CS 1810 2025 Midterm Practice Questions (Questions Only)

Midterm Topic List

The best way to prepare for the midterm is to review homeworks, section notes, lecture recaps, lecture concept checks, and the midterm practice questions.

The midterm will be conceptual and analytical, testing ideas and understanding. You are not expected to memorize formulas such as PDFs, or to memorize matrix cookbook rules, but you should be familiar with methods of probability theory (e.g. Bayes Rule) and the various models we've studied so far in the course.

Below is a brief list of topics that you might be asked about (this list emphasizes the main focus areas but is not fully inclusive):

- **Linear Regression:** Least squares loss, differentiating and solving for weights analytically, working with alternate (simple) loss functions, and understanding parametric vs. non-parametric regression.
- **Basis Functions:** General idea (not specific versions).
- **Generative Model of Linear Regression:** Incorporating noise and using maximum likelihood estimation.
- **Linear Classification:** Perceptron algorithm, hinge loss; logistic regression (understanding derivative without memorizing it); shapes of decision boundaries; and various loss functions (e.g., hinge vs. 0/1 vs. logistic).
- **Generative Classification:** Class-conditional distributions (e.g. Gaussian or categorical/Naive Bayes), using Bayes Rule for prediction, and maximum likelihood estimation. (Note: Logistic regression here is viewed as discriminative since it does not model $p(x, y)$.)
- **Bias-Variance Trade-off:** Understanding the roles of bias and variance and their connection to over-fitting.
- **Validation and Regularization:** Using (cross-)validation for model selection and avoiding over-fitting; understanding regularization (especially in linear regression problems).
- **Bayesian Methods:** Terminology, MAP estimation, posterior predictive, and the use of conjugate distributions (e.g., Beta-Bernoulli, Normal-Normal) without needing to memorize PDF forms.
- **Neural Networks:** Notation for weights and layers, use of sigmoid and ReLU activations (applied element-wise), forward propagation and back-propagation, and their use in classification and regression.
- **Support Vector Machines (SVM):** Hard max-margin formulation, soft margin formulation, dual formulation, and the kernel trick.

Midterm Practice Questions

1. Linear Regression

Consider a one-dimensional regression problem with training data $\{(x^{(n)}, y^{(n)})\}$. We seek to fit a linear model with no bias term:

$$\hat{y} = wx.$$

(a) Assume a squared loss

$$\frac{1}{2}\sum_{i=1}^{N}\left(y^{(i)} - \hat{y}^{(i)}\right)^2.$$

Solve for the optimal value of w^*.

(b) Suppose we have a generative model of the form

$$\hat{y} = wx + \epsilon,$$

where $\epsilon \sim N(0, \sigma^2)$ and w is known. Given a new x, what is the expression for the probability of $\hat{y}$?

Note: The univariate Gaussian PDF is

$$N(a|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(a-\mu)^2}{2\sigma^2}\right).$$

(c) Now assume that w is random and that we have a prior on w with known variance s_0^2:

$$w \sim N(0, s_0^2).$$

Write down the form of the posterior distribution over w. Take logs and drop terms that do not depend on the data D and prior parameters (you do not need to complete the square).

2. Regularization

Suppose we predict sales according to features of a sold item and its sales location using a linear regression model $y = w^T x$. We try three different losses:

(a) **No regularization:**

$$L(w) = \frac{1}{2}\sum_{n=1}^{N}(y_n - w^T x_n)^2.$$

(b) **Lasso regression:**

$$L(w) = \frac{1}{2}\sum_{n=1}^{N}(y_n - w^T x_n)^2 + \frac{\lambda}{2}\|w\|_1.$$

(c) **Ridge regression:**

$$L(w) = \frac{1}{2}\sum_{n=1}^{N}(y_n - w^T x_n)^2 + \frac{\lambda}{2}\|w\|_2^2.$$

The model is trained with each loss, resulting in different final coefficients (the coefficients are shown in plots in random order). Answer the following:

(a) Which plot of coefficients corresponds to which loss function? Why?

(b) How can we expect the plots to change as we increase λ?

3. Basis Functions

Basis functions $\phi(x)$ are important in both regression and classification tasks. For one-dimensional data x, consider

$$h(x; w) = w^T \phi(x).$$

Without basis functions, linear and logistic regression can only fit linear functions. Determine if the following choices of basis functions can linearly separate the data

$$D = \{(-\pi, 1), (0, -1), (\pi, 1)\},$$

assuming a logistic regression setup. If so, provide a setting of w that correctly classifies the data points.

(a) $\phi(x) = \begin{pmatrix} 1 \\ x \end{pmatrix}$.

(b) $\phi(x) = \begin{pmatrix} 1 \\ x \\ x^2 \end{pmatrix}$.

(c) $\phi(x) = \begin{pmatrix} 1 \\ x \\ x^4 \end{pmatrix}$.

(d) $\phi(x) = \begin{pmatrix} 1 \\ \cos x \end{pmatrix}$.

4. Probabilistic Linear Regression

In class, we derived the optimal w^* to maximize the likelihood of training data given normally distributed noise. In this problem, explore an alternative noise distribution for the labels y.

Assume one-dimensional data x, and that

$$\epsilon \sim \text{Lap}(0, 1),$$

with the model

$$y|x, \epsilon = wx + \epsilon.$$

The probability density function for a Laplace random variable $\text{Lap}(\mu, b)$ is

$$p(x) = \frac{1}{2b} \exp\left(-\frac{|x - \mu|}{b}\right).$$

You may assume that a linear transformation of a Laplace random variable results in a Laplace distribution with a shifted mean. Answer the following:

(a) What is the distribution of y given x?

(b) Given data $\{(x^{(i)}, y^{(i)})\}_{i=1}^N$, write an expression for the likelihood of the data in terms of the unknown parameter w.

(c) Write an expression for the negative log likelihood of the data.

(d) For normally distributed noise, maximizing the likelihood is equivalent to minimizing the squared error. What loss function $L(y, \hat{y})$ corresponds to minimizing the expression from part (c) for Laplacian noise?

(e) Given that $\frac{d}{da}|a| = \text{sign}(a)$ (with $\text{sign}(a) = 1$ if $a \geq 0$ and -1 if $a < 0$), take the gradient of the negative log likelihood with respect to w, leaving your expression in terms of the sign operator. Does this model seem more or less sensitive to outliers than the model with normally distributed noise? Explain.

5. Bayesian Linear Regression

Consider the following setup. Let

$$D = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}, \quad x^{(i)} \in \mathbb{R}^D, \quad y^{(i)} \in \mathbb{R}.$$

Consider the model

$$y^{(i)} \sim N\big(w^T x^{(i)}, \sigma^2\big).$$

The likelihood is given by

$$P(y|X, w) = N(Xw, \sigma^2 I) = \prod_{i=1}^{N} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\big(y^{(i)} - w^T x^{(i)}\big)^2}{2\sigma^2}\right).$$

Apply a conjugate Gaussian prior where each weight is i.i.d.:

$$P(w) = N(0, \sigma_0^2 I) = \prod_{j=1}^{D} \frac{1}{\sigma_0\sqrt{2\pi}} \exp\left(-\frac{w_j^2}{2\sigma_0^2}\right).$$

Answer the following:

(a) Find the MAP estimate for the weights as a simplified arg max or arg min expression (in non-matrix form). (Do not solve the equation.)

(b) What does the expression in part (a) remind you of?

(c) How does a wider (larger σ_0^2) versus a narrower (smaller σ_0^2) prior affect the posterior (in terms of both mean and variance)?

(d) The Gaussian prior has the form

$$P(w) \propto \prod_j \exp(-w_j^2),$$

while an alternative Laplace prior is given by

$$P(w) \propto \prod_j \exp\big(-\lambda|w_j|\big).$$

Conceptually, how would using a Laplacian prior instead of a Gaussian prior change the result from part (a) and its connection to regularization?

6. Multiclass Classification

Suppose we have a K-class classification problem with training data $\{(x^{(i)}, y^{(i)})\}_{i=1}^{n}$, where each $y^{(i)}$ is a one-hot column vector. Let C_k denote the one-hot vector with a 1 in the kth position.

We model this problem using a neural network with a single hidden layer containing d units, represented by

$$\phi(x; W, w_0) \in \mathbb{R}^d,$$

where each entry of ϕ is obtained by applying an activation function to a weighted input. The hidden layer output is then linearly combined and passed through a softmax function to yield K outputs. Let $v_\ell \in \mathbb{R}^d$ be the weights for the ℓth output. Thus,

$$p_{\text{model}}(y = C_k|x; \{v_\ell\}_{\ell=1}^{K}, W, w_0) = \frac{\exp\big(v_k^\top \phi\big)}{\sum_{\ell'=1}^{K} \exp\big(v_{\ell'}^\top \phi\big)}.$$

Answer the following:

(a) Suppose we add the same bias term v_0 to each final layer weight vector (i.e., replace $v_k^\top \phi$ with $v_k^\top \phi + v_0$ for all k). Does this increase the expressivity of the model? Why or why not?

(b) Assuming the sigmoid activation in the hidden layer with

$$\phi(x; W, w_0) = \sigma(Wx + w_0),$$

write down and simplify the log likelihood for a given observation $(x^{(i)}, y^{(i)})$, including constants.

(c) How might you train the parameters in this neural network? What roles do the loss function, sigmoid, and softmax functions play?

7. Probabilistic Generative Classification

Consider a Naive Bayes classifier for binary feature vectors $x \in \{0,1\}^D$ and two classes. The class-conditional distributions are:

$$p(x|y = C_k) = \prod_{j=1}^{D} \pi_{kj}^{x_j} (1 - \pi_{kj})^{1-x_j},$$

where $\pi_{kj} = p(x_j = 1|y = C_k)$, and assume the class priors are $p(y = C_1) = p(y = C_2) = \frac{1}{2}$.

(a) How is the quantity

$$\ln \frac{p(y = C_1|x)}{p(y = C_2|x)}$$

used for classifying a new example x?

(b) For $D = 1$ (a single binary feature), write out $\ln \frac{p(y=C_1|x)}{p(y=C_2|x)}$.

(c) Now suppose we use two redundant features (i.e., $x = [x, x]^\top$). What is $\ln \frac{p(y=C_1|x)}{p(y=C_2|x)}$ in terms of the value computed in part (a)?

(d) In terms of classifier performance, is this redundancy a bug or a useful property?

8. Overfitting and Underfitting

Harvard Insta-Ice Unit (HI2U) has developed a robot that delivers 24-hour shaved ice to student houses. To prevent collisions, they trained three classifiers to distinguish camera images of nearby tourists from open space. The performances are:

Classifier	Training Accuracy	Testing Accuracy
A	75.3%	74.8%
B	80.3%	77.8%
C	90.2%	60.0%

(a) For Classifier A, is it more likely overfitting or underfitting? Explain your reasoning.

(b) For Classifier C, is it more likely overfitting or underfitting? Explain your reasoning.

(c) Would more training examples significantly boost the test performance of Classifier A? Of Classifier C? Explain your answer in terms of bias and variance.

9. Neural Networks Part 1

Consider a 2-layer neural network that takes $x \in \mathbb{R}^2$ as input, has two ReLU hidden units, and a final sigmoid activation. There are no biases on the hidden units. For binary classification with labels $y \in \{0, 1\}$, the loss function is

$$L = -\Big(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})\Big).$$

(a) Suppose we update the network with stochastic gradient descent on a data point $x = [x_1, x_2]^T$.

(i) Calculate the gradient of the loss with respect to v_1.

(ii) Calculate the gradient of the loss with respect to w_{11}.

(b) Consider the classification of data points shown. Is it possible that this classification was generated by the weights

$$w_{11}, w_{12}, w_{21}, w_{22} = \{1, 0, 0, 1\}?$$

Why or why not? What if additional hidden layers were used (with these weights fixed)?

(c) (i) Why is it generally a bad idea to use ReLU as the activation function in the output layer?

(ii) Suppose we want to classify outputs into 5 categories. Why might it be problematic to use the label set $\{1, 2, 3, 4, 5\}$? What alternative encoding would you propose?

10. Neural Networks Part 2

Consider the following non-linearity for use in a neural network:

$$f_{0/1}(z) = \begin{cases} 1, & \text{if } z \geq 0, \\ 0, & \text{otherwise.} \end{cases}$$

Let $x \in \{0, 1\}^4$ be a binary feature vector. Define neural network A as:

$$\hat{y}_A = f_{0/1}(w^T x + w_0),$$

with weight vector $w \in \mathbb{R}^4$ and bias $w_0 \in \mathbb{R}$. Let $x_L = [x_1, x_2]$ and $x_R = [x_3, x_4]$. Define neural network B as:

$$\begin{aligned} h_1 &= f_{0/1}(t^T x_L + a), \\ h_2 &= f_{0/1}(u^T x_R + b), \\ h &= [h_1, h_2], \\ \hat{y}_B &= f_{0/1}(v^T h + c), \end{aligned}$$

with weight vectors $t, u, v \in \mathbb{R}^2$ and biases $a, b, c \in \mathbb{R}$. Answer the following:

(a) (i) Describe a logical formula on the inputs that can be expressed by network A but not by network B. Provide weights for w and w_0 that implement this formula (e.g., using ANDs, ORs).

(ii) Provide an argument for why network B cannot express this formula.

(iii) How might you change the architecture of network B to fix this issue? What downside might your modification have?

(b) What is the concern regarding training the networks as currently defined, and what modifications could alleviate this concern?

(c) State two ways in which a validation set can be used when training neural networks (one sentence for each).

11. Support Vector Machines (SVM)

Consider a binary classification problem using a Support Vector Machine (SVM) with the following three training points in $\mathbb{R}^2$:

$$\begin{aligned} x_1 &= (1, 1), \quad y_1 = +1, \\ x_2 &= (-1, -1), \quad y_2 = -1, \\ x_3 &= (2, 2), \quad y_3 = +1. \end{aligned}$$

Answer the following:

(a) Write down the optimization problem for a hard-margin SVM.

(b) Determine the equation of the optimal decision boundary assuming a linear SVM.

(c) Compute the margin of the classifier.

(d) Suppose we use a radial basis function (RBF) kernel

$$K(x, x') = \exp\big(-\gamma \|x - x'\|^2\big).$$

Explain why the decision boundary might become non-linear.