# CS 189 Summer 2019 Introduction to Machine Learning — Final Exam

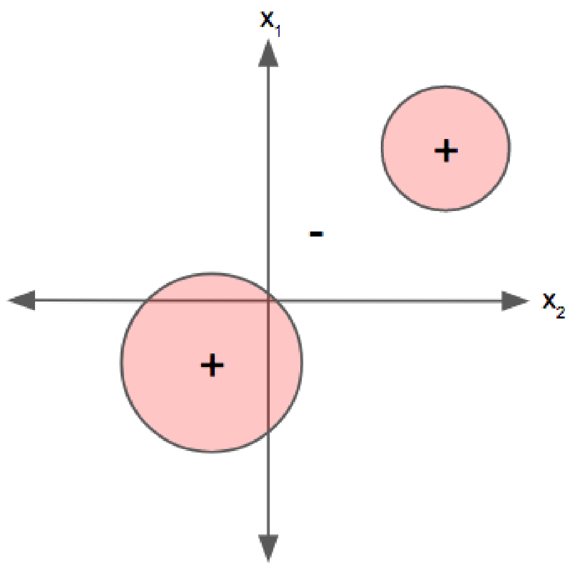
- Please do not open the exam before you are instructed to do so.
- The exam is closed book, closed notes except your two-page cheat sheet.
- **Electronic devices are forbidden on your person**, including cell phones, iPods, headphones, and laptops. Turn your cell phone off and **leave all electronics at the front of the room**, or **risk getting a zero** on the exam.
- You have 3 hours.
- Please write your initials at the top right of each page after this one (e.g., write "MK" if you are Marc Khoury). Finish this by the end of your 3 hours.
- Mark your answers on the exam itself in the space provided. Do **not** attach any extra sheets.
- The total number of points is 150. There are 26 multiple choice questions worth 3 points each, and 5 written questions worth a total of 72 points.
- For multiple answer questions, fill in the bubbles for **ALL correct choices:** there may be more than one correct choice, but there is always at least one correct choice. **NO partial credit** on multiple answer questions: the set of all correct answers must be checked.

| | |
|---|---|
| First name | |
| Last name | |
| SID | |
| First and last name of student to your left | |
| First and last name of student to your right | |

# Q1. [78 pts] Multiple Answer

Fill in the bubbles for **ALL correct choices**: there may be more than one correct choice, but there is always at least one correct choice. **NO partial credit**: the set of all correct answers must be checked.

**(a)** [3 pts] We would like to find the following decision boundary to classify points into two classes (Denoted +, - in the graph below) using a hard-margin SVM. Choose the **smallest** degree polynomial feature vector that can be used to find this decision boundary?



- ○ Degree 2 polynomial feature vector
- ○ Degree 3 polynomial feature vector
- [✓ Degree 4 polynomial feature vector]
- ○ Degree 5 polynomial feature vector
- ○ Degree 6 polynomial feature vector
- ○ Degree 7 polynomial feature vector

**(b)** [3 pts] Which of the following classification models can be designed to fit the decision boundary in question (a) arbitrarily closely?

- [✓ An arbitrarily deep decision tree with axis aligned splits.]
- [✓ An arbitrarily wide neural network with one hidden layer.]
- ○ Quadratic discriminant analysis
- [✓ SVM using a Gaussian kernel.]

**(c)** [3 pts] Given $n$ points $x_i \in \mathbb{R}^3$, we solve linear regression using Lasso. In other words, we find $w \in \mathbb{R}^3$ that minimizes $|Xw - y|^2 + ||w||_1$. In which situation does Lasso create the sparsest weight vector?

- [✓ An isocontour of $|Xw - y|^2$ is tangent to a vertex of an isocontour of $||w||_1$.]
- ○ An isocontour of $|Xw - y|^2$ is tangent to an edge of an isocontour of $||w||_1$.
- ○ An isocontour of $|Xw - y|^2$ is tangent to a face of an isocontour of $||w||_1$.
- ○ $||w||_1$ does not enforce sparsity.

**(d)** [3 pts] We run the $k$-means algorithm on $n$ points to find $y$ that minimizes the following objective function.

$$\sum_{i=1}^{k} \sum_{y_j=i} |X_j - u_i|^2$$

Which of the following always holds true?

○ Initial means $u_i$ can be chosen such that the algorithm does not terminate.

[✓ $k$-means algorithm may not halt at the optimal solution.]

○ Using the within-cluster objective function instead may change the output of $k$-means.

[✓ Initializing $k$-means with $k$-means++ generally allows the algorithm to converge to a lower objective value.]

**(e)** [3 pts] Which of the following is true about Adaboost?

[✓ Weight decreases for correctly classified points.]

○ Weight decreases for incorrectly classified points.

○ Weight increases for correctly classified points.

[✓ Weight increases for incorrectly classified points.]

**(f)** [3 pts] Which of the following statements are correct?

○ Any kernel matrix cannot have negative entries.

[✓ Any kernel matrix is positive semi-definitive.]

[✓ Applying a kernel in ridge regression gives the same solution as ridge regression with the kernel's corresponding feature transformation.]

○ Kernel ridge regression with polynomial kernel is always better than (least-square) linear regression in terms of population risk.

**(g)** [3 pts] Which of the following statements are correct?

[✓ In the linear regression setting with full row rank ($n < d$), taking $\lambda \to 0$ in the ridge regression solution gives the minimum norm solution that perfectly fits the data.]

○ Gradient descent with momentum decreases the function value in every iteration.

○ Adding momentum to gradient descent requires one more gradient computation in each iteration.

○ A 1-hidden-layer neural network with sigmoid activation under $L_2$ loss has a convex risk.

**(h)** [3 pts] Given training data consisting of an arbitrary set of points in $\mathbb{R}^d$ with arbitrary $\{-1, 1\}$ labels (assuming no two points coincide), which of the following statements are correct:

[✓ 1-nearest-neighbor can perfectly fit the data.]

[✓ There exists a multi-layer perceptron that can perfectly fit the data.]

○ AdaBoost (regardless the training accuracies of base learners), if trained long enough, can perfectly fit the data.

[✓ Polynomial regression, if using $\mathbb{I}\{f(x) \geq 0\}$ as the classification rule, can perfectly fit the data as long as the degree is large enough.]

**(i)** [3 pts] Which of the following neural network training techniques can improve the test accuracy?

- [✓ $L_2$ regularization]
- [✓ Early stopping]
- [✓ Dropout]
- ○ Pray

**(j)** [3 pts] Which of the following statements are correct:

- ○ Assuming the training set is drawn i.i.d. from the underlying distribution, empirical risk minimization for some model is the optimal way we could do to minimize the population risk.
- [✓ Random Forest is not equivalent to bagging decision trees.]
- ○ For a neural network trained with stochastic gradient methods, the randomness in initialization uniquely determines which local minimum the final solution is.
- ○ AdaBoost is always a better algorithm for combining decision trees than Random Forest in terms of test accuracy.

**(k)** [3 pts] Which of the following models are commonly used for classification?

- [✓ Random forests]
- [✓ $k$-Nearest Neighbors]
- [✓ Neural networks]
- [✓ Logistic Regression]

All can be used for classification

**(l)** [3 pts] Which of the following models are commonly used for regression?

- [✓ Random forests]
- [✓ Decision Trees]
- [✓ Neural networks]
- ○ $k$-Means

**(m)** [3 pts] Which of the following are first-order optimization algorithms?

- [✓ Gradient descent]
- ○ Newton's method
- [✓ Momentum]
- ○ Binary search

**(n)** [3 pts] Which of the following are true?

- [✓ Positive definite matrices are invertible.]
- ○ Symmetric matrices can have imaginary eigenvalues.
- [✓ Positive semidefinite matrices can be invertible.]
- [✓ $X^T X$ is positive semidefinite for any matrix $X$.]

**(o)** [3 pts] What is the major reasons entropy is used over the % misclassified function (number of points not part of the majority class) as the cost of a set of points when choosing split values for a decision tree?

- ○ Maximizing entropy encourages splits that have homogeneous children.
- [✓ Entropy is strictly concave]
- [✓ Entropy can differentiate between splits more easily than the % misclassified function.]
- ○ A decision tree that perfectly classifies all points will have the minimum possible entropy (0) at all of its leaves.

While many of these are true, the % misclassified function also satisfies many of these.

**(p)** [3 pts] What are characteristics of random forests?

- ○ Each tree is trained on subsets of data sampled without replacement from the training set.
- [✓ Random forests cannot strictly outperform the best possible decision tree on the training set.]
- [✓ Each split only considers axis-aligned splits on random subsamples of features.]
- ○ Random forests cannot learn nonlinear decision boundaries for classification.

The bottom left choice was thrown out.

For the next two questions, let $Z \sim \mathcal{N}(0, \Sigma)$ be a multivariate Gaussian in $\mathbb{R}^d$ with mean 0 and covariance $\Sigma$. Assume that $X$ has nonzero probability density on all of $\mathbb{R}^d$.

**(q)** [3 pts] Select all the powers of $\Sigma$ which "whiten" $Z$, i.e. select $\Sigma^c$ such that $\text{cov}[\Sigma^c X] = I_d$.

○ $\Sigma^{-1}$ [✓ $\Sigma^{-1/2}$] ○ $\Sigma^{1/2}$ ○ $\Sigma^{1}$ ○ $\Sigma^{2}$

Recall that $\text{cov}(AZ) = A\Sigma A^T$. If we want $A\Sigma A^T = I$ and $A = \Sigma^c$, then $\Sigma^{2c+1} = I$, i.e. $c = -1/2$.

**(r)** [3 pts] For a power $\Sigma^c$, consider the following optimization problem:

$$\underset{\|v\|_2 > 0}{\text{argmax}} \frac{v^\top \Sigma^c v}{v^\top v}$$

Select all powers of $\Sigma$ such that this optimization problem finds a direction of highest variance for $Z$.

○ $\Sigma^{-1}$ ○ $\Sigma^{-1/2}$ [✓ $\Sigma^{1/2}$] [✓ $\Sigma^{1}$] [✓ $\Sigma^{2}$]

For a power of $\Sigma$, optimizing the Rayleigh quotient will return the eigenvector corresponding to the largest eigenvalue of $\Sigma^c$. The eigenvectors don't depend on the powers, but the eigenvalues do, the eigenvalues of $\Sigma$ get mapped $\lambda \mapsto \lambda^c$ for $\Sigma^c$. The Rayleigh quotient $\Sigma$ returns the direction of highest variance by definition, and the largest eigenvalue for $\Sigma$ stays the largest in the general case iff $\lambda \mapsto \lambda^c$ is monotonoically increasing, which happens for $c = 1/2$ and $c = 2$.

**(s)** [3 pts] Which of the following statements are correct?

- [✓ The total explained variance of the $k$ highest-variance components is monotonically increasing with $k$]
- [✓ The components found by PCA are orthogonal]
- ○ The accuracy of PCA can be measured using a validation set
- ○ Dropping the lowest-variance component from PCA is equivalent to least squares linear regression, i.e. finding $\text{argmin}_w \|Xw - y\|$

**(t)** [3 pts] Let $X \in \mathbb{R}^{n \times n}$ be a real symmetric matrix. Given an eigendecomposition $X = W\Lambda W^\top$ and singular value decomposition $X = UDV^\top$, which of the following statements *must* be true the relationship between the two? (Recall the determinant of a diagonal matrix is the product of its diagonal entries.)

- [✓ The rank of $D$ is exactly equal to the rank of the matrix $\Lambda$.]
- [✓ The absolute value of the determinants, $|\det(\Lambda)|$ and $|\det(D)|$ are equal.]
- ○ The columns of $U$ are the same as the columns of $W$, although perhaps in a different order.
- ○ The non-zero entries of $D$ are equal to the non-zero entries $\Lambda$.

The spectral decomposition decomposition of $X$ gives a set of orthonormal vectors, the columns of $W$, which have the following property.

$$\begin{aligned} Xw_i &= \lambda_i w_i && \text{for all } 0 \leq i \leq n \\ &= |\lambda_i| \cdot \underbrace{\text{sgn}(\lambda_i) \cdot w_i}_{u_i} && \text{for all } 0 \leq i \leq n \\ &= |\lambda_i| u_i && \text{for all } 0 \leq i \leq n \end{aligned}$$

Recall from the spectral theorem, the eigenvalues of $X$ are real. Therefore, we have found a set of orthonormal vectors which map to another orthogonal basis after non-negative scaling. Since the singular values of a matrix are unique, this means the singular vectors of $X$ are the absolute values of its eigenvalues.

This shows that the first two options are true and the last is false. Now let $X$ be the 0 matrix; notice that $W$ can be any orthogonal matrix, so clearly option 3 cannot always be true.

**(u)** [3 pts] Consider the simple matrix $A \in \mathbb{R}^{2\times 2}$ defined below. Of the choices below, select all numbers which are singular values of the matrix below.

$$A = \begin{bmatrix} -1 & -1 \\ -1 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} -1 & -1 \end{bmatrix}$$

● 0　　○ $-1$

○ $\sqrt{2}$　　● 2

One way to find the singular values of this matrix is compute $A^\top A$ and solve for its eigenvalues. However, we can also use the following trick:

$$A = \begin{bmatrix} -1 & -1 \\ -1 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} -1 & -1 \end{bmatrix} = 2 \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

From this we can simply read off one singular value as 2. The other must be 0 since $A$ is non-invertible.

**(v)** [3 pts] Which of the following statements are true?

○ If the maximum information gain at each leaf of a decision tree is 0, then there does not exist a decision tree with higher training accuracy.

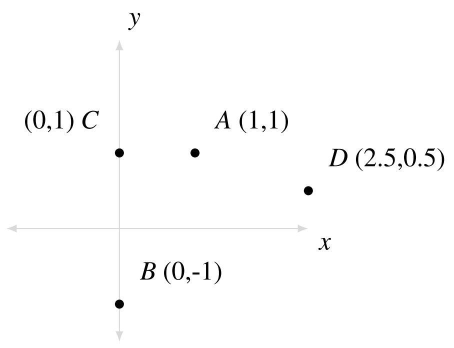
● If there does not exist a decision tree with higher training accuracy, then the maximum information gain at each leaf of a decision tree is 0.

○ If allowed arbitrary depth, axis-aligned decision trees cannot reach the same train accuracy as arbitrary linear boundary decision trees.

○ If depth is fixed, axis-aligned decision trees will always have the same train accuracy as arbitrary linear boundary decision trees.

Axis-aligned decision trees are less powerful than arbitrary linear boundary trees of the same depth, but when allowed arbitrary depth they can reach 100% train accuracy.

For the following questions, consider the following set of points in $\mathbb{R}^2$. We apply the agglomerative (bottom-up) clustering algorithm to the points below, stopping when there are two clusters remaining. (The distance between any two points is simply defined as their Euclidean distance.)



**(w)** [3 pts] Assume we use the centroid linkage distance function. Which of the following could be possible outputs?

○ $\{A, C\}$ and $\{B, D\}$

● $\{A, B, C\}$ and $\{D\}$

● $\{A, C, D\}$ and $\{B\}$

○ $\{A, D\}$ and $\{B, C\}$

When each cluster has a single point, the mean is simply the point itself. Therefore, the clusters $\{A\}$ and $\{C\}$ will get fused first. Notice that their mean is $(0.5, 1)$ which is equidistant from both $B$ and $D$. Therefore, either one could be fused with $\{A, C\}$ to end up with 2 clusters.

**(x)** [3 pts] Now instead assume we use the single linkage distance function. Which of the following could be possible outputs?

○ $\{A, C\}$ and $\{B, D\}$

○ $\{A, B, C\}$ and $\{D\}$

● $\{A, C, D\}$ and $\{B\}$

○ $\{A, D\}$ and $\{B, C\}$

We can simply follow the algorithm. First $A$ and $C$ will be combined, and then it will be merged with $D$

**(y)** [3 pts] You are an astronomer attempting to estimate the Hubble constant $H$. You sample 10 galaxies from the night sky, calculate $H$ for each of them $(\tilde{H}_1, \ldots, \tilde{H}_{10})$, and use as your estimate the sample average for $H$: $\hat{H} = \frac{1}{10}\sum_{i=1}^{10} \tilde{H}_i$.

For a galaxy that is $t$ light-years from Earth, $\tilde{H}(t) \sim \mathcal{N}(\mu = H - He^{-t}, \sigma^2 = t)$ independently for each galaxy. If you can only sample 10 galaxies that share the same distance $t$ and use their average $H$ as your estimate, what choice of $t$ minimizes the MSE $\mathbb{E}[(H - \hat{H})^2]$, assuming the true value of $H$ is $H = 1$?

*Hint:* Use the Bias-Variance decomposition to simplify the MSE.

○ $t = -\ln\left(\frac{1}{10}\right)$

○ $t = 0$

● $t = -\frac{1}{2}\ln\left(\frac{1}{20}\right)$

○ $t = -\frac{1}{10}\ln\left(\frac{1}{2}\right)$

Using the Bias-Variance decomposition gives

$$\mathbb{E}[(H - \hat{H})^2] = (H - \mathbb{E}[\hat{H}])^2 + \text{var}[\hat{H}]$$

For a mean of $n$ i.i.d variables, $\mathbb{E}[\hat{H}] = \tilde{H}(t) = H(1 - e^{-t})$, while $\text{var}[\hat{H}] = \frac{1}{10}\text{var}[\tilde{H}(t)] = \frac{t}{10}$. So the MSE can be written as $\mathbb{E}[(H - \hat{H})^2] = e^{-2t} + \frac{t}{10}$, a convex function. The derivative is $-2e^{-2t} + \frac{1}{10}$, setting that to 0 gives $t = -\frac{1}{2}\ln(\frac{1}{20})$.

**(z)** [3 pts] Assume the same setup for the previous problem, where for a galaxy $t$ light-years from Earth,

$$\tilde{H}(t) \sim \mathcal{N}(\mu = H - He^{-t}, \sigma^2 = t)$$

You sample $n$ different galaxies, except they are now at different distances. You would like to predict $H$ from $t$ by building a regression model from samples $\{(t_1, H_1), \ldots, (t_n, H_n)\}$. Assuming you can add one extra feature, which technique would perform the best for this problem? Note that you do not have to worry about the column of ones, you can assume it has already been added to the dataset.

○ Ordinary least-squares [✓ Weighted least-squares]

○ Ridge regression ○ LASSO

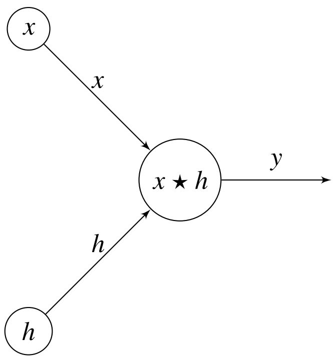
For Gaussian noise with different variances for the data points, the best performer would be weighted least squares. It would be easy to estimate, and would fit the data exactly. In this case, you would like to add the feature $\Phi(t) = e^{-t}$.

# Q2. [16 pts] Cross-Correlational Networks

Sometimes, the input data points contain information that is encoded by local groups of the input. To incorporate this relationship naturally into the model we train, let's define the **discrete-time cross-correlation node**. The node takes two inputs: $x$, an $m$ dimensional vector denoting our input point and $h$, a $m$ dimensional variable vector that we would like to tune via gradient descent. **Please show work in the space below the problem to be considered for partial credit** and all of the multiple choice questions have exactly **one** correct answer.
**For simplicity we will index into our vectors with 0 as the first index and not 1.** The output $y$ is also an $m$ dimensional vector such that:

$$y_j = (x \star h)[j] = \sum_{i=0}^{m-1} x_{(j+i) \bmod m} h_i$$

Recall that the $i$mod$m$ operator refers to the modulus operator, which is the integer remainder when $m$ divides $i$. When $i \geq m$, we wrap around to the front of our signal.



**(a)** [3 pts] Let's practice computing this operation:
If $x^\top = \begin{bmatrix} 1 & 2 & 3 & 2 & 1 \end{bmatrix}$ and $h^\top = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \end{bmatrix}$, what is $y$?

$y^\top = \begin{bmatrix} -1 & -1 & 1 & 1 & 0 \end{bmatrix}$

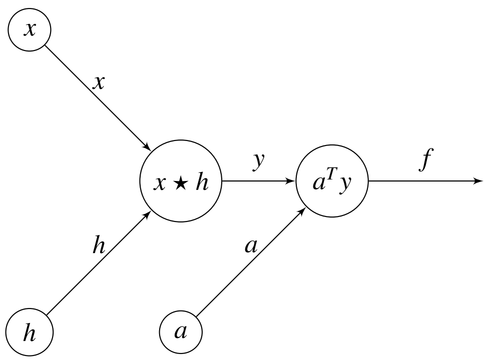
**(b)** [3 pts] What is $\frac{\partial y_j}{\partial h_i}$?

○ $h_{(j-i) \bmod m}$

○ $h_{(j+i) \bmod m}$

● $x_{(j+i) \bmod m}$

○ $x_{(j+i) \bmod m} h_{(j+i) \bmod m}$



**(c)** [4 pts] To integrate this information into a scalar output, we take a dot product of $y$ with another learned vector $a$. We will train this function with gradient descent.

What is $\nabla_h f$? You may leave your answer in terms of $\frac{\partial y_j}{\partial h_i}$ in the box below for partial credit.

| | | | |
|---|---|---|---|
| ○ $a$ | ○ $a \star x$ | ○ $h \star a$ | ○ $a \star x \star h$ |
| ○ $x$ | [✓ $x \star a$] | ○ $a \star h$ | ○ $x \star a \star h$ |
| ○ $y$ | ○ $x \star h \star a$ | ○ $h \star x$ | ○ $h \star a \star x$ |
| ○ $h$ | ○ $a \star h \star x$ | ○ $x \star h$ | ○ $h \star x \star a$ |

$\frac{\partial f}{\partial h_i} = \sum_{j=1}^{m} \frac{\partial f}{\partial y_j} \frac{\partial y_j}{h_i} = \sum_{j=1}^{m} x_{(j+i) \bmod m} a_j = (x \star a)[i]$

**(d)** [4 pts] What is $\nabla_a f$? You may express your answer in terms of $y$ and $a$.

$y$

**(e)** [2 pts] Is $f$ linear in $x$?
[✓ Yes] ○ No
Observe that cross correlation can be rewritten as matrix multiplication of $x$ with a circulant matrix containing entries from $h$. The next operation, a dot-product is also a linear transformation. So, $f$ is a linear function of $x$.

# Q3. [14 pts] MLE for Exponential Families

An *exponential family* is a set of probability distributions $\{p_\theta\}, \theta \in \mathbb{R}^p$ that can be written in the following form:

$$p_\theta(x) = h(x) \exp\left(\theta^\top T(x) - A(\theta)\right)$$

for some $h\colon \mathbb{R} \to \mathbb{R}$, $T\colon \mathbb{R} \to \mathbb{R}^p$, $A\colon \mathbb{R}^p \to \mathbb{R}$.

Exponential families come with some properties that make finding the MLE tractable in most cases.

**(a)** [6 pts] You observe $n$ samples $x_1, \dots, x_n$. Starting with the log-likelihood $L(\theta) = \ln \prod_{i=1}^n p_\theta(X_i)$, write the condition for the MLE $\hat{\theta}$ in the box. This may not be a closed-form solution for $\theta$.

You may use the fact that $\nabla_\theta A(\theta) = \mathbb{E}_\theta[T(X)]$ (the expectation of $T(X)$ under distribution $p_\theta$).

The log-likelihood is:

$$L(\theta) = \sum_{i=1}^n \ln p_\theta(x_i) = \sum_{i=1}^n \ln(h(x_i)) + \theta^\top T(x_i) - A(\theta) = -nA(\theta) + \sum_{i=1}^n \ln(h(x_i)) + \theta^\top \left(\sum_{i=1}^n T(x_i)\right)$$

When we take the gradient, we get:

$$\nabla_\theta L(\theta) = -n\nabla_\theta A(\theta) + \sum_{i=1}^n T(x_i)$$

Using the fact that $\nabla_\theta A(\theta) = \mathbb{E}_\theta[T(X)]$, we get:

$$\frac{1}{n}\sum_{i=1}^n T(x_i) = \mathbb{E}_\theta[T(X)]$$

**(b)** [4 pts] Find the Hessian of $L(\theta)$, $\nabla_\theta^2 L(\theta)$ and write it in the box. You may use the fact that $\nabla_\theta^2 A(\theta) = \mathrm{cov}_\theta[T(X)]$ (the covariance matrix of $T(X)$ under distribution $p_\theta$).

We take the derivative of $\nabla_\theta L(\theta) = -n\nabla_\theta A(\theta) + \sum_{i=1}^n T(x_i)$ once more:

$$\nabla_\theta^2 L(\theta) = -n\nabla_\theta^2 A(\theta) = -n \cdot \mathrm{cov}_\theta[T(X)]$$

**(c)** [4 pts] Argue that any local maximum of $L(\theta)$ is also a global maximum.

Covariance matrices are always PSD, so $n \cdot \mathrm{cov}_\theta[T(x)]$ is PSD for all $\theta$. Therefore $-L(\theta)$ is convex, so $L(\theta)$ is concave. Therefore, any local maximum of $L(\theta)$ is also a global maximum.

# Q4. [15 pts] Random Neural Network

Consider a one hidden layer neural network with $d$ input units, $m$ hidden units and 1 output unit:

$$f(x; V, w) = w^\top \sigma(V^\top x) = \sum_{i=i}^{m} w_i \sigma(x^\top V_i),$$

where $V = (V_1, V_2, \ldots, V_m) \in \mathbb{R}^{d\times m}$ is the weight matrix of the first layer, $w \in \mathbb{R}^m$ is the weight vector of the second layer, $\sigma(\cdot)$ is the sigmoid activation function.

In this problem, we are interested in using network to perform regression with $L_2$ regularization. Due to the limit of computational resources, we will randomly initialize $V$ as $V_0$ and freeze it throughout the training. Namely, we will only train the second layer $w$. To summarize, we are going to solve the following problem by stochastic gradient descent (with batch size 1):

$$\min_w \left\{ \frac{1}{n} \sum_{i=1}^{n} (f(X_i; V_0, w) - y_i)^2 + \lambda \|w\|^2 \right\},$$

where we have training data $(X, y)$ and $X \in \mathbb{R}^{n\times d}, y \in \mathbb{R}^n$.

**(a)** [2 pts] Is this problem convex or non-convex? Write your answer in the box and justify your answer in the blank area.

Convex.

It's basically linear regression with transformed features.

**(b)** [5 pts] Write down the stochastic gradient descent update (i.e, $w_{t+1} = w_t - \eta \cdot \underline{???}$) Please use $\sigma(\cdot)$ to represent the sigmoid function instead of using its explicit form.

$$w_{t+1} = w_t - 2\eta \left( \sigma(x_i^T V_i)(\sigma(x_i^T V_i) w_t - y_i) + \lambda w \right).$$

**(c)** [8 pts] Derive the final solution $f(x; V_0, w)$ that will be found by stochastic gradient descent with $\eta$ small enough. Please use $\sigma(\cdot)$ to represent the sigmoid function instead of using its explicit form, and try to write your answer in a succinct form without summation.

Writing the objective in a more succint way:

$$\min_w \left\{ \frac{1}{n} \|\sigma(XV)w - y\|^2 + \lambda \|w\|^2 \right\}$$

As pointed out in (a), this is basically ridge regression with transformed features. So the solution will be a kernel ridge regression solution with a kernel $K = \sigma(XV)\sigma(XV)^T$. Since the solution of ridge regression could be written as $\hat{w} = (X^\top X + \lambda I)^{-1} X^\top y = X^\top (XX^\top + \lambda I)^{-1} y$, we conclude that the final solution should be

$$\langle x, \sigma(XV)^\top (K + n\lambda I)^{-1} y \rangle,$$

or equivalently

$$\langle x, \left(\sigma(XV)^\top \sigma(XV) + n\lambda I\right)^{-1} \sigma(XV)^\top y \rangle,$$

where $K = \sigma(XV)\sigma(XV)^\top$.

# Q5. [15 pts] Sensitivity in Linear Models

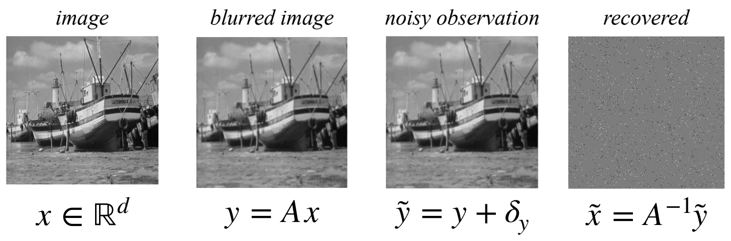
In machine learning, measurement error is often ignored or modelled as zero-mean Gaussian noise for simplicity. In this problem, we consider what happens at the other extreme, exploring the effect of worse-case measurement error on a simple linear model.

$$\begin{array}{cccccc} \text{(input)} & x & \longrightarrow & y = Ax & \text{(output)} \\ \text{(output)} & y & \longrightarrow & \tilde{y} = y + \delta_y & \text{(observation)} \end{array}$$

In this model, assume there is a linear relationship between some hidden features $x \in \mathbb{R}^d$ and an output $y \in \mathbb{R}^d$. However, due to corruption or noise we observe some $\tilde{y} = y + \delta_y$, where $\delta_y$ represents the measurement error. The goal is to recover the hidden features $x$ from the noisy observation $\tilde{y}$.

$$\text{(observation)} \quad \tilde{y} \quad \longrightarrow \quad \tilde{x} = A^{-1}\tilde{y} \quad \text{(estimate)}$$

When $A$ is invertible and known, a seemingly reasonably way to estimate $x$ is simply to compute $\tilde{x} = A^{-1}\tilde{y}$. Throughout this problem, we will try to understand how well this estimate performs. Throughout this problem, assume $A$ *is invertible.*



**(a)** [2 pts] First, let's explore the relationship between the input and output. What is the largest possible value of the ratio below, assuming $x$ is non-zero? Write your answer in terms of the singular values of $A$. Note this ratio is often called the *gain*, since it represents how large the output can be relative to the input.

$$\frac{\|y\|_2}{\|x\|_2}$$

$$\frac{\|y\|_2}{\|x\|_2} = \frac{\|Ax\|_2}{\|x\|_2} = \sqrt{\frac{x^\top A^\top Ax}{x^\top x}} \leq \sqrt{\lambda_{\max}(A^\top A)} = \sigma_{\max}(A)$$

**(b)** [2 pts] The recovery error ($\delta_x$) represents how far the estimate $\tilde{x}$ lies from the true input, $x$. More formally, we write $\delta_x = \tilde{x} - x$. Given this expression for $\delta_x$, write it in terms of $A$, $x$ and $\delta_y$ below. (Remember $A$ is invertible.)

$$\delta_x = \tilde{x} - x = A^{-1}(Ax + \delta_y) - x = A^{-1}\delta_y$$

**(c)** [4 pts] Now assume the measurement error $\delta_y$ can be arbitrary with norm at most $r$. Find the largest possible norm of the resulting recovery error. In other words, find

$$\max \|\delta_x\|_2 \qquad \text{s.t.} \qquad \left\|\delta_y\right\|_2 \leq r.$$

Write your answer in terms of $r$ and the singular values of $A$. For full credit, you must show your work and justify your answer.

We use the result of the previous part to express $\delta_x$ in terms of $\delta_y$. We can then work directly with the optimization problem, massaging it into a form we can immediately evaluate.

$$\begin{aligned}
\max_{\|\delta_y\|_2 \leq r} \|\delta_x\|_2 &= \max_{\|\delta_y\|_2 \leq r} \left\|A^{-1}\delta_y\right\|_2 \\
&= r \cdot \max_{\|u\|_2 \leq 1} \left\|A^{-1}u\right\|_2 && \text{where } \delta_y = ru \\
&= r \cdot \sigma_{\max}(A^{-1}) \\
&= \frac{r}{\sigma_{\min}(A)}
\end{aligned}$$

In the penultimate step, we noticed that the optimum of the program is simply equal to the largest singular value of $A^{-1}$. Since $A$ in invertible, this corresponds to the smallest singular value of $A$.

**(d)** [7 pts] Previous, we only focused on bounding the absolute error; however, error is usually more meaningful when normalized, taking into account the lengths of $x$ and $y$ respectively. Therefore, we are interested in bounding the relative error

$$\frac{\|\delta_x\|_2}{\|x\|_2} \qquad \text{when} \qquad \frac{\|\delta_y\|_2}{\|y\|_2} \leq r.$$

It turns out that we can prove this always less than or equal to $\kappa \cdot r$, for some scalar $\kappa$ which depends only on the singular values of $A$. Find the smallest possible $\kappa$.

*Note.* Partial credit will be awarded for providing this bound in terms of the result in part (b). In this case, please use $c(r, A)$ to denote your result in the previous part. (Note it was a function of both $r$ and $A$.)

We proceed directly. First notice that $\delta_y \leq r\,\|y\|_2$, so from the result above, we can immediately bound $\|\delta_x\|_2$.

$$\begin{aligned}
\frac{\|\delta_x\|_2}{\|x\|_2} &= \frac{1}{\|x\|_2}\,\|\delta_x\|_2 \\
&\leq \frac{1}{\|x\|_2} c(r\,\|y\|_2\,, A) \\
&= \frac{1}{\|x\|_2}\,\frac{r}{\sigma_{\min}(A)}\,\|y\|_2 \\
&= \frac{\|Ax\|_2}{\|x\|_2} \cdot \frac{r}{\sigma_{\min}(A)}
\end{aligned}$$

Now to finish, notice that bounding the first term is what we did in the first part of this problem. Therefore, we have:

$$\begin{aligned}
\frac{\|\delta_x\|_2}{\|x\|_2} &\leq \frac{\|Ax\|_2}{\|x\|_2} \cdot \frac{r}{\sigma_{\min}(A)} \\
&\leq \sigma_{\max}(A) \cdot \frac{r}{\sigma_{\min}(A)} \\
&= \underbrace{\frac{\sigma_{\max}(A)}{\sigma_{\min}(A)}}_{\kappa} \cdot r
\end{aligned}$$

Note that this quantity $\kappa$ is important; it is called the condition number of the matrix and it characterizes the numerical stability of the matrix $A$. In the images above, $A$ was a blurring matrix with an extremely large condition number. As a result, even adding a small amount of noise completely destroys the recovered image.[1]

[1] Source: Vandenberghe, L 2018, *Lecture 15: Problem condition*, ECE133A Applied Numerical Computing, University of California–Los Angeles.

## Q6. [12 pts] Estimation of Linear Models

In all of the following parts, write your answer as the solution to a norm minimization problem, potentially with a regularization term. **You do not need to solve the optimization problem.** Simplify any sums using matrix notation for full credit.
**Hint:** Recall that the MAP estimator maximizes $P(\theta|Y)$: $\hat{\theta} = \arg\max P(Y|\theta)P(\theta)/P(Y) = \arg\max_{\theta\in\mathbb{R}^d} P(Y|\theta)P(\theta)$. The difference between MAP and MLE is the inclusion of a prior distribution on $\theta$ in the objective function.

For the following problems assume you are given $X \in \mathbb{R}^{n\times d}$ and $y \in \mathbb{R}^n$ as training data.

**(a)** [3 pts] Let $y = X\theta + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \Sigma)$ for some positive definite, diagonal $\Sigma$. Write the MLE estimator of $\theta$ as the solution to a weighted least squares problem, potentially with a regularization term.

$\hat{\theta} = \arg\min_{\theta\in\mathbb{R}^d}$ ________________________________________

Observe that $y \sim \mathcal{N}(X\theta, \Sigma)$. From the PDF of y, maximizing the likelihood the data given $\theta$ is equivalent to minimizing:

$$\begin{aligned}\hat{\theta} &= \arg\min_{\theta\in\mathbb{R}^d}(y - X\theta)^\top\Sigma^{-1}(y - X\theta) \\ &= \arg\min_{\theta\in\mathbb{R}^d} \|\Sigma^{-\frac{1}{2}}(y - X\theta)\|_2^2\end{aligned}$$

Observe that this is weighted least squares.

**(b)** [3 pts] Let $y|\theta \sim \mathcal{N}(X\theta, \Sigma)$ for some positive definite, diagonal $\Sigma$. Let $\theta \sim \mathcal{N}(0, \lambda I_d)$ for some $\lambda > 0$ be the prior on $\theta$. Write the MAP estimator of $\theta$ as the solution to a weighted least squares minimization problem, potentially with a regularization term.

$\hat{\theta} = \arg\min_{\theta\in\mathbb{R}^d}$ ________________________________________

Observe that $(y, \theta) \sim \mathcal{N}(X\theta, \Sigma)\mathcal{N}(0, \lambda I_d)$. Finding the MAP estimator of $\theta$ is equivalent to minimizing:

$$\begin{aligned}\hat{\theta} &= \arg\min_{\theta\in\mathbb{R}^d} \frac{1}{2}(y - X\theta)^\top\Sigma^{-1}(y - X\theta) + \frac{1}{2\lambda}\theta^\top\theta \\ &= \arg\min_{\theta\in\mathbb{R}^d} \|\Sigma^{-\frac{1}{2}}(y - X\theta)\|_2^2 + \frac{1}{\lambda}\|\theta\|_2^2\end{aligned}$$

Observe that this is weighted ridge regression.

**(c)** [3 pts] Let $y = X\theta + \epsilon$ where $\epsilon_i \overset{i.i.d.}{\sim} Laplace(0, 1)$. Recall that the pdf for $Laplace(\mu, b)$ is $p(x) = \frac{1}{2b}\exp\left(-\frac{1}{b}|x - \mu|\right)$. Write down the MLE estimator of $\theta$ as the solution to a norm minimization optimization problem.

$\hat{\theta} = \arg\min_{\theta\in\mathbb{R}^d}$ ________________________________________

Observe that $y_i \sim Laplace(x_i^\top \theta, 1)$.

$$
\begin{aligned}
\hat{\theta} &= \arg\max_{\theta \in \mathbb{R}^d} \prod_{i=1}^{N} \frac{1}{2} \exp\left(-|y_i - x_i^\top \theta|\right) \\
&= \arg\min_{\theta \in \mathbb{R}^d} \sum_{i=1}^{N} |y_i - x_i^\top \theta| \\
&= \arg\min_{\theta \in \mathbb{R}^d} \|y - X\theta\|_1
\end{aligned}
$$

Observe that this estimator minimizes the sum of absolute differences.

**(d)** [3 pts] Let $y|\theta \sim \mathcal{N}(X\theta, \Sigma)$ for some positive definite, diagonal $\Sigma$. Let $\theta_i \overset{i.i.d.}{\sim} Laplace(0, \lambda)$ for some positive scalar $\lambda$. Write the MAP estimator of $\theta$ as the solution to a weighted least squares minimization problem, potentially with a regularization term.

$\hat{\theta} = \arg\min_{\theta \in \mathbb{R}^d}$ ________________________________________

$$\begin{aligned}\hat{\theta} &= \arg\min_{\theta \in \mathbb{R}^d} \sum_{i=1}^{d} \frac{1}{\lambda}|\theta_i| + \frac{1}{2}(y - X\theta)^\top \Sigma^{-1}(y - X\theta) \\ &= \arg\min_{\theta \in \mathbb{R}^d} \|\Sigma^{-\frac{1}{2}}(y - X\theta)\|_2^2 + \frac{2}{\lambda}\|\theta\|_1\end{aligned}$$

Observe that this is weighted LASSO regression.