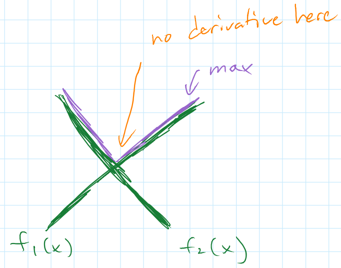
Converting to constrained problem

- non-smooth optimization is hard
- smooth constrained optimization often simpler

Trick to convert (1 of many):

If we have $\min_x \max_x \{f_i(x)\}$ … $\max\{x, -x\} = |x|$



$\Updownarrow$ equivalent

$\min_{x,v} v$, subject to $v \geq \max_i \{f_i(x)\}$ (one value)

$\Updownarrow$

$v \geq f_i(x), \forall_i$ (bigger than funtions individually)

Change of Basis

Original data

$$\begin{array}{c} \text{template} \\ \begin{bmatrix} 1 & x & x^2 \end{bmatrix} \end{array}$$

$$x = \begin{bmatrix} 1 \\ 0.5 \\ 2 \\ \vdots \\ 10 \end{bmatrix} \xrightarrow[\text{" } \Phi(X) \text{"}]{\text{make new X}} \underbrace{\begin{bmatrix} 1 & 1 & 1 \\ 1 & 0.5 & 0.25 \\ 1 & 2 & 4 \\ \vdots & \vdots & \vdots \\ 1 & 10 & 100 \end{bmatrix}}_{\text{new design matrix (then business as usual)}}$$

apply function Φ

$\bar{x}_i \rightsquigarrow \phi(\bar{x}_i)$

Cross Validation

- large 'k' : less chance of overestimating test error, but higher variance + running time
  - \# of folds
- idea of repeating CV with different random partitioning
  - lower variance

Ridge Regression

$$f(\bar{w}) = \overbrace{\frac{1}{2}\|X\bar{w} - Y\|^2}^{\text{log likelyhood}} + \overbrace{\frac{\lambda}{2}\|\bar{w}\|^2}^{\text{prior}}$$

lamda helps control change to X

what? shrink $\bar{w}_i$ towards 0

why? - solution is unique
- avoid overfitting due to huge values
- improving conditioning

"Magic": do a huge basis expansion (lets you use big design matrix without overfitting)

$\nabla f(\bar{w}) = X^T X\bar{w} - X^T y + \lambda w$

(set) $= 0$ (at the stationary point)

identity matrix to align dimensions

$(X^T X + \lambda I)\bar{w} = X^T Y$

$\boxed{\bar{w} = (X^T X + \lambda I) \backslash X^T Y}$ $\Longleftrightarrow$ matrix inversion lemma $\bar{w} = X^T(XX^T + \lambda I)^{-1} Y$

$\nabla^2 f(w) = \underbrace{X^T X + \lambda I}_{\text{solution is unique}} \succ 0$ positive definite

$X^T X$: "scatter matrix"

$1/N\ X^T X$: "covariance MLE"

$X X^T$: "Gram matrix"

MAP estimation

MLE: $\underset{h \in H}{\operatorname{argmax}}\ p(D|h)$ (hypothesis)

MAP: $\underset{h \in H}{\operatorname{argmax}}\ \underset{\text{posterior}}{p(h|D)} \propto \underset{\text{likelyhood}}{p(D|h)}\ \underset{\text{prior}}{p(h)}$ "is model h reasonable?"

proportional to. $f(x) \propto x$
$= cx$

$\Sigma p(x) = 1$ $f(x) \propto \frac{y^2 + \alpha^2}{\gamma} x$ still true

"exists constant, not dependent h, subject to $\sum_h p(h|D) = 1$"

$$\log p(h|D) = \log p(D|h) + \log p(h) + \text{const}$$

$$-\log p(h|D) = -\log p(D|h) - \log p(h) + \text{const}$$

$$= \sum_{i=1}^{N} (y_i - \bar{x}_i^T w)^2 + \underbrace{\frac{1}{2\sigma^2}\bar{w}^T\bar{w}}$$

"precision"

$$\lambda = \frac{1}{\sigma^2} \qquad \frac{\lambda}{2}\|\bar{w}\|^2$$

assumptions: gaussian prior

$$y_i | \bar{x}_i, \bar{w} \sim N(\bar{w}^T\bar{x}_i, 1)$$

$$w_i | \sigma^2 \sim N(\emptyset, \sigma^2)$$

mean of zero

$$p(w_i) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-1}{2}\frac{(w_i - \emptyset)^2}{\sigma^2}\right)$$

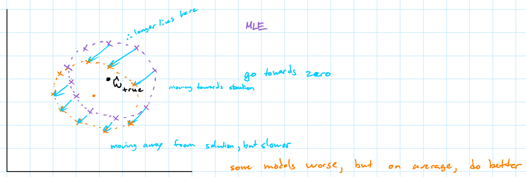
mean of zero

$$\sum_{i=1}^{d} w_i^2 = \|w\|^2$$

given σ^2

$$\log p(w_i) = -\frac{1}{2}\frac{(w_i)^2}{2\sigma^2} \underbrace{- \log(\sigma) - \log(\sqrt{2\pi})}_{\text{constant}}$$

stein's paradox



Kernels

We assume $\bar{x}_i \in \mathbb{R}^d$

what if we don't know representation?

Ex: "The cat is back"

"The black cat is back"

Longest common subsequence (cat is back)

Edit distance (∇ word to get from one to another)

"Kernel function"

$$k(\bar{x}_i, \bar{x}_j) \longmapsto \mathbb{R}$$ often a similarity

$x \in \mathcal{X}$

some set of all abstract space $\mathcal{X}$

"the cat is back" $\in \mathcal{X}$

Typically, $k(\bar{x}_i, \bar{x}_j) \geq 0$

$k(\bar{x}_i, \bar{x}_j) = k(\bar{x}_j, \bar{x}_i)$

Examples

vectors

"Linear" $\bar{x}_i^T\bar{x}_j$

"Poly" = $(\bar{x}_i^T\bar{x}_j + \alpha)^m$

"RBF" = $\exp\left(\frac{-\|\bar{x}_i - \bar{x}_j\|^2}{2\sigma^2}\right)$

section 14.2
- string kernels
- pyramid match kernel

How to use?

$$\phi(\bar{x}_i) = [k(\bar{x}_i, \bar{x}_1) \quad k(\bar{x}_i, \bar{x}_2) \quad \cdots \quad k(\bar{x}_i, \bar{x}_n)$$

$$[k(\bar{x}_i, z_1) \quad \cdots \quad k(\bar{x}_i, z_m)] \quad m < n$$

how?. possible course project

$$\Phi(X) = \begin{bmatrix} k(\bar{x}_1, \bar{x}_1) & \cdots & \\ \vdots & & \\ & & k(\bar{x}_n, \bar{x}_n) \end{bmatrix} \Rightarrow \text{Gram Matrix}$$

linear: $\Phi(X) = XX^T$

Kernel Trick

- Some kernels have an explicit feature map (inner product)

$$k(x, z) = (x^T z)^2$$

$$= (x_1 z_1 + x_2 z_2)^2$$

$$= x_1^2 z_1^2 + 2x_1x_2z_1z_2 + x_2^2 z_2^2$$

$$= (x_1^2, \sqrt{2}x_1x_2, x_2^2)^T (z_1^2, \sqrt{2}z_1z_2, z_2^2)$$

$$= \phi(x)^T \phi(z)$$ kernels between data points

$$\bar{w}_{MAP} = X^T(XX^T + \lambda I)^{-1} Y$$

$$f(X\bar{w}) + \frac{\lambda\|\bar{w}\|^2}{2}$$

anything written like this, can be kernelized

Test time

$$\hat{y} = \hat{X}\, \bar{w}_{MAP}$$

$$= \hat{X} X^T (\underbrace{XX^T}_{\text{gram matrix}} + \lambda I)^{-1} Y$$

$$k(\hat{X}, X) \qquad k(X, X)$$ kernel is n by n matrix

$$k(\hat{X}, X)\left(k(X,X) + \lambda I\right)^{-1} Y$$

"Gaussian Process"

Feature Selection

What if we don't know which features x are relevant

$$X = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

Search over features
- NP-hard
- 16, ok. 1 million? ☹