COMP 652 - Machine Learning
by Carlos G. Oliver, page 1 of 2

1 Math

Bayes rule: $P(A|B) = \frac{P(A,B)}{P(B)} = \frac{P(B|A)P(A)}{P(B)}$

Chain rule: $P(A_n,\dots,A_1) = P(A_n|A_{n-1},\dots,A_1)\cdot P(A_{n-1},\dots,A_1)$

Joint: $P\left(\bigcap_{k=1}^n A_k\right) = \prod_{k=1}^n P\left(A_k \,\middle|\, \bigcap_{j=1}^{k-1} A_j\right)$

Posterior: unobserved θ, observed x: $p(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)}$

Prior: prior belief in dist. of θ: $p(\theta)$

Likelihood: prob of observed given parameters: $p(x|\theta)$

MAP: $h_{MAP} = \arg\max_{h\in H} P(h|D) = \arg\max_{h\in H} P(D|h)P(h)$

Lagrange: $L(\mathbf{x},\lambda) = f(\mathbf{x}) + \lambda g(\mathbf{x})$ Set constraint $g(\mathbf{x})$ to zero and add multiplier.

$$\mathbf{xy^T} = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix}[x_1 \dots x_n] = \begin{bmatrix} x_1y_1 & \dots & x_1y_n \\ \vdots & \ddots & \vdots \\ x_my_1 & \dots & x_my_n \end{bmatrix}$$

$$\textbf{Matrix-vector product: } \mathbf{Ax} = \begin{bmatrix} a_1^T\mathbf{x} \\ \vdots \\ a_m^T\mathbf{x} \end{bmatrix}$$

Matrix mult': $(\mathbf{AB})_{ij} = \sum_{k=1}^m A_{ik}B_{kj}$

$$\mathbf{AB} = \begin{bmatrix} a_1^Tb_1 & a_1^Tb_2 & \dots & a_1^Tb_p \\ \vdots & \vdots & \ddots & \vdots \\ a_m^Tb_1 & a_m^Tb_2 & \dots & a_m^Tb_n \end{bmatrix}$$

$(n\times p)(p\times m) = n\times p$

2 todo

- Marginals
- Eigendecomposition
- $\sin(x)' = \cos x$
- $\cos(x)' = -\sin x$
-
- $(AB)^T = B^TA$
- $\log_b(x^y) = y\log_b(x)$
- Uniform distribution: $P_{\theta_1,\theta_2}(x) = \frac{1}{\theta_2-\theta_1}$
- $AA^{-1} = A^{-1}A = I$ for square matrices.
- $F'(x) = f'(g(x))g'(x)$
- $(f\cdot g)' = f'\cdot g + f\cdot g'$
- Log properties
- Convexity
- $||\mathbf{x}||_d = \sum_i x_i^d$
- Polynomial multiplication
- Matrix transpose and inverse tproperties
- Gradient vs partial derivative
- $\sigma(x)' = \sigma(x)(1-\sigma(x))$
- $E\left[\left(y-\hat{f}(x)\right)^2\right] = E\left[\hat{f}(x) - f(x)\right]^2 + E[\hat{f}(x)^2] - E[\hat{f}(x)]^2 + \epsilon^2$

3 Regression

3.1 L2 regularization

Weights do not reach zero. Faster.

$$J_w = \frac{1}{2}(\Phi\mathbf{w}-\mathbf{y})^T(\Phi\mathbf{w}-\mathbf{y}) + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w}$$

$$\mathbf{w} = (\Phi^T\Phi + \lambda\mathbf{I})^{-1}\Phi^Ty$$

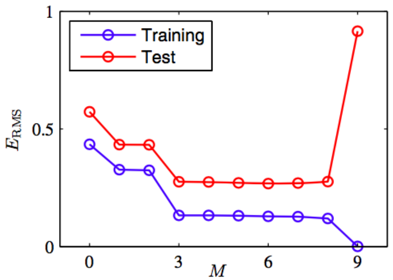
3.2 L1 regularization

Some weights set to zero. More expensive.

3.3 Gradient Descent

$\mathbf{w} \leftarrow \mathbf{w} - \alpha\nabla\log L(\mathbf{w})$ $\mathbf{w} \leftarrow \mathbf{w} - \alpha\nabla\log J(\mathbf{w})$

3.4 Bayesian regularization



4 Kernels

$k(\mathbf{x},\mathbf{z}) = \phi(\mathbf{x})\cdot\phi(\mathbf{z})$ **Mercer theorem:** $K(\mathbf{x},\mathbf{z})$ is kernel iff Gram matrix $\mathbf{K}$ symmetric and positive semidefinite: $\mathbf{K_{ij}} = \mathbf{K_{ji}}$ and $\mathbf{z}^T\mathbf{Kz} \geq 0$ $\mathbf{K} = \Phi\Phi^T$ where $\mathbf{K_{nm}} = \phi(\mathbf{x_n})^T\phi(\mathbf{x_m}) = k(\mathbf{x_n},\mathbf{x_m})$ Gram matrix size of input.

- Kernel properties
- Proving kernels

5 SVM

Minimize absolute error. More robust to outliers. Max margin is convex optimization.

$$h_{\mathbf{w}}(\mathbf{x}) = \text{sign}(\sum_{i=1}^m \alpha_i y_i(\mathbf{x_i x}) + \mathbf{w_0})$$

$\alpha_i > 0$ only for support vectors. Soft error SVM: $0 < \zeta \leq 1$ if inside margin. $\zeta > 1$ if misclassified. Total errors: $C\sum_i\zeta$. Large C higher variance.

6 EM/Active Learning/Missing Data