CS 194-10 Introduction to Machine Learning
Fall 2011 Stuart Russell Midterm Solutions

1. (20 pts.) Some Easy Questions to Start With

(a) (4) *True/False*: *In a least-squares linear regression problem, adding an L_2 regularization penalty cannot decrease the L_2 error of the solution $\hat{\mathbf{w}}$ on the training data.*
True. The L_2 error is already minimzed by the unregularized solution, so no form of regularization can improve on that.

(b) (4) *True/False*: *In a least-squares linear regression problem, adding an L_2 regularization penalty always decreases the expected L_2 error of the solution $\hat{\mathbf{w}}$ on unseen test data.*
False. Having such a guarantee would require predicting the future. Regularization can make things worse, e.g., if the true model requires large weights.

(c) (4) *True/False*: *In a regression problem, decision trees with constant leaves can fit every data set with zero training error.*
False. Cannot fit data with different y values for the same x.

(d) (4) *True/False*: *As the width b goes to 0 in locally weighted regression with a quadratic kernel $K(d) = max(0, 1 - d^2/b^2)$, the algorithm behaves exactly like 1-nearest-neighbor.*
False. As the width goes to 0, all the example weights go to zero, so the method fails. It does, however, perform like nearest neighbor when b is larger than the distance to the nearest neighbor but smaller than the distance to the second-nearest neighbor.

(e) (4) *True/False*: *In decision tree learning with noise-free data, starting with the wrong attribute at the root can make it impossible to find a tree that fits the data exactly.*
False. It can make the tree larger than necessary, but does not affect the expressive power of the model.

2. (10 pts.) Locally weighted regression
Can locally weighted regression exactly reproduce the learning behavior of ordinary least-squares regression, given a suitable kernel, for any data set? If so, how? If not, why not?
Yes; simple use the kernel $K(d) = c$ for any constant c. Then all examples are weighted equally and the algorithm behaves exactly like ordinary unweighted regression.

Some students insisted that such a solution would not work because kernels, according to a certain blue textbook, have to have a bounded integral. For locally weighted regression, this is not necessary, although for kernel density estimation it is. In these cass no credit was lost.

3. (10 pts.) Regularization
The standard form of the L_2-regularized L_2 loss function for linear regression is

$$L = (\mathbf{Y} - \mathbf{X}\mathbf{w})^T(\mathbf{Y} - \mathbf{X}\mathbf{w}) + \lambda \mathbf{w}^T\mathbf{w} \quad \text{where } \lambda > 0.$$

(a) (3) *Suppose we accidentally write $L = (\mathbf{Y} - \mathbf{X}\mathbf{w})^T(\mathbf{Y} - \mathbf{X}\mathbf{w}) + \lambda \mathbf{Y}^T\mathbf{Y}$ instead. Explain why this form of "regularization" has no effect.*
The optimal solution for $\mathbf{w}$ is determined by setting $\partial L/\partial \mathbf{w}$ to zero; since $\partial \mathbf{Y}^T\mathbf{Y}/\partial \mathbf{w} = 0$, it has no effect on the solution.

(b) (3) *Suppose we use the correct expression but accidentally choose $\lambda < 0$. Explain briefly how this defeats the purpose of regularization.*
L_2 regularization aims to avoid overfitting by keeping weights small. Setting $\lambda < 0$ has the opposite effect, so it is likely to encourage overfitting. Note that in this case it will not send $\mathbf{w}^T\mathbf{w}$ to infinity, because with L_2 error that will cause infinite error. The normal equations will still have a solution.

(c) (4) *In ordinary least squares, the squared-error loss measures error in the y direction. In* total least squares*, error is measured by the orthogonal distance from the point to the line (i.e., the length of the perpendicular). Explain briefly why regularizing with $\lambda < 0$ would be disastrous in this case. You may find it helpful to consider univariate regression ($y = w_0 + w_1 x$) as an example.*
For total least squares, the situation is different because it is possible for $\mathbf{w}^T\mathbf{w}$ to go to infinity while keeping the error bounded. Fix a point relative to the data set, e.g., the centroid. For any line through this point, the orthogonal distances are bounded. On the other hand, by sending w_0 and w_1 to $-\infty$ or $+\infty$, we can make a vertical line with $\mathbf{w}^T\mathbf{w} = \infty$. Thus, minimizing L with negative λ leads to this degenerate solution.

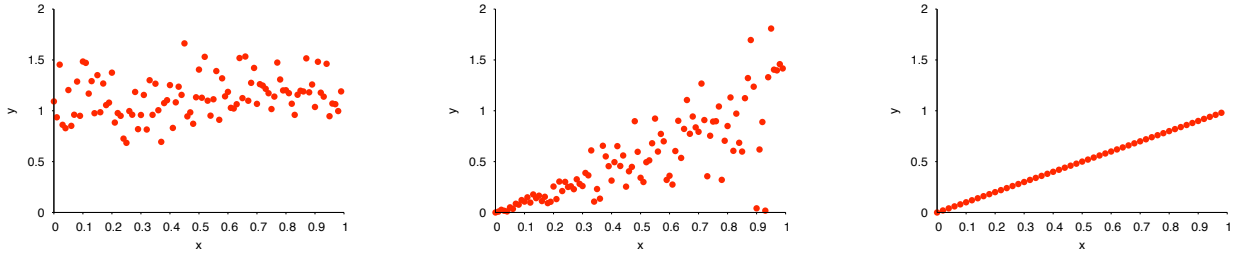

Figure 1: Three data sets.

4. (20 pts.) Input-dependent noise in regression

Ordinary least-squares regression is equivalent to assuming that each data point is generated according to a linear function of the input plus zero-mean, constant-variance Gaussian noise. In many systems, however, the noise variance is itself a positive linear function of the input (which is assumed to be non-negative, i.e., $x \geq 0$).

(a) (5) *Which of the following families of probability models correctly describes this situation in the univariate case? (Hint: only one of them does.)*

i.
$$P(y \mid x) = \frac{1}{\sigma x\sqrt{2\pi}} \exp\left(-\frac{(y - (w_0 + w_1 x))^2}{2x^2\sigma^2}\right)$$

ii.
$$P(y \mid x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(y - (w_0 + (w_1 + \sigma^2)x))^2}{2\sigma^2}\right)$$

iii.
$$P(y \mid x) = \frac{1}{\sigma\sqrt{2\pi x}} \exp\left(-\frac{(y - (w_0 + w_1 x))^2}{2x\sigma^2}\right)$$

(iii) is correct. In a Gaussian distribution over y, the variance is determined by the coefficient of y^2; so by replacing σ^2 by $x\sigma^2$, we get a variance that increases linearly with x. (Note also the change to the normalization "constant.") (i) has quadratic dependence on x; (ii) does not change the variance at all, it just renames w_1.

(b) (6) *Circle the plots in Figure 1 that could plausibly have been generated by some instance of the model family(ies) you chose.*
(ii) and (iii). (Note that (iii) works for $\sigma^2 = 0$.) (i) exhibits a large variance at $x = 0$, and the variance appears independent of x.

(c) (3) *True/False*: *Regression with input-dependent noise gives the same solution as ordinary regression for an infinite data set generated according to the corresponding model.*
True. In both cases the algorithm will recover the true underlying model.

(d) (6) *For the model you chose in part (a), write down the derivative of the negative log likelihood with respect to w_1.*

The negative log likelihood is

$$L = -\sum_{i=1}^{N} -\log \sigma\sqrt{2\pi x_i} - \frac{(y_i - (w_0 + w_1 x_i))^2}{2x_i\sigma^2}$$

and the derivative w.r.t. w_1 is

$$\frac{\partial L}{\partial w_1} = -\sum_{i=1}^{N} (y_i - (w_0 + w_1 x_i))/\sigma^2 \ .$$

Note that for lines through the origin ($w_0 = 0$), the optimal solution has the particularly simple form $\hat{w_1} = \bar{y}/\bar{x}$.

It is possible to take the derivative of the log without noticing that $\log \exp(x) = x$; we use log likelihoods for a good reason! Plus, they simplify the handling of multiple data points, because the product of probabilities becomes a sum of log probabilities.

5. (20 pts.) Classifying data

(i)

X_1	X_2	Y
1	1	+
4	2	−
4	5	−
5	5	+

(ii)

X_1	X_2	Y
1	1	+
5	5	−
4	5	−
5	5	+

(iii)

X_1	X_2	Y
1	1	+
4	2	−
4	5	+
5	5	+

(a) (3) *Multiple choice*: *Which data sets are linearly separable?*
Only (iii) is linearly separable. In (i) the two classes form opposite corners of a quadrilateral, while in (ii) the point (5,5) has two classes, so the classes cannot be separated by any line.

(b) (3) *Multiple choice*: *Which data sets have label noise?*
(ii).

(c) (3) *Multiple choice*: *Which data sets can be fit exactly by a decision tree?*
(i) and (iii). Any set without label noise can be fit exactly.

(d) (5) *A* 1-decision-list *is a decision tree in which the "yes" branch of every binary test is a leaf node. For a continuous attribute X_j, a test can be either $X_j > c$ or $X_j < c$. Continuous attributes can appear in multiple tests. Pick a data set and show a decision list that fits it exactly.*
Decision lists are "cond statements" in Lisp and Scheme, or if-then-elseif-then-elseif-then-elseif ... statements.
For data set (iii): If $X_2 > 3$ then + else if $X_1 > 3$ then – else +.

(e) (6) *In the absence of label noise, can any two-class data set in two dimensions be fit exactly by a decision list? Briefly explain why, or give a counterexample.*
One might generalize the idea used in (d), which is basically "pick off sets of + or – examples from the extreme values of any dimension until there are none left." But this does not always work—consider a square with alternating corners labeled + and –. Such data sets cannot be fit by decision lists. This is essentially the XOR example that caused problems for perceptrons in 1969; but note that decision lists as described are more expressive than perceptrons; e.g., they have no trouble with the example in (i).