1 Midterm Formulas - Dario Placencio

1.1 Learners

Learner	Hypothesis	Preference
Nearest Neighbor	Decomposition of space	Neighbors belong to same class
Decision Tree	Single feature, Axis parallels-splits	Identified by greedy search
Linear Regression	Linear function	Minimize squared error
Logistic Regression	Hyperplane Decision Bounderies	Lasso on Ridge can be used to prefer sparse on small weights

1.2 Concepts

- Discriminative Models: Ficys on predict the labels, given features.
- Generative Models: Focus on now the data was generated, with probabilistic approach.

1.3 Vector Norms

- L_1 norm: $\|x\|_1 = \sum_{i=1}^{n} |x_i|$
- L_2 norm: $\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$
- L_∞ norm: $\|x\|_\infty = \max_{i=1,\ldots,n} |x_i|$

1.4 Distances

- Hamming distance: $d(x, y) = \sum_{i=1}^{n} \mathbb{I}(x_i \neq y_i)$
- Euclidean distance: $d(x, y) = \|x - y\|_2 = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$
- Manhattan distance: $d(x, y) = \|x - y\|_1 = \sum_{i=1}^{n} |x_i - y_i|$

1.5 Probability

- Mean of a random variable: $\mathbb{E}[X] = \int_{-\infty}^{\infty} x f(x) dx$
- Variance of a random variable: $\text{Var}[X] = \mathbb{E}[(X - \mathbb{E}[X])^2]$
- Covariance of two random variables: $\text{Cov}[X, Y] = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])]$
- $P(A \cap B) = P(A)P(B|A)$
- $P(A \cup B) = P(A) + P(B) - P(A \cap B)$
- $P(A|B) = \frac{P(A \cap B)}{P(B)}$
- $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$
- $P(A|B) = \frac{P(B|A)P(A)}{P(B|A)P(A)+P(B|\bar{A})P(\bar{A})}$

1.6 Data Preprocessing

- Standardization: $x_i \leftarrow \frac{x_i - \mu_i}{\sigma_i}$
- Normalization: $x_i \leftarrow \frac{x_i - \min_i x_i}{\max_i x_i - \min_i x_i}$

1.7 Confusion Matrix

- True positive: $TP = \sum_{i=1}^{n} \mathbb{I}(y_i = 1, \hat{y}_i = 1)$
- False positive: $FP = \sum_{i=1}^{n} \mathbb{I}(y_i = 0, \hat{y}_i = 1)$
- True negative: $TN = \sum_{i=1}^{n} \mathbb{I}(y_i = 0, \hat{y}_i = 0)$
- False negative: $FN = \sum_{i=1}^{n} \mathbb{I}(y_i = 1, \hat{y}_i = 0)$
- Accuracy: $\frac{TP+TN}{TP+FP+TN+FN}$
- Precision: $\frac{TP}{TP+FP}$
- Recall: $\frac{TP}{TP+FN}$
- False positive rate: $\frac{FP}{FP+TN}$
- F1 score: $\frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$
- Confidence interval: $\hat{p} \pm z_{\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$

1.8 Decision Trees

- Entropy: $H(Y) = -\sum_{i=1}^{n} p_i \log_2 p_i$
- Joint entropy: $H(Y, X) = -\sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij} \log_2 p_{ij}$
- Conditional entropy: $H(Y|X) = -\sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij} \log_2 p_{j|i}$
- Mutual information: $I(Y; X) = H(Y) - H(Y|X)$
- Information gain: $IG(D, S) = H_D(Y) - H_D(Y|S)$ where D denotes empirical entropy and S denotes a split.
- Gain ratio: $GR(D, S) = \frac{IG(D,S)}{HD(S)} = \frac{IG(D,S)}{-\sum_{i=1}^{k} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}}$

1.9 Linear Regression

- $y = w^T x + b$
- $w = (X^T X)^{-1} X^T y$
- Gradient $w = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$
- $b = \bar{y} - w\bar{x}$
- Loss function: $\mathcal{L}(w, b) = \frac{1}{n}\sum_{i=1}^{n}(y_i - w^T x_i - b)^2$
- R2 score: $1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$
- Linear Regression as MLE, Gaussian Conditional Distribution $P(y|x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y - w^T x)^2}{2\sigma^2}\right)$

1.10 Ridge Regression

- $w = (X^T X + \lambda I)^{-1} X^T y$
- $w = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2 + \lambda}$
- Loss function: $\mathcal{L}(w, b) = \frac{1}{n}\sum_{i=1}^{n}(y_i - w^T x_i - b)^2 + \lambda\|w\|_2^2$

1.11 Lasso Regression

- $w = (X^T X + \lambda I)^{-1} X^T y$
- $w = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2 + \lambda}$
- Loss function: $\mathcal{L}(w, b) = \frac{1}{n}\sum_{i=1}^{n}(y_i - w^T x_i - b)^2 + \lambda\|w\|_1$

1.12 Polinomial Regression

- $y = w_0 + w_1 x + w_2 x^2 + \cdots + w_d x^d$
- $y = w^T \phi(x)$
- $w = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$
- Loss function: $\mathcal{L}(w, b) = \frac{1}{n} \sum_{i=1}^n (y_i - w^T \phi(x_i) - b)^2$

1.13 Logistic Regression

- $y = \sigma(w^T x + b)$
- Loss function: $-y \log \sigma(w^T x + b) - (1 - y) \log(1 - \sigma(w^T x + b))$
- $\sigma(z) = \frac{1}{1+e^{-z}}$
- $\sigma'(z) = \frac{e^{-z}}{(1+e^{-z})^2} = \frac{1}{1+e^{-z}} \cdot \frac{e^{-z}}{1+e^{-z}} = \sigma(z)(1 - \sigma(z))$
- $P(y_i|x_i) = \sigma(w^T x_i + b)^{y_i} (1 - \sigma(w^T x_i + b))^{1-y_i}$

1.14 Multiclass Logistic Regression

- Softmax function: $\sigma(z)_k = \frac{e^{z_k}}{\sum_{j=1}^K e^{z_j}} = \frac{e^{w_k^T x + b_k}}{\sum_{j=1}^K e^{w_j^T x + b_j}}$
- Cross Entropy Loss function: $\mathcal{L}(w, b) = -\frac{1}{n} \sum_{i=1}^n \sum_{k=1}^K \mathbb{I}(y_i = k) \log \frac{e^{w_k^T x_i + b_k}}{\sum_{j=1}^K e^{w_j^T x_i + b_j}}$ with n samples and K classes.
- KL Divergence: $D_{KL}(P||Q) = E_P[\log P] - E_P[\log Q] = \sum_{i=1}^n P(i) \log \frac{P(i)}{Q(i)}$ (The last for discrete distributions)

1.15 Gradient Descent

- $w^{(t+1)} = w^{(t)} - \eta \nabla \mathcal{L}(w^{(t)})$ Where η is the learning rate.
- For Linear Regression $\nabla \mathcal{L}(w) = \frac{2}{n} X^T (Xw - y)$
- For Logistic Regression $\nabla \mathcal{L}(w) = \frac{1}{n} X^T (\sigma(Xw) - y)$
- For Ridge Regression $\nabla \mathcal{L}(w) = \frac{2}{n} X^T (Xw - y) + 2\lambda w$
- For Lasso Regression $\nabla \mathcal{L}(w) = \frac{2}{n} X^T (Xw - y) + \lambda \text{sign}(w)$
- For Polinomial Regression $\nabla \mathcal{L}(w) = \frac{2}{n} \Phi^T (\Phi w - y)$
- Stochastic gradient descent: $w^{(t+1)} = w^{(t)} - \eta \nabla \mathcal{L}(w^{(t)}, x_i, y_i)$
- Lipschitzness $\|\nabla \mathcal{L}(w) - \nabla \mathcal{L}(w')\|_2 \leq L \|w - w'\|_2$

1.16 Maximum Likelihood Estimation

- $P(y|x) = \prod_{i=1}^n P(y_i|x_i)$
- $\log P(y|x) = \sum_{i=1}^n \log P(y_i|x_i)$
- $\log P(y|x) = \sum_{i=1}^n [y_i \log \sigma(w^T x_i + b) + (1 - y_i) \log(1 - \sigma(w^T x_i + b))]$

1.17 Maximum Posterior Estimation

- Goal: Find parameter θ that maximizes the posterior $P(\theta|y, x)$.
- Formula: $\hat{\theta}_{MAP} = \arg\max_\theta P(y|x; \theta) P(\theta)$.
- Combines likelihood $P(y|x; \theta)$ with prior $P(\theta)$.
- Different with MLE is the priori

1.18 Naïve Bayes

- Assumes independence of variables
- $P(X, Y) = P(Y) \prod_{i=1}^d P(X_i|Y)$
- Prediction $\hat{y} = \arg\max_Y P(Y|X)$
- Bernoulli $P(x_i|y) = \theta_{i|y}^{x_i} (1 - \theta_{i|y})^{1-x_i}$
- Multinomial $P(x_i|y) = \frac{N_{i|y}^{x_i}}{\sum_{j=1}^d N_{j|y}}$
- Gaussian $P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_{i|y}^2}} \exp\left(-\frac{(x_i - \mu_{i|y})^2}{2\sigma_{i|y}^2}\right)$
- Smoothing $P(x_i|y) = \frac{N_{i|y} + \alpha}{\sum_{j=1}^d N_{j|y} + \alpha d}$

1.19 Neural Networks

- Perceptron: $y = \sigma(w^T x + b)$
- Likelihood of Output: $P(y_i = 1|x_i) = \sigma(w^T x_i + b)$
- Perceptron Update Rule: $W_{t+1} = W_t + \eta(y_i - \hat{y}_i) x_i$
- Gradient Computation: $\nabla = \frac{\partial \mathcal{L}}{\partial w} = \frac{\partial \mathcal{L}}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial w}$

1.19.1 Activation Functions

- Threshold: $f(x) = \mathbb{I}(x > 0)$
- Sigmoid: $f(x) = \frac{1}{1+e^{-x}}$
- Tanh: $f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
- Relu: $f(x) = \max(0, x)$

1.19.2 Regularization

- Weight Decay: $\mathcal{L}(w) = \frac{1}{n} \sum_{i=1}^n (y_i - \sigma(w^T x_i + b))^2 + \lambda \|w\|_2^2$
- Dropout: $P(\text{keep}) = 1 - p$

1.19.3 Derivatives

- Sigmoid: $\sigma'(z) = \sigma(z)(1 - \sigma(z))$
- Tanh: $\tanh'(z) = 1 - \tanh^2(z)$
- Relu: $\text{relu}'(z) = \mathbb{I}(z > 0)$

1.20 Data Augmentation

- Transform and add new samples to data set.
- Images: Crop, Color, Rotations
- Text: Substitution, Bach Translation
- Adding Noise to pick a solution.
- Early stopping with validation printing.
- Droput probability of weight in testing.
- Convolution to reduce number of parameters.
- Padding to preserve edge information.