

Kernel Ridge Regression

Prof. Bennett
Based on Chapter 2 of
Shawe-Taylor and Cristianini



Outline

- Overview
 - Ridge Regression
 - Kernel Ridge Regression
 - Other Kernels
 - Summary
- 




Recall E&K model

$$R(t) = at^2 + bt + c$$

Is linear in its parameters

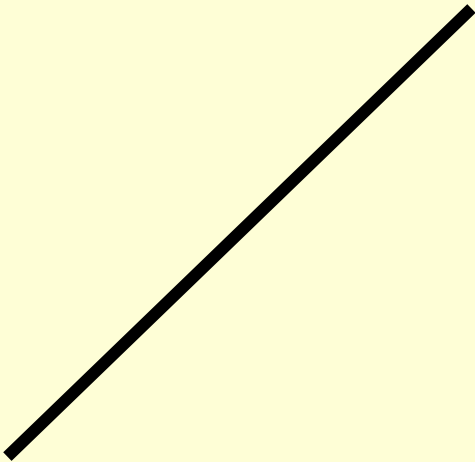
Define mapping $\theta(t)$ and make linear function in the $\theta(t)$ or feature space

$$\theta(t) = [t^2 \ t \ 1]' \quad s = [a \ b \ c]'$$

$$R(t) = \theta(t) \cdot s = [t^2 \ t \ 1]' \begin{bmatrix} a \\ b \\ c \end{bmatrix} = at^2 + bt + c$$


Linear versus Nonlinear

$$f(t)=bt+c$$



$$f(\theta(t))=at^2+bt+c$$





Kernel Method

Two parts (+):

- Mapping into embedding or feature space defined by kernel.
- Learning algorithm for discovering linear patterns in that space.

Illustrate using linear ridge regression.





Linear Regression in Feature Space

Key Idea:

Map data to higher dimensional space (feature space) and perform linear regression in embedded space.

Embedding Map:

$$\phi : \mathbf{x} \in R^n \rightarrow F \subseteq R^N \quad N \gg n$$



Nonlinear Regression in Feature Space

Input

$$\mathbf{x} = [r, s]$$

$$\langle \mathbf{x}, \mathbf{w} \rangle = w_1 r + w_2 s$$

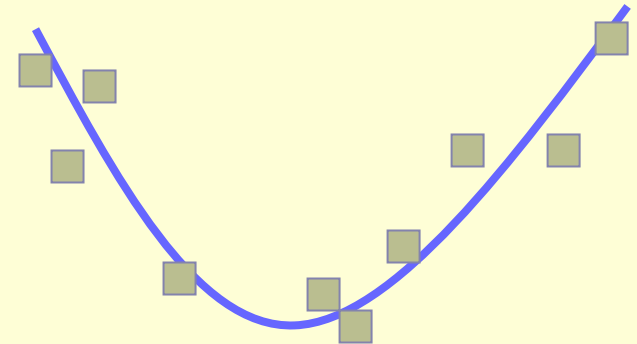
↓

Feature

$$\theta(\mathbf{x}) = [r^2, s^2, \sqrt{2}rs]$$

$$g(\mathbf{x}) = \langle \theta(\mathbf{x}), \mathbf{w} \rangle_F$$

$$= w_1 r^2 + w_2 s^2 + w_3 \sqrt{2}rs$$



Nonlinear Regression in Feature Space

Input

$$\mathbf{x} = [r, s]$$

$$\langle \mathbf{x}, \mathbf{w} \rangle = w_1 r + w_2 s$$

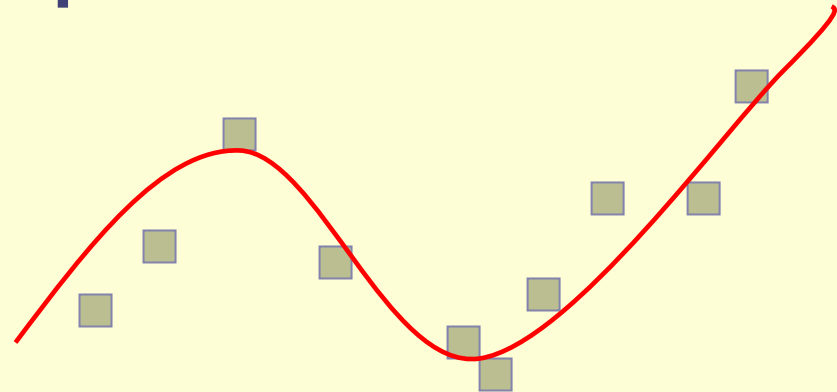
↓

Feature

$$\theta(\mathbf{x}) = [r^3, s^3, r^2, s^2, r^2 s, s^2 r, sr, s, r]$$

$$g(\mathbf{x}) = \langle \theta(\mathbf{x}), \mathbf{w} \rangle_F$$

$$= w_1 r^3 + w_2 s^3 + \dots + w_{10} \sqrt{2} r s$$





Let's try a quadratic on Aquasol

What are all the terms we need to add to our 525 dimensional space to map it into feature space?

Just do squared terms and cross terms.





Kernel and Duality to the rescue

- Duality

- Alternative but equivalent view of the problems

- Kernel trick – makes mapping to the feature space efficient






Linear Regression

Given training data:

$$S = \left((\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_i, y_i), \dots, (\mathbf{x}_\ell, y_\ell) \right)$$

points $\mathbf{x}_i \in R^n$ and labels $y_i \in R$

● Construct linear function:

$$g(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle = \mathbf{w}' \mathbf{x} = \sum_{i=1}^n w_i x_i$$


Least Squares Approximation

Want $g(x) \approx y$

Define error $f(\mathbf{x}, y) = y - g(\mathbf{x}) = \xi$

Minimize loss

$$L(g, s) = L(w, S) = \sum_{i=1}^{\ell} (y_i - g(\mathbf{x}_i))^2$$
$$= \sum_{i=1}^{\ell} \xi_i^2 = \sum_{i=1}^{\ell} l((\mathbf{x}_i, y_i), g)$$

Ridge Regression

- Use least norm solution for fixed $\lambda > 0$.

- **Regularized problem**

$$\min_{\mathbf{w}} L_{\lambda}(\mathbf{w}, S) = \lambda \|\mathbf{w}\|^2 + \|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2$$

- Optimality Condition:

$$\frac{\partial L_{\lambda}(\mathbf{w}, S)}{\partial \mathbf{w}} = 2\lambda \mathbf{w} - 2\mathbf{X}'\mathbf{y} + 2\mathbf{X}'\mathbf{X}\mathbf{w} = 0$$

$$(\mathbf{X}'\mathbf{X} + \lambda \mathbf{I}_n) \mathbf{w} = \mathbf{X}'\mathbf{y}$$

Requires $O(n^3)$ operations

Ridge Regression (cont)

- Inverse always exists for any $\lambda > 0$.

$$\mathbf{w} = (\mathbf{X}'\mathbf{X} + \lambda\mathbf{I})^{-1} \mathbf{X}'\mathbf{y}$$

- Alternative representation:

$$(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I})\mathbf{w} = \mathbf{X}'\mathbf{y} \Rightarrow \mathbf{w} = \lambda^{-1} (\mathbf{X}'\mathbf{y} - \mathbf{X}'\mathbf{X}\mathbf{w})$$

$$\Rightarrow \mathbf{w} = \lambda^{-1} \mathbf{X}'(\mathbf{y} - \mathbf{X}\mathbf{w}) = \mathbf{X}'\boldsymbol{\alpha}$$

$$\boldsymbol{\alpha} = \lambda^{-1} (\mathbf{y} - \mathbf{X}\mathbf{w})$$

Solving $l \times l$ equation is

Ridge Regression (cont)

- Inverse always exists for any $\lambda > 0$.

$$\mathbf{w} = (\mathbf{X}'\mathbf{X} + \lambda\mathbf{I})^{-1} \mathbf{X}'\mathbf{y}$$

- Alternative representation:

$$(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I})\mathbf{w} = \mathbf{X}'\mathbf{y} \Rightarrow \mathbf{w} = \lambda^{-1} (\mathbf{X}'\mathbf{y} - \mathbf{X}'\mathbf{X}\mathbf{w})$$

$$\Rightarrow \mathbf{w} = \lambda^{-1} \mathbf{X}'(\mathbf{y} - \mathbf{X}\mathbf{w}) = \mathbf{X}'\boldsymbol{\alpha}$$

$$\boldsymbol{\alpha} = \lambda^{-1} (\mathbf{y} - \mathbf{X}\mathbf{w})$$

$$\Rightarrow \lambda\boldsymbol{\alpha} = (\mathbf{y} - \mathbf{X}\mathbf{w}) = (\mathbf{y} - \mathbf{X}\mathbf{X}'\boldsymbol{\alpha})$$

$$\Rightarrow \mathbf{X}\mathbf{X}'\boldsymbol{\alpha} + \lambda\boldsymbol{\alpha} = \mathbf{y}$$

Solving $l \times l$ equation is

$$\Rightarrow \boldsymbol{\alpha} = (\mathbf{G} + \lambda\mathbf{I}_\ell)^{-1} \mathbf{y} \text{ where } \mathbf{G} = \mathbf{X}\mathbf{X}'$$

Ridge Regression (cont)

- Inverse always exists for any $\lambda > 0$.

$$\mathbf{w} = (\mathbf{X}'\mathbf{X} + \lambda\mathbf{I})^{-1} \mathbf{X}'\mathbf{y}$$

- Alternative representation:

$$(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I})\mathbf{w} = \mathbf{X}'\mathbf{y} \Rightarrow \mathbf{w} = \lambda^{-1} (\mathbf{X}'\mathbf{y} - \mathbf{X}'\mathbf{X}\mathbf{w})$$

$$\Rightarrow \mathbf{w} = \lambda^{-1} \mathbf{X}'(\mathbf{y} - \mathbf{X}\mathbf{w}) = \mathbf{X}'\boldsymbol{\alpha}$$

$$\boldsymbol{\alpha} = \lambda^{-1} (\mathbf{y} - \mathbf{X}\mathbf{w})$$

$$\Rightarrow \lambda\boldsymbol{\alpha} = (\mathbf{y} - \mathbf{X}\mathbf{w}) = (\mathbf{y} - \mathbf{X}\mathbf{X}'\boldsymbol{\alpha})$$

$$\Rightarrow \mathbf{X}\mathbf{X}'\boldsymbol{\alpha} + \lambda\boldsymbol{\alpha} = \mathbf{y}$$

Solving $l \times l$ equation is

$$\Rightarrow \boldsymbol{\alpha} = (\mathbf{G} + \lambda\mathbf{I}_\ell)^{-1} \mathbf{y} \text{ where } \mathbf{G} = \mathbf{X}\mathbf{X}'$$




Gram or Kernel Matrix

Gram Matrix

$$K = G = XX'$$

Composed of inner products of data

$$K_{i,j} = \langle x_i, x_j \rangle$$


Dual Ridge Regression

- To predict new point:

$$g(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle = \left\langle \sum_{i=1}^{\ell} \alpha_i \mathbf{x}_i, \mathbf{x} \right\rangle = \mathbf{y}' (\mathbf{G} + \lambda \mathbf{I})^{-1} \mathbf{z}$$

where $\mathbf{z}_i = \langle \mathbf{x}_i, \mathbf{x} \rangle$

- Note need only compute \mathbf{G} , the Gram Matrix $\mathbf{G} = \mathbf{X}\mathbf{X}'$ $G_{ij} = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$

Ridge Regression requires only
inner products between data points

Efficiency

- To compute

\mathbf{w} in primal ridge regression is $O(n^3)$

α in dual ridge regression is $O(l^3)$

- To predict new point \mathbf{x}

primal

$$g(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle = \sum_{i=1}^n w_i (\mathbf{x})_i \quad O(n)$$


dual

$$g(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle = \left\langle \sum_{i=1}^{\ell} \alpha_i \mathbf{x}_i, \mathbf{x} \right\rangle = \sum_{i=1}^{\ell} \alpha_i \left(\sum_{j=1}^n (\mathbf{x}_i)_j (\mathbf{x})_j \right) \quad O(nl)$$

Dual is better if $n \gg l$



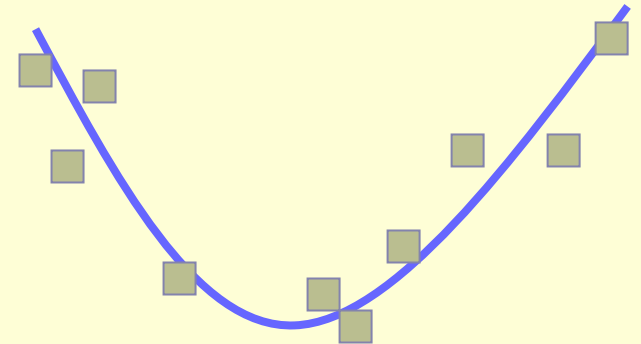
Notes on Ridge Regression

- “Regularization” is key to address stability and regularization.
 - Regularization lets method work when $n \gg p$.
 - Dual more efficient when $n \gg p$.
 - Dual only requires inner products of data.
- 

Nonlinear Regression in Feature Space

In dual representation:

$$\begin{aligned} g(\mathbf{x}) &= \langle \phi(\mathbf{x}), \mathbf{w} \rangle_F \\ &= \sum_{i=1}^{\ell} \alpha_i \langle \phi(\mathbf{x}), \phi(\mathbf{x}_i) \rangle \end{aligned}$$




So if we can efficiently compute inner product, our method is efficient



Let try it for our sample problem

$$\begin{aligned} & \langle \phi(\mathbf{u}), \phi(\mathbf{v}) \rangle \\ &= \left\langle (u_1^2, u_2^2, \sqrt{2}u_1u_2), (v_1^2, v_2^2, \sqrt{2}v_1v_2) \right\rangle \\ &= u_1^2v_1^2 + u_2^2v_2^2 + 2u_1u_2v_1v_2 \\ &= (u_1v_1 + u_2v_2)^2 \\ &= \langle \mathbf{u}, \mathbf{v} \rangle^2 \end{aligned}$$

Define: $K(\mathbf{u}, \mathbf{v}) = \langle \mathbf{u}, \mathbf{v} \rangle^2$






Kernel Function

- A kernel is a function K such that

$$K \langle \mathbf{x}, \mathbf{u} \rangle = \langle \phi(\mathbf{x}), \phi(\mathbf{u}) \rangle_F$$

where ϕ is a mapping from input space to feature space F .

- There are many possible kernels.
Simplest is linear kernel.

$$K \langle \mathbf{x}, \mathbf{u} \rangle = \langle \mathbf{x}, \mathbf{u} \rangle$$


Dual Ridge Regression

- To predict new point:

$$g(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle = \left\langle \sum_{i=1}^{\ell} \alpha_i \mathbf{x}_i, \mathbf{x} \right\rangle = \mathbf{y}' (\mathbf{G} + \lambda \mathbf{I})^{-1} \mathbf{z}$$

where $\mathbf{z}_i = \langle \mathbf{x}_i, \mathbf{x} \rangle$

- Note need only compute \mathbf{G} , the Gram Matrix $\mathbf{G} = \mathbf{X}\mathbf{X}'$ $G_{ij} = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$

Ridge Regression requires only
inner products between data points

Ridge Regression in Feature Space

● To predict new point:

$$g(\phi(\mathbf{x})) = \langle \mathbf{w}, \phi(\mathbf{x}) \rangle = \left\langle \sum_{i=1}^{\ell} \alpha_i \phi(\mathbf{x}_i), \phi(\mathbf{x}) \right\rangle = \mathbf{y}' (\mathbf{G} + \lambda \mathbf{I})^{-1} \mathbf{z}$$

where $\mathbf{z}_i = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}) \rangle$

● To compute the Gram Matrix

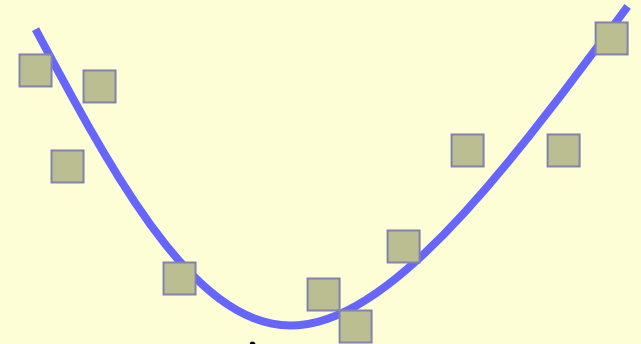
$$\mathbf{G} = \phi(\mathbf{X})\phi(\mathbf{X})' \quad G_{ij} = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle = K(\mathbf{x}_i, \mathbf{x}_j)$$

Use kernel to compute inner product

Nonlinear Regression in Feature Space

Kernel trick works for any dual representation:

$$\begin{aligned} g(\mathbf{x}) &= \langle \phi(\mathbf{x}), \mathbf{w} \rangle_F \\ &= \sum_{i=1}^{\ell} \alpha_i \langle \phi(\mathbf{x}), \phi(\mathbf{x}_i) \rangle \\ &= \sum_{i=1}^{\ell} \alpha_i K(\mathbf{x}, \mathbf{x}_i) \end{aligned}$$






Popular Kernels based on vectors

By Hilbert-Schmidt Kernels (Courant and Hilbert 1953)

$$\langle \theta(u), \theta(v) \rangle \equiv K(u, v)$$

for certain η and K , e.g.

$\theta(u)$	$K(u, v)$
Degree d polynomial	$(\langle u, v \rangle + 1)^d$
Radial Basis Function Machine	$\exp\left(-\frac{\ u - v\ ^2}{\sigma}\right)$
Two Layer Neural Network	$\text{sigmoid}(\eta \langle u, v \rangle + c)$





Kernels Intuition

- Kernels encode the notion of similarity to be used for a specific applications.
 - Document use cosine of “bags of text”.
 - Gene sequences can use edit distance.


- Similarity defines distance:

$$\| \mathbf{u} - \mathbf{v} \|^2 = (\mathbf{u} - \mathbf{v})'(\mathbf{u} - \mathbf{v}) = \langle \mathbf{u}, \mathbf{u} \rangle - 2\langle \mathbf{u}, \mathbf{v} \rangle + \langle \mathbf{v}, \mathbf{v} \rangle$$

- Trick is to get right encoding for domain.
- 




Important Points

- Kernel method =
linear method + embedding in feature space.
 - Kernel functions used to do embedding efficiently.
 - Feature space is higher dimensional space so must regularize.
 - Choose kernel appropriate to domain.
- 



Kernel Ridge Regression

- Simple to derive kernel method
 - Works great in practice with some finessing.
 - Next time:
 - Practical issues.
 - More standard dual derivation.
- 

Optimal Solution

Want: $\mathbf{y} \approx \mathbf{X}\mathbf{w} + b\mathbf{e}$ \mathbf{e} is a vector of ones

Mathematical Model:

$$\min_{\mathbf{w}} L(\mathbf{w}, b, S) = \|\mathbf{y} - (\mathbf{X}\mathbf{w} + \mathbf{e}b)\|^2 + \lambda \|\mathbf{w}\|^2$$

Optimality Conditions:

$$\frac{\partial L(\mathbf{w}, b, S)}{\partial \mathbf{w}} = 2\mathbf{X}'(\mathbf{y} - \mathbf{X}\mathbf{w} - \mathbf{e}b) + 2\lambda\mathbf{w} = 0$$

$$\frac{\partial L(\mathbf{w}, b, S)}{\partial b} = 2\mathbf{e}'(\mathbf{y} - \mathbf{X}\mathbf{w} - \mathbf{e}b) = 0$$



Let Try it: In class lab

Basic Ridge Regression $\lambda > 0$.

$$\min_{\mathbf{w}} L_{\lambda}(\mathbf{w}, S) = \lambda \|\mathbf{w}\|^2 + \|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2$$


• Optimality Condition:

$$\mathbf{w} = (\mathbf{X}'\mathbf{X} + \lambda\mathbf{I}_n)^{-1} \mathbf{X}'\mathbf{y}$$

• Dual equivalent

$$\boldsymbol{\alpha} = (\mathbf{G} + \lambda\mathbf{I})^{-1} \mathbf{y}$$

$$\text{where } G_{i,j} = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$$

$$\mathbf{w} = \mathbf{X}'\boldsymbol{\alpha}$$


Better model

Basic Ridge Regression $\lambda > 0$.

$$\min_{\mathbf{w}, b} L_{\lambda}(\mathbf{w}, S) = \lambda \|\mathbf{w}\|^2 + \|\mathbf{y} - (\mathbf{X}\mathbf{w} + \mathbf{e}b)\|^2$$

Center \mathbf{X} and \mathbf{Y} to get \mathbf{X}_c , \mathbf{Y}_c

$$\mathbf{w} = (\mathbf{X}_c' \mathbf{X}_c + \lambda \mathbf{I}_n)^{-1} \mathbf{X}_c' \mathbf{y}_c$$

Dual equivalent

$$\boldsymbol{\alpha} = (\mathbf{G} + \lambda \mathbf{I})^{-1} \mathbf{y}$$

$$\text{where } G_{i,j} = \langle \mathbf{x}_c_i, \mathbf{x}_c_j \rangle$$

$$\mathbf{w} = \mathbf{X}_c' \boldsymbol{\alpha}$$




Recenter Data

● Shift y by mean

$$\mu = \frac{1}{\ell} \sum_{i=1}^{\ell} y_i \quad yC_i := y_i - \mu$$

● Shift x by mean

$$\bar{\mathbf{x}} = \frac{1}{\ell} \sum_{i=1}^{\ell} \mathbf{x}_i \quad \mathbf{x}C_i := \mathbf{x}_i - \bar{\mathbf{x}}$$


Ridge Regression with bias

- Center data by \bar{x} and μ
 $\mathbf{X} = \mathbf{X}_c - \mathbf{e}\bar{x}'$ $y_c = y_c - \mu\mathbf{e}$

- Calculate w

$$\mathbf{w} = (\mathbf{X}_c'\mathbf{X}_c + \lambda\mathbf{I})^{-1}\mathbf{X}_c'y_c$$

- Calculate $b = \mu$

To predict new point

$$g(x) = (x - \bar{x})'w + b$$