Summary of Example Problems

This is not an exhaustive list

- Chpater 1: Intro to machine learning
  - Determine if a problem can benefit from machine learning
  - Identify the type of problem (supervised vs. unsupervised, classification / regression)
  - Describe simple supervised learning methods
- Chapter 2: Supervised Learning
  - Describe a supervised classification problem. Identify classes, predictors and training data.
  - Compute the prediction error of a classifier on training data. Select parameters to minimize the prediction error for simple classifiers
  - Describe a "doubt" region for a classifier
  - Define the VC dimension and compute it for simple classes of classifiers
  - Describe the multiple classification problem
  - Describe the regression problem, the empirical error and minimize the empirical error for simple estimators
  - Describe the concepts of over-fitting, under-fitting, generalization and the process of cross-validation.
- Chapter 3: Bayesian detection theory
  - Describe a classification problem in a Bayesian setting.
  - Compute the posterior probability from the likelihood and prior using Bayes' rule
  - Describe a risk function
  - Compute the parameters of a classifier by minimizing a risk function
  - Compute the parameters of a classifier by minimizing an empirical risk function based on training data.
- Chapter 4: Parametric estimation
  - Write the likelihood function of data $x = (x_1, \ldots, x_n)$ given an unknown parameter θ
  - Compute the MLE by maximizing the likelihood or log likelihood (this involves a derivative)
  - Compute the bias and variance of an estimate
  - Describe the role of a prior on data
  - Compute the MAP estimate of a parameter given a likelihood and prior
- Chpater 5: Multivariable methods
  - Describe the linear model $y = X\beta + \epsilon$
  - Compute the least-squares solution: $\hat{\beta} = (X^T X)^{-1} X^T y$
  - Compute the bias and variance of the estimate
- Chapter 6: Dimensionality reduction
  - Perform subset selection using an error function and the forward selection algorithm
  - Compute the PC vectors from the sample covariance matrix and its eigenvectors
  - Compute the Proportion of variance (PoV)
  - Describe problems with PCA and motivate LDA

- Compute the LDA vectors
- Chapter 7: Clustering
  - Describe a mixture distribution, $p(x|z = i)$
  - Compute mean and variance of a mixture.
  - Compute the posterior probability of a component given the measurement
  - Perform k-means clustering
  - Compute the Q function for the EM algorithm on a mixture distribution and perform the E- and M- steps
- Chapter 8: Non-parametric Estimation
  - Compute the non-parametric estimate of a function or density given a kernel and data points.
  - Compute the interpolated value of a function or a probability from the non-parametric function / density estimate
  - Describe the tradeoffs in selecting bandwidth. Describe the k-NN bandwidth selection
- Chapter 10: Linear Discrimination
  - Describe a linear classifier for K=2 and K>2 classes. Draw the classification regions
  - Describe a logistic regression classifier. Compute the class probabilities given the weights.
  - Describe the likelihood function of the weights and the gradient of the likelihood for a logistic classifier.
- Chapter 11: Multilayer perceptrons (MLP)
  - Describe a single-layer perceptron and training of the perceptron. (this is very similar to logistic classifier)
  - Describe online vs. batch learning
  - Describe a MLP
  - Determine coefficients for an MLP to approximate simple functions
  - Compute the gradient of a MLP using backpropagation
- Chapter 13: Kernel methods
  - Write a quadratic optimization to find an optimal separable hyperplane, for separable data
  - Write an optimization for non separable data using slack variables
  - Write the conditions for optimality using a Lagrangian
  - Write the optimization in terms of a hinge loss
  - Rewrite the optimization using a kernel function

Chap 1: 1.1 – 1.5

Machine Learning & Applications
- Learning associations
- Classification
- Unsupervised learning & reinforcement learning

Chap 2: Supervised learning

- Learning classes from examples
- Empirical error rate

$$E(h|x) = \frac{1}{N} \sum_{t=1}^{N} 1\{h(x^t) \neq r^t\}$$

- Doubt
- VC dim
- PAC learning
- Noise
- Multiple classes. Total empirical classification error rate
- Regression
- Interpolation, extrapolation
- Model selection, generalization underfitting, overfitting
- Cross validation

Chap. 3. Bayesian Detection Theory

- Problem: Evidence x, unknown class $C = 1, \ldots, K$
- Want to estimate C from x
- Assume <u>known</u> prior $P(C=i)$ and likelihood $p(x \mid C=i)$.
- Compute posterior

$$P(C=i \mid x) = \frac{P(x \mid C=i) P(C=i)}{\sum_j P(x \mid C=j) P(C=j)}$$

- Classifier $\hat{C}$ = estimate of C given x
- Risk fn. λ_{ij} = cost of selecting $\hat{C} = i$ when $C = j$
- ~~Avg.~~ Expected risk:

$$R(\hat{C}) = \sum \lambda_{ij} P(\hat{C} = i \mid C = j)$$

- ~~Likelihood ratio tests~~: Discriminant fn.
  - ~~Binary classification~~
  - Select $\hat{C} = \arg\max_i g_i(x)$ ← discriminant fn.
  - Often $g_i(x) = \ln P(x \mid C=i) + \alpha_i$ ← bias
  - For binary classification, this is a likelihood ratio test

BEOC

2.10 Supervised Machine Learning Algorithm

↳ Generally

$$\mathcal{X} = \{x^t, r^t\}_{t=1}^{N}$$ Sample is iid

t: indexes sample

$p(x, r)$ is the joint distribution of all of the samples.

Aim: Build approximation to r^t using model $g(x^t | \theta)$

① model the learning

$g(x | \theta)$, $g(\cdot)$ is model
x is input
θ parameters

② Loss function

$$E(\theta | \mathcal{X}) = \sum L(r^t, g(x^t | \theta))$$

Class learning checks for equality
Regression - squared error

③ Optimization procedure

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\, E(\theta | \mathcal{X})$$

- In polynomial regression, closed f
- Other cases, not so clear & easy
- Global soln or locally optimal?

Chapter 4 See examples in class notes

Parametric methods : $x^t \sim p(x|\theta)$

- Max Likelihood Estimation
- $\hat{\theta} = \text{argmax } p(x|\theta)$

$= \text{argmax} \prod_t p(x^t|\theta)$ IID

$\equiv \text{argmax} \log(p(x|\theta))$

$= \sum_t \log p(x^t|\theta)$

- 4.21 Read Bernoulli Example

4.2.2. Multinomial Example

4.2.3. Gaussian Example

4.3 Bias & Variance

$\text{Bias} = E_x(\hat{\theta}(x)) - \theta$

$\text{MSE} = E_x[(\hat{\theta}(x) - \theta)^2]$

Formally

$E(\hat{\theta}(x) - \theta | \theta)$

$E((\hat{\theta}(x) - \theta)^2 | \theta$

Look over page 69 & 70 + eqn 4.11

$\text{MSE} = \text{Bias}^2 + \text{variance}$

4.4 Bayes/MMSE Estimator

See eqn 4.13

$\hat{\theta}_{Bayes/MMSE} = E(\theta|x) = \int \theta \, p(\theta|x) \, d\theta$

$\hat{\theta}_{MAP} = \text{argmax } p(\theta|x)$

$= \text{argmax } p(x|\theta) p(\theta)$

$= \text{argmax} (\log p(x|\theta) + \log p(\theta))$

· Look at example on top of page 73

4.5 Parametric Classification
Look through equations

4.6 Regression
Look through Equations
:
make sure you can come up with
$$w = (D^T D)^{-1} D^T r$$

4.7 + 4.8 Tuning model complexity
Important

Review page 87

Ch 5: See class notes: Multivariate Data
Very straight forward
Multivariate examples
- Gaussian - examples with different variances
- Gaussian with equal variances
- Gaussian with equal variances & priors
  - nearest mean
  - template matching

Table 5.1
Note covariances & Number of parameters

5.6 Tuning Complexity
- RDA

5.7 + 5.8 = Understand Concepts & equations

Chapter 6: Dimensionality Reduction

→ Data matrix X $n \times p$

- Want to reduce dimension p.
- Subset selection:
  - Pick $k \ll p$ components
    $z_0 = (x_{\sigma(1)}, \ldots, x_{\sigma(k)})$
  - Minimize Error fn. Forward selection
- PCA:
  - Computing PC vectors
  - Prop. of variance PoV
- PCA and factor analysis representation.
- Computing PCA via SVD
- LDA
  - Problems with PCA
  - Computing LDA vectors

Chapter 7 Problems

(1) Consider a mixture of exponentials

$$p(x|z=i) = \lambda_i^{-1} \exp(-x/\lambda_i), \quad x \geq 0$$

$$P(z=i) = \pi_i$$

(a) What is $E(x)$ and $var(x)$?

$$E(x) = \sum_i \pi_i E(x|z=i) = \sum \pi_i \lambda_i = \mu$$

$$var(x) = \sum \pi_i \, var(x|z=i) + \sum \pi_i (\lambda_i - \mu)^2$$

$$= \sum \pi_i \left\{ \lambda_i^2 + (\lambda_i - \mu)^2 \right\}$$

(b) What is $P(z=i|x)$?

Bayes' rule:

$$P(z=i|x) = \frac{P(x|z=i) P(z=i)}{\sum P(x|z=j) P(z=j)}$$

$$= \frac{\pi_i \lambda_i^{-1} e^{-x/\lambda_i}}{\sum \pi_j \lambda_j^{-1} e^{-x/\lambda_j}}$$

(2) <u>k-means</u> ~~Comp~~ Perform k-means on the following data

$(0,0)$, $(0,2)$, $(3,0)$

(a) Start with clusters $\mu_1 = (0,0)$ and $\mu_2 = (0,2)$

→ Membership: $C_1 = \{(0,0),(3,0)\}$ $C_2 = \{(0,2)\}$

→ Means $\mu_1 = (1.5, 0)$, $\mu_2 = (0,2)$

→ Membership: Same as before. Algorithm converges

(b) Start with cluster centers $\mu_1 = (0,0)$ $\mu_2 = (3,0)$

→ Mem: $C_1 = \{(0,0),(0,2)\}$ $C_2 = \{(3,0)\}$

→ Means: $\mu_1 = (0,1)$, $\mu_2 = (3,0)$

→ Mem: Sam as before. Alg. converges

(c) Which sol'n in (a) or (b) has lower total in-class variance?

The in-class variance is

$$J = \sum_k \sum_{i \in C_k} \|x_i - \mu_k\|^2$$

Fo.

For the sol'n in (a):

$$J = \|(0,0)-(1.5,0)\|^2 + \|(3,0)-(1.5,0)\|^2 + \|(0,2)-(0,2)\|^2$$

$$= 1.5^2 + 1.5^2 + 0 = 2\left(\frac{3}{2}\right)^2 = 9/2$$

For (b):

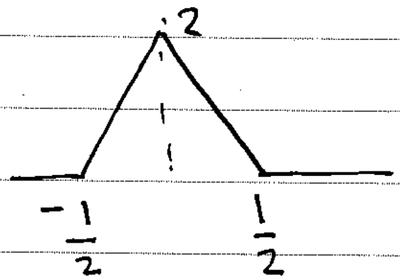
$$J = \|(0,0)-(0,1)\|^2 + \|(0,2)-(0,1)\|^2 + \|(3,0)-(3,0)\|^2$$

$$= 1 + 1 + 0 = 2$$

So (b) is a better sol'n.

Chapter 8 problem

(1) Consider a triangular kernel



$k(z)$

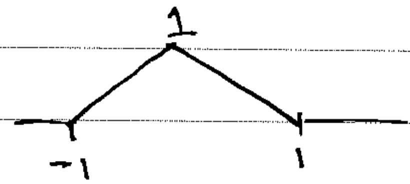
(a) Given data $X = \{0, 1, 2, 3\}$, draw the density estimate

$$\hat{p}(x) = \frac{1}{Nh} \sum_{i=1}^{N} k\left(\frac{x - x_i}{h}\right)$$

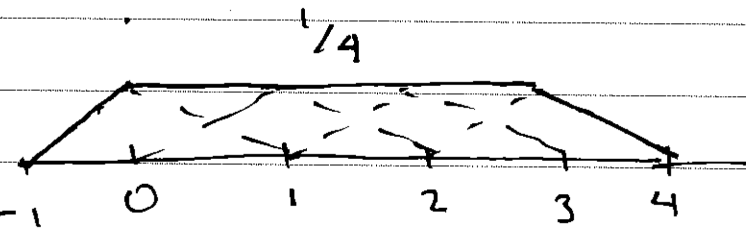
for $h = 2$.

This is a sum of overlapping triangles

$\frac{1}{h} k\left(\frac{z}{h}\right) \equiv$



$\hat{p}(x) =$



(b) What is $P(X \in [0, 2])$ given $\hat{p}(x)$

$$P(X \in [0,2]) = \int_0^2 \hat{p}(x)\,dx = \frac{1}{4}(2) = \cancel{2}\ \frac{1}{2}$$

Chapter 10 Problems

(1) Consider a general ~~binary~~ classification model where $r = 0$ or 1 and

$$\frac{P(r=1 \mid \bar{x}, \bar{w})}{P(r=0 \mid \bar{x}, \bar{w})} = g(\bar{w}^T \bar{x})$$

for some general fn $g(y)$.

(a) What is $P(r=1 \mid \bar{x}, \bar{w})$ and $P(r=0 \mid \bar{x}, \bar{w})$?

$$P(r=0 \mid \bar{x}, \bar{w}) = 1 - P(r=1 \mid \bar{x}, \bar{w})$$

$$\Rightarrow \frac{P(r=1)}{1 - P(r=1)} = g \Rightarrow P(r=1 \mid \bar{x}, \bar{w}) = \frac{g(\bar{w}^T \bar{x})}{1 + g(\bar{w}^T x)}$$

Similarly, $P(r=0 \mid \bar{x}, \bar{w}) = \dfrac{1}{1 + g(\bar{w}^T \bar{x})}$

(b) Given training data $(\bar{x}_i, r_i)$, What is the log likelihood

$$\mathcal{E}(\bar{w}) = \sum_{i=1}^{N} \ln p(r_i \mid \bar{x}_i, \bar{w})$$

We use the trick:

$$P(r_i \mid \bar{x}_i, \bar{w}) = \cancel{P(r_i = 1)} \begin{cases} \dfrac{g(y_i)}{1+g(y_i)} & r_i = 1 \\ \dfrac{1}{1+g(y_i)} & r_i = 0 \end{cases}$$

$$y_i = \bar{x}_i^T \bar{w}$$

$$\ln P(r_i \mid \bar{x}_i, \bar{w}) = r_i \ln g(y_i) - \ln(1 + g(y_i))$$

Hence $\mathcal{E}(\bar{w}) = \sum_{i=1}^{N} r_i \ln g(y_i) - \ln(1 + g(y_i))$

(c) What is the gradient $\nabla \mathcal{E}(\bar{w})$

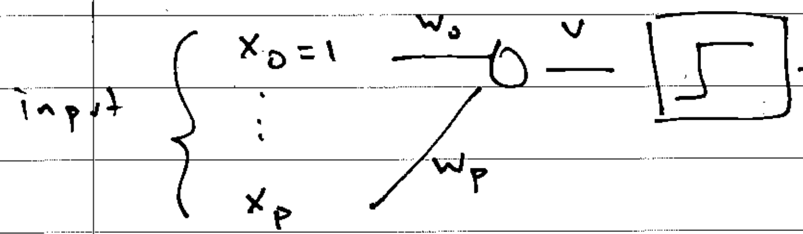
$$\frac{\partial \mathcal{E}(\bar{w})}{\partial w_j} = \sum_{i=1}^{N} \left[\frac{r_i}{g(y_i)} - \frac{1}{1+g(y_i)} \right] g'(y_i) \frac{\partial y_i}{\partial w_j}$$

$$= \sum_i \left[\frac{r_i}{g(y_i)} - \frac{1}{1+g(y_i)} \right] g'(y_i)\, x_{ij}$$

Chapter 11: Multi-Layer Perceptron

Single-layer

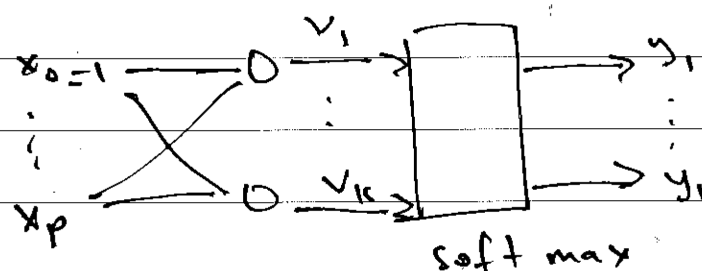
Binary case



$y = \text{sign}\{\sum w_i x_i\}$

= classification decision

Multi-class



$\bar{v} = \bar{W}\bar{x}$

$y_i = \dfrac{\exp(v_i)}{\sum \exp(v_i)}$ = "probability that $r_t = i$"

Training Single Layer

Given data $(\bar{x}_t^t, r_t^t)$ $t = 1, \ldots, N$

Loss fn:

$$\mathcal{E}(\bar{w}) = \sum_{t=1}^{N}\sum_{i=1}^{K} r_i^t \ln(y_i^t) = \text{cross-entropy}$$

$\Rightarrow$ I $\nabla \mathcal{E}(\bar{w})$ computed identically as Chapter 10.

Online training

→ Update $\bar{W}$ after each sample

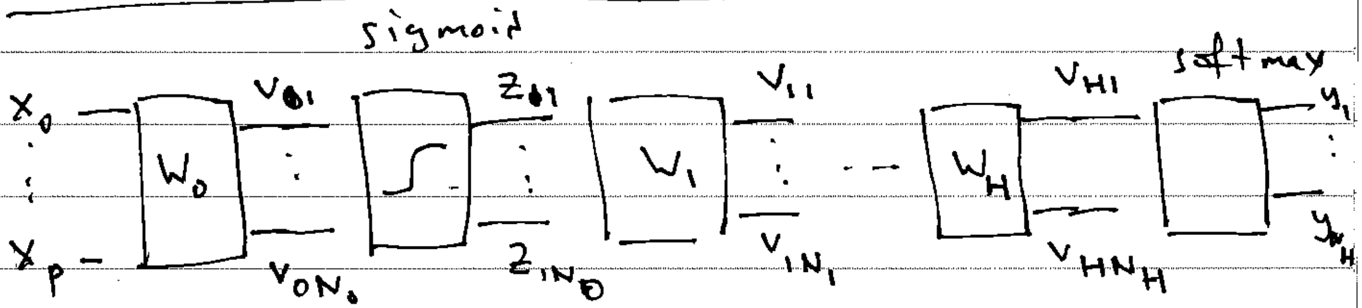
→ Continuous learning

$$\bar{W} \leftarrow \bar{W} - \eta^t \frac{\partial}{\partial \bar{W}} \left[\sum_{i=1}^{K} r_i^t \ln(y_i^t) \right]$$

→ Also called stochastic gradient descent since

$$\frac{\partial}{\partial W} \left[\sum_{i=1}^{K} r_i^t \ln(y_i^t) \right] = \nabla \mathcal{E}(\bar{w}) + \text{noise}$$

Multi-Layer Perceptron (MLP)



→ Can implement very general functions

→ Learn via backpropagation

(See HW4)

Chapter 13: Kernel methods / SVM

13.1 Optimal separating hyperplane

$w^T x^t + w_0 \geq 1$ when $r^t = 1$

$w^T x^t + w_0 \leq -1$ when $r^t = -1$

- min. $\|w\|^2$ s.t. $r^t(w^T x^t + w_0) \geq 1 \quad \forall t$
- Lagrangian formulation. Optimality conditions

13.2 Soft separation

- Data may not be linearly separable
- Min. slack

$$\min \frac{1}{2}\|w\|^2 + C\sum \xi^t \quad \text{s.t.} \quad r^t(w^T x^t + w_0) \geq 1 - \xi^t$$
$$\xi^t \geq 0$$

- Lagrangian formulation

13.5 Kernel methods

- Transform data $z = \phi(x)$.
- Classify using z.
- Rewrite optimization in terms of
  $K(x_1, x_2) = \phi(x_1)^T \phi(x_2)$ - Kernel
- Kernel examples