CPSC 340 Formula Sheet

Norms:

$$\|\mathbf{y}\|_2 = \sqrt{\sum_{i=1}^n y_i^2} \qquad \|\mathbf{y}\|_1 = \sum_{i=1}^n |y_i|$$

$$\|\mathbf{y}\|_0 = \sum_{i=1}^n (1 \text{ if } y_i \neq 0)$$

$$\|\mathbf{y}\|_\infty = \max(|y_1|, |y_2|, \ldots, |y_n|)$$

$$\|W\|_F = \sqrt{\sum_{j=1}^d \sum_{c=1}^k w_{jc}^2}$$

Supervised Learning

K-Nearest Neighbors

- Find k nearest values of $\mathbf{x}_i$ that are most similar to $\tilde{\mathbf{x}}_i$
- Use mode of corresponding y_i

Naive Bayes

$$P(y_i \mid \mathbf{x_i}) = \frac{P(\mathbf{x}_i \mid y_i)P(y_i)}{P(\mathbf{x}_i)} \propto P(\mathbf{x}_i \mid y_i)P(y_i)$$

Conditional Independence Assumption:

$$P(\mathbf{x}_i \mid y_i) \approx \prod_{j=1}^d P(x_{ij} \mid y_i)$$

Probability Assumption:

$$P(x_{ij} = k \mid y_i = c) = \frac{\#\text{ times } (y_i = c, x_{ij} = k)}{n}$$

Laplace Smoothing:

• Add β to numerator, and add 1 for each possible label to denominator.

Regression

Linear Regression:
$\hat{y}_i = \mathbf{w}^T \mathbf{x}$

Least Squares Objective:
$f(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^n (\mathbf{w}^T \mathbf{x}_i - y_i)^2$

Normal Equations:
$X^T X \mathbf{w} = X^T \mathbf{y}$

Huber Loss Approximation:

$$h(z) = \begin{cases} \frac{1}{2} z^2 & |z| < 1 \\ |z| - \frac{1}{2} & |z| > 1 \end{cases}$$

Log-sum-exp Approximation:
$\max_i \{z_i\} \approx \log\left(\sum_i \exp(z_i)\right)$

Gradient Descent:
$\mathbf{w}^{t+1} = \mathbf{w}^t - \alpha^t \nabla f(\mathbf{w}^t)$

General Polynomial Features ($d = 1$):

$$Z = \begin{bmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^p \\ 1 & x_2 & x_2^2 & \ldots & x_2^p \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^p \end{bmatrix}$$

Gaussian Radial Basis Functions:

$$g(\varepsilon) = \exp\left(-\frac{\varepsilon^2}{2\sigma^2}\right)$$

Gaussian Radial Basis Transform:

$$Z = \begin{bmatrix} g(\|\mathbf{x}_1 - \mathbf{x}_1\|) & \ldots & g(\|\mathbf{x}_1 - \mathbf{x}_n\|) \\ g(\|\mathbf{x}_2 - \mathbf{x}_1\|) & \ldots & g(\|\mathbf{x}_2 - \mathbf{x}_n\|) \\ \vdots & \ddots & \vdots \\ g(\|\mathbf{x}_n - \mathbf{x}_1\|) & \ldots & g(\|\mathbf{x}_n - \mathbf{x}_n\|) \end{bmatrix}$$

Gram Matrix:
$K = XX^T$

Kernel Trick:
$\hat{\mathbf{y}} = \tilde{Z}\mathbf{v} = \tilde{Z}Z^T (ZZ^T + \lambda I)\mathbf{y} = \tilde{K}(K + \lambda I)\mathbf{y}$

Kernel Trick with Polynomials:
$K_{ij} = (1 + \mathbf{x}_i^T \mathbf{x}_j)^p \qquad \tilde{K}_{ij} = (1 + \tilde{\mathbf{x}}_i^T \mathbf{x}_j)^p$

Linear Classifiers

Binary Classification:

- Encode using $y_i \in \{-1, 1\}$
- Use $\text{sign}(\mathbf{w}^T \mathbf{x}_i)$ as prediction.

0-1 Loss Function (# of classification errors):
$f(\mathbf{w}) = \|\text{sign}(X\mathbf{w}) - \mathbf{y}\|_0$

Hinge Loss (convex upper bound on 0-1 loss):
$f(\mathbf{w}) = \sum_{i=1}^n \max\{0, 1 - y_i \mathbf{w}^T \mathbf{x}_i\}$

Support Vector Machine:
$f(\mathbf{w}) = \sum_{i=1}^n \max\{0, 1 - y_i \mathbf{w}^T \mathbf{x}_i\} + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$

Logistic Loss:
$f(\mathbf{w}) = \sum_{i=1}^n \log\left(1 + \exp(-y_i \mathbf{w}^T \mathbf{x}_i)\right)$

Softmax for class c:

$$P(y_i = c) = \frac{\exp(\mathbf{w}_c^T \mathbf{x}_i)}{\sum_{c=1}^k \exp(\mathbf{w}_c^T \mathbf{x}_i)}$$

Softmax Loss for classes y_i:

$$f(\mathbf{w}) = -\mathbf{w}_{y_i}^T \mathbf{x}_i + \log\left(\sum_{c=1}^k \exp(\mathbf{w}_c^T \mathbf{x}_i)\right)$$

MLE and MAP

Maximum Likelihood Estimation:
$\hat{\mathbf{w}} \in \underset{\mathbf{w}}{\text{argmax}}\{P(D \mid \mathbf{w})\}$

Minimizing Negative Log Likelihood to maximize likelihood:
$\hat{\mathbf{w}} \in \underset{\mathbf{w}}{\text{argmin}}\{-\log(P(D \mid \mathbf{w}))\}$

Maximum a Posteriori Estimation:
$\hat{\mathbf{w}} \in \underset{\mathbf{w}}{\text{argmax}}\{P(\mathbf{w} \mid D)\}$

Minimizing Negative Log Likelihood to maximize a posteriori:

$$\mathbf{w} \in \underset{\mathbf{w}}{\text{argmin}}\left\{-\sum_{i=1}^n \log(P(\mathbf{D}_i \mid \mathbf{w})) - \log(P(\mathbf{w}))\right\}$$

Matrix Factorization

$$X \approx ZW \qquad \mathbf{x}_i \approx W^T \mathbf{z}_i \qquad x_{ij} \approx (\mathbf{w}^j)^T \mathbf{z}_i$$

Principal Component Analysis

PCA Objective Function:
$f(W, Z) = \sum_{i=1}^n \left\|W^T \mathbf{z}_i - \mathbf{x}_i\right\|_2^2 = \|ZW - X\|_F^2$

Prediction:

- Center: replace each $\tilde{x}_{ij}$ with $(\tilde{x}_{ij} - \mu_j)$
- Find $\tilde{Z}$ minimizing squared error: $\tilde{Z} = \tilde{X}W^T (WW^T)^{-1}$

Gradients:
$\nabla f(W, Z_0) = Z^T ZW - Z^T X$
$\nabla f(W_0, Z) = ZWW^T - XW^T$

Variance Explained:

$$1 - \frac{\|ZW - X\|_F^2}{\|X\|_F^2}$$

Non-Negative Matrix Factorization

Require Z, W to have non-negative values.

Projected Gradient Algorithm:
$\mathbf{w}^{t+1/2} = \mathbf{w}^t - \alpha^t \nabla f(\mathbf{w}^t)$
$\mathbf{w}_j^{t+1} = \max\{0, \mathbf{w}_j\}$

Multi-Dimensional Scaling

Traditional MDS cost function:
$f(Z) = \sum_{i=1}^n \sum_{j=i+1}^n (\|\mathbf{z}_i - \mathbf{z}_j\| - \|\mathbf{x}_i - \mathbf{x}_j\|)^2$

Neural Networks

Objective function for one hidden layer:
$f(\mathbf{v}, W) = \frac{1}{2} \sum_{i=1}^n (\mathbf{v}^T \mathbf{h}(W\mathbf{x}_i) - y_i)^2$

Updated April 18, 2021
https://github.com/DonneyF/formula-sheets