Features again

To move beyond input features, many ways to combine them:

- polynomials
- histograms
- binary collections

Together, these methods take a feature vector $\vec{x}$ and *lift* it into a higher dimensional space $\vec{x} \mapsto \Phi(\vec{x})$

How large of a lift is needed to get good fits?

Which features should be used?

RIDGE REVISITED

$$\underset{\vec{w}}{\text{minimize}} \quad \|\bar{X}\vec{w} - \vec{y}\|^2 + \gamma\|\vec{w}\|^2$$

complete the square:

$$\|\bar{X}\vec{w} - \vec{y}\|^2 + \gamma\|\vec{w}\|^2$$

$$= \vec{w}^T(\bar{X}^T\bar{X})w - 2\vec{w}^T(\bar{X}^T\vec{y}) + \gamma\vec{w}^T\vec{w} + \vec{y}^T\vec{y}$$

$$= \vec{w}(\bar{X}^T\bar{X} + \gamma I)w - 2\vec{w}(\bar{X}^T\vec{y}) + \vec{y}^T\vec{y}$$

$$= (\vec{w} - \vec{w}_\star)^T(\bar{X}^T\bar{X} + \gamma I)(\vec{w} - \vec{w}_\star)$$
$$- \vec{y}^T(\bar{X}(\bar{X}^T\bar{X} + \gamma I)^{-1}\bar{X}^T)\vec{y} + \vec{y}^T\vec{y}$$

where $\vec{w}_\star = (\bar{X}^T\bar{X} + \gamma I)^{-1}\bar{X}^T\vec{y}$ is the minimizer.

Note:

$$\vec{w}_\star = (\bar{X}^T\bar{X} + \gamma I_d)^{-1}\bar{X}^T\vec{y} = \bar{X}(\bar{X}\bar{X}^T + \gamma I_n)^{-1}\vec{y}$$

PROOF: (algebra)

In particular

$$\vec{w}_{RIDGE} = \sum_{i=1}^{n} c_i\vec{x}_i, \qquad \vec{c} = (\bar{X}\bar{X}^T + \gamma I)^{-1}\vec{y}$$

$$[\bar{X}\bar{X}^T]_{ij} = \vec{x}_i^T\vec{x}_j$$

$\bar{X}\bar{X}^T$ is called the <u>Gram matrix</u> of the data

For evaluation: Let $\vec{x}$ be a new data point:

$$\vec{w}_*^T \vec{x} = \sum_{i=1}^{n} c_i (\vec{x}_i^T \vec{x})$$

To compute $\vec{w}_*$ and evaluate predictions, only need to compute inner products.

Kernel "trick" never form feature vector $\Phi(\vec{x})$, but instead only use pairwise function $k(\vec{x}, \vec{z}) = \Phi(\vec{x})^T \Phi(\vec{z})$

k is called a "kernel function"

Kernels and Inner Products

Any positive semidefinite matrix $\bar{G}$ is a matrix of inner products:

$$\bar{G} = \bar{V}^T \bar{\Lambda} \bar{V} \qquad \bar{\Lambda} = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} \qquad \lambda_i \geq 0$$

$$\bar{V} = [\vec{v}_1, \ldots, \vec{v}_n]$$

Define $\vec{x}_i = \lambda_i^{1/2} \vec{v}_i$. Then $G_{ij} = \vec{x}_i^T \vec{x}_j$

Are there generic functions k s.t.

$K(\vec{x}_i, \vec{x}_j)$ form p.s.d. matrices for all $\{\vec{x}_i\}$?

Yes! Examples:

$k(\vec{x}, \vec{z}) = \vec{x}^T \vec{z}$

$k(\vec{x}, \vec{z}) = (1 + \vec{x}^T \vec{z})^p$ $\quad p$ an integer ≥ 1

$k(\vec{x}, \vec{z}) = \exp(-\alpha \|\vec{x} - \vec{z}\|^2)$

$k(\vec{x}, \vec{z}) = \exp(-\alpha \|\vec{x} - \vec{z}\|)$

Lifting: Suppose you have the explicit feature map $\Phi(x) = \begin{bmatrix} a_0 \\ a_1 x \\ a_2 x^2 \end{bmatrix}$

Then $\Phi(x)^T \Phi(z) = a_0^2 + a_1^2 xz + a_2^2 x^2 z^2$

Note that $(1+xz)^2 = 1 + 2xz + x^2 z^2$

So if $a_0 = 1$, $a_1 = \sqrt{2}$, $a_2 = 1$,

$k(x,z) = (1+xz)^2 = \Phi(\vec{x})^T \Phi(\vec{z})$

- Liftings and kernels are equivalent.
  - Every kernel has an associated lifting called the "feature space".

Gaussian Kernel has infinite dimensional feature space!

$$k(\vec{x},\vec{z}) = \exp\left(-\gamma \|\vec{x}-\vec{z}\|^2\right)$$

$$\propto \int \exp\left(-\frac{\|\vec{v}\|^2}{\gamma}\right) \exp\left(i\, \vec{v}^T (\vec{x}-\vec{z})\right)$$

$$= \int \exp\left(-\frac{\|\vec{v}\|^2}{\gamma}\right) \left[\cos(\vec{v}^T\vec{x}) \cos(\vec{v}^T\vec{z}) + \sin(\vec{v}^T\vec{x}) \sin(\vec{v}^T\vec{z})\right]$$

Representer Theorem

Consider the general problem

$$\underset{\vec{w}}{\text{minimize}} \quad \frac{1}{n}\sum_{i=1}^{n} \text{loss}(\vec{w}^T\vec{x}_i, y_i) + \gamma\|\vec{w}\|^2$$

By fundamental theorem of linear algebra we can always write

$$\vec{w} = \sum_{i=1}^{n} c_i \vec{x}_i + \vec{u} \qquad \text{w/ } \langle \vec{u}, \vec{x}_i \rangle = 0 \ \forall i$$

Plug in this form

$$\underset{\vec{c},\,\vec{u}}{\text{minimize}} \quad \frac{1}{n}\sum_{i=1}^{n} \text{loss}\left(\sum_{j=1}^{n} c_j (\vec{x}_j^T\vec{x}_i), y_i\right) + \gamma\left\|\sum_{i=1}^{n} c_i\vec{x}_i\right\|^2 + \gamma\|\vec{u}\|^2$$

Note that we only increase the cost if $\vec{u} \neq 0$.

Hence, we are left with the problem

$$\underset{\vec{c}}{\text{minimize}} \quad \frac{1}{n}\sum_{i=1}^{n} \text{loss}\left(\sum_{j=1}^{n} K_{ij} c_j, y_i\right) + \gamma\,\vec{c}^T K \vec{c}$$

Where $K = [\vec{x}_i^T\vec{x}_j]$ is the kernel matrix.

(a.k.a. the Gram matrix.)

Notes: Optimization problem only has n parameters no matter how large dimension.

Also,

$$\vec{w}_*^T \vec{x} = \sum_{i=1}^{n} c_i \left(\vec{x}_i^T \vec{x} \right)$$

so solution can be evaluated only with dot products.

Kernel Trick: replace $\vec{x}_i^T \vec{x}_j$ w/ $k(\vec{x}_i, \vec{x}_j)$ for appropriate kernel function.

$$\underset{\vec{c}}{\text{minimize}} \quad \frac{1}{n} \sum_{i=1}^{n} \text{loss}\left([\bar{K}\vec{c}]_i , y_i \right) + \gamma \vec{c}^T \bar{K} \vec{c}$$

$$K_{ij} = k(\vec{x}_i, \vec{x}_j)$$

Test: $f(\vec{x}) = \sum_{i=1}^{n} c_i \, k(\vec{x}_i, x)$

Kernels:

Pros:
- Never need more than n parameters
- Can engineer nonlinearity with minimal added complexity

Cons:
- Need to store entire data set for testing
- Kernel matrix of size $n \times n$ can be very large

For any $\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n$

$$G_{ij} = k(\vec{x}_i, \vec{x}_j) \Rightarrow G \text{ is p.s.d}$$

FACT

k_1, k_2 are kernels, $a > 0$

$\rightarrow$ $ak_1 + k_2$ is a kernel

$\rightarrow$ $k_1 \cdot k_2$ is a kernel

(**) G, H positive definite

$$K_{ij} = G_{ij} H_{ij} \Rightarrow K \text{ p.d.}$$