+1/1/60+

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Machine Learning – CS-433

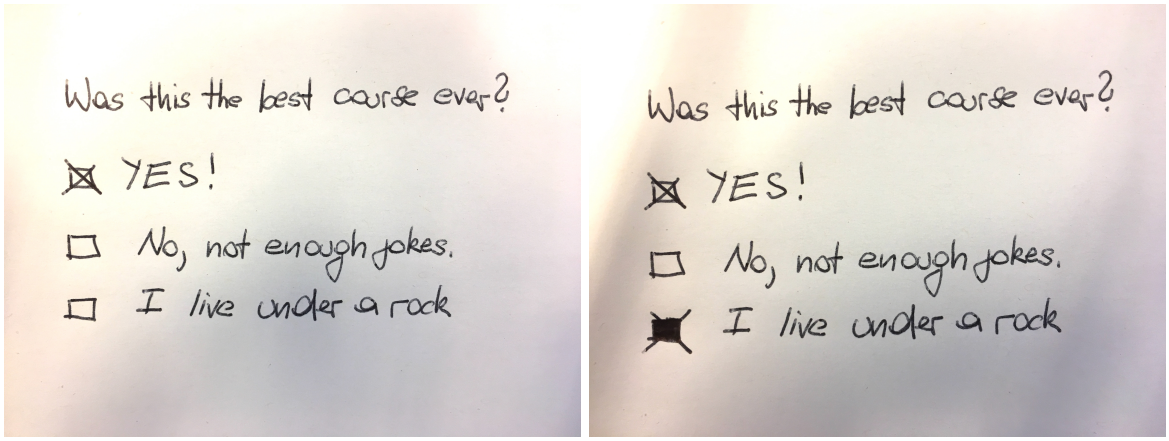
January 17, 2018

16h15-19h15 (STCC08328)

Important Notes

- This exam counts for 60 percent towards your final grade.
- This is a closed book exam. No electronic devices of any kind.
- Place on your desk: your student ID, writing utensils, one double-sided A4 page crypt sheet (handwritten or 11pt min font size) if you have one; place all other personal items below your desk or on the side.
- **Mark your answer with a thick 'X' in the corresponding box (see example below)**. If you want to change your answer, fill the box completely and mark the new answer with an 'X'. If you are unsure about an answer, perhaps just mark your best initial guess somewhere on the side before inserting the 'X' later on.
- Each page has a code on top of it (see the top of this page). Do not write on it.
- For the MCQ part, more than one answer can be correct. There are negative points for incorrect answers.
- You each have a different exam.
- For technical reasons, **do not use pencils for the MCQ part, only use pens**.

  Good luck!



The left figure shows how you mark an answer correctly.
Use a pen (and not a pencil) and make a **thick** "X". Make sure that it does not touch or cover other boxes.
The figure on the right shows how you can "erase" an answer.
Completely fill in the box corresponding to your old answer and then mark your new answer.

Room: **STCC**
Seat: **413**

Student Name / Sciper no.:
No It All
1

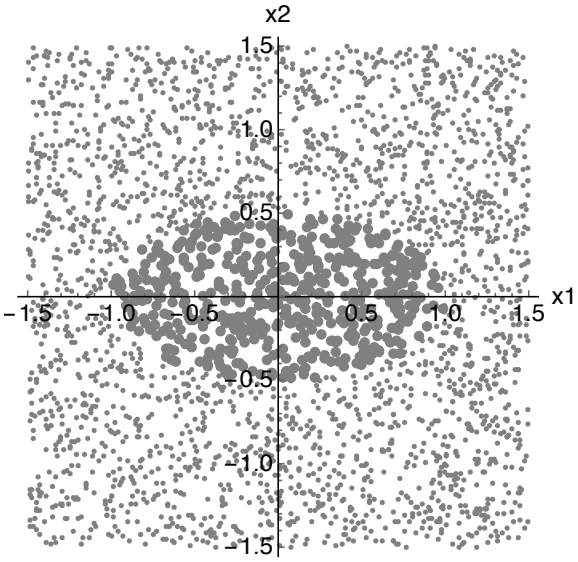


Figure 1: Some 2D data for classification. The two classes are indicated by different point sizes.

Problem 1 [1 point]

You have given the 2D data shown in Figure 1. You are allowed to add one component to your data (in addition to a constant component) and then must use a linear classifier. What component should you pick?

- ☐ $x_1 + x_2$
- ☐ $x_1^2 + 4x_2^2$
- ☐ $1/|x_1 + x_2|$
- ☐ $x_1 + 4x_2$
- ☐ $4x_1 + x_2$
- ☐ $x_1^2 x_2^2$
- ☐ $4x_1^2 + x_2^2$

Solution: The decision region is an ellipsoid with description $x_1^2 + 4x_2^2 = 1$.

Problem 2 [1.5 point]

Mark any of the following functions that have *unique maximizers*:

- ☐ $f(x) = -x^2, \quad x \in [-10, 10]$
- ☐ $f(x) = \ln(x), \quad x \in (0, 10]$
- ☐ $f(x) = x^2, \quad x \in [-10, 10]$
- ☐ $f(x) = \cos(2\pi x), \quad x \in [-1, 1]$
- ☐ $f(x) = \cos(2\pi x), \quad x \in [-\frac{1}{2}, \frac{1}{2}]$

Solution: The functions $f(x) = -x^2, x \in [-10, 10]$, $f(x) = \ln(x), x \in (0, 10]$, and $f(x) = \cos(2\pi x), x \in [-\frac{1}{2}, \frac{1}{2}]$ have a unique maximum.

Problem 3 [1 point]

We are given a data set $S = \{(\boldsymbol{x}_n, y_n)\}$ for a binary classification task where $\boldsymbol{x}_n$ in $\mathbb{R}^D$. We want to use a *nearest-neighbor* classifier. In which of the following situations do we have a reasonable chance of success with this approach? [Ignore the issue of complexity.]

- ☐ $n \to \infty$, D is fixed
- ☐ $n \to \infty$, $D \ll \ln(n)$
- ☐ $n = D^2$, $D \to \infty$
- ☐ n is fixed, $D \to \infty$

Solution: If the number of data points is exponential in the dimension then we have a chance that a nearest neighbor classifier works. Therefore when $n \to \infty$ and D is either fixed or very small compared to $\ln(n)$ then we have a chance.

Problem 4 [1 point]

Which of the following statements are true?

- ☐ The more training examples, the more accurate the prediction of a k-nearest-neighbor classifier.
- ☐ k-nearest-neighbors cannot be used for regression.
- ☐ A k-nearest-neighbor classifier is sensitive to outliers.
- ☐ Training a k-nearest-neighbor classifier takes more computational time than applying it / using it for prediction.

Solution: There is no traditional "training" involved when running a k-nearest-neighbor classifier. The more data we get the more accurate the classifier becomes. And such a scheme can also be used for regression. It is sensitive to outliers since around an outlier you will likely make the wrong prediction.

Problem 5 [1 point]

Consider the following joint distribution on X and Y, where both random variables take on the values $\{0,1\}$: $p(X=0,Y=0)=0.1$, $p(X=0,Y=1)=0.2$, $p(X=1,Y=0)=0.3$, $p(X=1,Y=1)=0.4$. You receive $X=1$. What is the largest probability of being correct you can achieve when predicting Y in this case?

- ☐ $\frac{1}{3}$
- ☐ $\frac{3}{4}$
- ☐ $\frac{1}{7}$
- ☐ 0
- ☐ 1
- ☐ $\frac{2}{3}$
- ☐ $\frac{6}{7}$
- ☐ $\frac{4}{7}$
- ☐ $\frac{3}{7}$
- ☐ $\frac{1}{4}$
- ☐ $\frac{2}{4}$

Solution: We have $p(Y=1|X=1)=p(Y=1,X=1)/p(X=1)=0.4/0.7=4/7$. Therefore, you should guess that $Y=1$ and you will be right a fraction $4/7$ of the time. This is the best you can do.

Problem 6 [1 point]

You are given a distribution on X, Y, and Z and you know that the joint distribution can be written in the form $p(x,y,z)=p(x)\,p(y|x)\,p(z|y)$. What conclusion can you draw? [Recall that $\perp$ means independent and $|\cdots$ means conditioned on $\cdots$.]

- ☐ $Y \perp Z$
- ☐ $X \perp Y \;|\, Z$
- ☐ $Y \perp Z \;|\, X$
- ☐ $X \perp Z$
- ☐ $X \perp Y$
- ☐ $X \perp Z \;|\, Y$

Solution: This is a Markov chain $X-Y-Z$ and so you have $X \perp Z \,|\, Y$. This is all you can say.

Problem 7 [1 point]

What is the gradient of $\boldsymbol{x}^\top \boldsymbol{W}^\top \boldsymbol{W}\boldsymbol{x}$ with respect to $\boldsymbol{x}$ (written as a vector)?

- ☐ $2\boldsymbol{W}^\top \boldsymbol{x}$
- ☐ $2\boldsymbol{W}^\top \boldsymbol{W}\boldsymbol{x}$
- ☐ $2\boldsymbol{W}\boldsymbol{W}^\top \boldsymbol{x}$
- ☐ $2\boldsymbol{W}$
- ☐ $2\boldsymbol{W}\boldsymbol{x}$

Solution: Writing out the expression explicitly with $\boldsymbol{A} = \boldsymbol{W}^\top \boldsymbol{W}$ we have $\sum_{i,j} \boldsymbol{A}_{i,j}\boldsymbol{x}_i\boldsymbol{x}_j$. Taking the derivative with respect to $\boldsymbol{x}_i$ we get two terms, one corresponding to $\boldsymbol{A}\boldsymbol{x}$ and one corresponding to $\boldsymbol{A}^\top \boldsymbol{x}$. But $\boldsymbol{A}$ is symmetric. So the final result is $2\boldsymbol{W}^\top \boldsymbol{W}\boldsymbol{x}$.

Problem 8 [2 points]

What is the derivative of $\boldsymbol{W}\boldsymbol{W}$ with respect to $\boldsymbol{W}_{i,j}$, where $\boldsymbol{W}$ is an $n \times n$ matrix.

- ☐ $\boldsymbol{W}_{i,j}$
- ☐ add to the $n \times n$ all-zero matrix the j-th row of $\boldsymbol{W}$ and the i-th column of $\boldsymbol{W}$
- ☐ the vector of length n equal to the sum of the j-th row of $\boldsymbol{W}$ and the i-th column of $\boldsymbol{W}$
- ☐ $\boldsymbol{W} + \boldsymbol{W}^\top$
- ☐ $\boldsymbol{W}_{i,j}^2$

Solution: Writing out the expression explicitly we have $\sum_j \boldsymbol{W}_{i,j}\boldsymbol{W}_{j,l}$. Note that this is an $n \times n$ matrix. So the derivative with respect to a particular parameter will also be an $n \times n$ matrix. Taking the derivative with respect to $\boldsymbol{W}_{i,j}$ we get two contributions. One will be an $n \times n$ matrix that has all-zeros except for the j-th row that is equal to the j-th row of $\boldsymbol{W}$ and one that is an $n \times n$ matrix that has all-zero except for the i-column that is equal to the i-th column of $\boldsymbol{W}$.

Problem 9 [1 point]

Assume that we have a convolutional neural net with L layers, K nodes per layer, and where each node is connected to k nodes in a previous layer. We ignore in the sequel the question of how we deal with the points at the boundary and assume that $k <<< K$ (much, much, much smaller). How does the complexity of the back-propagation algorithm scale in these parameters?

- ☐ $\Theta(Lk^K)$
- ☐ $\Theta(Lk^K)$
- ☐ $\Theta(LK^k)$
- ☐ $\Theta(LKk)$
- ☐ $\Theta(L^kK)$

Solution: For both the forward pass as well as the backward pass we have to send a "message" along each edge from one layer to the next. According to the description there are Kk such edges in each layer and there are L layers. So the complexity scaled like KkL. We cannot do better using the FFT given that $k <<< K$.

Problem 10 [1 point]

You are using a neural net with L layers, K nodes per layer, and ReLUs as activation functions. Your input data has components in $[-1, 0]$. You initialize all your weights to be Gaussians with mean 10 and variance 0.1 and all the bias terms are set to 0. You start optimizing using SGD. What will happen?

- ☐ the gradient is 0 and so nothing happens
- ☐ the gradient is very large and so your steps are likely too large for the algorithm to converge
- ☐ everything is fine
- ☐ you cannot use a neural net for features that have negative components

Solution: Due to this initialization the operation point will be almost surely 0 and all gradients will be 0 as well.

Problem 11 [1 point]

You are doing your ML project. It is a regression task under a square loss. Your neighbor uses linear regression and least squares. You are smarter. You are using a neural net with 10 layers and activations functions $f(x) = 3x$. You have a powerful laptop but not a supercomputer. You are betting your neighbor a beer at Satellite who will have a substantially better scores. What will happen? Who will pay?

- ☐ I will pay.
- ☐ It will essentially be a tie, so we decide to have two beers and both pay.
- ☐ My neighbor will pay.

Solution: Since I use a linear activation function I in fact use just a linear scheme. And since the problem is convex SGD will give the same result as least squares. So we will get exactly the same result.

Problem 12 [1 point]

What is the reason for the outcome of the previous bet?

- ☐ Because we use exactly the same scheme.
- ☐ Because it is almost impossible to train a network with 10 layers without a supercomputer.
- ☐ Because I should have used more layers.
- ☐ Because I should have used only one layer.

Solution: Since I use a linear activation function I in fact use just a linear scheme. And since the problem is convex SGD will give the same result as least squares. So we will get exactly the same result.

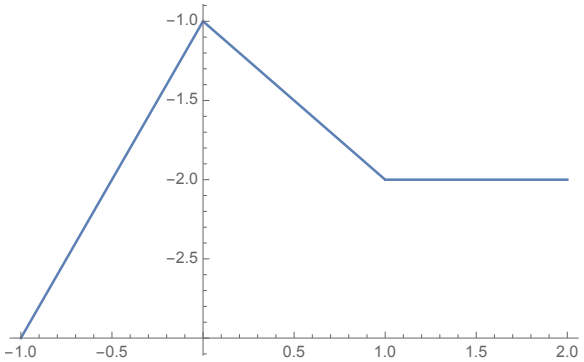


Figure 2: What is the subgradient of this function at $x = 1$?

Problem 13 [1 point]

Which of the following scalars g is a subgradient for the function shown in Figure 2 at the point $x = 1$?

- ☐ $g = -\frac{1}{2}$
- ☐ $g = -1$
- ☐ none exists
- ☐ $g = 0$

Solution: None exists. In order to be a subgradient the linear function defined by the operating point and the slope g should be a global lower bound on the function.

Problem 14 [1 point]

Consider the composite function $f(x) = g(h(x))$, where all functions are $\mathbb{R}$ to $\mathbb{R}$. Which of the following is the weakest condition that guarantees that $f(x)$ is convex?

- ☐ $g(x)$ and $h(x)$ are convex and $g(x)$ and $h(x)$ are increasing
- ☐ $g(x)$ is convex and $g(x)$ is increasing
- ☐ $g(x)$ and $h(x)$ are convex and $h(x)$ is increasing
- ☐ $g(x)$ and $h(x)$ are convex and $g(x)$ is increasing
- ☐ $g(x)$ is convex and $g(x)$ and $h(x)$ are increasing
- ☐ $h(x)$ is convex and $g(x)$ and $h(x)$ are increasing
- ☐ $g(x)$ is convex and $h(x)$ is increasing

Solution: The second derivative of the function $f(x)$ is equal to $f'' = g''(h')^2 + g'h''$. Therefore the minimal condition for convexity of $f(x)$ is that $g(x)$ and $h(x)$ are convex and $g(x)$ is increasing.

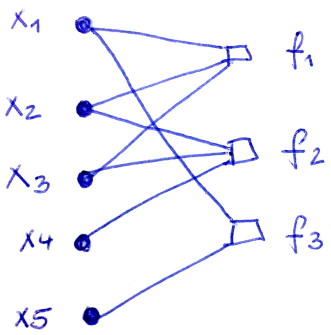


Figure 3: A factor graph.

Problem 15 [1 point]

Consider the factor graph shown in Figure 3. What is the implied factorization of the function $f(x_1, x_2, x_3, x_4, x_5)$ shown in this graph?

- ☐ $f_1(x_1)f_1(x_2)f_1(x_3)f_2(x_2)f_2(x_3)f_2(x_4)f_3(x_1)f_3(x_5)$
- ☐ $f_1(x_1, x_2, x_3, x_4x_5))f_2(x_2, x_3, x_4, x_5)f_3(x_3, x_4, x_5)$
- ☐ $f_1(x_1)f_2(x_2)f_3(x_3)f_4(x_4)f_5(x_5)$
- ☐ $f_1(x_1, x_2, x_3)f_2(x_2, x_3, x_4)f_3(x_1, x_5)$

Solution: $f_1(x_1, x_2, x_3)f_2(x_2, x_3, x_4)f_3(x_1, x_5)$

Problem 16 [1 point]

Consider again the factor graph shown in Figure 3. You are using the message-passing algorithm discussed in the course in order to compute the exact marginal $f(x_1)$. What is the complexity of the algorithm in terms of the alphabet size $|\mathcal{X}|$?

- ☐ $\Theta(|\mathcal{X}|^5)$.
- ☐ You cannot compute the marginals exactly on this graph via message-passing.
- ☐ $\Theta(|\mathcal{X}|^1)$.
- ☐ $\Theta(|\mathcal{X}|^4)$.
- ☐ $\Theta(|\mathcal{X}|^2)$.
- ☐ $\Theta(|\mathcal{X}|^3)$.

Solution: This graph has cycles. Hence you cannot apply the standard message algorithm direcly in order to compute the marginal exactly.

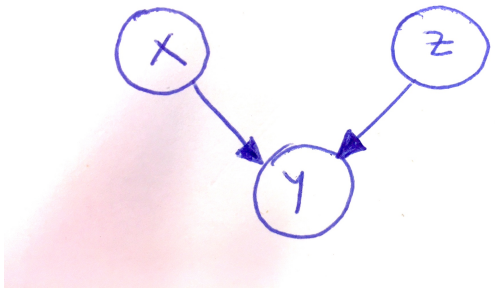


Figure 4: A directed acyclic graph.

Problem 17 [2.25 points]

Consider the directed acyclic graph shown in Figure 4. Which of the following factorizations of the random variables X, Y and Z are consistent with this graph. We say that a factorization is *consistent* with a Bayes net if every distribution characterized by the Bayes net can be written in the form of this factorization.

- ☐ $p(X)p(Z|X)p(Y|X,Z)$
- ☐ $p(Y)p(X,Z)$
- ☐ $p(X)p(Z)p(Y|X,Z)$
- ☐ $p(X)p(Z)p(Y|Z)$
- ☐ $p(X,Y,Z)$
- ☐ $p(X,Z)p(Y|X,Z)$
- ☐ $p(X)p(Y|X)p(Z)$
- ☐ $p(X)p(Y)p(Z)$
- ☐ $p(X)p(Z)p(Y|X)$

Solution: The following factorizations are consistent: $p(X)p(Z \mid X)p(Y|X,Z)$, $p(X)p(Z)p(Y|X,Z)$, $p(X,Z)p(Y|X,Z)$, $p(X,Y,Z)$

Problem 18 [1 point]

Consider a linear regression problem with N samples $\{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$, where each input $\boldsymbol{x}_n$ is a D-dimensional vector $\{-1,+1\}^D$, and all output values are $y_i \in \mathbb{R}$. Which of the following statements is correct?

- ☐ Linear regression always "works" very well for $N \ll D$
- ☐ A linear regressor works very well if the data is linearly separable.
- ☐ Linear regression always "works" very well for $D \ll N$
- ☐ None of the above.

Solution: D vs N does not matter at all. Linear separability is important for classification but does not imply anything for regression. So the correct answer is "None of the above."

Problem 19 [2 points]

Which of the following statements is correct?

- ☐ When applying stochastic gradient descent on the objective function $f(\boldsymbol{w}) := \sum_{n=1}^{30} \|\boldsymbol{w} - \boldsymbol{x}_n\|^2$ where $\boldsymbol{x}_n$ are the datapoints, a stochastic gradient step is roughly $30\times$ faster than a full gradient step.
- ☐ In practice, it could be good to let your model first overfit your task, and then apply drop-out or other regularization techniques.
- ☐ When applying stochastic gradient descent on the objective function $f(\boldsymbol{w}) := \sum_{n=1}^{30} n \cdot \|\boldsymbol{w}\|^2$, a stochastic gradient (for the n-th summand) is given by $2n \cdot \boldsymbol{w}$.
- ☐ The function $f(\boldsymbol{u}; \boldsymbol{v}) := g(\boldsymbol{u}\boldsymbol{v}^\top)$ is convex over the set of pairs of vectors $(\boldsymbol{u}; \boldsymbol{v}) \in \mathbb{R}^2 \times \mathbb{R}^2$, when $g : \mathbb{R}^{2\times 2} \to \mathbb{R}$ is defined as $g(\boldsymbol{X}) := X_{12} + X_{21}$.

Solution: mat fact: NO! stoch. grad: NO, need to multiply by 30 as it's not an average. SGD faster: YES. overfit: YES.

Problem 20 [1 point]

Is it true that K-means can be equivalently written as the following matrix factorization problem?
Here $\boldsymbol{X}$ denotes the $N \times D$ data matrix. The $\boldsymbol{\mu}_k$ denote columns of $\boldsymbol{M}$, rows of $\boldsymbol{Z}$, and $L(\boldsymbol{z}, \boldsymbol{\mu}) = \|\boldsymbol{X}^\top - \boldsymbol{M}\boldsymbol{Z}^\top\|_{\mathsf{Frob}}^2$.

$$\begin{aligned} &\min_{\boldsymbol{z},\boldsymbol{\mu}} L(\boldsymbol{z}, \boldsymbol{\mu}) \\ &\text{s.t. } \boldsymbol{\mu}_k \in \mathbb{R}^D, \\ &z_{nk} \in \{0, 1\}, \sum_{k=1}^{K} z_{nk} = 1. \end{aligned}$$

- ☐ yes
- ☐ no

Solution: yes

Problem 21 [2 points]

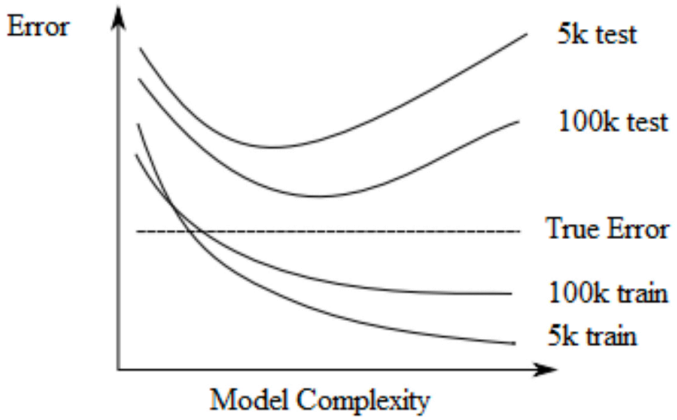
In Text Representation learning, which of the following statements is correct?

- ☐ Learning GloVe vectors can be done using SGD in a streaming fashion, by streaming through the input text only once.
- ☐ Every recommender systems algorithm for learning a matrix factorization $\boldsymbol{W}\boldsymbol{Z}^\top$ approximating the observed entries in least square sense does also apply to learn GloVe word vectors.
- ☐ FastText performs unsupervised learning of word vectors.
- ☐ If you fix all word vectors, and only train the remaining parameters, then FastText in the two-class case reduces to being just a linear classifier.

Solution: unsupervised fasttext: NO. MF algo for GloVe: YES. Glove streaming: NO. FastText linear classifier: YES.

Problem 22 [4 points]
(Due to Alex Smola) Assume that you have two data sets that contain iid samples from the same distribution, call them S_1 and S_2. S_1 contains 5000 samples, whereas S_2 contains 100000 samples. You randomly split each of the data sets into a training and a testing set, where eighty percent of the data is assigned to the training set. You then train and test on a family of increasing complexity.
In the figure below draw four curves, two that show the *training error* as a function of the model complexity (for S_1 and S_2) and two that show the *testing error* as a function of the model complexity (for S_1 and S_2). Label each of the 4 curves clearly. The constant curve labeled "true error" corresponds to the error due to the inherent noise in the samples and is drawn as a reference curve.

☐0 ☐0.5 ☐1 ☐1.5 ☐2 ☐2.5 ☐3 ☐3.5 ☐4 *Reserved, do not write on this box*



Solution:

- The training error decreases when increasing the model complexity, while the test error decreases first but then increases due to overfitting.
- Given the same model complexity, the model has larger training samples is less likely to overfit than the one with less samples. So the two curves representing 100k samples are nearer the dash line the other two curves.

Problem 23 [4 points]

Consider the Poisson distribution with parameter λ. It has a probability mass function given by $p(i) = \frac{\lambda^i e^{-\lambda}}{i!}$, $i = 0, 1, \cdots$.

(i) [2pts] Write $p(i)$ in the form of an exponential distribution $p(i) = h(i)e^{\eta\phi(i)-A(\eta)}$. Explicitly specify h, η, ϕ, and $A(\eta)$.

(ii) [2pts] Compute $\frac{dA(\eta)}{d\eta}$ and $\frac{d^2A(\eta)}{d\eta^2}$? Is this the result you expected?

☐0 ☐0.5 ☐1 ☐1.5 ☐2 ☐2.5 ☐3 ☐3.5 ☐4 *Reserved, do not write on this box*

(i) [2pts]

(a) $h(i) =$

(b) $\eta =$

(c) $\phi(i) =$

(d) $A(\eta) =$

(ii) [2pts]

(a) $\frac{dA(\eta)}{d\eta} =$

(b) $\frac{d^2A(\eta)}{d\eta^2} =$

(c) Explanation:

Solution:

(i) [2pts]

(a) $h(i) = \frac{1}{i!}$

(b) $\eta = \ln(\lambda)$

(c) $\phi(i) = i$

(d) $A(\eta) = \lambda = e^\eta$

(ii) [2pts]

(a) $\frac{dA(\eta)}{d\eta} = \lambda$

(b) $\frac{d^2A(\eta)}{d\eta^2} = \lambda$

(c) Explanation: We know that the first two derivatives of $A(\eta)$ give us the mean and the variance and for a Poisson distribution these are both λ. So the result is expected.

Problem 24 [4 points]
You are given your $D \times N$ data matrix $\boldsymbol{X}$, where D represents the dimension of the input space and N is the number of samples. We discussed in the course the singular value decomposition (SVD). Recall that the SVD is *not* invariant to scaling and that empirically it is a good idea to remove the mean of each feature (row of $\boldsymbol{X}$) and to normalize its variance to 1. Assume that $\boldsymbol{X}$ has this form except that the last row/feature is then multiplied by $\sqrt{2}$, i.e., it has variance (ℓ_2^2-norm) of 2 instead of 1.
Recall that the SVD allows us to write $\boldsymbol{X}$ in the form $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top$, where $\boldsymbol{U}$ and $\boldsymbol{V}$ are unitary and $\boldsymbol{S}$ is a $D \times N$ diagonal matrix with entries s_i that are non-negative and decreasing, called the singular values.
Assume now that you add a feature, i.e., you add a row to $\boldsymbol{X}$. Assume that this row is identical to the last row of $\boldsymbol{X}$, i.e., you just replicate the last feature. Call the new matrix $\tilde{\boldsymbol{X}}$. But assume also that for $\tilde{\boldsymbol{X}}$ we normalize all rows to have variance 1.
To summarize, $\boldsymbol{X}$ is the original data matrix, where all means have been taken out and all rows are properly normalized to have variance 1 except the last one that has variance 2. And $\tilde{\boldsymbol{X}}$ is the original data matrix with the last row replicated, and all means have been taken out and all rows are properly normalized.
Let $\boldsymbol{X} = \boldsymbol{U} \cdot \boldsymbol{S} \cdot \boldsymbol{V}^\top$ be the SVD of $\boldsymbol{X}$ and let. $\tilde{\boldsymbol{X}} = \tilde{\boldsymbol{U}} \cdot \tilde{\boldsymbol{S}} \cdot \tilde{\boldsymbol{V}}^\top$ be the SVD of $\tilde{\boldsymbol{X}}$.

1. Show that
   (a) $\tilde{\boldsymbol{V}} = \boldsymbol{V}$
   (b) $\tilde{\boldsymbol{S}}$ is equal to $\boldsymbol{S}$ with an extra all-zero row attached.
2. Based on the previous relationships and assuming that it is always best to run an SVD with "normalized" rows, what is better: If you *KNOW* that a feature is highly correlated to another feature a priori. Should you rather first run the SVD and then figure out what features to keep or should you first take the highly correlated feature out and then run the SVD? Explain.

☐0 ☐0.5 ☐1 ☐1.5 ☐2 ☐2.5 ☐3 ☐3.5 ☐4 *Reserved, do not write on this box*

Solution:

1. In the exercise you learned that the columns of $\boldsymbol{V}$ are the eigenvectors associated to $\boldsymbol{X}^\top\boldsymbol{X}$. And the non-zero singular values of $\boldsymbol{X}$ are the square roots of the non-zero eigenvalues of $\boldsymbol{X}^\top\boldsymbol{X}$. But for our case $\tilde{\boldsymbol{X}}^\top\tilde{\boldsymbol{X}} = \boldsymbol{X}^\top\boldsymbol{X}$, which proves the two claims.
2. It is better to first take out the highly correlated feature. If not, then the previous calculations show that this is the same as having a row that is not properly normalized.

Problem 25 [2 points]

You are given a training set $S = \{(x_n, y_n)\}_{n=1}^{N}$ for classification with $y_n \in \{0, 1\}$. Ninety percent of the labeled data has label 0. You split the data randomly into two equal parts, train on the first part, and then test on the second part. You get an accuracy of 85 percent. What is your reaction? Explain.

☐0 ☐0.5 ☐1 ☐1.5 ☐2 *Reserved, do not write on this box*

Solution: Better scratch your head, blame Martin and Ruediger for not teaching you properly, and try again. If you do not get an accuracy of at least 90 percent then you are not really doing anything since you can get ten percent by simply always outputting 0.