ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Introduction to Machine Learning

~~2010~~ 2009

Prof. J. M. Buhmann

Final Exam

February 10th, 2010

First and Last name: ______________________

ETH number: ______________________

Signature: ______________________

General Remarks

- Please check that you have all 21 pages of this exam.
- Remove all material from your desk which is not permitted by the examination regulations.
- Fill in your first and last name and your ETH number and sign the exam. Place your student ID in front of you.
- You have 2 hours for the exam. There are five questions, where you can earn a total of 150 points. Scoring 120 points (equivalent to solving four questions) guarantees you a grade of six.
- Write your answers directly on the exam sheets. If you need more space, put your name and ETH number on top of each supplementary sheet.
- Answer the questions in English. Do not use a pencil or red color pen.
- You may provide at most one valid answer per question. Invalid solutions must be canceled out clearly.

	Topic	Max. Points	Points Achieved	Visum
1	Density Estimation	30		
2	PCA	30		
3	Classifiers	30		
4	Regression	30		
5	Boosting	30		
Total		150		

Grade: ....

Question 1: Parametric Density Estimation (30 pts.)

When estimating densities from data, we assume that the data distribution is well approximated by a parametric model $P(x;\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ denotes the parameters of the model. Estimation of the density then reduces to estimation of the parameter $\boldsymbol{\theta}$.

a) In maximum-likelihood estimation (MLE), we usually start by assuming that the data is given as an i.i.d. sample $x_1, x_2, ..., x_N \sim P(x;\theta)$. Please explain how this assumption simplifies the calculations.

Answer: ① iid allows us to estimate a probability for each event independent from other giving a product

$$P(X|\theta) = \prod_{i=1}^{n} P(x_i|\theta) = P(x_1|\theta) \cdot P(x_2|\theta) \cdot P(x_3|\theta) \cdot \ldots \cdot P(x_n|\theta)$$

∴ This assumption would simplify to determine a mean and variance for the underlying distribution

2 pts.

b) We model a coin flip as a random variable X, where $X = 1$ denotes the outcome 'heads', and $X = 0$ denotes 'tails'. We model this event using the Bernoulli distribution, parametrized by μ that denotes $P(X = 1)$.

$$\text{Bern}(x|\mu) = \mu^x (1-\mu)^{1-x}.$$

Given a data set $\mathcal{X}_N = \{x_1, ..., x_N\}$ of N i.i.d. observations of X, show us step by step how to compute the MLE for μ. Comment your calculations where necessary.

$N - \sum x_i$ / $\sum(1-x_i)$

Answer: Likelihood: $P(X|\theta) = \prod_{i=1}^{N} \text{Bern}(x|\mu) = \prod_{i=1}^{N} \mu^{x_i}(1-\mu)^{1-x_i} \quad \left(= \mu^{\sum x_i}(1-\mu)\right)$

Take log of likelihood fxn:

$$\log P(X|\theta) = \log\left(\prod_{i=1}^{N} \text{Bern}(x_i|\mu)\right)$$
$$= \sum_{i=1}^{N} \log\left(\mu^{x_i}(1-\mu)^{1-x_i}\right)$$

$$P(\mathcal{X}|\mu) = \sum_{i=1}^{N}\left[\log\left[\mu^{x_i}\right] + \log\left[(1-\mu)^{1-x_i}\right]\right]$$

Take derivative wrt to μ. impose its equal.

$$\nabla_\mu \log P(\mathcal{X}|\mu) = \nabla_\mu \left[\sum_{i=1}^{N} x_i \log\mu + 1 - x_i \log(1-\mu)\right] \stackrel{!}{=} 0$$

$$= \sum_{i=1}^{N} x_i \cdot \frac{1}{\mu} + \frac{1-x_i}{1-\mu} \cdot -1 \Rightarrow \sum_{i=1}^{N} \frac{x_i}{\mu} - \sum_{i=1}^{N} \frac{1-x_i}{1-\mu} = 0$$

8 pts.

$$\sum_{i=1}^{N} \frac{x_i}{\mu} = \sum_{i=1}^{N} \frac{1-x_i}{1-\mu} \Rightarrow \frac{1}{\mu}\sum_{i=1}^{N} x_i = \frac{1}{1-\mu}\sum_{i=1}^{N}(1-x_i)$$

$$\Rightarrow \frac{1}{\mu}\sum_{i=1}^{N} x_i = \frac{1}{1-\mu}\left(\sum_{i=1}^{N} 1 - \sum_{i=1}^{N} x_i\right) \Rightarrow \frac{1}{\mu}\sum_{i=1}^{N} x_i = \frac{1}{1-\mu}\left(N - \sum_{i=1}^{N} x_i\right)$$

$$\Rightarrow \frac{1-\mu}{\mu} = \frac{N - \sum_{i=1}^{N} x_i}{\sum_{i=1}^{N} x_i} \qquad \frac{1}{\mu} - \cancel{\frac{\mu}{\mu}} = \frac{N}{\sum_{i=1}^{N} x_i} - \cancel{\frac{\sum_{i=1}^{N} x_i}{\sum_{i=1}^{N} x_i}}$$

$$\Rightarrow \frac{1}{\mu} - 1 = \frac{N}{\sum_{i=1}^{N} x_i} - 1 \Rightarrow \frac{1}{\mu} = \frac{N}{\sum_{i=1}^{N} x_i} - 1 + 1$$

$$\Rightarrow \boxed{\hat{\mu} = \frac{\sum_{i=1}^{N} x_i}{N}}$$

c) We flip our coin three times and obtain the dataset $\mathcal{X}_3 = \{1, 1, 1\}$.

1. What is the MLE for $\hat{\mu}$ for dataset $\mathcal{X}_3$? $N=3$

   Answer:

   $$\hat{\mu} = \frac{\sum_{i=1}^{3} x_i}{3} = \frac{1+1+1}{3} = 1$$

   $\hat{\mu} = 1$

2. State one problem with this estimate. What is a possible remedy (solution)?

   Answer:

   Problem. This estimate is ~~biased~~ not - $\hat{\mu} = 1$ versus $\hat{\mu} = \frac{1}{2}$. The MLE gave you a bias of $\hat{\mu} = 1$ unfair coin toss.

   High variance associated w/ a small population of values.

   Remedy: you more samples to get a better, less unfair estimate unless you take many more trials → solution

   $\hat{\mu}$

3 pts.

d) Assume that there are a total of m heads in $\mathcal{X}_N$, i.e., $\sum_{i=1}^{N} x_i = m$.

1. In terms of μ, N and m only, what is the probability of obtaining this dataset $P(\mathcal{X}_N|\mu)$? Explain the terms of your solution. coming from the same dataset $\mathcal{X}_N$

   Answer:

   $$P(\mathcal{X}_N|\mu) = \prod_{i=1}^{N} \mu^{x_i} (1-\mu)^{1-x_i} =$$

   $$\sum_{i}^{n} \log \left(\mu^{x_i} (1-\mu)^{1-x_i} \right)$$

   Distribute? $\sum x_i = m$ $\sum x_i = N$

   $$\log P(\mathcal{X}_N|\mu) = \sum_{i=1}^{N} \mu^{(x_i)} (1-\mu)^{\sum(1-x_i)} = \underbrace{\mu^{m}}_{\text{probability of getting exactly m heads}} \underbrace{(1-\mu)^{N-m}}_{\text{probability of getting exactly N-m tails}}$$

2. What is the expected value of m? (*Hint: For independent random variables, the expectation of a sum is the sum of expectations*)

   Answer:

   $$\mathbb{E}[m] = E\left[\sum_{i=1}^{N} x_i\right] = \sum_{i=1}^{N} E[x_i] = \sum_{i=1}^{N} \hat{\mu} = N \cdot \hat{\mu}$$

   $$\hat{\mu} = \frac{\sum_{i=1}^{N} x_i}{N} = \frac{m}{N}$$

   $m = \hat{\mu} \cdot n$

   $\boxed{\sum_{i}^{n} x_i = m}$

   Exponential propertyhat. probability

5 pts.

$$\sum_{i}^{n} \cancel{\log} \left(\sum \mu^{x_i} (1-\mu)^{1-x_i} \right) = \cancel{\log} \left(\mu^{\sum_{i}^{n} x_i} (1-\mu)^{\sum_{i}^{n}(1-x_i)} \right)$$

$$= \cancel{\log} \left(\mu^{m} (1-\mu)^{n-m} \right)$$

We now turn to Bayesian estimation of μ given a data set $\mathcal{X}$, by introducing a prior distribution $P(\mu)$. Here, we use the (conjugate) prior given by the Beta distribution:

$$\text{Beta}(\mu|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\mu^{\alpha-1}(1-\mu)^{\beta-1},$$

where $\Gamma(x)$ denotes the gamma function (not necessary). Some useful properties:

1. Mean $\mathbb{E}(\mu) = \frac{\alpha}{\alpha+\beta}$
2. Variance $\mathbb{V}(\mu) = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$

e) With the mean and variance of μ, we can understand how the parameters influence the prior. Select the best (α_0, β_0) pair for the prior that emphasizes the following beliefs.

✓ 1. My coin is biased towards 'tails'. more probability of tails low variance, high bias

☐(0.5, 0.5) ☒(1, 4) ☐(4, 1) ☐(2, 2)

✓ 2. My coin is most probably biased (but I don't know heads or tails)

☒(0.5, 0.5) ☐(1, 4) ☐(4, 1) ☐(2, 2) high-variance, low bias (not much info about the system)

3 pts.

f) Show that if the prior is Beta distributed

$$P(\mu) = \text{Beta}(\alpha_0, \beta_0),$$

then the posterior takes again the same functional dependency on μ as the prior, i.e.,

$$P(\mu|\mathcal{X}_N) \propto \mu^{\alpha_0+m-1}(1-\mu)^{\beta_0+N-m-1}. \quad P(\mu|X_N) = P(X_N|\mu)\,P(\mu)$$

Beta → prior likelihood → Bern

Answer:

$$\text{Beta}(\mu|\alpha_0\beta_0) = \frac{\cancel{\Gamma(\alpha_0+\beta_0)}}{\cancel{\Gamma(\alpha_0)\Gamma(\beta_0)}}\mu^{\alpha_0-1}(1-\mu)^{\beta_0-1}$$

$$\text{Bern}(X_N|\mu) = \mu^m(1-\mu)^{N-m}$$

$$P(\mu|X_n) = (\mu^m(1-\mu)^{N-m})(\mu^{\alpha_0-1}(1-\mu)^{\beta_0-1})$$

~~Fisher~~

$$\boxed{P(\mu|X_n) = \mu^{m+\alpha_0-1}(1-\mu)^{\beta_0+N-m-1}}$$ ✓

0.5, 0.5

mean. $\frac{0.5}{0.5+0.5} = \frac{\frac{1}{2}}{1} = \frac{1}{2}$

var. $\frac{0.25}{(1)^2(2)} = \frac{0.25}{2} = \frac{\frac{1}{4}}{2} = \frac{1}{8}$

2, 2

mean $\frac{2}{2+2} = \frac{2}{4} = \frac{1}{2}$

$\frac{4}{(4)^2(5)} = \frac{4}{16.5} = \frac{4}{16.5}$ = small

5 pts.

prior $= \frac{1}{2}$

g) In fact, the posterior is again a Beta distribution, and we obtain $P(X = 1|\mathcal{X}) = \frac{m+\alpha_0}{N+\alpha_0+\beta_0}$. $= \frac{1}{2}$

1. For $\mathcal{X}_3 = \{1, 1, 1\}$, use a choice of (α_0, β_0) which assumes a fair coin, and obtain your estimate for $P(X = 1|\mathcal{X})$. How does this estimate differ from the MLE?

   Answer:

   Fair coin: $\hat{\mu} = \frac{1}{2}$: $\frac{1}{2} = \frac{\alpha}{\alpha+\beta}$ $\quad \alpha+\beta = 2\alpha \quad \beta = 2\alpha - \alpha \Rightarrow \beta = \alpha \quad 1 = 1$

   $\sum_{i=1}^{N} x_i = m = 3$

   $P(X=1|\mathcal{X}) = \frac{3+\alpha_0}{3+\alpha_0+\beta_0}$

   $\alpha_0 = \beta_0 = 1$

   $P(X=1|\mathcal{X}) = \frac{m+1}{N+1+1} = \frac{m+1}{N+2} = \frac{3+1}{3+2} = \frac{4}{5} = 0.8$

   $1 - 0.8 - 0.5$

   MLE posterior prior

   MLE $\rightarrow \hat{\mu} = 1$ with the prior information of a fair coin, this tells us our estimate indicates the our MLE estimate is a pretty harsh biased coin. You can ~~Always~~ be sure that the MLE is ~~not~~ unbiased but to know if its truely biased you need a bayesian approach

2. What happens to the posterior as $N \to \infty$? How does this relate to the MLE?

   Answer:

   $\sum x_i$ gets larger, constants become insignificant

   $$\lim_{N\to\infty} \left[P(\mu|\mathcal{X}_n) = \mu^{m+\alpha_0-1} (1-\mu)^{\beta_0 + N - m - 1} \right]$$

   $N - \sum x_i$

   ✓

   $\uparrow N$

   $\rightsquigarrow \mu^{m}(1-\mu)^{N-m} = P(\mathcal{X}_n|\mu) \Rightarrow$ MLE

   $\hat{\mu} = \frac{m}{N}$

   $\uparrow N \uparrow m \uparrow$

   4 pts.

   As posterior $N \to \infty$, then becomes MLE equation

Question 2: Principal Component Analysis (PCA) (30 pts.)

Principal component analysis is a widely applied method in data analysis and dimensionality reduction. Given a dataset $\mathbf{X} \in \mathbb{R}^{D \times N}$ (observations as columns), where D is the number of dimensions and N is the number of observations, a linear transformation using an orthonormal matrix $\mathbf{P} \in \mathbb{R}^{M \times D}$ is applied to make a *change of basis*, to obtain a (usually) lower-dimensional dataset $\mathbf{Y} \in \mathbb{R}^{M \times N}$.

a) We begin by reviewing the steps of applying PCA to a dataset $\mathbf{X}$. Please complete each step below by providing the appropriate formula to compute the desired quantity.

1. Define the zero-mean dataset $\tilde{\mathbf{X}}$ in terms of the original dataset $\mathbf{X}$. For this purpose, introduce the data mean as a column-vector $\boldsymbol{\mu} \in \mathbb{R}^{D \times 1}$.
   Answer:

2. Define the covariance matrix $\mathbf{C}_{\tilde{\mathbf{X}}}$ in terms of the zero-mean dataset $\tilde{\mathbf{X}}$.
   Answer:

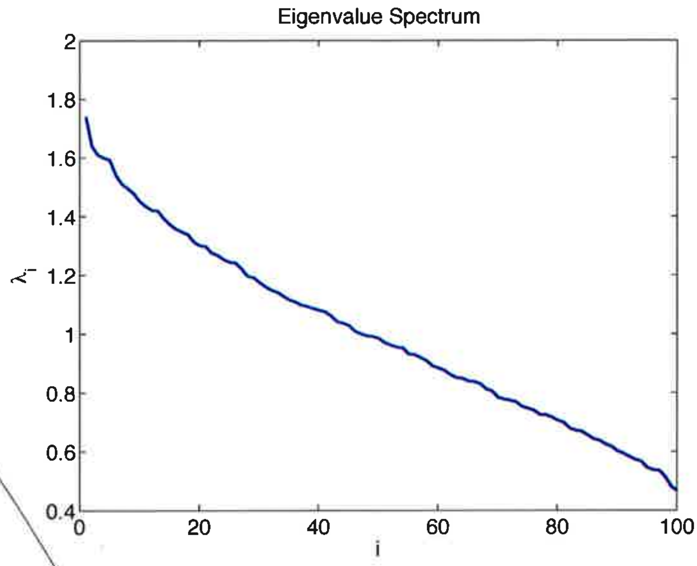
3. Write down the eigen-decomposition of the covariance matrix $\mathbf{C}_{\tilde{\mathbf{X}}}$, in terms of the eigenvector matrix $\mathbf{E}$ (eigenvectors as columns) and the diagonal matrix of eigenvalues $\mathbf{D}$.
   Answer:

4. Define the PCA transformation matrix $\mathbf{P}$ in terms of the eigenvector matrix $\mathbf{E}$. (Note: assume we don't want to reduce the dimensionality of the transformed dataset. Further assume that the eigenvectors in $\mathbf{E}$ have already been sorted according to the corresponding eigenvalues, in decreasing order.)
   Answer:

5. Define the transformed dataset $\mathbf{Y}$ in terms of the original dataset $\mathbf{X}$ and the transformation matrix $\mathbf{P}$.
   Answer:

6 pts.

b) Assume we have applied PCA to some dataset ($D = 100$). We observe the following eigenvalue spectrum of the covariance matrix of the data. (λ_i: eigenvalues)



1. Is the intrinsic dimensionality of this dataset low or high? Why?
   Answer:

2. Can this dataset be expressed in few dimensions with low approximation error? Why?
   Answer:

3. If yes, which dimensionality (approximately) should be chosen for the transformed dataset and why?
   Answer:

5 pts.

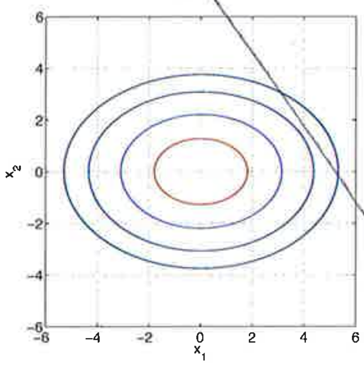
c) Assume you have observed 2D data $\mathbf{X} \in \mathbb{R}^{2\times N}$ (observations as columns). The first row of $\mathbf{X}$ corresponds to the first dimension x_1, the second row corresponds to x_2. For each of the three covariance matrices $\mathbf{C_X}$ below, please choose the iso-line plot (A-E) corresponding to the covariance matrix. (Note the axis labels on the figures)

1. $\begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$ Answer: ( )

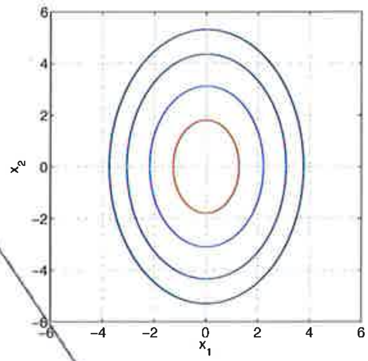
2. $\begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}$ Answer: ( )

3. $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ Answer: ( )

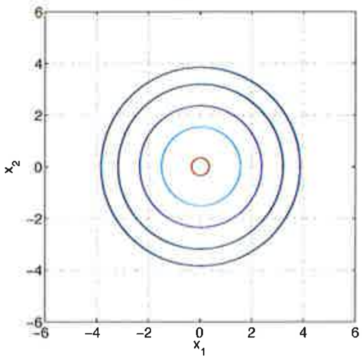
A



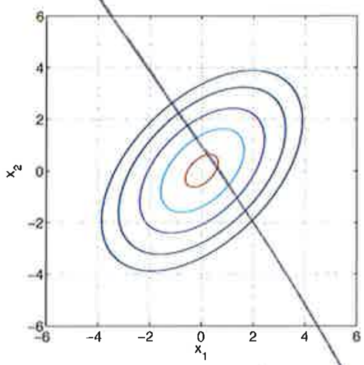
B



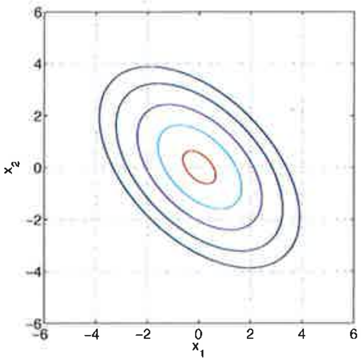
C



D



E



3 pts.

d) PCA can be derived as follows: Given a dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\} \in \mathbb{R}^{D \times N}$, find a new set of basis vectors such that the variance of the projected dataset is maximized. We begin by finding the first projection direction $\mathbf{e}_1 \in \mathbb{R}^D$, $||\mathbf{e}_1||_2 = 1$, which satisfies our goal.

1. Please describe in words why we are interested to find a projection direction such that the projected data has large variance.
   Answer:

2. Write down formally the objective function for maximizing the variance of the projected dataset. For this purpose, define the variance of the projected dataset in terms of the original data $\mathbf{x}_n$ and the first projection direction $\mathbf{e}_1$. (Hint: introduce the mean $\boldsymbol{\mu}$ of the data $\mathbf{X}$.)
   Answer:

5 pts.

e) PCA transforms a dataset $\mathbf{X}$ into a dataset $\mathbf{Y}$ by defining a new basis using the eigenvectors of the covariance matrix $\mathbf{C_X}$ of the dataset $\mathbf{X}$. With this particular choice of a new basis, the covariance matrix $\mathbf{C_Y}$ of the transformed dataset $\mathbf{Y}$ is *diagonalized*. (Note: For this exercise, assume a zero-mean dataset.)

1. Please explain in words, why we desire the covariance matrix of the transformed dataset to be diagonal.
   Answer:

2. Show that $\mathbf{C_Y} = \mathbf{P}\mathbf{C_X}\mathbf{P}^T$, i.e., that the covariance matrix $\mathbf{C_Y}$ of the transformed dataset can be written in terms of the covariance matrix $\mathbf{C_X}$ of the original dataset.
   Answer:

3. Show that the choice $\mathbf{P} = \mathbf{E}^T$ for the PCA transformation matrix, where $\mathbf{E}$ is the matrix of eigenvectors of the covariance matrix $\mathbf{C_X}$, actually diagonalizes the covariance matrix $\mathbf{C_Y}$ of the transformed dataset $\mathbf{Y}$. Use the eigen decomposition of $\mathbf{C_X}$ and the fact that $\mathbf{P}^{-1} = \mathbf{P}^T$.
   Answer:

11 pts.

Question 3: Linear Classifiers (30 pts.)

The simplest representation of a linear discriminant function is obtained by taking a linear function of the input vector $\boldsymbol{x}$ so that

$$y(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b$$

a) Define the term decision boundary. Explain what is the relation between the vector $\boldsymbol{w}$ and the decision boundary.

Answer:
Decision boundary: A hyperplane separateng different classes (linearly separaple). points). A decision boundary is always $\perp$ to the $\vec{w}$.

2 pts.

b) Consider the following set $S \subseteq \mathbb{R}^2 \times \{+1, -1\}$:

$$S = \{((3,3),+1),((1,4),+1),((2,1),-1),((5,1),-1),((6,2),+1),((4,3),+1),((4,1),-1)\}$$

Prove that there exist no homogeneous half-space that separates S with no errors (recall that a half-space is homogeneous if its separating hyperplane passes through the origin). Assume that the classification of a point is given by $y(\boldsymbol{x}) = \text{sign}(\boldsymbol{w}^T\boldsymbol{x} + b)$.

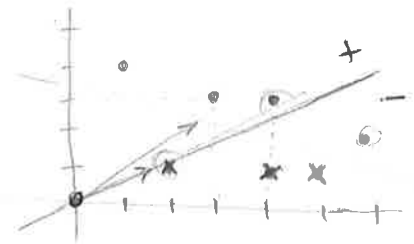
Answer:

3 points

X_1 ((2,1), -1)

X_2 ((4,3), +1)

X_3 ((6,2), +1)



$y(X_1) = (3,2)\begin{pmatrix}2\\1\end{pmatrix} + 0 = \text{sign}(6+2) = + \times$

$(3,2)\begin{pmatrix}4\\3\end{pmatrix} + 0 = \text{sign}(12+6) = + \checkmark$

5 pts.

The simple form of the online perceptron algorithm is defined as follows:

Initialize by setting $\boldsymbol{w}^{(0)} = \mathbf{0}$.

$w^T x$

$$\text{Update } \boldsymbol{w}^{(t+1)} = \begin{cases} \boldsymbol{w}^{(t)} & y_k(\boldsymbol{w}^{(t)^T}\boldsymbol{x}_k) > 0 \\ \boldsymbol{w}^{(t)} + y_k\boldsymbol{x}_k & \text{otherwise.} \end{cases}$$

c) We would like to use the perceptron algorithm to find a separating half-space that makes no errors on the set S. Recall that you have just proven that this is impossible using a homogeneous half-space. Make the necessary adjustments and perform 7 iterations of the perceptron algorithm. For each iteration write down the training example and the current vector $\boldsymbol{w}^{(t)}$. Assume that the points are given in the same order as they appear in S.
Answer:

necessary adjustment is homogeneous coordinates

	x_1	x_2	x_3 bias	ω_1	ω_2	ω_3	y_k	y_k	
0	3	3	1	0	0	0	1	0	✗
1	1	4	1	3	3	1	1	16	✓
2	2	1	1	3	3	1	-1	10	✗
3	5	1	1	1	2	0	-1	7	✗
4	6	2	1	-1	1	-1	1	-23	✗
5	4	3	1	2	3	0	1	17	✓
6	4	1	1	2	3	0	-1	11	✗
				-2	2	-1			

max $\frac{1}{\|w\|}$ → min $\|w\|$

5 pts.

Consider the primal form of the soft margin SVM

$$\min \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$
$$\text{s.t.} \quad y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 - \xi_i, \qquad i = 1,\ldots,n$$

maximize margin; Distance of two lines; regularization term; "regularization" term; Allow error in classification, mistakes, over training examples; "slack" variables; loss fxn

d) Indicate which of the following statements hold as we increase the value of the parameter C (relative to any starting value, $C > 0$). Use

C = ∞ enforces all constraints, hard margin

D - if the validity of the statement depends on the situation,
T - if the statement is necessary true
N - if the statement is never true

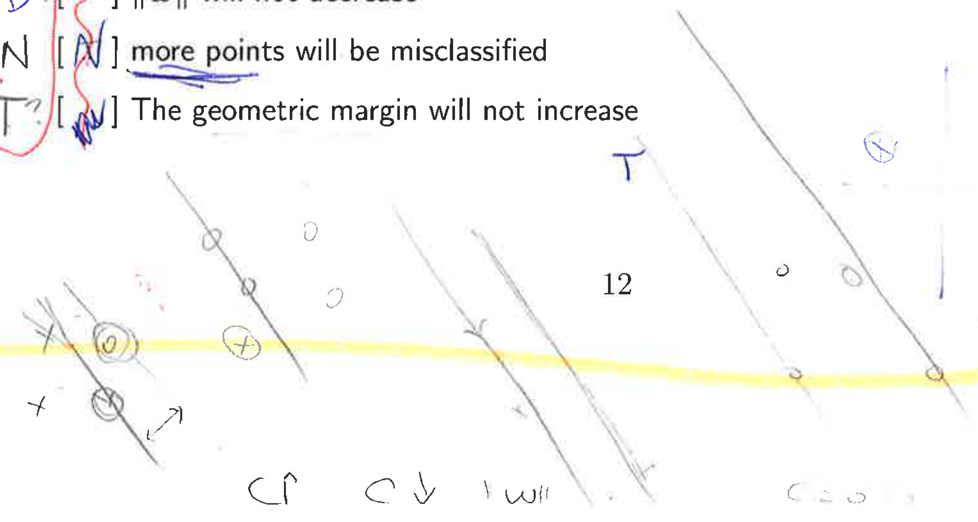
large C makes constraints hard to ignore → narrow margin
↑ C = 1000 → hard-margin SVM
C = 10 → wider margin
C = 0.1 → widest margin
small C, allows constraints to be easily ignored, large margin

EMAIL TA

D [T] b will not increase — Not in the obj fxn only in the constraint which is fixed.
D [N] $\|\boldsymbol{w}\|$ increases — minimizing $\frac{1}{2}\|w\|^2$
D [D] $\|\boldsymbol{w}\|$ will not decrease
N [N] more points will be misclassified
T? [N] The geometric margin will not increase

T

10 pts.



The dual soft margin SVM is given below

$$\max \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)$$
$$s.t. \quad 0 \le \alpha_i \le C, \qquad i = 1, \dots, n$$

We would like to solve the SVM problem using the kernel $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = 1 + (\boldsymbol{x}_i^T \boldsymbol{x}_j)^2$.

$z \in \mathbb{R}^2$ $\quad x = (x_1, x_2)$ $\quad z = (z_1, z_2)$

e) What is the feature representation $\phi(\boldsymbol{x})$ corresponding to this kernel for $\boldsymbol{x} \in \mathbb{R}^2$?

Answer: $K(x, z) = 1 + (x_1 z_1 + x_2 z_2)^2$ $\quad (a+b)^2 = (a+b)(a+b)$ $\quad (a^2 + 2ab + b^2)$

$= 1 + ((x_1 z_1)^2 + 2 x_1 z_1 x_2 z_2 + (x_2 z_2)^2)$

$\phi(x)^T \phi(z) = (1, x_1^2, \sqrt{2} x_1 x_2, x_2^2)^T \cdot (1, z_1^2, \sqrt{2} z_1 z_2, z_2^2)$

$\boxed{\phi(x) = (1, x_1^2, \sqrt{2} x_1 x_2, x_2^2)}$ V

2 pts.

f) Using the dual variables α's and $\phi(\boldsymbol{x})$, give an explicit form for the primal variable $\boldsymbol{w}$.

Answer:

$w^* = \sum_{i=1}^{n} \alpha^* y_i \phi(x)$

2 pts.

We solved the dual SVM problem with the above kernel. We used the following training set $\boldsymbol{x}_1 = (1, 1), y_1 = +1$, $\boldsymbol{x}_2 = (3, 0), y_2 = -1$, $\boldsymbol{x}_3 = (-2, 1), y_3 = -1$.
Below are the α's and b values
$\alpha_1 = 0.1$, $\alpha_2 = 0$, $\alpha_3 = 0.1$, $b = 1.5$

g) Use these values to compute the kernelized discriminant function $y(\boldsymbol{x})$.

Answer: $x \cdot w + b = \sum_{i=1}^{3} \alpha_i y_i (x \cdot x_i) + b$ $\qquad K(x, z) = 1 + (x^T z)^2$

$\phi(x) = \begin{bmatrix} x_1 x_1 \\ x_1 x_2 \\ x_1 x_2 \\ x_2 x_1 \\ x_2 x_2 \\ x_2 x_3 \\ x_3 x_1 \\ x_3 x_2 \\ x_3 x_3 \end{bmatrix}$

$= (\alpha_1 y_1 (x \cdot x_1) + b) + (\alpha_2 y_2 (x \cdot x_2) + b) + (\alpha_3 y_3 (x \cdot x_3) + b)$

$y(x) = \phi(x) w^* + b$.

4 pts.

Question 4: Regression (30 pts.)

In *univariate regression analysis*, we study the *statistical* dependency of the output variable $y \in \mathbb{R}$ on the input variable $x \in \mathbb{R}$.

linear combination of non-linear fcns of input variables $\phi^i(x) \rightarrow$ basis fcns

a) Complete the equation

$$y = \sum_{i=1}^{N} w^i \phi^i(x) + w_0 \quad \varepsilon \tag{1}$$

nonlinear regression

where the right hand side consists of two terms. Define each term in the equation.
Note: Be general, don't restrict yourself to linear least-squares regression.

Answer:

$y = \boxed{w^T x + w_0} + \varepsilon$

$\phi^i(x)$ are basis fcns

w^i = weights/coefficients

w_0 = offset

$\mathcal{N} \sim (0, \sigma^2)$

$E[w^T x + w_0 + \varepsilon]$ $E[\varepsilon] = 0$.

3 pts.

b) Regression analysis is only meaningful if there actually is a true dependency between x and y. Prove that if x and y are *statistically independent*, there is no linear dependency:

1. Assume that x and y are statistically independent.
2. Begin with the definition of *covariance*,

$$\mathrm{cov}(x, y) = \mathbb{E}\left[(x - \mu_x)(y - \mu_y)\right], \qquad f: X \rightarrow Y$$

where $\mathbb{E}$ is the expectation operator and μ_x is the mean of x.

3. Show that x and y are uncorrelated. $E[x] = \mu_x$ mean of x.

Note: Comment your calculations where necessary.

Answer:

$$E[(x-\mu_x)(y-\mu_y)] = E[xy - x\mu_y - y\mu_x + \mu_x\mu_y]$$

$$= E[xy] - E[x\mu_y] - E[y\mu_x] + E[\mu_x\mu_y]$$

independent

$$= E[x]E[y] - E[x]\mu_y - E[y]\mu_x + \mu_x\mu_y$$

$$= E[x]E[y] - \mu_x\mu_y - \mu_y\mu_x + \mu_x\mu_y$$

$$= \mu_x\mu_y - \mu_x\mu_y$$

$$= 0$$

Covariance is 0, which means uncorrelated.

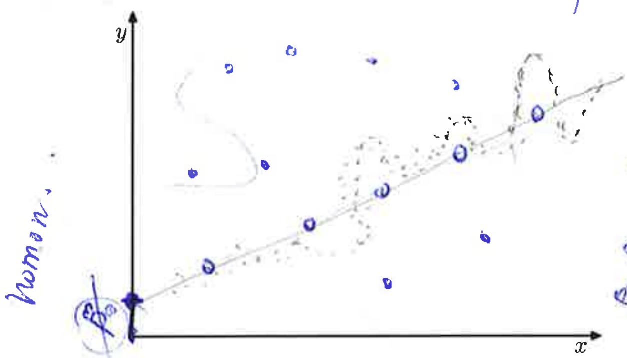
5 pts.

c) You have shown that eq. (1) consists of two terms. Illustrate two conceptually different cases where a linear least-squares fit is inappropriate for the true dependency between x and y, one related to the first term in eq. (1), and one related to the second term in eq. (1):

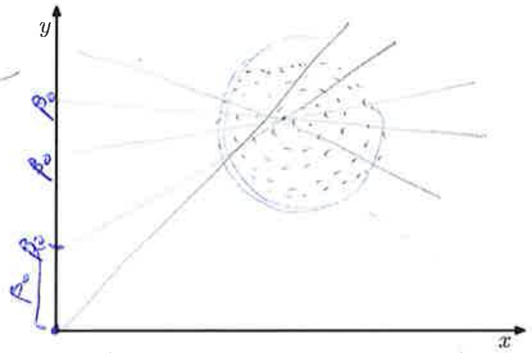
1. Draw a scatter plot of a data sample.
2. Fit the linear least-squares solution into the sample (qualitatively).
3. Explain why the solution does not capture the true dependency between x and y.

Answer:

w_0



The linear fit does not capture the complex relationship between x and y. this classifier is considered to "underfit"



with such a distribution there a multiple ways of determining β_0

4 pts.

So far, we considered a single input variable x. In multivariate linear regression, we assume a weighted linear functional dependency

$$y = w_0 + w_1 x_1 + \cdots + w_D x_D \tag{2}$$

between D input variables $x_1, \ldots, x_D \in \mathbb{R}$ and output variable y.

d) Define the input data matrix $\mathbf{X}$, weight vector $\mathbf{w}$ and output data vector $\mathbf{y}$ such that eq. (2) for all N data points of the sample $(x_1^n, x_2^n, \ldots, x_D^n, y^n), n = 1, \ldots, N$ can be written as

> homogeneous coordinates

$$\mathbf{y} = \mathbf{X}\mathbf{w}.$$

Answer:

$$\mathbf{X} := \underbrace{\begin{bmatrix} 1, x_1^1, x_2^1, x_3^1, \ldots, x_D^1 \\ 1, x_1^2, x_2^2, x_3^2, \ldots, x_D^2 \\ \vdots \\ 1, x_1^N, x_2^N, x_3^N, \ldots, x_D^N \end{bmatrix}}_{N \times (D+1)} \quad \mathbf{w} := \underbrace{\begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_D \end{bmatrix}}_{D \times 1} \quad \mathbf{y} := \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_N \end{bmatrix}}_{N \times 1}$$

$D+1$ N

3 pts.

The linear least-squares prediction $\hat{\mathbf{y}}$ is the *orthogonal projection* of $\mathbf{y}$ onto the space spanned by the columns of $\mathbf{X}$, and can be written as

$$\hat{y} = \boxed{\cancel{W}^T X} = \underbrace{\hat{\mathbf{y}}}_{=W^T X} \overset{X^T y}{=} \mathbf{P}\mathbf{y}. \qquad \hat{y} = W^T X \Rightarrow W = (X^T X)^{-1} X y$$

e) Analytically derive $\mathbf{P}$, where $\hat{\mathbf{y}}$ is the least-squares prediction vector. To achieve this:

1. Derive the optimal weight vector $\hat{\mathbf{w}}$ that minimizes the least-squares objective function

$$\|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2 \tag{3}$$

Note: we have $\frac{\partial}{\partial \mathbf{w}}\mathbf{w}^\top\mathbf{S}\mathbf{w} = 2\mathbf{S}\mathbf{w}$ *for symmetric matrix* $\mathbf{S}$*, and* $\frac{\partial}{\partial \mathbf{w}}\mathbf{w}^\top\mathbf{a} = \mathbf{a}$.

2. Using your result for $\hat{\mathbf{w}}$, give the definition for $\mathbf{P}$.

$X = n \times d$
$W = d \times 1$
$Y = n \times 1$

Answer:

$$\underset{\vec{w}}{\mathrm{argmin}} \|Xw - y\|_2^2 = (Xw - y)^T (Xw - y)$$

$$= w^T X^T X w - w^T X^T y - y^T X w - y^T y \quad \frac{d}{dw} \qquad X^T X = S$$

$$0 = 2X^T X w - X^T y - y^T X$$

$$0 = 2X^T X w - 2X^T y$$

$$X^T X w = X^T y$$

$$w = (X^T X)^{-1} X^T y \qquad \boxed{P = X(X^T X)^{-1} X^T} \; y$$

$$y = X^T P$$

5 pts.

f) Prove that $\mathbf{P}$ is an orthogonal projector, by showing that $\mathbf{P} = \mathbf{P}\mathbf{P}$ and $\mathbf{P} = \mathbf{P}^\top$.

Answer:

$$(X(X^T X)^{-1} X^T)(X(X^T X)^{-1} X^T) \qquad (A^{-1})^T = (A^T)^{-1}$$

$$P^T = (X(X^T X)^{-1} X^T)^T = X^T($$

2 pts.

g) Computing $\hat{\mathbf{w}}$ involves the inversion of the symmetric matrix $\mathbf{S} := \mathbf{X}^\top\mathbf{X}$, which we assume to be *positive definite*. Give a mathematical condition (in terms of the eigenvalues of $\mathbf{S}$), when this inversion is numerically unstable.

Answer: Defn of a positive definite matrix → eigendecomposition All diagonal elements, eigenvalues $\lambda_i > 0$ positive and real. if not then inversion is unstable.

All its eigenvalues are positive

$$\begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \lambda \end{bmatrix}$$

2 pts.

h) Ridge regression finds the regularized weight vector $\hat{\mathbf{w}}_{\text{ridge}}$ that minimizes eq. (3) with an additional norm penalty,

$$\|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{w}\|_2^2 .$$

Explain why choosing $\lambda > 0$ improves stability of the inversion: choosing a $\lambda > 0$ will shrink w's to not grow so big.

1. Derive the regularized optimal weight vector $\hat{\mathbf{w}}_{\text{ridge}}$.
   Answer:

$$\arg\min_{w} \|xw - y\|_2^2 + \lambda\|w\|_2^2 \Rightarrow \left(w^T x^T x w - w^T x^T y - y^T x w - y^T y + \lambda w^T w\right)\frac{d}{dw}$$

$$0 = 2x^T x w - 2x^T y + \lambda w$$

$$2x^T y = 2x^T x w + \lambda w$$

$$2x^T y = w(2x^T x + \lambda)$$

$$w_{Ridge} = \frac{2x^T y}{2x^T x + \lambda} = \frac{x^T y}{x^T x + \frac{1}{2}\lambda}$$

$$\boxed{w_{Ridge} = \left(x^T x + \frac{\lambda}{2}\right)^{-1} x^T y}$$

2 pts.

2. What is the effect of increasing λ on
   (a) the norm $\|\hat{\mathbf{w}}_{\text{ridge}}\|_2$ and
   (b) the numerical stability of computing $\hat{\mathbf{w}}_{\text{ridge}}$?

   Provide **arguments** for your answers.
   Answer:

(a) increasing λ will shrink w coefficients to not grow so big and prevent overfitting

(b)

4 pts.

Question 5: Bagging and AdaBoost (30 pts.)

Bagging and AdaBoost are two ensemble methods used frequently in classification.

a) State two essential differences between bagging and AdaBoost.

Answer:

	Bagging	AdaBoost
choice of data 1	varies the training sets using resampling to train weak learners	Trains weak learners with the same training set on every iteration
choice of weights 2	gives the same importance to every prediction	weights the prediction of every weak classifier according to its accuracy

2 pts.

b) Explain how bagging reduces the variance of an estimator?

Answer:

bagging uses the method of boot strap to create different training sets from the original distribution. in order to reduce the variance, this simulates increasing sample size.

3 pts.

c) State two factors that may influence the performance of AdaBoost?

Answer:

1) if the number of outliers is large, then the emphasis placed on these outliers can become large.

2 pts.

d) How can we use AdaBoost to detect outliers (mislabeled or ambiguously labeled examples)?

Answer:

By finding the classifier with the highest weight

2 pts.

For a binary classification problem (± 1), assume we have $c_1, ..., c_B$ successive base classifiers obtained via AdaBoost, and let $\alpha_1, ..., \alpha_B$ be the corresponding weights. The ensemble classifier is $\hat{c}_B(x) = \text{sgn}(\sum_{b=1}^{B} \alpha_b c_b(x))$.

Reusing the same base classifiers $c_1, ..., c_B$, we now train a linear classifier, by finding *new* $\tilde{\alpha}_b$'s that minimize the average exponential loss on the training examples, and obtain $\tilde{c}_B(x) = \text{sgn}(\sum_{b=1}^{B} \tilde{\alpha}_b c_b(x))$.

e) Is $\tilde{c}_B$ equivalent to $\hat{c}_B$? Justify your answer.

Answer:

$$\hat{C}_B(x) = \text{sgn}\left(\sum_{b=1}^{B} \alpha_b c_b(x)\right) \rightarrow \tilde{C}_B(x) = \text{sgn}\left(\sum_{b=1}^{B} \tilde{\alpha}_b c_b(x)\right)$$

Adaboost = linear classifier.

yes, it is equivalent because the sum of classifiers given weights is a linear fxn αc

4 pts.

Consider the following simplified ensemble learning algorithm:

We consider a dataset $\{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$ of N i.i.d. samples, where $y_n \in \{-1, +1\}$. After each iteration b of this algorithm, each training sample x_n has an associated weight $w_{b,n}$. At the beginning, $w_{0,n} = 1$ for $n = 1, ..., N$. Let $0 < \gamma < \frac{1}{2}$ and $\beta = \frac{1+2\gamma}{1-2\gamma}$.

For each iteration $b = 1$ to B:

- The algorithm finds a new weak learner c_b on the weighted training set that is guaranteed to have an error of at most $\frac{1}{2} - \gamma$ on the weighted training set.
- We then update the weights of the training samples:
  - If x_n is misclassified by c_b, i.e., if $c_b(x_n) \neq y_n$: Set $w_{b,n} := \beta w_{b-1,n}$.
  - Otherwise the weight remains unchanged: $w_{b,n} := w_{b-1,n}$.

The ensemble classifier decides by combining the weak learners as follows:

$$\hat{c}_B(x) = \text{sgn}\left(\sum_{b=1}^{B} c_b(x)\right).$$

Let $W_b = \sum_{n=1}^{N} w_{b,n}$ be the sum of the weights at the end of iteration b.

f) What is the difference between AdaBoost and this algorithm in the way they combine the weak learners?

Answer: The don't directly place coefficients with each weak learner, but instead place weights on training samples

Adaboost assigns a weight to each classifier based on performance.

2 pts.

g) Derive a bound on W_B, the sum of the weights after the final iteration. Given that at each iteration b, the sum of the weights increases by at most a factor of $1 + 2\gamma$:

$$\frac{W_b}{W_{b-1}} \leq 1 + 2\gamma,$$

show that for the final sum of weights it holds that

$$W_B \leq N(1 + 2\gamma)^B.$$

Answer:

$W_B \leq (1+2\gamma) W_{b-1}$

4 pts.

h) Prove the following lower bound on the final weight of a training sample, if it is misclassified by the ensemble:

If $\hat{c}_B(x_n) \neq y_n$, then $w_{B,n} \geq \beta^{B/2}$.

Answer:

4 pts.

i) Finally, we will show a bound on the empirical error rate on the training data. Let $M = |\{n : \hat{c}_B(x_n) \neq y_n\}|$ be the number of training samples misclassified by the ensemble classifier.

1. Plugging the two previous bounds together, we have

$$M\beta^{B/2} \leq \sum_{n=1}^{N} w_{n,B} = W_B \leq N(1+2\gamma)^B.$$

   Starting from this inequality, show that the empirical error rate on the training data is bounded by the following:

$$\frac{M}{N} \leq e^{-2B\gamma^2}.$$

   (*Hint: note that* $(1+a)^c \leq e^{ac}$ *for* $c \geq 0$, *and recall that* $\beta = \frac{1+2\gamma}{1-2\gamma}$)
   Answer:

2. Explain what happens to this bound when we
   (a) increase B (the number of training iterations),
   (b) increase γ (have a very good learner).
   Answer:

7 pts.

This page has been intentionally left blank.

Supplementary Sheet

Supplementary Sheet

1

Mean $\frac{\alpha}{\alpha+\beta}$