STANFORD UNIVERSITY
CS 229, Autumn 2015
Midterm Examination

Wednesday, November 4, 6:00pm-9:00pm

Question	Points
1 Short Answers	/26
2 More Linear Regression	/10
3 Generalized Linear Models	/17
4 Naive Bayes and Logistic Regression	/17
5 Anomaly Detection	/15
6 Learning Theory	/15
Total	/100

Name of Student: ______________________________

SUNetID: ______________________________@stanford.edu

The Stanford University Honor Code:

I attest that I have not given or received aid in this examination, and that I have done my share and taken an active part in seeing to it that others as well as myself uphold the spirit and letter of the Honor Code.

Signed: ______________________________

1. **[26 points] Short answers**

The following questions require a reasonably short answer (usually at most 2-3 sentences or a figure, though some questions may require longer or shorter explanations).

To discourage random guessing, one point will be deducted for a wrong answer on true/false or multiple choice questions! Also, no credit will be given for answers without a correct explanation.

(a) [6 points] Suppose you are fitting a fixed dataset with m training examples using linear regression, $h_\theta(x) = \theta^T x$, where $\theta, x \in \mathbb{R}^{n+1}$. After training, you realize that the variance of your model is relatively high (i.e. you are overfitting). For the following methods, indicate true if the method can mitigate your overfitting problem and false otherwise. Briefly explain why.

i. [3 points] Add additional features to your feature vector.
Answer: False. More features will make our model more complex, which will capture more outliers in the training set and overfit more.

ii. [3 points] Impose a prior distribution on θ, where the distribution of θ is of the form $\mathcal{N}(0, \tau^2 I)$, and we derive θ via maximum a posteriori estimation.
Answer: True. By imposing a prior belief on the distribution of θ, we are effectively limiting the norm of θ, since larger norm will have a lower probability. Thus, it makes our model less susceptible to overfitting.

(b) [3 points] Choosing the parameter C is often a challenge when using SVMs. Suppose we choose C as follows: First, train a model for a wide range of values of C. Then, evaluate each model on the test set. Choose the C whose model has the best performance on the test set. Is the performance of the chosen model on the test set a good estimate of the model's generalization error?

Answer: No it is not because C will be selected using the test set, meaning that the test set is no longer separate from model development. As a result, the choice of C might be over-fit to the test set and therefore might not generalize well on a new example, but there will be no way to figure this out because the test set was used to choose C.

(c) [11 points] For the following, provide the VC-dimension of the described hypothesis classes and briefly explain your answer.

i. [3 points] Assume $\mathcal{X} = \mathbb{R}^2$. $\mathcal{H}$ is a hypothesis class containing a single hypothesis h_1 (i.e. $\mathcal{H} = \{h_1\}$)

Answer: $VC(\mathcal{H}) = 0$. The VC dimension of a single hypothesis is always zero because a single hypothesis can only assign one labeling to a set of points.

ii. [4 points] Assume $\mathcal{X} = \mathbb{R}^2$. Consider $\mathcal{A}$ to be the set of all convex polygons in $\mathcal{X}$. $\mathcal{H}$ is the class of all hypotheses $h_P(x)$ (for $P \in \mathcal{A}$) such that

$$h_P(x) = \begin{cases} 1 & \text{if } x \text{ is contained within polygon P} \\ 0 & \text{otherwise} \end{cases}$$

Hint: Points on the edges or vertices of P are included in P

Answer: $VC(\mathcal{H}) = \infty$. For any positive integer n, take n points from $\mathcal{A}$. Suppose we place the n points $\{x_1, x_2, ..., x_n\}$ uniformly spaced on the unit circle. Then for each of the 2^n subsets of this data set, there is a convex polygon with vertices at these n points. For each subset, the convex polygon contains the set and excludes its complement. Therefore, $\forall n$, the shattering coefficient is 2^n and thus the VC dimension is infinite.

iii. [4 points] $\mathcal{H}$ is the class of hypotheses $h_{(a,b)}(x)$ such that each hypothesis is represented by a single open interval in $\mathcal{X} = \mathbb{R}$ as follows:

$$h_{(a,b)}(x) = \begin{cases} 1 & \text{if } a < x < b \\ 0 & \text{otherwise} \end{cases}$$

Answer: $VC(\mathcal{H}) = 2$. Take for example two points $\{0, 2\}$. We can shatter these two points by choosing the following set of intervals for our hypotheses $\{(3,5), (-1,1), (1,3), (-1,3)\}$. These correspond to the labellings: $\{(0,0), (1,0), (0,1), (1,1)\}$. We cannot shatter any set of three points $\{x_1, x_2, x_3\}$ such that $x_1 < x_2 < x_3$ because the labelling $x_1 = x_3 = 1, x_2 = 0$ cannot be realized. More generally, alternate labellings of consecutive points cannot be realized.

(d) [3 points] Consider a sine function $f(x) = \sin(x)$ such that $x \in [-\pi, \pi]$. We use two different hypothesis classes such that $\mathcal{H}_0$ contains all constant hypotheses of the form, $h(x) = b$ and $\mathcal{H}_1$ contains all linear hypotheses of the form $h(x) = ax + b$. Consider taking a very large number of training sets, $S_i, i = 1, ..., N$ such that each S_i contains only two points $\{(x_1, y_1), (x_2, y_2)\}$ sampled iid from $f(x)$. In other words, each (x, y) pair is drawn from a distribution such that $y = f(x) = \sin(x)$ is satisfied. We train a model from each hypothesis class using each training set such that we have a collection of N models from each class. We then compute a mean-squared error between each model and the function $f(x)$.

It turns out that the average expected error of all models from $\mathcal{H}_0$ is significantly lower than the average expected error of models from $\mathcal{H}_1$ even though $\mathcal{H}_1$ is a more complex hypothesis class. Using the concepts of bias and variance, provide an explanation for why this is the case.

Answer: Consider what happens when we plot all of the possible hypotheses on top of the function $f(x)$. This can be seen in Figure 1. We can see that because our training set only consists of two points, the variance in linear hypotheses is far greater than that of the constant hypotheses. Even though the constant hypotheses have higher bias, the overall average expected error is less than for the linear hypotheses because of the huge difference in variance.

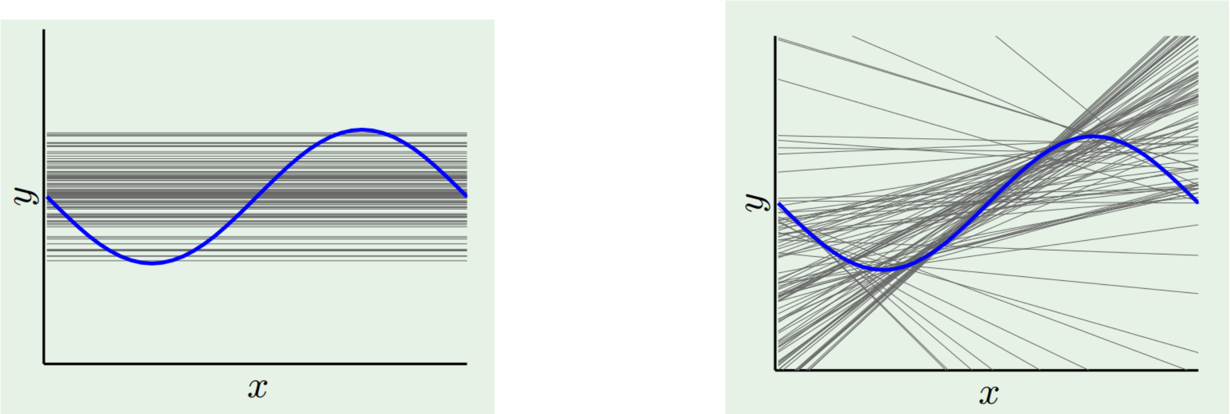

Figure 1: Many hypotheses from $\mathcal{H}_0$ and $\mathcal{H}_1$ plotted on top of $f(x)$

(e) [3 points] In class when we discussed the decision boundary for logistic regression $h_\theta(x) = g(\theta^T x)$, we did not require an explicit intercept term because we could define $x_0 = 1$ and let θ_0 be the intercept. When discussing SVMs, we dropped this convention and had $h_{w,b}(x) = g(w^T x + b)$ with b as an explicit intercept term. Consider an SVM where we now write $h_w(x) = g(w^T x)$ and define $x_0 = 1$ such that w_0 is the intercept. If the primal optimization objective remains $\frac{1}{2}\left|\left|w\right|\right|^2$, can we change the intercept in this way without changing the decision boundary found by the SVM? Justify your answer.

Answer: We cannot make this change, because it will change the decision boundary. In the original SVM, the $\frac{1}{2}\left|\left|w\right|\right|^2$ term did not penalize the intercept, so b could be chosen as large as possible to satisfy the constraints. If we treat, w_0 as the intercept, it will be regularized, thus changing the optimal solution.

2. **[10 + 3 Extra Credit points] More Linear Regression**
In our homework, we saw a variant of linear regression called locally-weighted linear regression. In the problem below, we consider a regularized form of locally-weighted linear regression where we favor smaller parameter vectors by adding a complexity penalty term to the cost function. Additionally, we consider the case where we are trying to predict multiple outputs for each training example. Our dataset is:

$$\mathcal{S} = \{(x^{(i)}, y^{(i)}), i = 1, ..., m\}, x^{(i)} \in \mathbb{R}^n, y^{(i)} \in \mathbb{R}^p$$

Thus for each training example, $y^{(i)}$ is a real-valued vector with p entries. We wish to use a linear model to predict the outputs by specifying the parameter matrix θ, where $\theta \in \mathbb{R}^{n \times p}$. You can assume $x^{(i)}$ contains the intercept term (i.e. $x_0 = 1$). The cost function for this model is:

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{p} w^{(i)} \left((\theta^T x^{(i)})_j - y_j^{(i)}\right)^2 + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{p}(\theta_{ij})^2 \tag{1}$$

As before, $w^{(i)}$ is the “weight” for a specific training example i.

(a) [2 points] Show that $J(\theta)$ can be written as

$$J(\theta) = \frac{1}{2}\mathrm{tr}\left((X\theta - Y)^T W (X\theta - Y)\right) + \frac{1}{2}\mathrm{tr}(\theta^T\theta)$$

Answer:

$$\begin{aligned}
J(\theta) &= \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{p} w^{(i)} \left((\theta^T x^{(i)})_j - y_j^{(i)}\right)^2 + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{p}(\theta_{ij})^2 & (2)\\
&= \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{p} w^{(i)}(X\theta - Y)_{ij}^2 + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{p}(\theta_{ij})^2 & (3)\\
&= \frac{1}{2}\sum_{i=1}^{m} w^{(i)}\left((X\theta - Y)^T(X\theta - Y)\right)_{ii} + \frac{1}{2}\sum_{i=1}^{n}(\theta^T\theta)_{ii} & (4)\\
&= \frac{1}{2}\mathrm{tr}\left((X\theta - Y)^T W (X\theta - Y)\right) + \frac{1}{2}\mathrm{tr}(\theta^T\theta) & (5)
\end{aligned}$$

(b) [5 points] Derive a closed form expression for the minimizer θ^* that minimizes $J(\theta)$ from part (a).

Answer: We compute the gradient of the first and second term separately. We start with the first term, $J_1(\theta) = \frac{1}{2}\mathrm{tr}\left((X\theta - Y)^T W(X\theta - Y)\right)$.

$$\begin{aligned}
\nabla_\theta J_1(\theta) &= \nabla_\theta \frac{1}{2}\mathrm{tr}\left((X\theta - Y)^T W(X\theta - Y)\right) && (6)\\
&= \frac{1}{2}\nabla_\theta \mathrm{tr}\Big(\theta^T X^T W X\theta - \theta^T X^T W Y - Y^T W X\theta - Y^T W Y\Big) && (7)\\
&= \frac{1}{2}\nabla_\theta \left[\mathrm{tr}(\theta^T X^T W X\theta) - \mathrm{tr}(\theta^T X^T W Y) - \mathrm{tr}(Y^T W X\theta) - \mathrm{tr}(Y^T W Y)\right] && (8)\\
&= \frac{1}{2}\nabla_\theta \left[\mathrm{tr}(\theta^T X^T W X\theta) - 2\mathrm{tr}(Y^T W X\theta) - \mathrm{tr}(Y^T W Y)\right] && (9)\\
&= \frac{1}{2}(X^T W X\theta - 2X^T W Y + X^T W X\theta) && (10)\\
&= X^T W X\theta - X^T W Y && (11)
\end{aligned}$$

Now we find the gradient of the second term, $J_2(\theta) = \frac{1}{2}\mathrm{tr}(\theta^T\theta)$:

$$\nabla_\theta J_2(\theta) = \nabla_\theta \frac{1}{2}\mathrm{tr}(\theta^T\theta) = \frac{1}{2}\nabla_\theta \mathrm{tr}(\theta^T\theta) = \frac{1}{2}(2\theta) = \theta \qquad (12)$$

Combining the gradient of both terms gives us the final gradient:

$$\begin{aligned}
\nabla_\theta J(\theta) &= \nabla_\theta J_1(\theta) + \nabla_\theta J_2(\theta)\\
&= X^T W X\theta - X^T W Y + \theta
\end{aligned} \qquad (13)$$

We can then set this equal to zero and find the optimal θ which optimizes $J(\theta)$.

$$\begin{aligned}
0 &= X^T W X\theta - X^T W Y + \theta\\
X^T W Y &= X^T W X\theta + \theta\\
X^T W Y &= (X^T W X + I)\theta\\
\theta^* &= (X^T W X + I)^{-1} X^T W Y
\end{aligned} \qquad (14)$$

where I is the $n \times n$ identity matrix.

(c) [3 points] Given the dataset $\mathcal{S}$ above, which of the following cost functions will lead to higher accuracy on the training set? Briefly explain why this is the case. If there is insufficient information, explain what details are needed to make a decision.

i. $J_1(\theta) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{p}\left((\theta^T x^{(i)})_j - y_j^{(i)}\right)^2$

ii. $J_2(\theta) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{p}\left((\theta^T x^{(i)})_j - y_j^{(i)}\right)^2 + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{p}(\theta_{ij})^2$

iii. $J_3(\theta) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{p}\left((\theta^T x^{(i)})_j - y_j^{(i)}\right)^2 + 100\sum_{i=1}^{n}\sum_{j=1}^{p}(\theta_{ij})^2$

Answer: (i) The regularization terms prevents overfitting by penalizing high weights. Excluding the regularization terms will allow the model to overfit on the training set and achieve a higher training accuracy.

(d) [3 Extra Credit points] Suppose we want to weight the regularization penalty on a per element basis. For this problem, we use the following cost function:

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{p} w^{(i)}\left((\theta^T x^{(i)})_j - y_j^{(i)}\right)^2 + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{p}((\Gamma\theta)_{ij})^2 \tag{15}$$

Here, $\Gamma \in \mathbb{R}^{n\times n}$ where $\Gamma_{ij} > 0$ for all i, j. Derive a closed form solution for $J(\theta)$ and θ^* using this new cost function.

Answer: We first write $J(\theta)$ in matrix-vector notation to make the gradient derivation easier.

$$\begin{aligned}
J(\theta) &= \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{p} w^{(i)}\left((\theta^T x^{(i)})_j - y_j^{(i)}\right)^2 + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{p}((\Gamma\theta)_{ij})^2 && (16)\\
&= \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{p} w^{(i)}(X\theta - Y)^2_{ij} + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{p}((\Gamma\theta)_{ij})^2 && (17)\\
&= \frac{1}{2}\sum_{i=1}^{m} w^{(i)}\left((X\theta - Y)^T(X\theta - Y)\right)_{ii} + \frac{1}{2}\sum_{i=1}^{n}\left((\Gamma\theta)^T(\Gamma\theta)\right)_{ii} && (18)\\
&= \frac{1}{2}\text{tr}\left((X\theta - Y)^T W(X\theta - Y)\right) + \frac{1}{2}\text{tr}\left((\Gamma\theta)^T(\Gamma\theta)\right) && (19)
\end{aligned}$$

The gradient of the first term is the same as part (a):

$$\begin{aligned}
\nabla_\theta J_1(\theta) &= \nabla_\theta \frac{1}{2}\text{tr}\left((X\theta - Y)^T W(X\theta - Y)\right) && (20)\\
&= X^T W X\theta - X^T W Y && (21)
\end{aligned}$$

Now we find the gradient of the second term, denoted as $J_2(\theta)$:

$$\begin{aligned}
\nabla_\theta J_2(\theta) &= \nabla_\theta \frac{1}{2}\text{tr}\left((\Gamma\theta)^T(\Gamma\theta)\right) && (22)\\
&= \frac{1}{2}\nabla_\theta \text{tr}\left(\theta^T\Gamma^T\Gamma\theta\right) && (23)\\
&= \frac{1}{2}\left((\Gamma^T\Gamma)^T\theta + \Gamma^T\Gamma\theta\right) && (24)\\
&= \frac{1}{2}\left(2\Gamma^T\Gamma\theta\right) && (25)\\
&= \Gamma^T\Gamma\theta && (26)
\end{aligned}$$

The jump from (23) to (24) can be made using Equation 5 from lecture notes 1, where $A^T = \theta, A = \theta^T, B = \Gamma^T\Gamma$ and C be the identity matrix. Combining the gradient of both terms gives us the final gradient:

$$\nabla_\theta J(\theta) = X^T W X\theta - X^T W Y + \Gamma^T\Gamma\theta \tag{27}$$

We can then set this equal to zero and find the optimal θ which optimizes $J(\theta)$.

$$\begin{aligned} 0 &= X^TWX\theta - X^TWY + \Gamma^T\Gamma\theta \\ X^TWY &= X^TWX\theta + \Gamma^T\Gamma\theta \\ X^TWY &= (X^TWX + \Gamma^T\Gamma)\theta \\ \theta^* &= (X^TWX + \Gamma^T\Gamma)^{-1}X^TWY \end{aligned} \tag{28}$$

3. **[17 points] Generalized Linear Models**

In class we showed that the Gaussian distribution is in the Exponential Family. However, a simplification we made to make the derivation easier was to set the variance term $\sigma^2 = 1$. This problem will investigate a more general form for the Exponential Family. First, recall that the Gaussian distribution can be written as follows:

$$p(y|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left\{-\frac{1}{2\sigma^2}(y-\mu)^2\right\} \tag{29}$$

(a) [6 points] Show that the Gaussian distribution (without assuming unit variance) is an exponential family distribution. In particular, please specify $b(y)$, η, $T(y)$, $a(\eta)$. Recall that the standard form for the exponential family is given by

$$p(y;\eta) = b(y)\exp\{\eta^\top T(y) - a(\eta)\} \tag{30}$$

Hint: since σ^2 is now a variable, η and $T(y)$ will now be two dimensional vectors; for consistent notation denote $\eta = \begin{bmatrix}\eta_1 & \eta_2\end{bmatrix}^\top$. For full credit, please ensure $a(\eta)$ is expressed in terms of η_1 and η_2.

Answer: We can rearrange the Gaussian distribution as follows:

$$\begin{aligned} p(y|\mu, \sigma^2) &= \frac{1}{\sqrt{2\pi}\sigma}\mathsf{exp}\left\{-\frac{1}{2\sigma^2}(y-\mu)^2\right\} \\ &= \frac{1}{\sqrt{2\pi}}\mathsf{exp}\left\{\frac{\mu}{\sigma^2}y - \frac{1}{2\sigma^2}y^2 - \frac{1}{2\sigma^2}\mu^2 - \mathsf{ln}\ \sigma\right\} \end{aligned}$$

This is now in the exponential family form, and we can note that:

$$\begin{aligned} \eta &= \begin{bmatrix}\eta_1 \\ \eta_2\end{bmatrix} = \begin{bmatrix}\mu/\sigma^2 \\ -1/2\sigma^2\end{bmatrix} \\ T(y) &= \begin{bmatrix}y \\ y^2\end{bmatrix} \\ a(\eta) &= \frac{\mu^2}{2\sigma^2} + \mathsf{ln}\ \sigma = -\frac{\eta_1^2}{4\eta_2} - \frac{1}{2}\mathsf{ln}(-2\eta_2) \\ b(y) &= \frac{1}{\sqrt{2\pi}} \end{aligned}$$

(b) [4 points] Suppose you are given an IID training set $\{(x^{(i)}, y^{(i)}), i = 1, ..., m\}$. Starting with the expression in (30) for $p(y; \eta)$, derive the general expression for the Hessian of the log-likelihood $\ell(\theta) = \sum_{i=1}^{m} \log p(y^{(i)}|x^{(i)}; \theta)$. Your answer should be in terms of x, η_1 and η_2.

Answer: The log-likelihood is given by

$$\begin{aligned}
\ell(\theta) &= \sum_{i=1}^{m} \text{log } p(y^{(i)}|x^{(i)}; \theta) \\
&= \sum_{i=1}^{m} \text{log}(b(y)) + \eta^{(i)} T(y) - a(\eta^{(i)})
\end{aligned}$$

We now take partials with respect to θ_i and θ_j as follows. Recall that our standard GLM assumption is that $\eta = \theta^\top x$.

$$\begin{aligned}
\frac{\partial}{\partial \theta_j} \ell(\theta) &= \sum_{i=1}^{M} T(y) x_j^{(i)} - \frac{\partial}{\partial \eta} a(\eta^{(i)}) x_j^{(i)} \\
\frac{\partial^2}{\partial \theta_j \partial \theta_k} \ell(\theta) &= \sum_{i=1}^{M} - \frac{\partial^2}{\partial \eta^2} a(\eta^{(i)}) x_j^{(i)} x_k^{(j)} \\
&= H_{jk}
\end{aligned}$$

Thus the Hessian is described by each element H_{jk} which is itself a 2x2 symmetric matrix obtained by expanding the partial derivative with respect to η as follows:

$$\begin{aligned}
\frac{\partial^2}{\partial \eta_1^2} a(\eta^{(i)}) &= \frac{\partial}{\partial \eta_1} \left(-\frac{2\eta_1}{4\eta_2} \right) = -\frac{1}{2\eta_2} \\
\frac{\partial^2}{\partial \eta_2^2} a(\eta^{(i)}) &= \frac{\partial}{\partial \eta_2} \left(\frac{-2\eta_2 + \eta_1^2}{4\eta_2^2} \right) = \frac{\eta_2 - \eta_1^2}{2\eta_2^3} \\
\frac{\partial^2}{\partial \eta_1 \partial \eta_2} a(\eta^{(i)}) &= \frac{\partial}{\partial \eta_2} \left(-\frac{2\eta_1}{4\eta_2} \right) = \frac{\eta_1}{2\eta_2^2} \\
\frac{\partial^2}{\partial \eta_2 \partial \eta_1} a(\eta^{(i)}) &= \frac{\partial}{\partial \eta_1} \left(\frac{-2\eta_2 + \eta_1^2}{4\eta_2^2} \right) = \frac{\eta_1}{2\eta_2^2}
\end{aligned}$$

(c) [5 points] Using your result from the part (b), show that the Hessian is negative semi-definite, i.e., $z^\top Hz \leq 0$.

Answer: 5 points given to everyone because difficulty was harder than originally anticipated.

(d) [2 points] It turns out there is a more general definition for the exponential family given by

$$p(y;\eta,\tau) = b(a,\tau)\text{exp}\left\{\frac{\eta^\top T(y) - a(\eta)}{c(\tau)}\right\}$$

In particular $c(\tau)$ is the dispersion function, where τ is called the *dispersion parameter*. Show that the Gaussian distribution can be written in this more general form with $c(\tau) = \sigma^2$.

Answer: In part a, we wrote the Gaussian distribution in the exponential family form as:

$$p(y|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi}}\text{exp}\left\{\frac{\mu}{\sigma^2}y - \frac{1}{2\sigma^2}y^2 - \frac{1}{2\sigma^2}\mu^2 - \ln\,\sigma\right\}$$

We can manipulate this slightly to get it into the general exponential family form, as follows:

$$p(y|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi}}\text{exp}\left\{\frac{\mu}{\sigma^2}y - \frac{1/2}{\sigma^2}y^2 - \frac{1/2}{\sigma^2}\mu^2 - \frac{\sigma^2}{\sigma^2}\ln\,\sigma\right\}$$

Thus we have that the dispersion function $c(\tau) = \sigma^2$, as desired.

4. **[17 points] Naive Bayes and Logistic Regression**
For this entire problem assume that the input features x_j, $j = 1, ..., n$ are discrete binary-valued variables such that $x_j \in \{0, 1\}$ and $x = [x_1\ x_2\ ...\ x_n]$. For each training example $x^{(i)}$, assume that the output target variable $y^{(i)} \in \{0, 1\}$.

(a) [2 points] Consider the Naive Bayes model, given the above context. This model can be parameterized by $\phi_{j|y=0} = p(x_j = 1|y = 0)$, $\phi_{j|y=1} = p(x_j = 1|y = 1)$ and $\phi_y = p(y = 1)$. Write down the expression for $p(y = 1|x)$ in terms of $\phi_{j|y=0}, \phi_{j|y=1}$, and ϕ_y.
Answer:
We first use the fact that each x_j has a binomial distribution:

$$\begin{aligned} p(x|y=0) &= \prod_{j=1}^{n} p(x_j|y=0) \\ &= \prod_{j=1}^{n} (\phi_{j|y=0})^{x_j} (1 - \phi_{j|y=0})^{1-x_j} \end{aligned} \tag{31}$$

$p(x|y = 1)$ can be written the same way with $y = 0$ replaced by $y = 1$. Now we can derive $p(y = 1|x)$ using Bayes rule:

$$p(y=1|x) = \frac{\phi_y (\prod_{j=1}^{n} (\phi_{j|y=1})^{x_j} (1 - \phi_{j|y=1})^{1-x_j})}{\phi_y (\prod_{j=1}^{n} (\phi_{j|y=1})^{x_j} (1 - \phi_{j|y=1})^{1-x_j}) + (1 - \phi_y)(\prod_{j=1}^{n} (\phi_{j|y=0})^{x_j} (1 - \phi_{j|y=0})^{1-x_j})} \tag{32}$$

Points are given for this or any equivalent solution.

(b) [7 points] Show that the conditional likelihood expression you obtained in part (a) can be simplified to the same form as the hypothesis for logistic regression:

$$p(y=1|x) = \frac{1}{1+e^{-\theta^T x}}. \tag{33}$$

Hint: Modify the definition of x to include the intercept term $x_0 = 1$

Answer:

We begin by dividing the numerator and denominator in the expression from part (a) by the numerator:

$$\begin{aligned} p(y=1|x) &= \frac{\phi_y(\prod_{j=1}^n(\phi_{j|y=1})^{x_j}(1-\phi_{j|y=1})^{1-x_j})}{\phi_y(\prod_{j=1}^n(\phi_{j|y=1})^{x_j}(1-\phi_{j|y=1})^{1-x_j}) + (1-\phi_y)(\prod_{j=1}^n(\phi_{j|y=0})^{x_j}(1-\phi_{j|y=0})^{1-x_j})} \\ &= \frac{1}{1+\frac{(1-\phi_y)(\prod_{j=1}^n(\phi_{j|y=0})^{x_j}(1-\phi_{j|y=0})^{1-x_j})}{\phi_y(\prod_{j=1}^n(\phi_{j|y=1})^{x_j}(1-\phi_{j|y=1})^{1-x_j})}} \\ &= \frac{1}{1+\mathsf{exp}(\mathsf{log}\ \frac{1-\phi_y}{\phi_y} + \sum_{j=1}^n x_j(\mathsf{log}\ \frac{\phi_{j|y=0}}{\phi_{j|y=1}}) + (1-x_j)(\mathsf{log}\ \frac{1-\phi_{j|y=0}}{1-\phi_{j|y=1}}))} \\ &= \frac{1}{1+\mathsf{exp}(\mathsf{log}\ (\frac{1-\phi_y}{\phi_y} + \frac{1-\phi_{j|y=0}}{1-\phi_{j|y=1}}) + \sum_{j=1}^n x_j(\mathsf{log}\ \frac{\phi_{j|y=0}}{\phi_{j|y=1}} - \mathsf{log}\ \frac{1-\phi_{j|y=0}}{1-\phi_{j|y=1}}))} \end{aligned} \tag{34}$$

Now we remember that $x_0 = 1$, so if we set:

$$\begin{aligned} \theta_0 &= -\mathsf{log}\ \frac{1-\phi_y}{\phi_y} - \sum_{j=1}^n \mathsf{log}\ \frac{1-\phi_{j|y=0}}{1-\phi_{j|y=1}} \\ \theta_j &= -\mathsf{log}\ \frac{\phi_{j|y=0}}{\phi_{j|y=1}} + \mathsf{log}\ \frac{1-\phi_{j|y=0}}{1-\phi_{j|y=1}} \qquad \forall j = 1...n \end{aligned} \tag{35}$$

.

we arrive at the final form:

$$p(y=1|x) = \frac{1}{1+e^{-\theta^T x}} \tag{36}$$

[More space for (b)]

(c) [6 points] In part (b) you showed that the discrete Naive Bayes decision boundary has the same form as that of the logistic regression. Now consider a dataset S_1 with m training examples of the form: $\{(x^{(i)}, y^{(i)}), i = 1, ..., m\}$ with each $x^{(i)} \in \mathbb{R}^{n+1}$. Note that for this problem, S_1 satisfies the Naive Bayes assumption: $p(x_1, \ldots, x_n|y) = \prod_{j=1}^{n} p(x_j|y)$.
Suppose a second dataset, S_2, is given to you again with m training examples $\{(x^{(i)}, y^{(i)}), i = 1, \cdots, m\}$, but now each $x^{(i)} \in \mathbb{R}^{n+2}$ because each $x^{(i)}$ contains the same n conditionally-independent features and an additional feature x_{n+1} such that $x_{n+1} = x_n$. Each $x^{(i)}$ contains the intercept term $x_0 = 1$.

i. [2 points] You train two Naive Bayes classifiers independently on S_1 and S_2. Test data is generated according to the true distribution (i.e. $p(x_1, ..., x_n, y) = p(x_1, ..., x_n, x_{n+1}, y) = p(y)p(x_1, ..., x_n|y)$, where $x_{n+1} = x_n$). Would you expect the test error of the classifier trained on S_1 to be larger or smaller than that trained on S_2? You may assume that m is very large. Briefly justify your answer.
Answer: The expected testing error of the classifier trained on S_1 will be less than the error of the classifier trained on S_2. This is because S_2 violates the conditional independence assumption made by Naive Bayes between features x_n and x_{n+1}, so the model will learn an incorrect joint distribution.

ii. [4 points] Now we will look at a similar situation regarding how logistic regression is affected by copies of features. In order to simplify the math, let's assume a more basic case where S_1 still has m training examples, but now has one feature x_1. S_2 has m training examples but has two features x_1 and x_2 where $x_2 = x_1$. The logistic regression model trained on S_1 therefore has associated parameters $\{\theta_0, \theta_1\}$ and the model trained on S_2 has parameters $\{\theta_0, \theta_1, \theta_2\}$. Here, θ_0 is associated with the intercept term $x_0 = 1$. Testing data is generated the same way (from the original true distribution). How will the error of the classifier trained on S_1 compare to that of the classifier trained on S_2? For this question you need to prove your result mathematically. (Hint: compare the forms of the log-likelihood for each classifier)

Answer: The log-likelihood for the model trained on D_1 can be written as follows: (for simplicity we substitute $g(\theta^T x)$ for $h_\theta(x)$:

$$L_1(\theta) = \sum_i y^{(i)} \text{log } g(\theta_0 + \theta_1 x_1^{(i)}) + (1 - y^{(i)})(\text{log } (1 - g(\theta_0 + \theta_1 x_1^{(i)}))) \quad (37)$$

The log-likelihood for the model trained on D_2 can be similarly written as:

$$L_2(\theta) = \sum_i y^{(i)} \text{log } g(\theta'_0 + \theta'_1 x_1^{(i)} + \theta'_2 x_2^{(i)}) + (1 - y^{(i)})(\text{log } (1 - g(\theta'_0 + \theta'_1 x_1^{(i)} + \theta'_2 x_2^{(i)}))) \quad (38)$$

But we know that x_2 is a duplication of x_1, so when we maximize $L_2(\theta)$. It is equivalent to maximizing:

$$\begin{aligned} L_2(\theta) &= \sum_i y^{(i)} \text{log } g(\theta'_0 + \theta'_1 x_1^{(i)} + \theta'_2 x_1^{(i)}) + (1 - y^{(i)})(\text{log } (1 - g(\theta'_0 + \theta'_1 x_1^{(i)} + \theta'_2 x_1^{(i)}))) \\ &= \sum_i y^{(i)} \text{log } g(\theta'_0 + (\theta'_1 + \theta'_2) x_1^{(i)}) + (1 - y^{(i)})(\text{log } (1 - g(\theta'_0 + (\theta'_1 + \theta'_2) x_1^{(i)}))) \end{aligned} \quad (39)$$

Looking at this, we see that $L_2(\theta) = L_1(\theta)$ if we choose $\theta_0 = \theta'_0$ and $\theta_1 = \theta'_1 + \theta'_2$. This basically means that the logistic regression trained on D_2 will split the weight of θ_1 between θ'_1 and θ'_2, but the decision boundary will be the same. Therefore, the accuracies of both classifiers will be the same.

(d) [2 points] In general, if we assume that the number of training examples m is very large, which classifier will have a lower generalization error? Briefly justify why.

Answer: The logistic regression will in general have a lower asymptotic error rate because the Naive Bayes classifier makes the conditional independence assumption about the data. Therefore, the logistic regression can learn the Naive Bayes decision boundary; however, the reverse is not true. It turns out that this generalization error relationship is true for any generative-discriminative pair because of the fact that the generative model makes stronger modeling assumptions.

In lecture we showed that if we make assumptions about the distribution of $p(x|y)$ (specifically that this follows a multivariate gaussian), then we can either perform classification using Gaussian Discriminant Analysis or Logistic Regression, and still arrive at a logistic form for the conditional distribution $p(y|x)$. Therefore, we can think of GDA and Logistic Regression as being a generative-discriminative pair where the discriminative model directly estimates the boundary between the class-conditionals that is learned by the generative classifier through Bayes' rule. In this problem, you have shown that a similar generative-discriminative pair property can be derived between Naive Bayes and Logistic Regression for binary classification. This kind of case analysis can provide a lot of insights into the intricate differences between generative and discriminative models.

5. **[15 points] Anomaly Detection**

Consider the following optimization problem:

$$\begin{aligned} \underset{r,z,\xi}{\text{minimize}} \quad & r^2 + C\sum_{i=1}^{m} \xi_i \\ \text{s.t.} \quad & \left|\left|x^{(i)} - z\right|\right|_2^2 \le r^2 + \xi_i \;\; i = 1, \ldots, m. \\ & \xi_i \ge 0, \;\; i = 1, \ldots, m. \end{aligned} \tag{40}$$

where ξ_i are the slack variables.

(a) [2 points] Write down the Lagrangian for the optimization problem above. We suggest using two sets of Lagrange multipliers α_i and η_i corresponding to the two inequality constraints so that the Lagrangian would be written as $\mathcal{L}(r, z, \xi, \alpha, \eta)$.

Answer: Using the definition of the Lagrangian, we obtain:

$$\mathcal{L}(r, z, \xi, \alpha, \eta) = r^2 + C\sum_{i=1}^{m} \xi_i - \sum_{i=1}^{m} \alpha_i(r^2 - \left|\left|x^{(i)} - z\right|\right|_2^2 + \xi_i) - \sum_{i=1}^{m} \eta_i \xi_i \tag{41}$$

with $\alpha, \eta \ge 0$.

(b) [7 points] Assuming a non-trivial solution ($r > 0$), derive the dual optimization problem using the Lagrangian from part (a).

Answer: We start by taking derivatives of the Lagrangian with respect to r, z, and ξ and set them to 0:

$$\partial_r \mathcal{L} = 2r - 2\sum_{i=1}^m \alpha_i r = 0 \implies \sum_{i=1}^m \alpha_i = 1 \tag{42}$$

$$\partial_z \mathcal{L} = \sum_{i=1}^m \alpha_i 2(x^{(i)} - z) = 0 \implies z = \frac{\sum_{i=1}^m \alpha_i x^{(i)}}{\sum_{i=1}^m \alpha_i} = \sum_{i=1}^m \alpha_i x^{(i)} \tag{43}$$

$$\partial_\xi \mathcal{L} = C - \alpha_i - \eta_i = 0 \implies \eta_i = C - \alpha_i \tag{44}$$

The result in (43) was obtained by using the result from (42). The last equality (44) shows that we can eliminate η_i by substituting for α_i and instead add the constraint $0 \leq \alpha_i \leq C, \quad \forall i$. We now substitute these values back into the Lagrangian and simplify as follows:

$$\mathcal{L}(r, z, \xi, \alpha, \eta) = r^2 - \sum_{i=1}^m \alpha_i r^2 + C\sum_{i=1}^m \xi_i - \sum_{i=1}^m \alpha_i \xi_i - \sum_{i=1}^m (C - \alpha_i)\xi_i \\ + \sum_{i=1}^m \alpha_i \left|\left| x^{(i)} - \sum_{j=1}^m \alpha_j x_j \right|\right|_2^2 \tag{45}$$

Using (42), we see that the first two terms cancel, and because of our substitution from (44), all of the next 3 terms cancel leaving us with the dual problem:

$$\begin{aligned} \text{maximize}_{\alpha} \quad & \sum_{i=1}^m \alpha_i \left|\left| x^{(i)} - \sum_{j=1}^m \alpha_j x_j \right|\right|_2^2 , \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq C, \ \ i = 1 \ldots m. \\ & \sum_{i=1}^m \alpha_i = 1, \ \ i = 1 \ldots m. \end{aligned} \tag{46}$$

Points are given for this or any simplified version of this.

(c) [3 points] Show that the dual problem from (b) can be kernelized.
Answer: If we look at the dual from (b) we can simplify the objective further as follows:

$$\sum_{i=1}^{m} \alpha_i \left|\left| x^{(i)} - \sum_{j=1}^{m} \alpha_j x^{(j)} \right|\right|_2^2 = \sum_{i=1}^{m} \left(\alpha_i \langle x^{(i)}, x^{(i)} \rangle - 2\alpha_i \langle x^{(i)}, \sum_{j=1}^{m} \alpha_j x^{(j)} \rangle + \alpha_i \langle \sum_{j=1}^{m} \alpha_j x^{(j)}, \sum_{j=1}^{m} \alpha_j x^{(j)} \rangle \right) \tag{47}$$

$$= \sum_{i=1}^{m} \alpha_i \langle x^{(i)}, x^{(i)} \rangle - \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle \tag{48}$$

The last equality was simplified by using (42). Now we can see that the dual objective can be written in terms of inner products of the training data, so the problem can be kernelized by mapping the input data into a higher dimensional feature space using a function $\phi(x)$ and computing the kernel sas $K(x^{(i)}, x^{(j)}) = \langle \phi(x^{(i)}), \phi(x^{(j)}) \rangle$.

(d) [3 points] Now consider the following dual optimization problem

$$\begin{aligned}\max_{\alpha} \quad & -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j\langle x^{(i)}, x^{(j)}\rangle \\ \text{s.t} \quad & \sum_{i=1}^{m}\alpha_i = 1, \;\; i = 1, \ldots, m.\end{aligned} \tag{49}$$

Assume that we choose K such that it is a Gaussian Kernel. How does this dual compare with the dual you derived in part (c)?.

Answer: The objective in (48) only differs from the dual in this part by a constant $\frac{1}{2}$ in front of the second term and the first term $\sum_{i=1}^{m}\alpha_i\langle x^{(i)}x^{(i)}\rangle$ term. However, this inner product is a constant because for a gaussian kernel, $K(x, x) = \kappa$ (a constant), and by the constraint (42) this first term is just a constant. Therefore, the two optimization problems only differ by constant factors and are actually equivalent optimization problems.

The optimization problem defined in part (a) is known as the minimum enclosing ball (MEB) problem, which is commonly used for anomaly detection. The optimization is used to learn a minimum enclosing "ball" defined by a radius r which packs most of the training data close to its center defined by z. A new data point will be considered anomalous if it is outside this ball. The dual in part (d) is actually the dual of the much less intuitive but much more commonly used one-class SVM which is also used for anomaly detection. As you have shown in part (d), it turns out that these optimization problems are actually equivalent when using an isotropic kernel like the Gaussian Kernel.

6. **[15 points] Learning Theory**

Consider a finite hypothesis class $\mathcal{H}$ with size $k = |\mathcal{H}|$ and $h^\star = \operatorname*{argmin}_{h \in \mathcal{H}} \epsilon(h)$.

(a) [7 points] Assume that the best hypothesis $h^\star$ has generalization error $\epsilon(h^\star) = B$ such that B is a constant with $0 \leq B \leq 1$. Prove that the joint probability of the expected risk minimizer $\hat{h}$ having large generalization error and the best hypothesis h^* having small training error can be bounded as:

$$P(\epsilon(\hat{h}) > B + 2\gamma, \hat{\epsilon}(h^\star) \leq B + \gamma) \leq \sum_{h \in \mathcal{H}} P(\epsilon(h) > B + 2\gamma, \hat{\epsilon}(h) \leq B + \gamma) \quad (50)$$

Answer: By the definition of ERM, we know that $\hat{\epsilon}(h^\star) \geq \hat{\epsilon}(\hat{h})$, so that

$$P(\epsilon(\hat{h}) > B + 2\gamma, \hat{\epsilon}(h^\star) \leq B + \gamma) \leq P(\epsilon(\hat{h}) > B + 2\gamma, \hat{\epsilon}(\hat{h}) \leq B + \gamma).$$

Using union bound, we can bound the RHS of last expression as

$$\begin{aligned} P(\epsilon(\hat{h}) > B + 2\gamma, \hat{\epsilon}(\hat{h}) \leq B + \gamma) &\leq P(\exists h \text{ s.t. } \epsilon(h) > B + 2\gamma, \hat{\epsilon}(h) \leq B + \gamma) \\ &\leq \sum_{h \in \mathcal{H}} P(\epsilon(h) > B + 2\gamma, \hat{\epsilon}(h) \leq B + \gamma). \end{aligned}$$

For any hypothesis $h' \in \mathcal{H}$ with high generalization error (i.e. $\epsilon(h') > B' + \tau$), the probability that it has low training error (i.e. $\hat{\epsilon}(h') \leq B'$) is bounded by:

$$P(\hat{\epsilon}(h') \leq B' \mid \epsilon(h') > B' + \tau) \leq \exp\left\{\frac{-m\tau^2}{2(B' + 4\tau/3)}\right\} \tag{51}$$

for any $B' \in (0, 1)$ and $\tau > 0$.

(b) [8 points] Using (51) and the result from part (a), show that:

$$P(\epsilon(\hat{h}) > B + 2\gamma, \hat{\epsilon}(h^\star) \leq B + \gamma) \leq k \exp\left\{\frac{-m\gamma^2}{2(B + 7\gamma/3)}\right\}. \tag{52}$$

Answer: Using the definition of conditional probability and applying (51) with $B' = B + \gamma$ and $\tau = \gamma$, we get

$$\begin{aligned} P(\epsilon(h) > B + 2\gamma, \hat{\epsilon}(h) \leq B + \gamma) &= P(\hat{\epsilon}(h) \leq B + \gamma | \epsilon(h) > B + 2\gamma) P(\epsilon(h) > B + 2\gamma) \\ &\leq P(\hat{\epsilon}(h) \leq B + \gamma | \epsilon(h) > B + 2\gamma) \\ &\leq \exp\left\{\frac{-m\tau^2}{2(B + 7\tau/3)}\right\}. \end{aligned}$$

We can then use the fact that the summation over all hypotheses in $\mathcal{H}$ is from 1 to k to show that

$$P(\epsilon(\hat{h}) > B + 2\gamma, \hat{\epsilon}(h^\star) \leq B + \gamma) \leq k \exp\left\{\frac{-m\gamma^2}{2(B + 7\gamma/3)}\right\}.$$