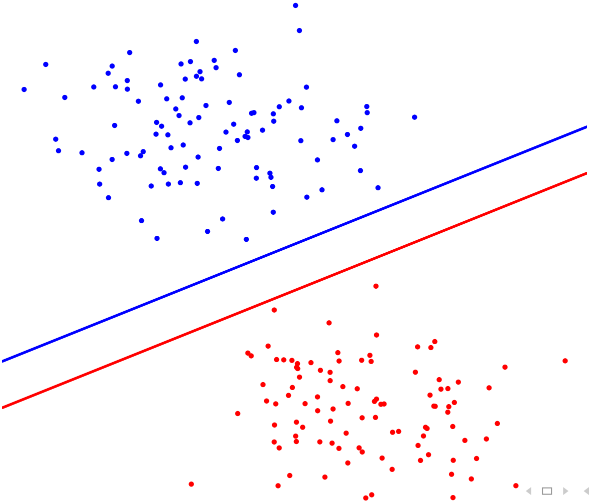
Support Vector Machines: Maximum Margin Classifier (Bishop 2006)



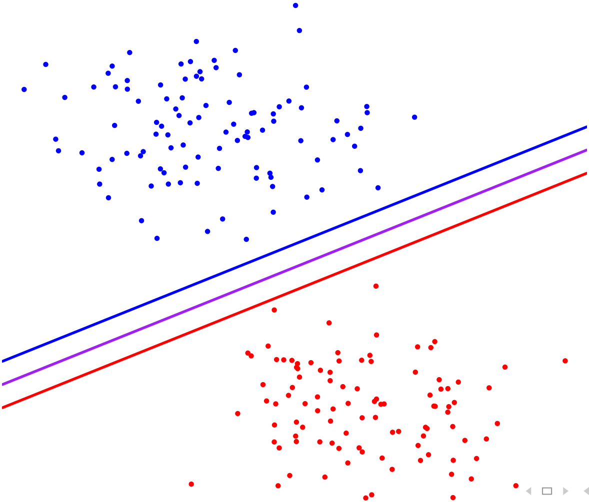
Support Vector Machines: Maximum Margin Classifier (Bishop 2006)



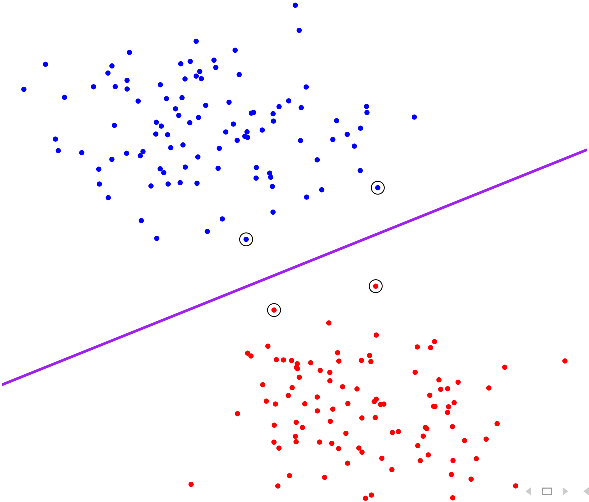
Support Vector Machines: Maximum Margin Classifier (Bishop 2006)



Support Vector Machines: Maximum Margin Classifier (Bishop 2006)



Support Vector Machines: Maximum Margin Classifier (Bishop 2006)



Support Vector Machines: Algebra (Bishop 2006)

Goal create a score to classify:

$$s(\boldsymbol{x}_i) \quad = \quad \boldsymbol{\beta}^{'}\boldsymbol{x}_i + b$$

- $\boldsymbol{\beta}$ Determines orientation of surface (slope)
- b determines location (moves surface up or down)
- If $s(\boldsymbol{x}_i) > 0 \rightarrow$ class 1
- If $s(\boldsymbol{x}_i) < 0 \rightarrow$ class 2
- $\frac{|s(\boldsymbol{x}_i)|}{||\boldsymbol{\beta}||} =$ Document distance from decision surface (margin)

Support Vector Machines: Algebra (Bishop 2006)

Objective function: maximum margin

Support Vector Machines: Algebra (Bishop 2006)

Objective function: maximum margin

$\text{min}_i[\ |(s(\boldsymbol{x}_i)|\ ]$: Point closest to decision surface

Support Vector Machines: Algebra (Bishop 2006)

Objective function: maximum margin

min$_i$[$|(s(\boldsymbol{x}_i)|$]: Point closest to decision surface

We want to identify $\boldsymbol{\beta}$ and b to maximize the margin:

Support Vector Machines: Algebra (Bishop 2006)

Objective function: maximum margin

$\min_i[\ |(s(\boldsymbol{x}_i)|\ ]$: Point closest to decision surface

We want to identify $\boldsymbol{\beta}$ and b to maximize the margin:

$$\text{arg max}_{\boldsymbol{\beta},b} \left\{ \frac{1}{||\boldsymbol{\beta}||} \min_i[\ |(s(\boldsymbol{x}_i)|\ ] \right\}$$

Support Vector Machines: Algebra (Bishop 2006)

Objective function: maximum margin

$\text{min}_i[\ |(s(\boldsymbol{x}_i)|\ ]$: Point closest to decision surface

We want to identify $\boldsymbol{\beta}$ and b to maximize the margin:

$$\text{arg max}_{\boldsymbol{\beta},b}\left\{\frac{1}{||\boldsymbol{\beta}||}\ \text{min}_i[\ |(s(\boldsymbol{x}_i)|\ ]\right\}$$

$$\text{arg max}_{\boldsymbol{\beta},b}\left\{\frac{1}{||\boldsymbol{\beta}||}\ \text{min}_i[\ |\boldsymbol{\beta}^{'}\boldsymbol{x}_i + b|\ ]\right\}$$

Support Vector Machines: Algebra (Bishop 2006)

Objective function: maximum margin

min$_i$[$|(s(\boldsymbol{x}_i)|$]: Point closest to decision surface

We want to identify $\boldsymbol{\beta}$ and b to maximize the margin:

$$\text{arg max}_{\boldsymbol{\beta},b} \left\{ \frac{1}{||\boldsymbol{\beta}||} \ \text{min}_i[\ |(s(\boldsymbol{x}_i)|\ ] \right\}$$

$$\text{arg max}_{\boldsymbol{\beta},b} \left\{ \frac{1}{||\boldsymbol{\beta}||} \ \text{min}_i[\ |\boldsymbol{\beta}^{'}\boldsymbol{x}_i + b|\ ] \right\}$$

Constrained optimization problem $\rightsquigarrow$ Quadratic programming problem

What About Overlap? (Bishop 2006)

- Rare that classes are separable.

What About Overlap? (Bishop 2006)

- Rare that classes are separable.
- Define:

What About Overlap? (Bishop 2006)

- Rare that classes are separable.
- Define:

$$\xi_i \quad = \quad 0 \text{ if correctly classified}$$

What About Overlap? (Bishop 2006)

- Rare that classes are separable.
- Define:

$$\begin{aligned} \xi_i &= 0 \text{ if correctly classified} \\ \xi_i &= |s(\boldsymbol{x}_i)| \text{ if incorrectly classified} \end{aligned}$$

What About Overlap? (Bishop 2006)

- Rare that classes are separable.
- Define:

$$\begin{aligned}\xi_i &= 0 \text{ if correctly classified} \\ \xi_i &= |s(\boldsymbol{x}_i)| \text{ if incorrectly classified}\end{aligned}$$

Tradeoff:

What About Overlap? (Bishop 2006)

- Rare that classes are separable.
- Define:

$$\xi_i = 0 \text{ if correctly classified}$$
$$\xi_i = |s(\boldsymbol{x}_i)| \text{ if incorrectly classified}$$

Tradeoff:

- Maximize margin between correctly classified groups

What About Overlap? (Bishop 2006)

- Rare that classes are separable.
- Define:

$$\begin{aligned} \xi_i &= 0 \text{ if correctly classified} \\ \xi_i &= |s(\boldsymbol{x}_i)| \text{ if incorrectly classified} \end{aligned}$$

Tradeoff:

- Maximize margin between correctly classified groups
- Minimize error from misclassified documents

What About Overlap? (Bishop 2006)

- Rare that classes are separable.
- Define:

$$\begin{aligned}\xi_i &= 0 \text{ if correctly classified} \\ \xi_i &= |s(\boldsymbol{x}_i)| \text{ if incorrectly classified}\end{aligned}$$

Tradeoff:

- Maximize margin between correctly classified groups
- Minimize error from misclassified documents

$$\text{arg max}_{\boldsymbol{\beta},b} \left\{ C \sum_{i=1}^{N} \xi_i + \frac{1}{||\boldsymbol{\beta}||} \text{ min}_i[\ |\boldsymbol{\beta}^{'} \boldsymbol{x}_i + b|\ ] \right\}$$

What About Overlap? (Bishop 2006)

- Rare that classes are separable.
- Define:

$$
\begin{aligned}
\xi_i &= 0 \text{ if correctly classified} \\
\xi_i &= |s(\boldsymbol{x}_i)| \text{ if incorrectly classified}
\end{aligned}
$$

Tradeoff:

- Maximize margin between correctly classified groups
- Minimize error from misclassified documents

$$
\text{arg max}_{\boldsymbol{\beta},b} \left\{ C \sum_{i=1}^{N} \xi_i + \frac{1}{||\boldsymbol{\beta}||} \text{ min}_i [\ |\boldsymbol{\beta}^{'} \boldsymbol{x}_i + b|\ ] \right\}
$$

C captures tradeoff

What About Overlap? (Bishop 2006)

- Rare that classes are separable.
- Define:

$$\begin{aligned} \xi_i &= 0 \text{ if correctly classified} \\ \xi_i &= |s(\boldsymbol{x}_i)| \text{ if incorrectly classified} \end{aligned}$$

Tradeoff:

- Maximize margin between correctly classified groups
- Minimize error from misclassified documents

$$\text{arg max}_{\boldsymbol{\beta},b} \left\{ C \sum_{i=1}^{N} \xi_i + \frac{1}{||\boldsymbol{\beta}||} \text{min}_i [\ |\boldsymbol{\beta}^{'} \boldsymbol{x}_i + b|\ ] \right\}$$

C captures tradeoff

How to Handle Multiple Comparisons?

- Rare that we only want to classify two categories

How to Handle Multiple Comparisons?

- Rare that we only want to classify two categories
- How to handle classification into K groups?

How to Handle Multiple Comparisons?

- Rare that we only want to classify two categories
- How to handle classification into K groups?
    1) Set up K classification problems:

How to Handle Multiple Comparisons?

- Rare that we only want to classify two categories
- How to handle classification into K groups?
    1) Set up K classification problems:
        - Compare each class to all other classes

How to Handle Multiple Comparisons?

- Rare that we only want to classify two categories
- How to handle classification into K groups?
  1) Set up K classification problems:
     - Compare each class to all other classes
     - Problem: can lead to inconsistent results

How to Handle Multiple Comparisons?

- Rare that we only want to classify two categories
- How to handle classification into K groups?
  1) Set up K classification problems:
     - Compare each class to all other classes
     - Problem: can lead to inconsistent results
     - Solution(?): select category with largest "score"

How to Handle Multiple Comparisons?

- Rare that we only want to classify two categories
- How to handle classification into K groups?
  1) Set up K classification problems:
     - Compare each class to all other classes
     - Problem: can lead to inconsistent results
     - Solution(?): select category with largest "score"
     - Problem: scales are not comparable

How to Handle Multiple Comparisons?

- Rare that we only want to classify two categories
- How to handle classification into K groups?
  1) Set up K classification problems:
     - Compare each class to all other classes
     - Problem: can lead to inconsistent results
     - Solution(?): select category with largest "score"
     - Problem: scales are not comparable
  2) Common solution: set up $K(K-1)/2$ classifications

How to Handle Multiple Comparisons?

- Rare that we only want to classify two categories
- How to handle classification into K groups?
  1) Set up K classification problems:
     - Compare each class to all other classes
     - Problem: can lead to inconsistent results
     - Solution(?): select category with largest "score"
     - Problem: scales are not comparable
  2) Common solution: set up $K(K-1)/2$ classifications
  - Perform vote to select class (still suboptimal)

How to Handle Multiple Comparisons?

- Rare that we only want to classify two categories
- How to handle classification into K groups?
  1) Set up K classification problems:
     - Compare each class to all other classes
     - Problem: can lead to inconsistent results
     - Solution(?): select category with largest "score"
     - Problem: scales are not comparable
  2) Common solution: set up $K(K-1)/2$ classifications
  - Perform vote to select class (still suboptimal)
  3) Simultaneous estimation possible, much slower

R Code to Run SVMs

```
library(e1071)
fit<- svm(T . , as.data.frame(tdm) , method ='C',
kernel='linear')
```

where: `method = 'C'` $\rightarrow$ Classification

`kernel='linear'` $\rightarrow$ allows for distortion of feature space. Options:

- Linear
- Polynomial
- Radial
- sigmoid

```
preds<- predict(fit, data =
as.data.frame(tdm[-c(1:no.train),]))
```

SVMs ⇝ Political Science Research

Hillard, Purpura, Wilkerson: SVMs to code topic/sub topics for policy agendas project

TABLE 3. Bill Title Interannotator Agreement for Five Model Types

	SVM	MaxEnt	Boostexter	Naïve Bayes
Major topic *N* = 20	88.7% (.881)	86.5% (.859)	85.6% (.849)	81.4% (.805)
Subtopic *N* = 226	81.0% (.800)	78.3% (.771)	73.6% (.722)	71.9% (.705)

Kernel Trick (Huge literature in machine learning) and KRLS (Hazlett and Hainmueller 2014; Hazlett inspired slides)

We want flexible models

- Recover complicated functional form
- Recover systematic features of data

Introduction to Flexible Regression

Prerequisite 1: Feature Maps

- $y_i \in \Re$ Dependent variable
- $\boldsymbol{x}_i \in \Re^J$ is $J \times 1$ covariate

Feature Map

A *feature map* $\phi(x_i)$ is a mapping from $\Re^J \rightarrow \Re^{J'}$, usually with $J' >> J$

For example,

$$\phi([x_1, x_2]) = [x_1, x_2, x_1^2, x_2^2, x_1 \cdot x_2]^T$$

- You are used to models linear in x:

$$f(x_i) = \sum_{j=1}^{J} (x_i)^{(j)} \beta_j$$

- We'll be working with models linear in $\phi(x)$

$$f(x_i) = \sum_{j=1}^{J'} \phi(x_i)^{(j)} \theta_j$$

Prerequisite 1: Feature Maps

- $y_i \in \Re$ Dependent variable
- $\boldsymbol{x}_i \in \Re^J$ is $J \times 1$ covariate

Feature Map

A *feature map* $\phi(x_i)$ is a mapping from $\Re^J \to \Re^{J'}$, usually with $J' >> J$

For example,

$$\phi([x_1, x_2]) = [x_1, x_2, x_1^2, x_2^2, x_1 \cdot x_2]^T$$

- You are used to models linear in x:

$$f(x_i) = \sum_{j=1}^{J} (x_i)^{(j)} \beta_j$$

- We'll be working with models linear in $\phi(x)$

$$f(x_i) = \sum_{j=1}^{J'} \phi(x_i)^{(j)} \theta_j$$

Prerequisite 1: Feature Maps

- $y_i \in \Re$ Dependent variable
- $\boldsymbol{x}_i \in \Re^J$ is $J \times 1$ covariate

Feature Map

A *feature map* $\phi(x_i)$ is a mapping from $\Re^J \rightarrow \Re^{J'}$, usually with $J' >> J$

For example,

$$\phi([x_1, x_2]) = [x_1, x_2, x_1^2, x_2^2, x_1 \cdot x_2]^T$$

- You are used to models linear in x:

$$f(x_i) = \sum_{j=1}^{J} (x_i)^{(j)} \beta_j$$

- We'll be working with models linear in $\phi(x)$

$$f(x_i) = \sum_{j=1}^{J'} \phi(x_i)^{(j)} \theta_j$$

Prerequisite 1: Feature Maps

- $y_i \in \Re$ Dependent variable
- $\boldsymbol{x}_i \in \Re^J$ is $J \times 1$ covariate

Feature Map

A *feature map* $\phi(x_i)$ is a mapping from $\Re^J \rightarrow \Re^{J'}$, usually with $J' >> J$

For example,

$$\phi([x_1, x_2]) = [x_1, x_2, x_1^2, x_2^2, x_1 \cdot x_2]^T$$

- You are used to models linear in x:

$$f(x_i) = \sum_{j=1}^{J} (x_i)^{(j)} \beta_j$$

- We'll be working with models linear in $\phi(x)$

$$f(x_i) = \sum_{j=1}^{J'} \phi(x_i)^{(j)} \theta_j$$

Prerequisite 1: Feature Maps

- $y_i \in \Re$ Dependent variable
- $\boldsymbol{x}_i \in \Re^J$ is $J \times 1$ covariate

Feature Map

A *feature map* $\phi(x_i)$ is a mapping from $\Re^J \rightarrow \Re^{J'}$, usually with $J' >> J$

For example,

$$\phi([x_1, x_2]) = [x_1, x_2, x_1^2, x_2^2, x_1 \cdot x_2]^T$$

- You are used to models linear in x:

$$f(x_i) = \sum_{j=1}^{J} (x_i)^{(j)} \beta_j$$

- We'll be working with models linear in $\phi(x)$

$$f(x_i) = \sum_{j=1}^{J'} \phi(x_i)^{(j)} \theta_j$$

Prereqs 2: Inner Products

Consider vector $u, v, w \in \mathbb{R}^J$, and scalar a.

- Standard inner-product: $\langle u, v \rangle = u^T v$.
- More broadly, an inner-product $\langle u, v \rangle_\star$ satisfies:
  1. Symmetry: $\langle u, v \rangle = \langle v, u \rangle$
  2. Linearity: $\langle au, v \rangle = a\langle u, v \rangle$, and $\langle u + w, v \rangle = \langle u, v \rangle + \langle w, v \rangle$
  3. Positive Definite: $\langle u, u \rangle \geq 0$.
- Some common uses:
  - Orthogonality: if $\overline{u}, \overline{v} = 0$, u, v orthogonal if $\langle u, v \rangle = 0$
  - Length of a vector, e.g. $||u|| = \sqrt{\langle u, u \rangle}$

    where $|| \cdot ||$ is a norm, in this case the Euclidean norm
  - Distance between vectors, e.g. $||u - v|| = \sqrt{\langle u - v, u - v \rangle}$

Prereqs 2: Inner Products

Consider vector $u, v, w \in \mathbb{R}^J$, and scalar a.

- Standard inner-product: $\langle u, v \rangle = u^T v$.
- More broadly, an inner-product $\langle u, v \rangle_\star$ satisfies:
  1. Symmetry: $\langle u, v \rangle = \langle v, u \rangle$
  2. Linearity: $\langle au, v \rangle = a\langle u, v \rangle$, and $\langle u + w, v \rangle = \langle u, v \rangle + \langle w, v \rangle$
  3. Positive Definite: $\langle u, u \rangle \geq 0$.

- Some common uses:
  - Orthogonality: if $\overline{u}, \overline{v} = 0$, u, v orthogonal if $\langle u, v \rangle = 0$
  - Length of a vector, e.g. $||u|| = \sqrt{\langle u, u \rangle}$

    where $|| \cdot ||$ is a norm, in this case the Euclidean norm
  - Distance between vectors, e.g. $||u - v|| = \sqrt{\langle u - v, u - v \rangle}$

Prereqs 2: Inner Products

Consider vector $u, v, w \in \mathbb{R}^J$, and scalar a.

- Standard inner-product: $\langle u, v \rangle = u^T v$.
- More broadly, an inner-product $\langle u, v \rangle_{\star}$ satisfies:
  1. Symmetry: $\langle u, v \rangle = \langle v, u \rangle$
  2. Linearity: $\langle au, v \rangle = a \langle u, v \rangle$, and $\langle u + w, v \rangle = \langle u, v \rangle + \langle w, v \rangle$
  3. Positive Definite: $\langle u, u \rangle \geq 0$.
- Some common uses:
  - Orthogonality: if $\overline{u}, \overline{v} = 0$, u, v orthogonal if $\langle u, v \rangle = 0$
  - Length of a vector, e.g. $||u|| = \sqrt{\langle u, u \rangle}$

    where $|| \cdot ||$ is a norm, in this case the Euclidean norm
  - Distance between vectors, e.g. $||u - v|| = \sqrt{\langle u - v, u - v \rangle}$

Prereqs 2: Inner Products

Consider vector $u, v, w \in \mathbb{R}^J$, and scalar a.

- Standard inner-product: $\langle u, v \rangle = u^T v$.
- More broadly, an inner-product $\langle u, v \rangle_\star$ satisfies:
  1. Symmetry: $\langle u, v \rangle = \langle v, u \rangle$
  2. Linearity: $\langle au, v \rangle = a\langle u, v \rangle$, and $\langle u + w, v \rangle = \langle u, v \rangle + \langle w, v \rangle$
  3. Positive Definite: $\langle u, u \rangle \geq 0$.
- Some common uses:
  - Orthogonality: if $\overline{u}, \overline{v} = 0$, u, v orthogonal if $\langle u, v \rangle = 0$
  - Length of a vector, e.g. $||u|| = \sqrt{\langle u, u \rangle}$

    where $|| \cdot ||$ is a norm, in this case the Euclidean norm
  - Distance between vectors, e.g. $||u - v|| = \sqrt{\langle u - v, u - v \rangle}$

Prereqs 2: Inner Products

Consider vector $u, v, w \in \mathbb{R}^J$, and scalar a.

- Standard inner-product: $\langle u, v \rangle = u^T v$.
- More broadly, an inner-product $\langle u, v \rangle_\star$ satisfies:
  1. Symmetry: $\langle u, v \rangle = \langle v, u \rangle$
  2. Linearity: $\langle au, v \rangle = a\langle u, v \rangle$, and $\langle u + w, v \rangle = \langle u, v \rangle + \langle w, v \rangle$
  3. Positive Definite: $\langle u, u \rangle \geq 0$.

- Some common uses:
  - Orthogonality: if $\overline{u}, \overline{v} = 0$, u, v orthogonal if $\langle u, v \rangle = 0$
  - Length of a vector, e.g. $||u|| = \sqrt{\langle u, u \rangle}$
    where $|| \cdot ||$ is a norm, in this case the Euclidean norm
  - Distance between vectors, e.g. $||u - v|| = \sqrt{\langle u - v, u - v \rangle}$

Prereqs 2: Inner Products

Consider vector $u, v, w \in \mathbb{R}^J$, and scalar a.

- Standard inner-product: $\langle u, v \rangle = u^T v$.
- More broadly, an inner-product $\langle u, v \rangle_\star$ satisfies:
  1. Symmetry: $\langle u, v \rangle = \langle v, u \rangle$
  2. Linearity: $\langle au, v \rangle = a\langle u, v \rangle$, and $\langle u + w, v \rangle = \langle u, v \rangle + \langle w, v \rangle$
  3. Positive Definite: $\langle u, u \rangle \geq 0$.

- Some common uses:
  - Orthogonality: if $\overline{u}, \overline{v} = 0$, u, v orthogonal if $\langle u, v \rangle = 0$
  - Length of a vector, e.g. $||u|| = \sqrt{\langle u, u \rangle}$
    where $|| \cdot ||$ is a norm, in this case the Euclidean norm
  - Distance between vectors, e.g. $||u - v|| = \sqrt{\langle u - v, u - v \rangle}$

Prereqs 2: Inner Products

Consider vector $u, v, w \in \mathbb{R}^J$, and scalar a.

- Standard inner-product: $\langle u, v \rangle = u^T v$.
- More broadly, an inner-product $\langle u, v \rangle_{\star}$ satisfies:
  1. Symmetry: $\langle u, v \rangle = \langle v, u \rangle$
  2. Linearity: $\langle au, v \rangle = a\langle u, v \rangle$, and $\langle u + w, v \rangle = \langle u, v \rangle + \langle w, v \rangle$
  3. Positive Definite: $\langle u, u \rangle \geq 0$.
- Some common uses:
  - Orthogonality: if $\overline{u}, \overline{v} = 0$, u, v orthogonal if $\langle u, v \rangle = 0$
  - Length of a vector, e.g. $||u|| = \sqrt{\langle u, u \rangle}$
    where $|| \cdot ||$ is a norm, in this case the Euclidean norm
  - Distance between vectors, e.g. $||u - v|| = \sqrt{\langle u - v, u - v \rangle}$

Prereqs 2: Inner Products

Consider vector $u, v, w \in \mathbb{R}^J$, and scalar a.

- Standard inner-product: $\langle u, v \rangle = u^T v$.
- More broadly, an inner-product $\langle u, v \rangle_\star$ satisfies:
  1. Symmetry: $\langle u, v \rangle = \langle v, u \rangle$
  2. Linearity: $\langle au, v \rangle = a\langle u, v \rangle$, and $\langle u + w, v \rangle = \langle u, v \rangle + \langle w, v \rangle$
  3. Positive Definite: $\langle u, u \rangle \geq 0$.

- Some common uses:
  - Orthogonality: if $\overline{u}, \overline{v} = 0$, u, v orthogonal if $\langle u, v \rangle = 0$
  - Length of a vector, e.g. $||u|| = \sqrt{\langle u, u \rangle}$

    where $|| \cdot ||$ is a norm, in this case the Euclidean norm
  - Distance between vectors, e.g. $||u - v|| = \sqrt{\langle u - v, u - v \rangle}$

Prereqs 3: Kernels

Kernel

A kernel is a function $\Re^J \times \Re^J \rightarrow \Re$

$$k(x_i, x_l) \rightarrow \Re$$

Interpretable as an inverse distance metric.

Gaussian Kernel

$$k(x_l, x_i) = e^{-\frac{||x_l - x_i||^2}{\sigma^2}}$$

where $||x_l - x_i||$ is the Euclidean distance between x_l and x_i

Prereqs 3: Kernels

Kernel

A kernel is a function $\Re^J \times \Re^J \rightarrow \Re$

$$k(x_i, x_l) \rightarrow \Re$$

Interpretable as an inverse distance metric.

Gaussian Kernel

$$k(x_l, x_i) = e^{-\frac{||x_l - x_i||^2}{\sigma^2}}$$

where $||x_l - x_i||$ is the Euclidean distance between x_l and x_i

Prereqs 3b: Positive Semi-Definite Kernels

- Construct "kernel matrix" K s.t. $K_{l,i} = k(x_l, x_i)$.
- What are some properties of K? What are its dimensions?

Definition: Positive Semi-definite Kernels

A kernel function $k(\cdot, \cdot)$ is positive semi-definite (PSD) if and only if for any $u \in \Re^N$, $u^T K u \geq 0$.

- Fact: for any Positive Semi-Definite kernel k there exists some choice of $\phi(\cdot)$ s.t.

$$k(x_i, x_l) = \langle \phi(x_i), \phi(x_l) \rangle$$

Prereqs 3b: Positive Semi-Definite Kernels

- Construct "kernel matrix" K s.t. $K_{l,i} = k(x_l, x_i)$.
- What are some properties of K? What are its dimensions?

Definition: Positive Semi-definite Kernels

A kernel function $k(\cdot,\cdot)$ is positive semi-definite (PSD) if and only if for any $u \in \Re^N$, $u^T K u \geq 0$.

- Fact: for any Positive Semi-Definite kernel k there exists some choice of $\phi(\cdot)$ s.t.

$$k(x_i, x_l) = \langle \phi(x_i), \phi(x_l) \rangle$$

Prereqs 3b: Positive Semi-Definite Kernels

- Construct "kernel matrix" K s.t. $K_{l,i} = k(x_l, x_i)$.
- What are some properties of K? What are its dimensions?

Definition: Positive Semi-definite Kernels

A kernel function $k(\cdot, \cdot)$ is positive semi-definite (PSD) if and only if for any $u \in \Re^N$, $u^T K u \geq 0$.

- Fact: for any Positive Semi-Definite kernel k there exists some choice of $\phi(\cdot)$ s.t.

$$k(x_i, x_l) = \langle \phi(x_i), \phi(x_l) \rangle$$

Prereqs 3b: Positive Semi-Definite Kernels

- Construct "kernel matrix" K s.t. $K_{l,i} = k(x_l, x_i)$.
- What are some properties of K? What are its dimensions?

Definition: Positive Semi-definite Kernels

A kernel function $k(\cdot, \cdot)$ is positive semi-definite (PSD) if and only if for any $u \in \Re^N$, $u^T K u \geq 0$.

- Fact: for any Positive Semi-Definite kernel k there exists some choice of $\phi(\cdot)$ s.t.

$$k(x_i, x_l) = \langle \phi(x_i), \phi(x_l) \rangle$$

Some examples of kernels as inner-products

- Take vectors (observations) $\boldsymbol{x} = [x_1, x_2]^{'}$, and $\boldsymbol{y} = [y_1, y_2]^{'}$.
- Suppose you construct

$$\phi(\boldsymbol{x}) = [1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, \sqrt{2}x_1x_2, x_2^2]^T$$

- Define $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^{'}\boldsymbol{y}$
- Then

$$\begin{aligned}
\langle \phi(\boldsymbol{x}), \phi(\boldsymbol{y}) \rangle = \phi(x) &= [1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, \sqrt{2}x_1x_2, x_2^2] \begin{bmatrix} 1 \\ \sqrt{2}y_1 \\ \sqrt{2}y_2 \\ y_1^2 \\ \sqrt{2}y_1y_2 \\ y_2^2 \end{bmatrix} \\
&= 1 + 2x_1y_1 + 2x_2y_2 + x_1^2y_1^2 + 2x_1x_2y_1y_2 + x_2^2y_2^2 \\
&= (1 + \langle \boldsymbol{x}, \boldsymbol{y} \rangle)^2
\end{aligned}$$

- So $k(\boldsymbol{x}, \boldsymbol{y}) = (1 + \langle x, y \rangle)^2$ is same as $\langle \phi(x), \phi(y) \rangle$, without having to do the mapping

Some examples of kernels as inner-products

- Take vectors (observations) $\boldsymbol{x} = [x_1, x_2]^{'}$, and $\boldsymbol{y} = [y_1, y_2]^{'}$.
- Suppose you construct

$$\phi(\boldsymbol{x}) = [1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, \sqrt{2}x_1x_2, x_2^2]^T$$

- Define $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^{'}\boldsymbol{y}$
- Then

$$\langle \phi(\boldsymbol{x}), \phi(\boldsymbol{y}) \rangle = \phi(x) = [1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, \sqrt{2}x_1x_2, x_2^2] \begin{bmatrix} 1 \\ \sqrt{2}y_1 \\ \sqrt{2}y_2 \\ y_1^2 \\ \sqrt{2}y_1y_2 \\ y_2^2 \end{bmatrix}$$

$$= 1 + 2x_1y_1 + 2x_2y_2 + x_1^2y_1^2 + 2x_1x_2y_1y_2 + x_2^2y_2^2$$

$$= (1 + \langle \boldsymbol{x}, \boldsymbol{y} \rangle)^2$$

- So $k(\boldsymbol{x}, \boldsymbol{y}) = (1 + \langle x, y \rangle)^2$ is same as $\langle \phi(x), \phi(y) \rangle$, without having to do the mapping

Some examples of kernels as inner-products

- Take vectors (observations) $\boldsymbol{x} = [x_1, x_2]^{'}$, and $\boldsymbol{y} = [y_1, y_2]^{'}$.
- Suppose you construct

$$\phi(\boldsymbol{x}) = [1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, \sqrt{2}x_1x_2, x_2^2]^T$$

- Define $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^{'}\boldsymbol{y}$
- Then

$$\begin{aligned}
\langle \phi(\boldsymbol{x}), \phi(\boldsymbol{y}) \rangle = \phi(x) &= [1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, \sqrt{2}x_1x_2, x_2^2] \begin{bmatrix} 1 \\ \sqrt{2}y_1 \\ \sqrt{2}y_2 \\ y_1^2 \\ \sqrt{2}y_1y_2 \\ y_2^2 \end{bmatrix} \\
&= 1 + 2x_1y_1 + 2x_2y_2 + x_1^2y_1^2 + 2x_1x_2y_1y_2 + x_2^2y_2^2 \\
&= (1 + \langle \boldsymbol{x}, \boldsymbol{y} \rangle)^2
\end{aligned}$$

- So $k(\boldsymbol{x}, \boldsymbol{y}) = (1 + \langle x, y \rangle)^2$ is same as $\langle \phi(x), \phi(y) \rangle$, without having to do the mapping

Some examples of kernels as inner-products

- Take vectors (observations) $\boldsymbol{x} = [x_1, x_2]^{'}$, and $\boldsymbol{y} = [y_1, y_2]^{'}$.
- Suppose you construct

$$\phi(\boldsymbol{x}) = [1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, \sqrt{2}x_1x_2, x_2^2]^T$$

- Define $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^{'}\boldsymbol{y}$
- Then

$$\begin{aligned}
\langle \phi(\boldsymbol{x}), \phi(\boldsymbol{y}) \rangle = \phi(x) &= [1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, \sqrt{2}x_1x_2, x_2^2] \begin{bmatrix} 1 \\ \sqrt{2}y_1 \\ \sqrt{2}y_2 \\ y_1^2 \\ \sqrt{2}y_1y_2 \\ y_2^2 \end{bmatrix} \\
&= 1 + 2x_1y_1 + 2x_2y_2 + x_1^2y_1^2 + 2x_1x_2y_1y_2 + x_2^2y_2^2 \\
&= (1 + \langle \boldsymbol{x}, \boldsymbol{y} \rangle)^2
\end{aligned}$$

- So $k(\boldsymbol{x}, \boldsymbol{y}) = (1 + \langle x, y \rangle)^2$ is same as $\langle \phi(x), \phi(y) \rangle$, without having to do the mapping

Some examples of kernels as inner-products

Some other kernels:

- More generally $k(\boldsymbol{x}, \boldsymbol{y}) = (1 + \langle x, y \rangle)^d$ maps to d-order polynomials
- Linear kernel: $k(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^{'}\boldsymbol{y}$, so $\phi(\boldsymbol{x}) = \boldsymbol{x}$
- Gaussian kernel maps to infinite-dimensional $\phi(\cdot)$

Why do you care?

The Kernel Trick

If you can write an algorithm that uses the data only as inner-products, you can operate in high or infinite dimensional feature space without ever computing $\phi(\boldsymbol{x})$.

Some examples of kernels as inner-products

Some other kernels:

- More generally $k(\boldsymbol{x}, \boldsymbol{y}) = (1 + \langle x, y \rangle)^d$ maps to d-order polynomials
- Linear kernel: $k(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^{'}\boldsymbol{y}$, so $\phi(\boldsymbol{x}) = \boldsymbol{x}$
- Gaussian kernel maps to infinite-dimensional $\phi(\cdot)$

Why do you care?

The Kernel Trick

If you can write an algorithm that uses the data only as inner-products, you can operate in high or infinite dimensional feature space without ever computing $\phi(\boldsymbol{x})$.

Some examples of kernels as inner-products

Some other kernels:

- More generally $k(\boldsymbol{x}, \boldsymbol{y}) = (1 + \langle x, y \rangle)^d$ maps to d-order polynomials
- Linear kernel: $k(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^{'}\boldsymbol{y}$, so $\phi(\boldsymbol{x}) = \boldsymbol{x}$
- Gaussian kernel maps to infinite-dimensional $\phi(\cdot)$

Why do you care?

The Kernel Trick

If you can write an algorithm that uses the data only as inner-products, you can operate in high or infinite dimensional feature space without ever computing $\phi(\boldsymbol{x})$.

Some examples of kernels as inner-products

Some other kernels:

- More generally $k(\boldsymbol{x}, \boldsymbol{y}) = (1 + \langle x, y \rangle)^d$ maps to d-order polynomials
- Linear kernel: $k(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^{'}\boldsymbol{y}$, so $\phi(\boldsymbol{x}) = \boldsymbol{x}$
- Gaussian kernel maps to infinite-dimensional $\phi(\cdot)$

Why do you care?

The Kernel Trick

If you can write an algorithm that uses the data only as inner-products, you can operate in high or infinite dimensional feature space without ever computing $\phi(\boldsymbol{x})$.

Using the Kernel Trick for Regression

- A feature map, $\phi : \Re^J \mapsto \Re^{J'}$, such that: $k(x_i, x_l) = \langle \phi(x_i), \phi(x_l) \rangle$
- A linear model in the new features: $f(x_i) = \phi(x_i)^{'}\theta$, $\theta \in \mathbb{R}^{J'}$
- Regularized (ridge) regression:

$$\underset{\theta \in \mathbb{R}^{J'}}{\operatorname{argmin}} \sum_{i=1}^{N} (y_i - \phi(x_i)^{'}\theta)^2 + \lambda \langle \theta, \theta \rangle$$

- Solve the F.O.C.s:

$$R(\theta) = \sum_{i=1}^{N} (y_i - \phi(x_i)^{'}\theta)^2 + \lambda \theta^{'}\theta$$

$$\frac{\partial R(\theta)}{\partial \theta} = -2 \sum_{i=1}^{N} \phi(x_i)(y_i - \phi(x_i)^{'}\theta) + 2\lambda\theta = 0$$

Using the Kernel Trick for Regression

- A feature map, $\phi : \Re^J \mapsto \Re^{J'}$, such that: $k(x_i, x_l) = \langle \phi(x_i), \phi(x_l) \rangle$
- A linear model in the new features: $f(x_i) = \phi(x_i)^{'}\theta$, $\theta \in \mathbb{R}^{J'}$
- Regularized (ridge) regression:

$$\underset{\theta \in \mathbb{R}^{J'}}{\operatorname{argmin}} \sum_{i=1}^{N} (y_i - \phi(x_i)^{'}\theta)^2 + \lambda \langle \theta, \theta \rangle$$

- Solve the F.O.C.s:

$$R(\theta) = \sum_{i=1}^{N} (y_i - \phi(x_i)^{'}\theta)^2 + \lambda \theta^{'}\theta$$

$$\frac{\partial R(\theta)}{\partial \theta} = -2 \sum_{i=1}^{N} \phi(x_i)(y_i - \phi(x_i)^{'}\theta) + 2\lambda\theta = 0$$

Using the Kernel Trick for Regression

- A feature map, $\phi : \Re^J \mapsto \Re^{J'}$, such that: $k(x_i, x_l) = \langle \phi(x_i), \phi(x_l) \rangle$
- A linear model in the new features: $f(x_i) = \phi(x_i)^{'}\theta$, $\theta \in \mathbb{R}^{J'}$
- Regularized (ridge) regression:

$$\underset{\theta \in \mathbb{R}^{J'}}{\text{argmin}} \sum_{i=1}^{N} (y_i - \phi(x_i)^{'}\theta)^2 + \lambda \langle \theta, \theta \rangle$$

- Solve the F.O.C.s:

$$R(\theta) = \sum_{i=1}^{N} (y_i - \phi(x_i)^{'}\theta)^2 + \lambda \theta^{'}\theta$$

$$\frac{\partial R(\theta)}{\partial \theta} = -2 \sum_{i=1}^{N} \phi(x_i)(y_i - \phi(x_i)^{'}\theta) + 2\lambda\theta = 0$$

Using the Kernel Trick for Regression

- A feature map, $\phi : \Re^J \mapsto \Re^{J'}$, such that: $k(x_i, x_l) = \langle \phi(x_i), \phi(x_l) \rangle$
- A linear model in the new features: $f(x_i) = \phi(x_i)^{'}\theta$, $\theta \in \mathbb{R}^{J'}$
- Regularized (ridge) regression:

$$\underset{\theta \in \mathbb{R}^{J'}}{\text{argmin}} \sum_{i=1}^{N} (y_i - \phi(x_i)^{'}\theta)^2 + \lambda \langle \theta, \theta \rangle$$

- Solve the F.O.C.s:

$$R(\theta) = \sum_{i=1}^{N} (y_i - \phi(x_i)^{'}\theta)^2 + \lambda \theta^{'}\theta$$

$$\frac{\partial R(\theta)}{\partial \theta} = -2 \sum_{i=1}^{N} \phi(x_i)(y_i - \phi(x_i)^{'}\theta) + 2\lambda\theta = 0$$

Infinite Ridge Regression

$$\theta = \frac{1}{\lambda}\sum_{i}^{N}(y_i - \phi(x_i)^{'}\theta)\phi(x_i)$$

- Looks scary, but $y_i - \phi(x_i)^{'}\theta$ is just N scalars.
- Let $c_i = \frac{1}{\lambda}(y_i - \phi(x_i)^{'}\theta)$, then

$$\theta = \sum_{i=1}^{N} c_i \phi(x_i) \tag{2.1}$$

This is great! Despite being possibly infinite-dimensional,

- Solution for θ is in the span of features at observed points
- And has just N parameters
- This result given more directly by Representer Theorem

Infinite Ridge Regression

$$\theta = \frac{1}{\lambda}\sum_{i}^{N}(y_i - \phi(x_i)'\theta)\phi(x_i)$$

- Looks scary, but $y_i - \phi(x_i)'\theta$ is just N scalars.
- Let $c_i = \frac{1}{\lambda}(y_i - \phi(x_i)'\theta)$, then

$$\theta = \sum_{i=1}^{N} c_i\phi(x_i) \qquad (2.1)$$

This is great! Despite being possibly infinite-dimensional,

- Solution for θ is in the span of features at observed points
- And has just N parameters
- This result given more directly by Representer Theorem

Infinite Ridge Regression

$$\theta = \frac{1}{\lambda}\sum_{i}^{N} (y_i - \phi(x_i)^{'}\theta)\phi(x_i)$$

- Looks scary, but $y_i - \phi(x_i)^{'}\theta$ is just N scalars.
- Let $c_i = \frac{1}{\lambda}(y_i - \phi(x_i)^{'}\theta)$, then

$$\theta = \sum_{i=1}^{N} c_i \phi(x_i) \tag{2.1}$$

This is great! Despite being possibly infinite-dimensional,

- Solution for θ is in the span of features at observed points
- And has just N parameters
- This result given more directly by Representer Theorem

Infinite Ridge Regression

$$\theta = \frac{1}{\lambda}\sum_{i}^{N}(y_i - \phi(x_i)^{'}\theta)\phi(x_i)$$

- Looks scary, but $y_i - \phi(x_i)^{'}\theta$ is just N scalars.
- Let $c_i = \frac{1}{\lambda}(y_i - \phi(x_i)^{'}\theta)$, then

$$\theta = \sum_{i=1}^{N} c_i\phi(x_i) \tag{2.1}$$

This is great! Despite being possibly infinite-dimensional,

- Solution for θ is in the span of features at observed points
- And has just N parameters
- This result given more directly by Representer Theorem

Infinite Ridge Regression

$$\theta = \frac{1}{\lambda}\sum_{i}^{N}(y_i - \phi(x_i)'\theta)\phi(x_i)$$

- Looks scary, but $y_i - \phi(x_i)'\theta$ is just N scalars.
- Let $c_i = \frac{1}{\lambda}(y_i - \phi(x_i)'\theta)$, then

$$\theta = \sum_{i=1}^{N} c_i\phi(x_i) \tag{2.1}$$

This is great! Despite being possibly infinite-dimensional,

- Solution for θ is in the span of features at observed points
- And has just N parameters
- This result given more directly by Representer Theorem

Infinite Ridge Regression

$$\theta = \frac{1}{\lambda} \sum_{i}^{N} (y_i - \phi(x_i)^{'} \theta) \phi(x_i)$$

- Looks scary, but $y_i - \phi(x_i)^{'} \theta$ is just N scalars.
- Let $c_i = \frac{1}{\lambda}(y_i - \phi(x_i)^{'} \theta)$, then

$$\theta = \sum_{i=1}^{N} c_i \phi(x_i) \tag{2.1}$$

This is great! Despite being possibly infinite-dimensional,

- Solution for θ is in the span of features at observed points
- And has just N parameters
- This result given more directly by Representer Theorem

Infinite Ridge Regression

$$\theta = \frac{1}{\lambda}\sum_{i}^{N}(y_i - \phi(x_i)^{'}\theta)\phi(x_i)$$

- Looks scary, but $y_i - \phi(x_i)^{'}\theta$ is just N scalars.
- Let $c_i = \frac{1}{\lambda}(y_i - \phi(x_i)^{'}\theta)$, then

$$\theta = \sum_{i=1}^{N} c_i\phi(x_i) \tag{2.1}$$

This is great! Despite being possibly infinite-dimensional,

- Solution for θ is in the span of features at observed points
- And has just N parameters
- This result given more directly by Representer Theorem

Infinite Dimensional Ridge Regression: Solution

To get $f(x)$ we never need to see $\phi(x)$:

$$
\begin{aligned}
f(x_i) &= \phi(x_i)^{'}\theta \\
&= \phi(x_i)^{'}\sum_{j=1}^{N} c_j \phi(x_j) \\
&= \sum_{j=1}^{N} c_j \langle \phi(x_i), \phi(x_j) \rangle \\
&= \sum_{j}^{N} c_j k(x_j, x_i)
\end{aligned}
$$

- Or in vectors,

$$y = Kc$$

- And the regularizer, $\langle \theta, \theta \rangle = ||\theta||^2$

$$\langle \theta, \theta \rangle = \langle \sum_{i=1}^{N} c_i \phi(x_i), \sum_{i=1}^{N} c_i \phi(x_i) \rangle = c^T K c$$

Infinite Dimensional Ridge Regression: Solution

To get $f(x)$ we never need to see $\phi(x)$:

$$\begin{aligned} f(x_i) &= \phi(x_i)^{'}\theta \\ &= \phi(x_i)^{'}\sum_{j=1}^{N} c_j \phi(x_j) \\ &= \sum_{j=1}^{N} c_j \langle \phi(x_i), \phi(x_j) \rangle \\ &= \sum_{j} c_j k(x_j, x_i) \end{aligned}$$

- Or in vectors,

$$y = Kc$$

- And the regularizer, $\langle \theta, \theta \rangle = ||\theta||^2$

$$\langle \theta, \theta \rangle = \langle \sum_{i=1}^{N} c_i \phi(x_i), \sum_{i=1}^{N} c_i \phi(x_i) \rangle = c^T K c$$

Infinite Dimensional Ridge Regression: Solution

To get $f(x)$ we never need to see $\phi(x)$:

$$
\begin{aligned}
f(x_i) &= \phi(x_i)^{'}\theta \\
&= \phi(x_i)^{'}\sum_{j=1}^{N} c_j\phi(x_j) \\
&= \sum_{j=1}^{N} c_j\langle\phi(x_i),\phi(x_j)\rangle \\
&= \sum_{j}^{N} c_j k(x_j, x_i)
\end{aligned}
$$

- Or in vectors,

$$y = Kc$$

- And the regularizer, $\langle\theta,\theta\rangle = ||\theta||^2$

$$\langle\theta,\theta\rangle = \langle\sum_{i=1}^{N} c_i\phi(x_i), \sum_{i=1}^{N} c_i\phi(x_i)\rangle = c^T K c$$

Infinite Dimensional Ridge Regression

The key formula there is $f(x) = \sum_j^N c_j k(x_j, x)$, or $y = Kc$.

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_N) \\ k(x_2, x_1) & k(x_2, x_2) & \dots & k(x_2, x_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_N, x_1) & k(x_N, x_2) & \dots & k(x_N, x_N) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix}$$

Infinite Dimensional Ridge Regression

The key formula there is $f(x) = \sum_j^N c_j k(x_j, x)$, or $y = Kc$.

$$
\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_N) \\ k(x_2, x_1) & k(x_2, x_2) & \dots & k(x_2, x_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_N, x_1) & k(x_N, x_2) & \dots & k(x_N, x_N) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix}
$$

Ridge Regression in Feature Space

- Back once more to the minimization problem, we now have:

$$\underset{c \in \mathbb{R}^N}{\operatorname{argmin}} \, (y - Kc)^T (y - Kc) + \lambda c^T K c$$

- And we can get these c's in closed-form:

Closed-form solution for choice coefficients

$$c = (K + \lambda I)^{-1} y \tag{2.2}$$

- Summary: we can do ridge regression $\phi(x)$,
  - Closed form: $y = Kc$, with $c = (K + \lambda I)^{-1} y$
  - Even if $\phi(\cdot)$ is infinite-dimensional
  - We don't need to compute $\phi(x)$, just $\langle \phi(x_i) \phi(x_j) \rangle = k(x_i, x_j)$
  - No matter the dimension, just N values to solve.
  - This representation is *linear in the columns of K.*
  - Without regularization this would not have worked!

Ridge Regression in Feature Space

- Back once more to the minimization problem, we now have:

$$\underset{c \in \mathbb{R}^N}{\operatorname{argmin}} (y - Kc)^T (y - Kc) + \lambda c^T K c$$

- And we can get these c's in closed-form:

Closed-form solution for choice coefficients

$$c = (K + \lambda I)^{-1} y \tag{2.2}$$

- Summary: we can do ridge regression $\phi(x)$,
  - Closed form: $y = Kc$, with $c = (K + \lambda I)^{-1} y$
  - Even if $\phi(\cdot)$ is infinite-dimensional
  - We don't need to compute $\phi(x)$, just $\langle \phi(x_i)\phi(x_j) \rangle = k(x_i, x_j)$
  - No matter the dimension, just N values to solve.
  - This representation is *linear in the columns of K.*
  - Without regularization this would not have worked!

Ridge Regression in Feature Space

- Back once more to the minimization problem, we now have:

$$\underset{c \in \mathbb{R}^N}{\operatorname{argmin}} (y - Kc)^T (y - Kc) + \lambda c^T K c$$

- And we can get these c's in closed-form:

Closed-form solution for choice coefficients

$$c = (K + \lambda I)^{-1} y \tag{2.2}$$

- Summary: we can do ridge regression $\phi(x)$,
  - Closed form: $y = Kc$, with $c = (K + \lambda I)^{-1} y$
  - Even if $\phi(\cdot)$ is infinite-dimensional
  - We don't need to compute $\phi(x)$, just $\langle \phi(x_i) \phi(x_j) \rangle = k(x_i, x_j)$
  - No matter the dimension, just N values to solve.
  - This representation is *linear in the columns of K.*
  - Without regularization this would not have worked!

From RLS to KRLS

- All that sounds powerful and generalizes well
- The Gaussian kernel is often a good choice:
  - Terrific empirical performance
  - $||f||_K^2$ penalizes high-frequencies
  - Has close links to other methods (Gaussian processes)
- But we want to:
  - Develop intuitions for this space of functions?
  - Get from a good fit to useful quantitites of interest?
  - Do inference on those QoIs?
- "KRLS" is a particular set of choices and interpretational machinery to accomplish these
  1. Intuitions: Similarity-based, Gaussian superposition
  2. Choices: Gaussian kernel, $\sigma^2 = J$
  3. Interpretation: pointwise and average marginal effects

From RLS to KRLS

- All that sounds powerful and generalizes well
- The Gaussian kernel is often a good choice:
  - Terrific empirical performance
  - $||f||_K^2$ penalizes high-frequencies
  - Has close links to other methods (Gaussian processes)
- But we want to:
  - Develop intuitions for this space of functions?
  - Get from a good fit to useful quantitites of interest?
  - Do inference on those QoIs?
- "KRLS" is a particular set of choices and interpretational machinery to accomplish these
  1. Intuitions: Similarity-based, Gaussian superposition
  2. Choices: Gaussian kernel, $\sigma^2 = J$
  3. Interpretation: pointwise and average marginal effects

From RLS to KRLS

- All that sounds powerful and generalizes well
- The Gaussian kernel is often a good choice:
  - Terrific empirical performance
  - $||f||_K^2$ penalizes high-frequencies
  - Has close links to other methods (Gaussian processes)
- But we want to:
  - Develop intuitions for this space of functions?
  - Get from a good fit to useful quantitites of interest?
  - Do inference on those QoIs?
- "KRLS" is a particular set of choices and interpretational machinery to accomplish these
  1. Intuitions: Similarity-based, Gaussian superposition
  2. Choices: Gaussian kernel, $\sigma^2 = J$
  3. Interpretation: pointwise and average marginal effects

From RLS to KRLS

- All that sounds powerful and generalizes well
- The Gaussian kernel is often a good choice:
  - Terrific empirical performance
  - $||f||_K^2$ penalizes high-frequencies
  - Has close links to other methods (Gaussian processes)
- But we want to:
  - Develop intuitions for this space of functions?
  - Get from a good fit to useful quantitites of interest?
  - Do inference on those QoIs?
- "KRLS" is a particular set of choices and interpretational machinery to accomplish these
  1. Intuitions: Similarity-based, Gaussian superposition
  2. Choices: Gaussian kernel, $\sigma^2 = J$
  3. Interpretation: pointwise and average marginal effects

From RLS to KRLS

- All that sounds powerful and generalizes well
- The Gaussian kernel is often a good choice:
  - Terrific empirical performance
  - $||f||_K^2$ penalizes high-frequencies
  - Has close links to other methods (Gaussian processes)

- But we want to:
  - Develop intuitions for this space of functions?
  - Get from a good fit to useful quantitites of interest?
  - Do inference on those QoIs?

- "KRLS" is a particular set of choices and interpretational machinery to accomplish these
  1. Intuitions: Similarity-based, Gaussian superposition
  2. Choices: Gaussian kernel, $\sigma^2 = J$
  3. Interpretation: pointwise and average marginal effects

From RLS to KRLS

- All that sounds powerful and generalizes well
- The Gaussian kernel is often a good choice:
  - Terrific empirical performance
  - $||f||_K^2$ penalizes high-frequencies
  - Has close links to other methods (Gaussian processes)

- But we want to:
  - Develop intuitions for this space of functions?
  - Get from a good fit to useful quantitites of interest?
  - Do inference on those QoIs?

- "KRLS" is a particular set of choices and interpretational machinery to accomplish these
  1. Intuitions: Similarity-based, Gaussian superposition
  2. Choices: Gaussian kernel, $\sigma^2 = J$
  3. Interpretation: pointwise and average marginal effects

From RLS to KRLS

- All that sounds powerful and generalizes well
- The Gaussian kernel is often a good choice:
  - Terrific empirical performance
  - $||f||_K^2$ penalizes high-frequencies
  - Has close links to other methods (Gaussian processes)
- But we want to:
  - Develop intuitions for this space of functions?
  - Get from a good fit to useful quantitites of interest?
  - Do inference on those QoIs?
- "KRLS" is a particular set of choices and interpretational machinery to accomplish these
  1. Intuitions: Similarity-based, Gaussian superposition
  2. Choices: Gaussian kernel, $\sigma^2 = J$
  3. Interpretation: pointwise and average marginal effects

From RLS to KRLS

- All that sounds powerful and generalizes well
- The Gaussian kernel is often a good choice:
  - Terrific empirical performance
  - $||f||_K^2$ penalizes high-frequencies
  - Has close links to other methods (Gaussian processes)
- But we want to:
  - Develop intuitions for this space of functions?
  - Get from a good fit to useful quantitites of interest?
  - Do inference on those QoIs?
- "KRLS" is a particular set of choices and interpretational machinery to accomplish these
  1. Intuitions: Similarity-based, Gaussian superposition
  2. Choices: Gaussian kernel, $\sigma^2 = J$
  3. Interpretation: pointwise and average marginal effects

From RLS to KRLS

- All that sounds powerful and generalizes well
- The Gaussian kernel is often a good choice:
  - Terrific empirical performance
  - $||f||^2_K$ penalizes high-frequencies
  - Has close links to other methods (Gaussian processes)
- But we want to:
  - Develop intuitions for this space of functions?
  - Get from a good fit to useful quantitites of interest?
  - Do inference on those QoIs?
- "KRLS" is a particular set of choices and interpretational machinery to accomplish these
  1. Intuitions: Similarity-based, Gaussian superposition
  2. Choices: Gaussian kernel, $\sigma^2 = J$
  3. Interpretation: pointwise and average marginal effects

From RLS to KRLS

- All that sounds powerful and generalizes well
- The Gaussian kernel is often a good choice:
  - Terrific empirical performance
  - $||f||_K^2$ penalizes high-frequencies
  - Has close links to other methods (Gaussian processes)
- But we want to:
  - Develop intuitions for this space of functions?
  - Get from a good fit to useful quantitites of interest?
  - Do inference on those QoIs?
- "KRLS" is a particular set of choices and interpretational machinery to accomplish these
  1. Intuitions: Similarity-based, Gaussian superposition
  2. Choices: Gaussian kernel, $\sigma^2 = J$
  3. Interpretation: pointwise and average marginal effects

From RLS to KRLS

- All that sounds powerful and generalizes well
- The Gaussian kernel is often a good choice:
  - Terrific empirical performance
  - $||f||_K^2$ penalizes high-frequencies
  - Has close links to other methods (Gaussian processes)
- But we want to:
  - Develop intuitions for this space of functions?
  - Get from a good fit to useful quantitites of interest?
  - Do inference on those QoIs?
- "KRLS" is a particular set of choices and interpretational machinery to accomplish these
  1. Intuitions: Similarity-based, Gaussian superposition
  2. Choices: Gaussian kernel, $\sigma^2 = J$
  3. Interpretation: pointwise and average marginal effects

From RLS to KRLS

- All that sounds powerful and generalizes well
- The Gaussian kernel is often a good choice:
  - Terrific empirical performance
  - $||f||_K^2$ penalizes high-frequencies
  - Has close links to other methods (Gaussian processes)
- But we want to:
  - Develop intuitions for this space of functions?
  - Get from a good fit to useful quantitites of interest?
  - Do inference on those QoIs?
- "KRLS" is a particular set of choices and interpretational machinery to accomplish these
  1. Intuitions: Similarity-based, Gaussian superposition
  2. Choices: Gaussian kernel, $\sigma^2 = J$
  3. Interpretation: pointwise and average marginal effects

From RLS to KRLS

- All that sounds powerful and generalizes well
- The Gaussian kernel is often a good choice:
  - Terrific empirical performance
  - $||f||_K^2$ penalizes high-frequencies
  - Has close links to other methods (Gaussian processes)
- But we want to:
  - Develop intuitions for this space of functions?
  - Get from a good fit to useful quantitites of interest?
  - Do inference on those QoIs?
- "KRLS" is a particular set of choices and interpretational machinery to accomplish these
  1. Intuitions: Similarity-based, Gaussian superposition
  2. Choices: Gaussian kernel, $\sigma^2 = J$
  3. Interpretation: pointwise and average marginal effects

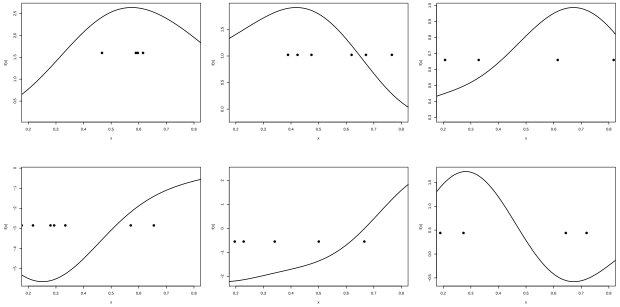
Intuition 1: Similarity

Think of the KRLS function space as built on similarity:

$$f(x^\star) = \sum_{i=1}^{N} c_i k(x^\star, x_i)$$

$$f(x^\star) = c_1(\text{similarity of } x^\star \text{ to } x_1) + \ldots + c_N(\text{similarity of } x^\star \text{ to } x_N)$$

Some random functions from this space:



You can also see it written out this way:
(recalling that $k(x_1, x_2)$ is similarity of x_1 to x_2)

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \ldots & k(x_1, x_N) \\ k(x_2, x_1) & k(x_2, x_2) & \ldots & k(x_2, x_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_N, x_1) & k(x_N, x_2) & \ldots & k(x_N, x_N) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix}$$

Or on new (test) data, we have:

$$y_{test} = K_{test}c \quad = \begin{bmatrix} k(x_{test1}, x_1) & k(x_{test1}, x_2) & \ldots & k(x_{test1}, x_N) \\ k(x_{test2}, x_1) & k(x_{test2}, x_2) & \ldots & k(x_{test2}, x_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_{Ntest}, x_1) & k(x_{Ntest}, x_2) & \ldots & k(x_{Ntest}, x_N) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix} \tag{3.1}$$

You can also see it written out this way:
(recalling that $k(x_1, x_2)$ is similarity of x_1 to x_2)

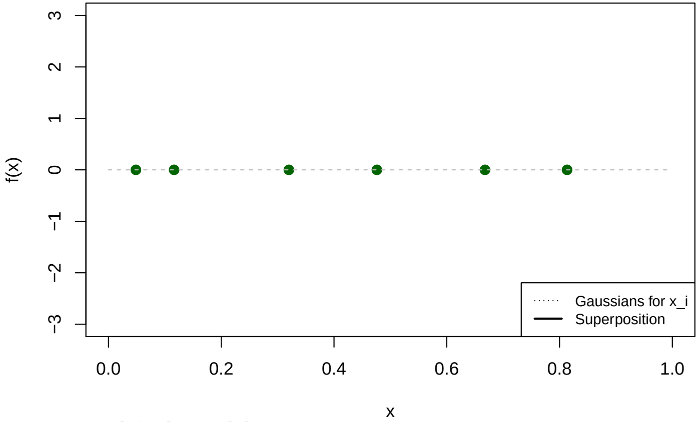
$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \ldots & k(x_1, x_N) \\ k(x_2, x_1) & k(x_2, x_2) & \ldots & k(x_2, x_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_N, x_1) & k(x_N, x_2) & \ldots & k(x_N, x_N) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix}$$

Or on new (test) data, we have:

$$y_{test} = K_{test} c \quad = \begin{bmatrix} k(x_{test1}, x_1) & k(x_{test1}, x_2) & \ldots & k(x_{test1}, x_N) \\ k(x_{test2}, x_1) & k(x_{test2}, x_2) & \ldots & k(x_{test2}, x_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_{Ntest}, x_1) & k(x_{Ntest}, x_2) & \ldots & k(x_{Ntest}, x_N) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix} \tag{3.1}$$

Intuition 2: Superposition of Gaussians

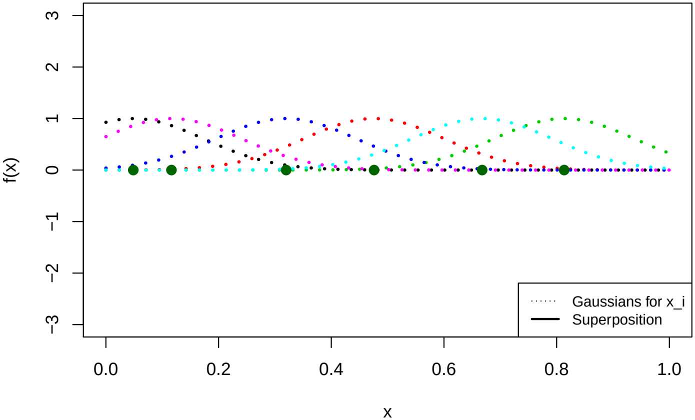
$$f(\cdot) = c_1 k(\cdot, x_1) + c_2 k(\cdot, x_2) + \ldots + c_N k(\cdot, x_N)$$



- Clarifies that $E[y|x^{\star}] \to E[y]$ for $x^{\star}$ far from training data.

Intuition 2: Superposition of Gaussians

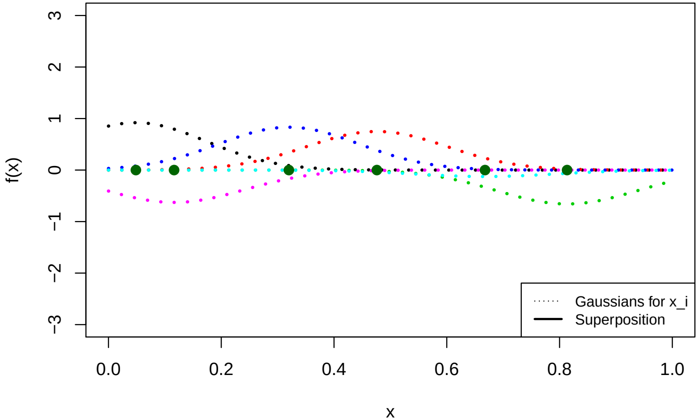
$$f(\cdot) = c_1 k(\cdot, x_1) + c_2 k(\cdot, x_2) + \ldots + c_N k(\cdot, x_N)$$



- Clarifies that $E[y|x^\star] \to E[y]$ for $x^\star$ far from training data.

Intuition 2: Superposition of Gaussians

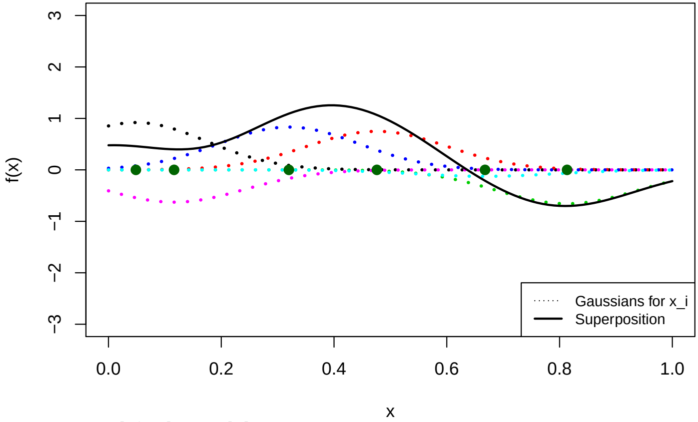
$$f(\cdot) = c_1 k(\cdot, x_1) + c_2 k(\cdot, x_2) + \ldots + c_N k(\cdot, x_N)$$



- Clarifies that $E[y|x^{\star}] \to E[y]$ for $x^{\star}$ far from training data.

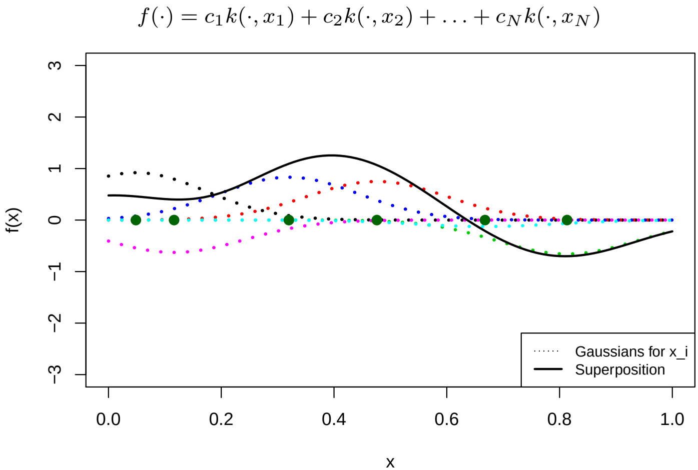
Intuition 2: Superposition of Gaussians

$$f(\cdot) = c_1 k(\cdot, x_1) + c_2 k(\cdot, x_2) + \ldots + c_N k(\cdot, x_N)$$



- Clarifies that $E[y|x^{\star}] \rightarrow E[y]$ for $x^{\star}$ far from training data.

Intuition 2: Superposition of Gaussians



- Clarifies that $E[y|x^{\star}] \rightarrow E[y]$ for $x^{\star}$ far from training data.

Other Choices

- Standardize data before analysis then transformed back
- λ is chosen by GCV
- σ^2 is chosen to be J.

  After standardizing, $E[||x_i - x_l||^2] = 2J$. Since $k(x_l, x_i) = e^{-\frac{||x_l - x_i||^2}{\sigma^2}}$, choosing $\sigma^2 \propto J$ ensures reasonable spread of similarities.

Other Choices

- Standardize data before analysis then transformed back
- λ is chosen by GCV
- σ^2 is chosen to be J.

  After standardizing, $E[||x_i - x_l||^2] = 2J$. Since $k(x_l, x_i) = e^{-\frac{||x_l - x_i||^2}{\sigma^2}}$, choosing $\sigma^2 \propto J$ ensures reasonable spread of similarities.

Other Choices

- Standardize data before analysis then transformed back
- λ is chosen by GCV
- σ^2 is chosen to be J.

  After standardizing, $E[||x_i - x_l||^2] = 2J$. Since $k(x_l, x_i) = e^{-\frac{||x_l - x_i||^2}{\sigma^2}}$, choosing $\sigma^2 \propto J$ ensures reasonable spread of similarities.

Quantities of Interest:

- $\hat{y}_i = E[y_i|x_i]$, for training or test points

- In KRLS, partial derivatives vary freely by point:
  Let $x^{(d)}$ be a particular variable. Then, for a single observation, j, we have:

$$\frac{\partial y}{\partial x_l^{(j)}} \approx \frac{-2}{\sigma^2} \sum_i c_i e^{\frac{-||x_i - x_l||^2}{\sigma^2}} (x_i^{(j)} - x_l^{(j)})$$

- Can summarize as you like
  - Scatter plots, regress on original X
  - Histograms
  - Sample average partial derivatives

$$E_N\left[\frac{\partial y}{\partial x_l^{(j)}}\right] \approx \frac{-2}{\sigma^2 N} \sum_l \sum_i c_i e^{\frac{-||x_i - x_l||^2}{\sigma^2}} (x_i^{(j)} - x_l^{(j)}).$$

Quantities of Interest:

- $\hat{y}_i = E[y_i|x_i]$, for training or test points

- In KRLS, partial derivatives vary freely by point:
Let $x^{(d)}$ be a particular variable. Then, for a single observation, j, we have:

$$\frac{\partial y}{\partial x_l^{(j)}} \approx \frac{-2}{\sigma^2} \sum_i c_i e^{\frac{-||x_i - x_l||^2}{\sigma^2}} (x_i^{(j)} - x_l^{(j)})$$

- Can summarize as you like
  - Scatter plots, regress on original X
  - Histograms
  - Sample average partial derivatives

$$E_N\left[\frac{\partial y}{\partial x_l^{(j)}}\right] \approx \frac{-2}{\sigma^2 N} \sum_l \sum_i c_i e^{\frac{-||x_i - x_l||^2}{\sigma^2}} (x_i^{(j)} - x_l^{(j)}).$$

Quantities of Interest:

- $\hat{y}_i = E[y_i|x_i]$, for training or test points
- In KRLS, partial derivatives vary freely by point:
  Let $x^{(d)}$ be a particular variable. Then, for a single observation, j, we have:

$$\frac{\partial y}{\partial x_l^{(j)}} \approx \frac{-2}{\sigma^2} \sum_i c_i e^{\frac{-||x_i - x_l||^2}{\sigma^2}} (x_i^{(j)} - x_l^{(j)})$$

- Can summarize as you like
  - Scatter plots, regress on original X
  - Histograms
  - Sample average partial derivatives

$$E_N \left[\frac{\partial y}{\partial x_l^{(j)}} \right] \approx \frac{-2}{\sigma^2 N} \sum_l \sum_i c_i e^{\frac{-||x_i - x_l||^2}{\sigma^2}} (x_i^{(j)} - x_l^{(j)}).$$

Quantities of Interest:

- $\hat{y}_i = E[y_i|x_i]$, for training or test points

- In KRLS, partial derivatives vary freely by point:
  Let $x^{(d)}$ be a particular variable. Then, for a single observation, j, we have:

$$\frac{\partial y}{\partial x_l^{(j)}} \approx \frac{-2}{\sigma^2} \sum_i c_i e^{\frac{-||x_i - x_l||^2}{\sigma^2}} (x_i^{(j)} - x_l^{(j)})$$

- Can summarize as you like
  - Scatter plots, regress on original X
  - Histograms
  - Sample average partial derivatives

$$E_N\left[\frac{\partial y}{\partial x_l^{(j)}}\right] \approx \frac{-2}{\sigma^2 N} \sum_l \sum_i c_i e^{\frac{-||x_i - x_l||^2}{\sigma^2}} (x_i^{(j)} - x_l^{(j)}).$$

Quantities of Interest:

- $\hat{y}_i = E[y_i|x_i]$, for training or test points

- In KRLS, partial derivatives vary freely by point:
  Let $x^{(d)}$ be a particular variable. Then, for a single observation, j, we have:

$$\frac{\partial y}{\partial x_l^{(j)}} \approx \frac{-2}{\sigma^2} \sum_i c_i e^{\frac{-||x_i - x_l||^2}{\sigma^2}} (x_i^{(j)} - x_l^{(j)})$$

- Can summarize as you like
  - Scatter plots, regress on original X
  - Histograms
  - Sample average partial derivatives

$$E_N \left[\frac{\partial y}{\partial x_l^{(j)}} \right] \approx \frac{-2}{\sigma^2 N} \sum_l \sum_i c_i e^{\frac{-||x_i - x_l||^2}{\sigma^2}} (x_i^{(j)} - x_l^{(j)}).$$

Quantities of Interest:

- $\hat{y}_i = E[y_i|x_i]$, for training or test points

- In KRLS, partial derivatives vary freely by point:
  Let $x^{(d)}$ be a particular variable. Then, for a single observation, j, we have:

$$\frac{\partial y}{\partial x_l^{(j)}} \approx \frac{-2}{\sigma^2} \sum_i c_i e^{\frac{-||x_i - x_l||^2}{\sigma^2}} (x_i^{(j)} - x_l^{(j)})$$

- Can summarize as you like
  - Scatter plots, regress on original X
  - Histograms
  - Sample average partial derivatives

$$E_N\left[\frac{\partial y}{\partial x_l^{(j)}}\right] \approx \frac{-2}{\sigma^2 N} \sum_l \sum_i c_i e^{\frac{-||x_i - x_l||^2}{\sigma^2}} (x_i^{(j)} - x_l^{(j)}).$$

Flexible Interactions: Golder/Brambor

- Brambor et al 2006 argues for multiplicative interaction terms
- Example from Golder 2006: *"short-coattails" hypothesis*:

  temporally-proximate presidential elections reduce the effective number of legislative parties if and only if the number of presidential candidates is sufficiently low.

- Model:

$$ElectoralParties = \beta_0 + \beta_1 Proximity + \beta_2 PresidentialCandidates + \beta_3 (Proximity \cdot PresidentialCandidates) + \beta_4 Controls + \epsilon$$

- Thus, model asserts:

$$\frac{\partial parties}{\partial proximity} = \beta_1 + \beta_3 PresidentialCandidates$$

Flexible Interactions: Golder/Brambor

- Brambor et al 2006 argues for multiplicative interaction terms
- Example from Golder 2006: *"short-coattails" hypothesis*:

  temporally-proximate presidential elections reduce the effective number of legislative parties if and only if the number of presidential candidates is sufficiently low.

- Model:

$$ElectoralParties = \beta_0 + \beta_1 Proximity + \beta_2 PresidentialCandidates + \beta_3(Proximity \cdot PresidentialCandidates) + \beta_4 Controls + \epsilon$$

- Thus, model asserts:

$$\frac{\partial parties}{\partial proximity} = \beta_1 + \beta_3 PresidentialCandidates$$

Flexible Interactions: Golder/Brambor

- Brambor et al 2006 argues for multiplicative interaction terms
- Example from Golder 2006: *"short-coattails" hypothesis*:

  temporally-proximate presidential elections reduce the effective number of legislative parties if and only if the number of presidential candidates is sufficiently low.

- Model:

$$ElectoralParties = \beta_0 + \beta_1 Proximity + \beta_2 PresidentialCandidates + \beta_3(Proximity \cdot PresidentialCandidates) + \beta_4 Controls + \epsilon$$

- Thus, model asserts:

$$\frac{\partial parties}{\partial proximity} = \beta_1 + \beta_3 PresidentialCandidates$$

Flexible Interactions: Golder/Brambor

- Brambor et al 2006 argues for multiplicative interaction terms
- Example from Golder 2006: *"short-coattails" hypothesis*:

  > *temporally-proximate presidential elections reduce the effective number of legislative parties if and only if the number of presidential candidates is sufficiently low.*

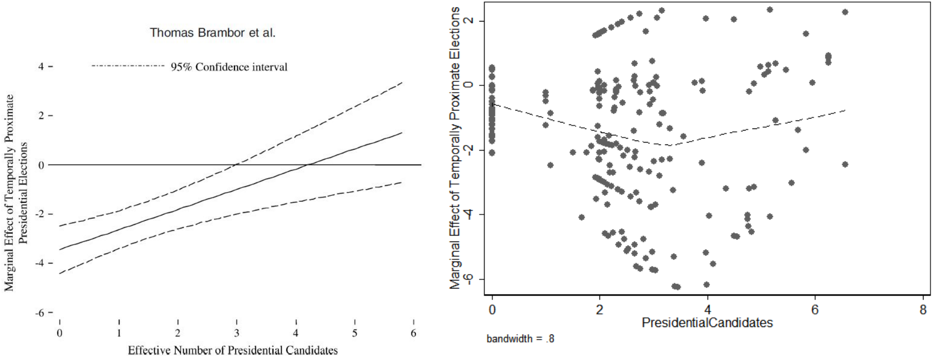
- Model:

$$
\begin{aligned}
ElectoralParties = {} & \beta_0 + \beta_1 Proximity + \beta_2 PresidentialCandidates + \\
& \beta_3 (Proximity \cdot PresidentialCandidates) + \beta_4 Controls + \epsilon
\end{aligned}
$$

- Thus, model asserts:

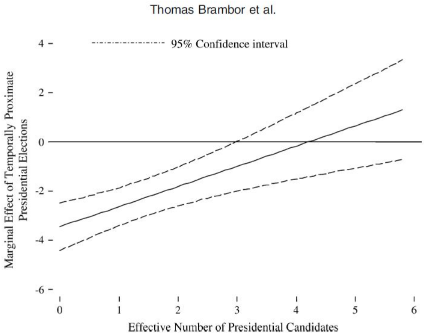
$$
\frac{\partial parties}{\partial proximity} = \beta_1 + \beta_3 PresidentialCandidates
$$

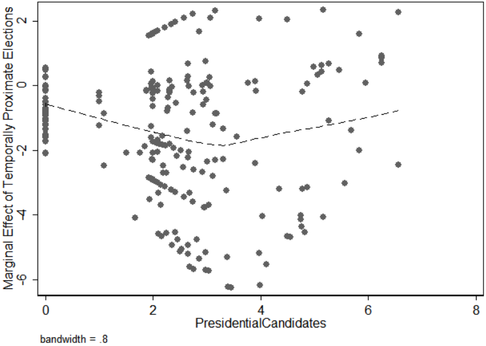
Flexible Interactions: Golder/Brambor



- *Left*: Figure from Brambor 2006.
- *Right*: scatterplot of KRLS estimates of $\frac{\partial parties}{\partial proximity}$. Agrees with the Brambor result only where pres. candidates > 2. At ≤ 2 (70% of the data), we see opposite effect.

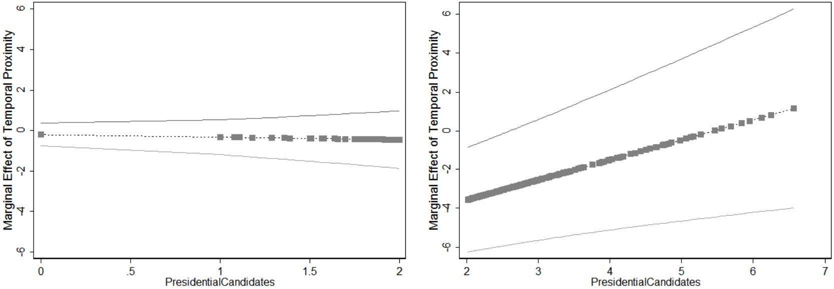
Flexible Interactions: Golder/Brambor





- *Left*: Figure from Brambor 2006.
- *Right*: scatterplot of KRLS estimates of $\frac{\partial parties}{\partial proximity}$. Agrees with the Brambor result only where pres. candidates > 2. At ≤ 2 (70% of the data), we see opposite effect.

Taking this insight back to OLS models:



- At ≤ 2 candidates, zero/opposite effect
- OLS results from > 2 candidates matches Brambor results closely

Conclusion

1) SVM: Classification Surfaces
2) Kernel Regression: A Flexible response surface
3) KRLS: Approach for estimating social science effects