CS181 Practice Questions: Generative Classification

1. **MLE for Probabilistic Classification (Bishop 4.9)**

Consider a generative classification model for K classes defined by prior class probabilities $p(\mathcal{C}_k) = \pi_k$ and general class-conditional densities $p(\phi|\mathcal{C}_k)$ where ϕ is the input feature vector. Suppose we are given a training data set $\{\phi_n, \mathbf{t}_n\}$ where $n = 1, \ldots, N$ and $\mathbf{t}_n$ is a binary target vector of length K that uses the 1-of-K coding scheme, so that it has components $t_{nj} = I_{jk}$ if pattern n is from class $\mathcal{C}_k$. Assuming that the data points are drawn independently from this model, show that the maximum-likelihood solution for the prior probabilities is given by:

$$\pi_k = \frac{N_k}{N},$$

where N_k is the number of data points assigned to class $\mathcal{C}_k$.

The likelihood function is pretty straightforward to represent here because of our 1-of-K coding scheme:

$$p(\{\phi_n, \mathbf{t}_n\}|\{\pi_k\}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \{p(\phi_n|\mathcal{C}_k)\pi_k\}^{t_{nk}}$$

We take the log-likelihood, since maximizing this is the same as maximizing the likelihood:

$$\ln p(\{\phi_n, \mathbf{t}_n\}|\{\pi_k\}) = \sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk}\{\ln p(\phi_n|\mathcal{C}_k) + \ln \pi_k\}$$

To maximize with respect to π_k, we need to preserve the constraint $\sum_k \pi_k = 1$. This can be done with Lagrange multipliers. Let us introduce multiplier λ:

$$\ln p(\{\phi_n, t_n\}|\{\pi_k\}) + \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right).$$

Setting the derivative with respect to π_k equal to zero, we have:

$$\sum_{n=1}^{N} \frac{t_{nk}}{\pi_k} + \lambda = 0$$

Solving gives us:

$$-\pi_k \lambda = \sum_{n=1}^{N} t_{nk} = N_k$$

Summing both sides over k, we get that $\lambda = -N$. Then, we get that:

$$\pi_k = \frac{N_k}{-\lambda} = \frac{N_k}{N},$$

as desired.

2. **MLE for Gaussian Probabilistic Classification (Bishop 4.10)**

Consider the classification model of the previous exercise and now suppose that class-conditional densities are given by Gaussian distributions with a shared covariance matrix, so that

$$p(\phi|\mathcal{C}_k) = \mathcal{N}(\phi|\mu_k, \mathbf{\Sigma})$$

Show that the maximum likelihood solution for the mean of the Gaussian distribution for class $\mathcal{C}_k$ is given by

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} t_{nk} \phi_n$$

which represents the mean of those feature vectors assigned to class $\mathcal{C}_k$. Similarly, show that the maximum likelihood solution for the shared covariance matrix is given by

$$\mathbf{\Sigma} = \sum_{k=1}^{K} \frac{N_k}{N} \mathbf{S}_k$$

where

$$\mathbf{S}_k = \frac{1}{N_k} \sum_{n=1}^{N} t_{nk} (\phi_n - \mu_k)(\phi_n - \mu_k)^\mathsf{T}.$$

Thus $\mathbf{\Sigma}$ is given by a weighted average of the covariances of the data associated with each class, in which the weighting coefficients are given by the prior probabilities of the classes.

The log-likelihood, using the probability density of the multivariate Gaussian, we have:

$$\ln p(\{\phi_n, t_n\}|\{\pi_k\}) = \\ -\frac{1}{2} \sum_{n=1}^{n} \sum_{k=1}^{K} t_{nk} \{\ln |\mathbf{\Sigma}| + (\phi_n - \mu_k)^\mathsf{T} \mathbf{\Sigma}^{-1} (\phi - \mu)\},$$

where we have dropped terms independent of $\{\mu_k\}$ and $\mathbf{\Sigma}$. Taking the derivative with respect to μ_k, we get:

$$\sum_{n=1}^{n} \sum_{k=1}^{K} t_{nk} \Sigma^{-1} (\phi_n - \mu_k) = 0.$$

We can rearrange this to get μ_k, as desired. The RHS of the log-likelihood can actu-

ally be expressed as follows:

$$-\frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk}\{\ln|\mathbf{\Sigma}| + \mathrm{Tr}[\mathbf{\Sigma}^{-1}(\phi_n - \mu_k)(\phi - \mu_k)^{\mathsf{T}}]\}$$

Taking the derivative with respect to Σ^{-1}, we get:

$$\frac{1}{2}\sum_{n=1}^{N}\sum_{k}^{T} t_{nk}\{\mathbf{\Sigma} - (\phi_n - \mu_n)(\phi_n - \mu_k)^{\mathsf{T}}\} = 0$$

Once again, we rearrange this to get the desired $\mathbf{\Sigma}$.

3. **Gaussian Decision Boundaries (Murphy 4.21)**

Consider a two-class, generative classification model where $p(x \mid \mathcal{C}_j) = \mathcal{N}(x \mid \mu_j, \sigma_j^2)$. Let $\mu_1 = 0$, $\sigma_1^2 = 1$, $\mu_2 = 1$, $\sigma_2^2 = 10^6$, and let the class priors be $p(\mathcal{C}_1) = p(\mathcal{C}_2) = \frac{1}{2}$. Find the decision region $\mathcal{R}_1 = \{x \mid p(\mathcal{C}_1 \mid x) \geq p(\mathcal{C}_2 \mid x)\}$.
Hint: find the solutions of the equation $p(x|\mu_1, \sigma_1^2) = p(x|\mu_2, \sigma_2^2)$, and recall that to solve a quadratic equation $ax^2 + bx + c = 0$, we use

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

Because our priors on each class are equal, we have that $\mathcal{R}_1 = \{x \mid p(x|\mu_1, \sigma_1^2) \geq p(x|\mu_2, \sigma_2^2)\}$. We can then solve for the decision boundary by setting the probabilities to be equal:

$$\begin{aligned}
&p(x|\mu_1, \sigma_1^2) = p(x|\mu_2, \sigma_2^2) \\
&\Rightarrow \mathcal{N}(x|\mu_1, \sigma_1^2) = \mathcal{N}(x|\mu_2, \sigma_2^2) \\
&\Rightarrow \frac{1}{\sqrt{2\pi\sigma_1^2}} \exp\left(\frac{-1}{2\sigma_1^2}(x - \mu_1)^2\right) = \frac{1}{\sqrt{2\pi\sigma_2^2}} \exp\left(\frac{-1}{2\sigma_2^2}(x - \mu_2)^2\right) \\
&\Rightarrow -\ln\sigma_1 - \frac{1}{2\sigma_1^2}(x - \mu_1)^2 = -\ln\sigma_2 - \frac{1}{2\sigma_2^2}(x - \mu_2)^2 \quad \text{[take logs]} \\
&\Rightarrow -0 - \frac{1}{2}x^2 = -\ln 10^3 - \frac{1}{2 \cdot 10^6}(x - 1)^2 \quad \text{[substitute known values]} \\
&\Rightarrow 0 = x^2 - \frac{x^2}{10^6} + \frac{2x}{10^6} - \frac{1}{10^6} - 2\ln 10^3 \\
&\Rightarrow x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \approx \pm 3.7169,
\end{aligned}$$

where $a = (1 - \frac{1}{10^6})$, $b = \frac{2}{10^6}$, $c = -\frac{1}{10^6} - 2\ln 10^3$. Therefore, $\mathcal{R}_1$ is the region between -3.7169 and 3.7169. Note that this implies that $\mathcal{R}_2$ is composed of discontiguous regions!

4. **Logistic Regression vs LDA/QDA (Murphy 4.20)**

Suppose we train the following binary classifiers via maximum likelihood:

- GaussI: A generative classifier, where the class conditional densities are Gaussian, with both covariance matrices set to $\boldsymbol{I}$, i.e., $p(\boldsymbol{x}|\mathcal{C}_j) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_j, \boldsymbol{I})$.
- GaussX: same as GaussI, but the covariance matrices are unconstrained, i.e., $p(\boldsymbol{x}|\mathcal{C}_j) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$.
- LinLog: A logistic regression model with linear features.
- QuadLog: A logistic regression model, using linear and quadratic features.

Now suppose that after training, we evaluate the conditional log-likelihood of the *training set* under each model. That is, for each model we evaluate

$$L(\theta, M) = \sum_{n=1}^{N} \ln p(t_n|\phi(\boldsymbol{x}_n), \theta),$$

where θ are our MLE parameters and M the model in question. For each of the following pairs of models, indicate which model will have lower (or equal) $L(\theta, M)$, or indicate that no such statement can be made. Explain your answers.

a. GaussI, LinLog

b. GaussX, QuadLog

c. LinLog, QuadLog

d. GaussI, QuadLog

Finally, is it in general true that a classifier that gives a higher $L(\theta, M)$ on the training set will have fewer classification errors on the training set?

a. GaussI $\leq$ LinLog. As Bishop indicates, the conditional log-likelihood of both models will have the same form (i.e., sum of logistic functions), but training LinLog will optimize this criterion directly, and so will be at least as high as GaussI.

b. GaussX $\leq$ QuadLog. Similar to the previous, both of these have logistic conditional log-likelihoods with quadratic features, but QuadLog optimizes the conditional log-likelihood directly.

c. LinLog $\leq$ QuadLog. QuadLog has all the features of LinLog and more, and so will do at least as well as *on the training data*.

d. GaussI $\leq$ QuadLog. Implied by above inequalities.

No, it isn't. It's easiest to see that a classifier M_1 can have *as many* classification errors as a classifier M_2, even if $L(\theta, M_1) > L(\theta, M_2)$. For instance, an unregularized

logistic regression model might assign infinite log-probability to a linearly separable dataset, and a generative classifier might not, even if both models misclassify no training examples. Or, similarly, two classifiers might make the same predictions (and thus have the same classification error), but one might be much more certain than the other. (This can improve conditional log-likelihood if the classifier is both certain and right, or make it worse if it is certain and wrong).