MLPR Tutorial[1] Sheet 5

In week 3 we covered *generative* or *Bayes* classifiers, which model the joint distribution $p(y, \mathbf{x}) = p(\mathbf{x} \mid y)\, P(y)$, and then use Bayes rule to derive the class probabilities $P(y \mid \mathbf{x})$.

Week 7 covers logistic regression, a *discriminative* approach, where we directly model $P(y \mid \mathbf{x})$. Logistic regression is widely used, and the basis for deeper neural network classifiers.

We need gradient-based optimizers to fit logistic regression, but it often gives more accurate predictions than simple generative models. This tutorial compares the computation of the models, and explores logistic regression a bit further.

---

1. **The costs of some classifiers:**

   We have a training set of N examples, with D-dimensional features and binary labels. This question gets you to revisit what computations you need to do to build and use a classifier.

   Assume the following computational complexities: matrix-matrix multiplication AB costs $O(LMN)$ for $L \times M$ and $M \times N$ matrices A and B. Inverting an $N \times N$ matrix G and/or finding its determinant costs $O(N^3)$.[2]

   a) What is the computational complexity of training a "Bayes classifier" that models the features of each class by a maximum likelihood Gaussian fit (a Gaussian matching the mean and covariances of the features)?

   b) What is the computational complexity of assigning class probabilities to a test feature vector?

   c) In *linear discriminant analysis* we assume the classes just have different means, and share the same covariance matrix. Show that given its parameters θ, the log likelihood ratio for a binary classifier,

   $$\log \frac{p(\mathbf{x} \mid y=1,\, \theta)}{p(\mathbf{x} \mid y=0,\, \theta)} \;=\; \log p(\mathbf{x} \mid y=1,\, \theta) - \log p(\mathbf{x} \mid y=0,\, \theta),$$

   is a linear function of $\mathbf{x}$ (as opposed to quadratic).

   d) When the log likelihood ratio is linear in $\mathbf{x}$, the log posterior ratio is also linear. Reviewing Q3 in tutorial 4, we can recognize that when the log posterior ratio is linear, the predictions have the same form as logistic regression:

   $$P(y=1 \mid \mathbf{x},\, \theta) = \sigma(\mathbf{w}^\top \mathbf{x} + b),$$

   but the parameters $\mathbf{w}(\theta)$ and $b(\theta)$ are fitted differently.

   How do your answers to a) and b) change when the classes share one covariance? What can you say about the cost of the linear discriminant analysis method compared to logistic regression?

2. **Gradient descent:**

   Let $E(\mathbf{w})$ be a differentiable function. Consider the gradient descent procedure

   $$\mathbf{w}^{(t+1)} \leftarrow \mathbf{w}^{(t)} - \eta \nabla_{\mathbf{w}} E.$$

---

1. Parts of this tutorial sheet are based on previous versions by Amos Storkey, Charles Sutton, and Chris Williams
2. In fact, good implementations usually take a factorization of G rather than inverting it, which also costs $O(N^3)$. Given that factorization, we can compute $G^{-1}H$ in $O(N^2K)$, where H is $N \times K$, and find the (log) determinant in $O(N)$. I'm using "big-O notation". A good introduction on this notation in the context of computational complexity are the notes from our second year Inf2b course. Applied areas like machine learning usually use big-O notation more sloppily than in that note; it's rare to see Ω or Θ.

a) Are the following true or false? Prepare a clear explanation, stating any necessary assumptions:

i) Let $\mathbf{w}^{(1)}$ be the result of taking one gradient step. Then the error never gets worse, i.e., $E(\mathbf{w}^{(1)}) \leq E(\mathbf{w}^{(0)})$.

ii) There exists some choice of the step size η such that $E(\mathbf{w}^{(1)}) < E(\mathbf{w}^{(0)})$.

b) A common programming mistake is to forget the minus sign in either the descent procedure or in the gradient evaluation. As a result one unintentionally writes a procedure that does $\mathbf{w}^{(t+1)} \leftarrow \mathbf{w}^{(t)} + \eta \nabla_{\mathbf{w}} E$. What happens?

3. **Maximum likelihood and logistic regression:**

Maximum likelihood logistic regression maximizes the log probability of the labels,

$$\sum_n \log P(y^{(n)} \,|\, \mathbf{x}^{(n)}, \mathbf{w}),$$

with respect to the weights $\mathbf{w}$. As usual, $y^{(n)}$ is a binary label at input location $\mathbf{x}^{(n)}$.

The training data is said to be *linearly separable* if the two classes can be completely separated by a hyperplane. That means we can find a decision boundary

$$P(y^{(n)} = 1 \,|\, \mathbf{x}^{(n)}, \mathbf{w}, b) = \sigma(\mathbf{w}^\top \mathbf{x}^{(n)} + b) = 0.5, \qquad \text{where } \sigma(a) = \frac{1}{1+e^{-a}},$$

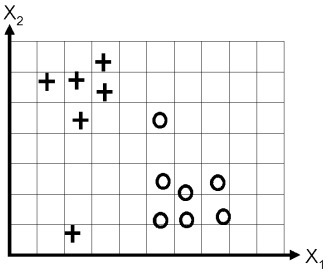
such that all the $y=1$ labels are on one side (with probability greater than 0.5), and all of the $y \neq 1$ labels are on the other side.

a) Show that if the training data is linearly separable with a decision hyperplane specified by $\mathbf{w}$ and b, the data is also separable with the boundary given by $\tilde{\mathbf{w}}$ and $\tilde{b}$, where $\tilde{\mathbf{w}} = c\mathbf{w}$ and $\tilde{b} = cb$ for any scalar $c > 0$.

b) What consequence does the above result have for maximum likelihood training of logistic regression for linearly separable data?

4. **Logistic regression and maximum likelihood:** (Murphy, Exercise 8.7, by Jaaakkola.)

Consider the following data set:



a) Suppose that we fit a logistic regression model with a bias weight w_0, that is $p(y = 1 \,|\, \mathbf{x}, \mathbf{w}) = \sigma(w_0 + w_1 x_1 + w_2 x_2)$, by maximum likelihood, obtaining parameters $\hat{\mathbf{w}}$. Sketch a possible decision boundary corresponding to $\hat{\mathbf{w}}$. Is your answer unique? How many classification errors does your method make on the training set?

b) Now suppose that we regularize only the w_0 parameter, that is, we minimize

$$J_0(\mathbf{w}) = -\ell(\mathbf{w}) + \lambda w_0^2,$$

where ℓ is the log-likelihood of $\mathbf{w}$ (the log-probability of the labels given those parameters).

Suppose λ is a very large number, so we regularize w_0 all the way to 0, but all other parameters are unregularized. Sketch a possible decision boundary. How many classification errors does your method make on the training set? Hint: consider the behaviour of simple linear regression, $w_0 + w_1 x_1 + w_2 x_2$ when $x_1 = x_2 = 0$.

c) Now suppose that we regularize only the w_1 parameter, i.e., we minimize

$$J_1(\mathbf{w}) = -\ell(\mathbf{w}) + \lambda w_1^2.$$

Again suppose λ is a very large number. Sketch a possible decision boundary. How many classification errors does your method make on the training set?

d) Now suppose that we regularize only the w2 parameter, i.e., we minimize

$$J_2(\mathbf{w}) = -\ell(\mathbf{w}) + \lambda w_2^2.$$

Again suppose λ is a very large number. Sketch a possible decision boundary. How many classification errors does your method make on the training set?