STAT 154 Spring 2019: Sample Final Exam

Instructor: Prof. Bin Yu
May 16, 7 PM–10 PM

Name: Raaz Dwivedi Student ID: ~

Row Number: ~ Seat Number: ~

Maximum Points: 100 Time: 3 hours

Instructions (please read carefully)

- Do not turn the page until you are told do so. Just count the number of sheets, **you should have** 11 **sheets** (22 pages counting both sides). If you have any issues contact the GSI/instructor immediately.
- Write your student-id clearly on top of each page.
- Your answer **will be graded only if** it is written in the space provided after the question. Use the blank pages at the end for rough work. There is a help-sheet at the end for your comfort.
- **When time is up**, please take your exam in your **left hand** and raise it. When asked pass it on to your left (facing the board). Please maintain the sequence of your seating and **Do not leave until instructed to do so**. Instructor/GSI will collect the copies from the leftmost student in the row.

Pre-Exam Questions

1. What are your favorite restaurants in Berkeley?

2. Express your feelings for the class.

1 Single Choice Questions (no justification, 13 parts, 26 pts)

In the following questions, each part has **EXACTLY one** correct choice. Please darken the corresponding bubble properly. e.g., How often do you like teaching this class?

● Always ○ Sometimes ○ Rarely ○ Never

(i) Select the **CORRECT** statement.

○ The 3-partition protocol in the construction of a prediction rule guarantees that the prediction rule will work for future data.

○ Data collection process usually has no-to-little influence on the outcome of a prediction problem.

○ Results from prediction algorithms are always connected to reality in data because they are run on data.

● The dotted red circle in the 3-circle representation of the prediction problem is necessary since the future is almost always different from present data.

(ii) Which of the following methods **DOES NOT** make probabilistic assumptions related to the generation of the data?

○ LDA ○ QDA ○ Logistic Regression ● SVM

(iii) Select the **INCORRECT** statement from the following.

○ For multi-class LDA, the decision boundary is piecewise linear.

● For two-class classification, the decision boundary for LDA, logistic regression and SVM is linear ONLY if the data is linearly separable.

○ The decision boundary in a Decision tree is often non-linear.

○ Training ~~error~~ loss (exponential) in boosting is non-increasing as the iterations progress.

(iv) Which of the following statements are **TRUE** about kernels?

○ Even for a valid kernel, there might exist a set of points such that the corresponding kernel matrix can be negative semi-definite.

○ If k_1 and k_2 are valid kernels, then $k = 2k_1 + k_2$ need not be a valid kernel.

○ If k_1 and k_2 are valid kernels, then $k = \alpha k_1 + \beta k_2$ is a valid kernel for any $\alpha, \beta \in \mathbb{R}$. ($\beta = -1$, $\alpha = 0$)

● Given an arbitrary feature map ϕ, if we choose the kernel function k such that $k(\mathbf{x}, \mathbf{z}) = \phi(\mathbf{x})^\top \phi(\mathbf{z})$ for any $\mathbf{x}, \mathbf{z}$, then k is guaranteed to be a valid kernel function.

$n=1000,\ d=10$

(v) Suppose $\mathbf{X} \in \mathbb{R}^{1000\times 10}$ and $\mathbf{y} \in \mathbb{R}^{1000}, \beta \in \mathbb{R}^{10}, \alpha \in \mathbb{R}$. Which of the following commands will take the **largest time** to execute in R?

- ○ `β - t(X) %*% (X %*% β - y)` $O(nd)$
- ● `X %*% t(X)` $O(n^2 d)$
- ○ `(X %*% β - y)` $O(nd)$
- ○ `t(X) %*% X` $O(nd^2)$

$mxk \quad kxp \Rightarrow mkp$

(vi) Let $\mathbf{A} = \begin{bmatrix}\mathbf{a}_1 & \mathbf{a}_2\end{bmatrix} \in \mathbb{R}^{d\times 2}$, where $\mathbf{a}_1, \mathbf{a}_2 \in \mathbb{R}^d$ denote the columns of the matrix $\mathbf{A}$. Select the **CORRECT** statement:

- ○ The matrix $\mathbf{A}^\top\mathbf{A}$ is a projection matrix. $A(A^TA)^{-1}A^T$
- ○ The matrix $\mathbf{a}_1\mathbf{a}_1^\top + \mathbf{a}_2\mathbf{a}_2^\top$ need not be a PSD matrix. ✗
- ○ The matrix $\mathbf{a}_1\mathbf{a}_1^\top + \mathbf{a}_2\mathbf{a}_2^\top$ is always a projection matrix. ✗
- ● The matrix $\mathbf{a}_1\mathbf{a}_1^\top + \mathbf{a}_2\mathbf{a}_2^\top$ is always a PSD matrix but is a projection matrix only when a_1 and a_2 are orthonormal vectors. ✓

(vii) Which of the following statements **IS TRUE** with regards to logistic regression for a two-class classification?

→ when X is not full rank.

- ○ Logistic regression (with no-regularization) always has unique solution. ✗
- ○ ℓ_1-regularized logistic regression always has a unique solution. ✗ → when multiple sparse
- ● ℓ_2-regularized logistic regression always has a unique solution. → strongly convex
- ○ The loss function for logistic regression (with no-regularization) can be non-convex. ✗ → always convex

(viii) Which of the following methods is **guaranteed** to perform well on the future test data once they are tuned and trained to achieved zero training error and a small enough validation error?

- ○ Random Forest
- ○ Deep Neural Network
- ○ Methods based on boosting
- ● There does not exist a universal solution to all problems. ✓

(ix) Which of the following statements is **TRUE** about loss functions?

- ○ Loss functions are usually learned from the data set.
- ○ Cross-entropy loss and mean-squared loss are always identical.
- ○ The lower the value of the loss function at the end of training, the better the test error.
- ● MSE loss can be derived by doing maximum likelihood on a linear regression model (with Gaussian i.i.d. noise).

(x) "Regularization" is a critical aspect of machine learning. Which of the following statements about regularization is **FALSE**?

- ○ Doing dimensionality reduction as a part of the learning process has a regularizing effect.
- ○ Adding noisy copies of training data points is regularizing.
- ○ Adding a penalty term on the learned weights can be regularizing.
- ● Using kernelized methods is always more regularizing than using direct methods.

(xi) The form of the decision boundary for a Linear Discriminant Analysis (LDA) based classifier

- ● is always linear.
- ○ can be quadratic.
- ○ depends on if the means for each class are are all the same.
- ○ depends on if the covariance matrices are diagonal.

(xii) Both PCA and Lasso can be used for feature selection. Which of the following statements is **TRUE**?

- ○ PCA and Lasso both allow you to specify how many features are chosen.
- ○ PCA produces features that are non-linear combinations of the original features.
- ○ PCA and Lasso are the same if you use the kernel trick.
- ○ PCA always produces interpretable features.
- ○ Features selected by lasso are not interpretable.
- ● Lasso selects a subset (not necessarily a strict subset) of the original features.

(xiii) Which of the following is **TRUE** about bagging?

- ● In bagging, we choose random subsamples of the input points with replacement.
- ○ Bagging is ineffective with logistic regression, because all of the learners learn exactly the same decision boundary.
- ○ The main purpose of bagging is to decrease the bias of learning algorithms.
- ○ In bagging, we use a subset of features to train each model.

2 Playing with Kernels (2+3+3+2=10 pts)

1. What is the primary motivation for using the kernel methods in machine learning algorithms?
   Ans. If we want to map sample points to a very high-dimensional feature space, the kernel trick can save us from having to compute those features explicitly, thereby saving a lot of time. (Alternative solution: the kernel trick enables the use of infinite-dimensional feature spaces.)

2. Given a design matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$, show that the Gram/Kernel matrix of the linear kernel (with $\Phi : x \mapsto x$) is positive semi-definite.
   Ans. In this case, the gram matrix is simply $\mathbf{X}\mathbf{X}^\top$ which is clearly PSD:

$$\mathbf{z}^\top \mathbf{X}\mathbf{X}^\top \mathbf{z} = \left\| \mathbf{X}^\top \mathbf{z} \right\|_2^2 \geq 0.$$

3. Suppose a machine learning algorithm contains the following line of code (update equation),

$$\mathbf{w} \leftarrow \mathbf{w} + \mathbf{X}^\top \mathbf{M}\mathbf{X}\mathbf{X}^\top \mathbf{u}, \tag{1}$$

where $\mathbf{X} \in \mathbb{R}^{n\times d}$ is the design matrix, $\mathbf{w} \in \mathbb{R}^d$ is the weight vector, $\mathbf{M}$ is a matrix unrelated to $\mathbf{X}$ and $\mathbf{u} \in \mathbb{R}^n$ is a vector unrelated to $\mathbf{X}$. Suppose that we know that we can express $\mathbf{w}$ as a linear combination of samples $\mathbf{x}_i$ (rows of $\mathbf{X}$). That is, there exists a vector $\mathbf{a} \in \mathbb{R}^n$, such that $\mathbf{w} = \mathbf{X}^\top a$. Show that we can write the update equation (1) only in terms of $\mathbf{a}$ (so that $\mathbf{w}$ does not appear).

Ans. When we replace $\mathbf{w} = \mathbf{X}^\top \mathbf{a}$, we obtain:

$$\mathbf{X}^\top \mathbf{a} \leftarrow \mathbf{X}^\top \mathbf{a} + \mathbf{X}^\top \mathbf{M}\mathbf{X}\mathbf{X}^\top \mathbf{u},$$

and thus it suffices to just run

$$\mathbf{a} \leftarrow \mathbf{a} + \mathbf{M}\mathbf{X}\mathbf{X}^\top \mathbf{u}$$

4. Can the update equation in terms of $\mathbf{a}$ in the previous question be kernelized? If so, show how. If not, explain why.

Ans. Yes. When we use features $\mathbf{x} \mapsto \phi(\mathbf{x})$, the matrix $\mathbf{X}\mathbf{X}^\top$ gets replaced by $\mathbf{\Phi}\mathbf{\Phi}^\top$ which is simply the Kernel/Gram Matrix $\mathbf{K}$ and thus we can write the update as:

$$\mathbf{a} \leftarrow \mathbf{a} + \mathbf{M}\mathbf{K}\mathbf{u}.$$

3 When testing is unfair (2+3+7+4+3=19 pts)

In this problem we explore what happens when we train an estimator on a data set which has a different distribution than the test set. Suppose the training data is $\{\mathbf{x}_i, y_i\}_{i=1}^n$ generated as

$$y_i = \mathbf{x}_i^\top \mathbf{w}^\star + \varepsilon_i, \quad i = 1, \ldots, n$$

where $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$ and the noise variables satisfy $\varepsilon_i \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$. We use the standard matrix-vector notation for observations and features:

$$\mathbf{y} \in \mathbb{R}^n, \quad \mathbf{X} \in \mathbb{R}^{n \times d} \quad \text{and} \quad \mathbf{\Sigma} = \mathbf{X}^\top \mathbf{X} \in \mathbb{R}^{d \times d}$$

where $\mathbf{X}$ has $\mathbf{x}_i^\top$ as the rows. We assume $\mathbf{X}$ is full column rank and

$$\mathbf{X} = \mathbf{U}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{V}^\top \quad \text{where} \quad \mathbf{U} \in \mathbb{R}^{n \times d}, \mathbf{V} \in \mathbb{R}^{d \times d}, \text{ and } \mathbf{\Lambda} = \operatorname{diag}(\lambda_1, \ldots, \lambda_d).$$

We use OLS on $\mathbf{X}, \mathbf{y}$ to obtain a linear fit with estimate $\widehat{\mathbf{w}}$.

(a) **Write down the expression of least squares estimator $\widehat{\mathbf{w}}$ in terms of $\mathbf{w}^\star, \mathbf{V}, \mathbf{\Lambda}$ and $\widetilde{\varepsilon} = \mathbf{U}^\top \varepsilon$.** *No need to derive the expression for OLS.*
Ans. Note that the eigen-decomposition of $\mathbf{\Sigma} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$ reads and therefore

$$\begin{aligned}
\widehat{\mathbf{w}} &= (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y} \\
&= \mathbf{\Sigma}^{-1} \mathbf{X}^\top (\mathbf{X}\mathbf{w}^\star + \varepsilon) \\
&= \mathbf{w}^\star + \mathbf{V}\mathbf{\Lambda}^{-1}\mathbf{\Lambda}^{1/2}\mathbf{U}^\top \varepsilon \\
&= \mathbf{w}^\star + \mathbf{V}\mathbf{\Lambda}^{-1/2}\widetilde{\varepsilon}
\end{aligned}$$

(b) **What is the mean and covariance matrix of the Gaussian random vector $\widetilde{\varepsilon} = \mathbf{U}^\top \varepsilon$?**
Ans. Use the fact that

$$\begin{aligned}
\mathbb{E}_\varepsilon[\widetilde{\varepsilon}\widetilde{\varepsilon}^\top] &= \mathbb{E}_\varepsilon[\mathbf{U}^\top \varepsilon \varepsilon^\top \mathbf{U}] \\
&= \mathbf{U}^\top \mathbb{E}_\varepsilon[\varepsilon\varepsilon^\top]\mathbf{U} \\
&= \mathbf{I}_d
\end{aligned}$$

and mean is zero by linearity of expectation.

(c) We are now given a different test set $\mathbf{X}_{\text{test}}$ with same left/right singular vectors but different singular values, i.e., $\mathbf{X}_{\text{test}} = \mathbf{U}\mathbf{\Lambda}_{\text{test}}^{1/2}\mathbf{V}^\top$. Use previous parts to **show that the expected prediction error on $\mathbf{X}_{\text{test}}$ can be simplified as:**

$$\frac{1}{n}\mathbb{E}_\varepsilon\|\mathbf{X}_{\text{test}}\widehat{\mathbf{w}} - \mathbf{X}_{\text{test}}\mathbf{w}^\star\|^2 = \frac{1}{n}\sum_{i=1}^{n}\frac{\lambda_i^{\text{test}}}{\lambda_i}.$$

Note that the eigenvalues, λ_i^{test}, correspond to the same i-th eigenvector in $\mathbf{V}$, for $i = 1, \ldots, d$ and we take the expectation over the training noise variables ε.

Ans. Noting that $\mathbf{\Sigma}_{\text{test}} = \mathbf{V}\mathbf{\Lambda}_{\text{test}}\mathbf{V}^\top$, the error on the new $\mathbf{X}_{\text{test}}$ with covariance matrix $\mathbf{\Sigma}_{\text{test}}$ is

$$\begin{aligned}
\mathbb{E}_\varepsilon\|\mathbf{X}_{\text{test}}\widehat{\mathbf{w}} - \mathbf{X}_{\text{test}}\mathbf{w}^\star\|^2 &= \mathbb{E}(\widehat{\mathbf{w}} - \mathbf{w}^\star)^\top\mathbf{\Sigma}_{\text{test}}(\widehat{\mathbf{w}} - \mathbf{w}^\star) \\
&= \mathbb{E}\widetilde{\varepsilon}^\top\mathbf{\Lambda}^{-1}\mathbf{\Lambda}_{\text{test}}\widetilde{\varepsilon} \\
&= \text{trace}(\mathbb{E}[\widetilde{\varepsilon}\widetilde{\varepsilon}^\top]\mathbf{\Lambda}^{-1}\mathbf{\Lambda}_{\text{test}}) \\
&= \text{trace}(\mathbf{\Lambda}^{-1}\mathbf{\Lambda}_{\text{test}}) = \sum_{i=1}^{n}\frac{\lambda_i^{\text{test}}}{\lambda_i}
\end{aligned}$$

where the third equality follows from the linearity of expectation and the hint, and the fourth equality follows from (b).

(d) In practice, we sometimes have a choice of training sets. Let's consider a concrete scenario with $d = 2$. We assume for simplicity that $\mathbf{V} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ for all covariance matrices to follow. Assume that we can choose to obtain noisy observations $\mathbf{y} = \mathbf{X}^\top \mathbf{w}^\star + \varepsilon$ for three possible training feature matrices $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3$ whose covariance matrices have the following diagonal eigenvalue matrices

$$\mathbf{\Lambda}_1 = \begin{pmatrix} 10 & 0 \\ 0 & 10 \end{pmatrix} \qquad \mathbf{\Lambda}_2 = \begin{pmatrix} 1 & 0 \\ 0 & 100 \end{pmatrix} \qquad \mathbf{\Lambda}_3 = \begin{pmatrix} 100 & 0 \\ 0 & 0.1 \end{pmatrix}.$$

First, we are given a test set $\mathbf{X}_{\text{test}}$ (with the same singular vectors as $\mathbf{X}$) with diagonal eigenvalue matrix $\mathbf{\Lambda}_{\text{test}} = \begin{pmatrix} 0.01 & 0 \\ 0 & 100 \end{pmatrix}$ and are asked to **minimize the average expected prediction error $\frac{1}{n}\mathbb{E}_\varepsilon \|\mathbf{X_{test}}\widehat{\mathbf{w}} - \mathbf{X_{test}}\mathbf{w}^\star\|_2^2$. Which training feature matrix do you choose and why?**

Ans. For test set: The best choice is feature matrix number 2 is the best. Evident from formula in previous subpart as the sum of the ratio is minimized among the different diagonal matrices (first has sum ~ 10, second has sum ~ 1, third has sum ~ 100) - but also intuitively because the second eigenvector is weighted heavily in the test set, and thus needs to be better "explored" in the training set.

(e) Now you are asked to make a choice of training data from the previous part's $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3$ which **minimizes the worst-case expected error under any possible $\mathbf{x_{test}}$ with unit norm. Which feature matrix do you choose and why?**

Ans. For worst case: The worst case error only depends on the sum of inverse eigenvalues. In that case feature matrix number 1 is the best because the first has error ~ 0.1, second has ~ 1, third has ~ 10.

$$\mathbb{E} \max_{\|x\|=1} \left(x^\top(\hat{w} - w^*)\right)^2 = \mathbb{E}\, \|\hat{w} - w^*\|^2$$

Cauchy Schwarz

$$= \sum_{i=1}^{d} \frac{1}{\lambda_i}$$

$$\max_{\|x\|=1} |x^\top a| = \|a\|$$

4 Generalized IRWLS (2+2+4+3, 11 pts)

Suppose that given n data points $\mathcal{D} = \{(\mathbf{x}_i, y_i) \in \mathbb{R}^d \times \mathbb{N}^*\}_{i=1}^n$ we fit a generalized linear model with a Poisson distribution and a log-linear link function. Here $\mathbb{N}^*$ denotes the set of all natural numbers and zero. In simple words, given a fixed feature $\mathbf{x}$ we model the distribution of the response variable as a Poisson random variable,

$$P(Y = y|\lambda) = e^{-\lambda}\frac{\lambda^y}{y!}$$

where the parameter λ is related to the feature $\mathbf{x}$ via the log-linear link function, i.e.,

$$g(\lambda) = \ln \lambda = \mathbf{x}^\top \beta.$$

This set-up is called GLM with a log-linear link function.

(a) Assuming the data $\mathcal{D}$ is i.i.d., derive the expression for the log-likelihood $\mathcal{L}(\beta)$ of the data in terms of $\mathbf{x}_i, y_i$ and β.

$\log P(Y=y|\lambda) = -\lambda + y\ln\lambda - \ln y!$

$\Rightarrow \mathcal{L}(\beta) = \sum_{i=1}^n \log P(Y=y_i | x=x_i;\beta) = \sum_{i=1}^n \log P(Y=y_i|\lambda = e^{x_i^T\beta})$

$= \sum_{i=1}^n \left(-e^{x_i^T\beta} + y_i x_i^T\beta - \ln(y_i!)\right)$

(b) Does the maximum likelihood estimator for β have a closed form expression? Justify.

No. Because $\nabla_\beta \mathcal{L}(\beta) = 0$ does not have closed form solu.

$\nabla_\beta \mathcal{L}(\beta) = \sum_{i=1}^n -e^{x_i^T\beta}x_i + y_i x_i = 0$

$\Rightarrow \sum_{i=1}^n e^{x_i^T\beta}x_i = \sum_{i=1}^n y_i x_i$

$\longrightarrow$ no direct solution.

(c) Derive the Newton update for maximizing $\mathcal{L}$. Recall that the update is given by

$$\beta_{k+1} = \beta_k - [\nabla^2_\beta \mathcal{L}(\beta_k)]^{-1} \nabla_\beta \mathcal{L}(\beta_k).$$

$$\nabla_\beta \ell(\beta) = \sum_{i=1}^{n} -e^{x_i^T\beta} x_i + y_i x_i = X^T(y - p_k) \quad \Big| \quad p_k = \begin{bmatrix} e^{x_1^T\beta_k} \\ \vdots \\ e^{x_n^T\beta_k} \end{bmatrix}$$

$$\nabla^2_\beta \ell(\beta) = \sum_{i=1}^{n} -e^{x_i^T\beta} x_i x_i^T = -X^T W_k X \quad \Big| \quad W_k = \text{diag}(p_k)$$

$$\beta_{k+1} = \beta_k + (X^T W_k X)^{-1} X^T (y - p_k)$$

(d) Discuss why the update obtained in previous part can be seen as an iterative re-weighted least squares algorithm. You should express β_{k+1} in the form $(\mathbf{X}^\top \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{W} \mathbf{z}$ for appropriate choices of $\mathbf{W}$ and $\mathbf{z}$.

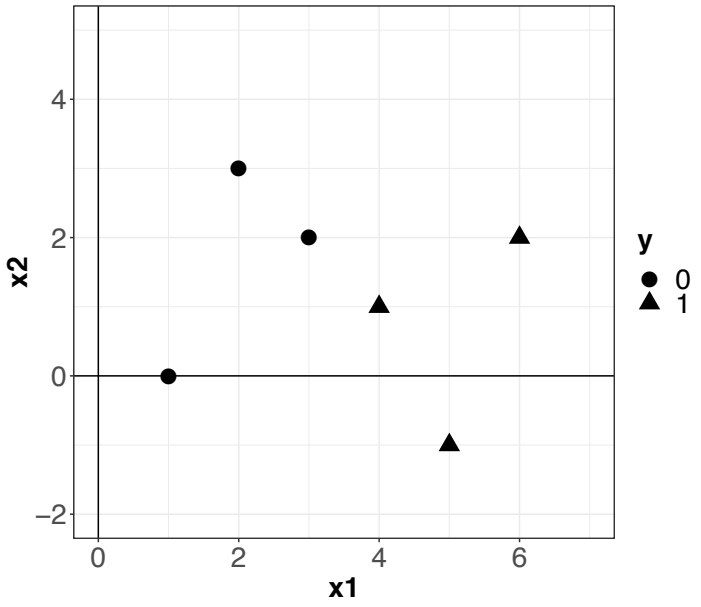
Clearly $\beta_{k+1} = (X^T W_k X)^{-1} \left[(X^T W_k X) \beta_k + X^T W_k W_k^{-1} (y - p_k) \right]$

$$= (X^T W_k X)^{-1} X^T W_k \underbrace{\left(X\beta_k + W_k^{-1}(y - p_k) \right)}_{z_k}$$

which is the solution of a WLS problem. Since W_k and z_k change with k, its called iterative - reweighted LS.

5 Is linear classification easy?, (1*3+3+3+4+2+3, 18 pts))

Consider the toy-dataset displayed below. We have 6 data points, with two dimensional feature vector and the labels taking discrete values in $\{-1, 1\}$. In short, we have $(\mathbf{x}_i, y_i) \in \mathbb{R}^2 \times \{0, 1\}$ for $i = 1, \ldots, 6$. The scatter plot for the data is shown below.



We now investigate the performance of different methods on this dataset. Assume that we use **all six** data-points for training our models. We use the notation $\mathbf{x}_i = \begin{bmatrix} x_{i,1} \\ x_{i,2} \end{bmatrix}$ to denote the two features of the i-th sample point.

(a) Both LDA and QDA will get 100% training accuracy on this dataset.

● True ○ False

(b) Using linear regression for this problem does not make sense since the goal here is to do classification.

○ True ● False

(c) We can not use Soft-margin SVM here since the data is linearly separable.

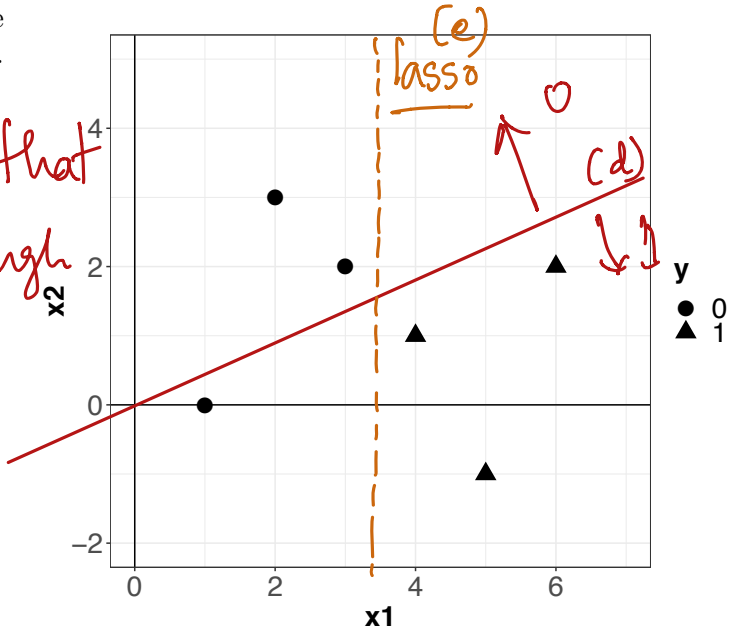
○ True ● False

(d) Raaz decides to use logistic regression for this data but forgets to add intercept in his model. That is he tries to solve the following optimization problem:

$$\min_{\beta_1,\beta_2} \sum_{i=1}^{n} -y_i(x_{i,1}\beta_1 + x_{i,2}\beta_2) + \log(1 + e^{x_{i,1}\beta_1 + x_{i,2}\beta_2}) \tag{2}$$

What can go wrong with such a mistake? What is the **best possible decision boundary** that his model (2) problem can find and what is the corresponding accuracy?

Draw your boundary in the figure and report your accuracy below. Provide a brief reasoning.



No intercept implies that The line has to pass through origin.

error: 1/6 ≈ 16% | Accuracy = 5/6 ≈ 84%

Read the next few pages for more details.

(e) Yuansi likes sparse models and decides to use ℓ_1-regularization where $\lambda > 0$ denotes the penalization factor. Let β_0 denote the intercept and β_1, β_2 the usual coefficients as in equation (2). **What loss function does he need to optimizer?** Your answer should be a slight modification of equation (2). Also suggest a solution that Yuansi's model would find when λ is large (but not too large).

$$\sum_{i=1}^{n} -y_i(\beta_0 + x_{i,1}\beta_1 + x_{i,2}\beta_2) + \log(1 + e^{\beta_0 + x_{i,1}\beta_1 + x_{i,2}\beta_2}) + \lambda(|\beta_1| + |\beta_2|)$$ ← no penalty for β_0.

λ large → sparse solution (here vertical / horizontal axis)

boundary $\Rightarrow x_1 - 3.5 \overset{14}{=} 0 \Rightarrow \beta_0 = -3.5$, $\beta_1 = 1$, $\beta_2 = 0$

Read the next few pages for more details.

Logistic Regression, LDA and SVM on (raw features) are all linear methods, in the sense that they have linear decision boundary for predicting y.

LDA: Decision boundary is given by solving

$$P(y=1 \mid X=x) = P(y=0 \mid X=x)$$

$$\Rightarrow \underbrace{x^T \hat{\Sigma}^{-1}(\hat{\mu}_1 - \hat{\mu}_0)}_{\hat{\beta}} + \underbrace{\left[\ln \frac{P(Y=0)}{P(Y=1)} - \frac{\hat{\mu}_1^T \hat{\Sigma}^{-1} \hat{\mu}_1 - \hat{\mu}_0 \hat{\Sigma}^{-1} \hat{\mu}_0}{2}\right]}_{\hat{\beta}_0} = 0$$

(notes 18 lda_qda)

$$\Rightarrow \boxed{x^T \hat{\beta} + \hat{\beta}_0 = 0}$$ Linear Boundary.

Logistic Regression: You learn $\hat{\beta}$ (and $\hat{\beta}_0$ if fitting an intercept) by maximizing log-likelihood (or minimizing logistic loss) and then the prediction is done by an appropriate choice of threshold α

$$\hat{Y} = \begin{cases} 1 & \text{if } P(Y=1 \mid X=x; \hat{\beta}, \hat{\beta}_0) > \alpha \\ 0 & \text{if } P(Y=0 \mid X=x; \hat{\beta}, \hat{\beta}_0) \leq \alpha \end{cases}$$

So the decision boundary is

$$P(Y=1 \mid X=x; \hat{\beta}, \hat{\beta}_0) = \alpha$$

$$\Rightarrow \quad \frac{e^{\hat{\beta}^T x + \hat{\beta}_0}}{1 + e^{\hat{\beta}^T x + \hat{\beta}_0}} = \alpha$$

$$\Rightarrow \quad \hat{\beta}^T x + \hat{\beta}_0 = \ln\left(\frac{\alpha}{1-\alpha}\right)$$

↑ linear boundary in x

SVM: The problem formulation is already as a linear classifier (the labels are traditionally chosen as ± 1 in 2-class SVM)

$$\hat{y} = \text{sign}(\hat{\beta}^T x + \hat{\beta}_0) = \begin{cases} +1 & \text{if } \hat{\beta}^T x + \hat{\beta}_0 > 0 \\ -1 & \text{if } \hat{\beta}^T x + \hat{\beta}_0 < 0 \end{cases}$$

and the boundary is

$$\hat{\beta}^T x + \hat{\beta}_0 = 0 \leftarrow \text{linear } //$$

The three methods above differ in the way $\hat{\beta}$ (and $\hat{\beta}_0$) are estimated.

LDA: You estimate $\hat{\mu}_1$, $\hat{\mu}_0$ and $\hat{\Sigma}$ using MLE with Gaussian assumption on $X|Y$ and then $\hat{\beta}$ and $\hat{\beta}_0$ pop-out automatically.

Logistic: You estimate $\hat{\beta}_1$ and $\hat{\beta}_0$ by MLE with assumption $Y|X \sim \text{Ber}\left(\frac{e^{\beta^T x+\beta_0}}{1+e^{\beta^T x+\beta_0}}\right)$

SVM: You estimate $\hat{\beta}_1$ and $\hat{\beta}_0$ by minimizing the loss: $(y_i \in \{\pm 1\})$

$$\min_{\beta, \beta_0} \sum_{i=1}^{n} \max\{0, 1 - y_i(\beta^T x_i + \beta_0)\} + \lambda \|\beta\|^2$$

Continued ...

Regularized logistic Regression: the loss for Logistic Regression is given by

$$\min_{\beta, \beta_0} \underbrace{\sum_{i=1}^{n} -y_i (x_i^T \beta + \beta_0) + \log(1 + e^{x_i^T \beta + \beta_0})}_{L(\beta, \beta_0)}$$

Like OLS, we can add penalty here to regularize the model.

If we want sparsity objective becomes

$$\min_{\beta, \beta_0} L(\beta, \beta_0) + \lambda \|\beta\|_1$$

- Note that the decision boundary still remains linear, just that β may (not necessarily) become sparse.

→ One can also add l_2-regularization

$$\min_{\beta} L(\beta, \beta_0) + \lambda \|\beta\|^2$$

to control the norm of β.

⇒ In either case, the optimal $\hat{\beta}$ would change but the boundary (for decision rule) will remain linear!

→ In part (e), the question asked for a solution that lasso might find. We cannot guess the solution looking at the objective, but we can guess a sparse solution. What does the boundary for a sparse solution look like

$$X_1\beta_1 + X_2\beta_2 + X_3\beta_3 + \cdots + X_d\beta_d + \beta_0 = 0$$

Some of them can become zero

in our case $d=2$: so two options for 1-sparse

$X_1\beta_1 + \beta_0 = 0 \Rightarrow$ vertical boundary

or $X_2\beta_2 + \beta_0 = 0 \Rightarrow$ horizontal boundary

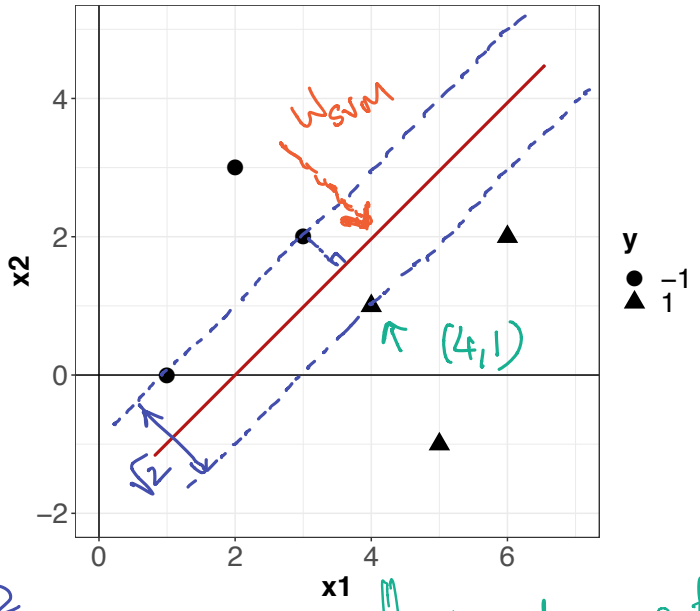
or 2 sparse $\Rightarrow \beta_0 = 0 \leftarrow$ trivial boundary

The question asked to suggest some sparse solution and we can see a vertical boundary would do well, and thus a "possible" sparse solution is $X_1 - 35 = 0$

Ofcourse we cannot guarantee if lasso would converge to it since lasso does not have a closed form expression so its only a guess albeit an informed one!

(f) And now comes your turn. You recall that SVM makes not much assumptions and works well on linearly separable data and decide to train a Hard-margin SVM. Draw the decision boundary for your classifier. Also compute the exact expression for the decision boundary. (We have changed the y-labels to ± 1 to avoid any confusion.)

Draw your boundary in the figure and report your accuracy below. Use the space below to derive the equation for the decision boundary.



Read the next few pages for more details.

Accuracy = 100%
(linearly separable data)

① margin $\Rightarrow m = \frac{1}{\sqrt{2}}$

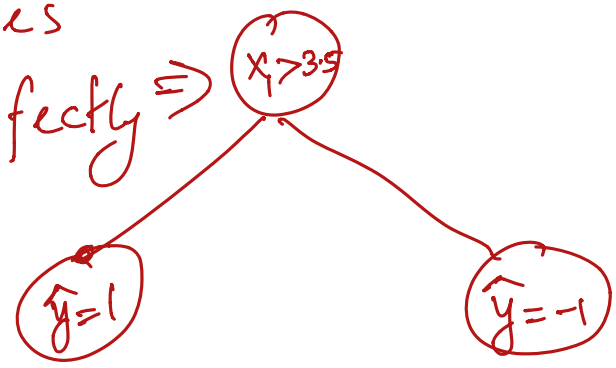
② $w = (a, -a)$ & $\|w\| = \sqrt{2}$
$\Rightarrow w = (1, -1)$

$w^T x + b \big|_{x=(4,1)} = 1$
$\Rightarrow x_1 - x_2 + b = 1 \Rightarrow b = 2$ $\Rightarrow$ $x_1 - x_2 - 2 = 0$

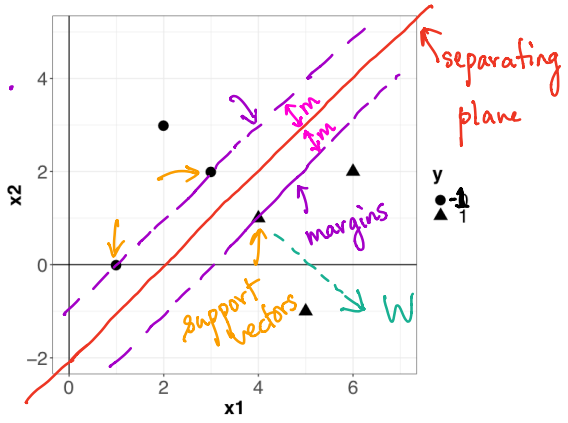
Direct computation from graph also acceptable.

(g) Bin likes simple rules and decides to train a decision tree using stumps on this dataset. **Describe the decision tree that she would obtain**.

Stump classifies the data perfectly $\Rightarrow$



(f): For SVM: We have to guess the nearest points and then draw the boundary and margins.



From the figure we see that w should be in the direction

$$\frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix}$$

and its norm is chosen $m = \frac{1}{\|w\|}$

where m is $\frac{1}{2}$ the distance between nearest points b/w the classes.

$$\Rightarrow \quad m = \frac{1}{2}\sqrt{2} = \frac{1}{\sqrt{2}}$$

$$\Rightarrow \quad \|w\| = \sqrt{2} \qquad \text{direction} = \begin{pmatrix}1\\-1\end{pmatrix}\frac{1}{\sqrt{2}}$$

$$\Rightarrow \quad w = \|w\|\,\hat{w} = \begin{pmatrix}1\\-1\end{pmatrix}$$

In hard-margin SVM,
The intercept is chosen such that

$y(w^T x + b) = 1$ for support vectors

picking the point $x = \begin{pmatrix} 4 \\ +1 \end{pmatrix}$ and $y = +1$

we get

$$1\left((1,-1) \begin{pmatrix} 4 \\ +1 \end{pmatrix} + b \right) = 1$$

$$\Rightarrow \quad b = -2$$

$\Rightarrow$ the equation becomes

$$w_1 x_1 + w_2 x_2 + b = 0$$

$$\Rightarrow \quad x_1 - x_2 - 2 = 0$$

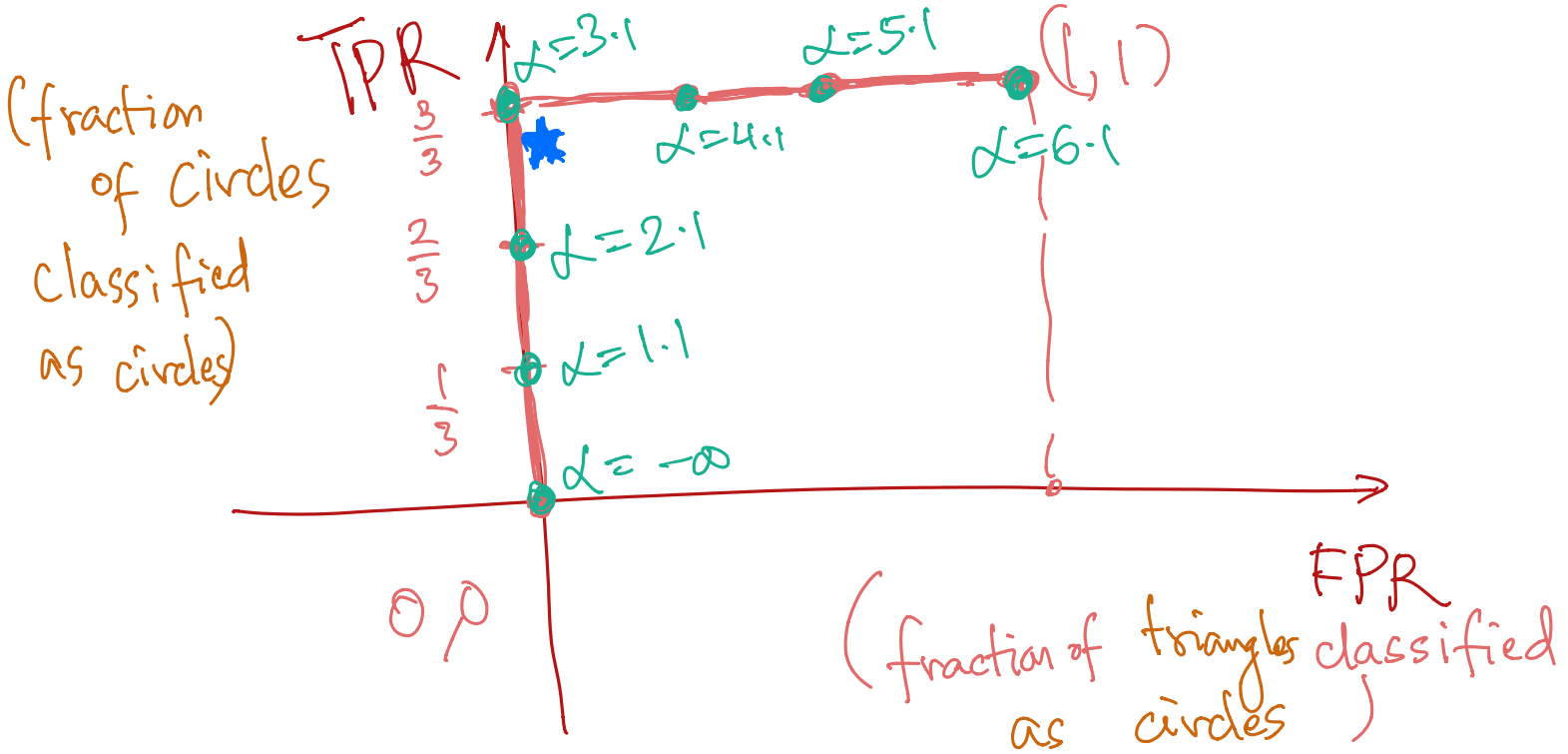
(h) Suppose we want to investigate the ROC curve for prediction of class −1, for the following simple classification rule:

circles

Circles

$$\widehat{y}_i(\alpha) = \begin{cases} -1 & \text{if } x_{i,1} < \alpha \\ 1 & \text{if } x_{i,1} \geq \alpha \end{cases}$$

Draw the ROC curve for true and false positive rates for this set of functions with respect to the class −1. Can you find a good model from this ROC curve? Show work.



The rates change only at first coordinates of data points.

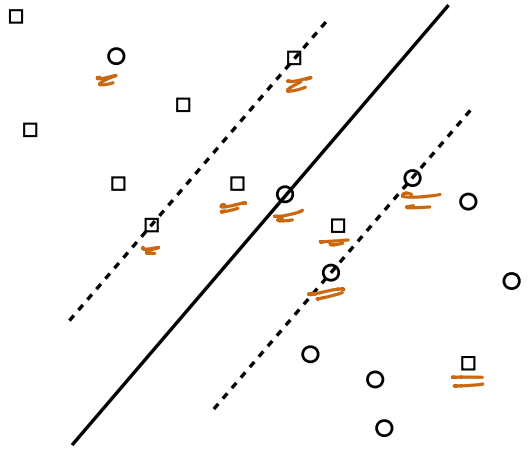
Clearly $\alpha = 3.1$ is a good point!

6 Support vector machine (16 pts)

Recall the soft margin SVM formulation:

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n}\xi_i$$
$$\text{s.t. } \xi_i \geq 0 \quad \text{and} \quad y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \text{for all } i = 1, \ldots n.$$

Here ξ_i denotes the slack for data point i. In the next figure, the training data is labeled as $y_i \in \{-1, 1\}$ and is represented as circles and squares respectively for clarity.



* for hard-margin support vectors are on margin (dotted lines)

* for soft-margin all points on margin & inside it (mis-classified) are support vectors.

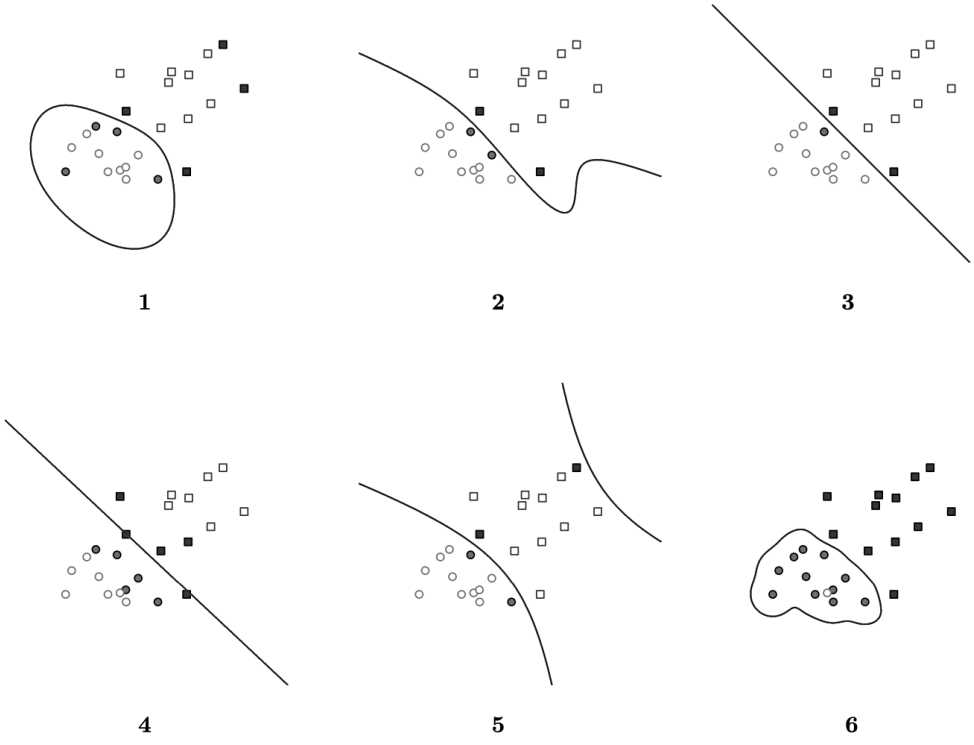
(a) (6 pts) In the figure above, determine the number of

underlined above →

(i)	support vectors	2+2+3+2=9
(ii)	points with $\xi_i = 0$	7+6=13
(iii)	points with $\xi_i \in (0, 1)$	0+1=1
(iv)	points with $\xi_i = 1$	1+0=1
(v)	points with $\xi_i > 1$	1+2=3
(vi)	points with $\xi_i < 0$	0

} 18

(b) (5*2=10 pts) In the figure below, there are different Kernel SVMs with different shapes/patterns of decision boundaries. The training data is labeled as $y_i \in \{-1, 1\}$, represented as the shape of circles and squares respectively. The points corresponding to support vectors have been darkened (solid circles and solid squares). Match the scenarios described below to one of the 6 plots (note that one of the plots does not match to anything). Each scenario should be matched to a unique plot. Explain in less than two sentences why it is the case for each scenario.



(i) A soft-margin linear SVM with $C = 0.02$

4

(bad classifier)

linear classifiers

(ii) A soft-margin linear SVM with $C = 20$.

3

(like hard-margin)

(iii) A hard-margin kernel SVM with Kernel $k(\mathbf{x}, \mathbf{z}) = (1 + \mathbf{x}^\top \mathbf{z})^2$

perfectly separated, ellipse like boundary

1/5 or hyperbola

(iv) A hard margin kernel SVM with Kernel $k(\mathbf{x}, \mathbf{z}) = e^{-5\|\mathbf{x}-\mathbf{z}\|_2^2}$

perf. separated, very strong classifier ✓ sharp boundary

6

(v) A hard margin kernel SVM with Kernel $k(\mathbf{x}, \mathbf{z}) = e^{-\frac{1}{5}\|\mathbf{x}-\mathbf{z}\|_2^2}$

perf. Separated, Smooth classifier ✓

1

Help-sheet

- Given features $\mathbf{X} \in \mathbb{R}^{n \times d}$ and responses $\mathbf{y} \in \mathbb{R}^n$, if the feature matrix $\mathbf{X}$ is full column rank, the OLS solution is given by

$$\widehat{\beta}^{\text{OLS}} = \arg\min_{\beta} \|\mathbf{X}\beta - \mathbf{y}\|_2^2 = (\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top \mathbf{y}.$$

- The pdf/PMF of a few distributions is given below:

Distribution	**Notation**	**pdf** (p) **/ PMF** $(\mathbb{P})$
Multi-variate Gaussian	$Z \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	$p(Z = \mathbf{z}) = \frac{1}{\sqrt{\det(2\pi\boldsymbol{\Sigma})}} \exp\left(-\frac{(\mathbf{z}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{z}-\boldsymbol{\mu})}{2}\right)$
Exponential	$Z \sim \text{Exponential}(\lambda)$	$p(Z = z) = \lambda e^{-\lambda z}, \quad z \geq 0.$
Bernoulli	$Z \sim \text{Bernoulli}(\alpha)$	$\mathbb{P}(Z = z) = \alpha^z (1-\alpha)^{(1-z)}, \quad z \in \{0, 1\}$
Poisson	$Z \sim \text{Poisson}(\lambda)$	$\mathbb{P}(Z = z) = e^{-\lambda} \frac{\lambda^z}{z!}, \quad z \in \{0, 1, 2, \ldots\}.$

- Jensen's inequality for concave function $g : \mathbb{R} \to \mathbb{R}$

$$g(\sum_{i=1}^n \alpha_i x_i) \geq \sum_{i=1}^n \alpha_i g(x_i), \quad \text{where} \quad \alpha_i \geq 0, \ \sum_{i=1}^n \alpha_i = 1.$$

- Taylor expansion.

$$\begin{aligned} \exp(x) &= \sum_{n=0}^{\infty} \frac{x^n}{n!} \\ &= 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \cdots . \end{aligned}$$

Fun Space

Feel free to draw/write something if you want to or give us suggestions or complaints. You can also use this space to report anything suspicious that you might have noticed.

Space for rough work (will NOT be graded)

Space for rough work (will NOT be graded)