Solutions Manual

Introduction to Machine Learning with Applications in Information Security

by Mark Stamp

May 9, 2020

A Note to Instructors

For my previous book, *Information Security: Principles and Practice*, published by Wiley, I provided a solutions manual with almost every problem solved in detail. After a short period of time, many students began submitting exactly these same solutions. It's probably inevitable that solutions will become available, but I don't want those solutions to be mine.

For this "solutions" manual, I've only provided brief (or nonexistent) solutions, which (at best) are designed to confirm that you are on the right track. Many of the solutions include blanks, like this: $\boxed{}$. The blanks can represent digits or words, for example, and whatever is missing should be reasonably clear from context. I've included a few comments and references where it seemed appropriate.

Believe it or not, I find it more difficult to write these partial solutions than to provide complete solutions. As a result, there are quite a few problems here that have no solution. I plan to add more solutions (or hints), so you might want to check for an updated version periodically.

Chapter 2

1. a) For example,

$$
\begin{aligned}
P(\mathcal{O}, X = CHC) &= 1.0 \cdot 0.2 \cdot 0.4 \cdot 0.1 \cdot 0.3 \cdot 0.1 = 0.00024 \\
P(\mathcal{O}, X = CCH) &= 1.0 \cdot 0.2 \cdot 0.6 \cdot 0.7 \cdot 0.4 \cdot 0.5 = 0.00504
\end{aligned}
$$

The sum is 0.0□4□□.

b) For example,

$$
\begin{aligned}
\alpha_0(1) &= 1.0 \cdot 0.2 = 0.2 \\
\alpha_2(0) &= (0.008 \cdot 0.7 + 0.084 \cdot 0.4) \cdot 0.5 = 0.0392
\end{aligned}
$$

Again, sum is 0.0□4□□.

c) The work factor for the brute force approach is $2TN^T$, while the work factor for the forward algorithm is N^2T. Since T is typically large and N is small, the forward algorithm (aka α pass) is the way to go.

2. a) The best hidden state sequence, in the HMM sense is $(X_0, X_1, X_2) = (C, C, H)$.

b) The best hidden state sequence, in the DP sense is also $(X_0, X_1, X_2) = (C, C, H)$, so it's a trick question.

3. a) In this case, there are $3^4 = 81$ observation sequences. Students would be well advised to write a program to solve this, as working it by hand would take forever and it would be highly error-prone. A correct solution would require all 81 probabilities, and they must sum to 1. As a couple of random examples, $P(1, 1, 0, 1) = 0.009757$ and $P(2, 1, 0, 1) = 0.010458$.

b) This is the same as part a), except that the forward algorithm is used instead of a direct calculation. Again, students need to provide all 81 probabilities and show that they sum to 1. When I teach this class, I require that students submit a working program for this part.

4. Derivation.

5. a) The backward algorithm computes the probabilities of partial sequences beginning at the end of the observation sequence and working towards the beginning. Hence,

$$
P(\mathcal{O} \mid \lambda) = \sum_{i=0}^{N-1} \boxed{}.
$$

b) Write a program and verify that using the formula in part a) gives the same result as the forward algorithm.

6. a) For example, the re-estimation formula for π_i can be written as

$$\pi_i = \gamma_0(i) = \sum_{j=0}^{N-1} \gamma_t(i,j) = \sum_{j=0}^{N-1} \frac{\alpha_0(i)a_{ij}b_j(\mathcal{O}_1)\beta_1(j)}{P(\mathcal{O}\,|\,\lambda)}.$$

The re-estimation formulas for the elements of the A and B matrices are slightly more complex.

b) For example, from part a), we have

$$\pi_i = \sum_{j=0}^{N-1} \frac{\alpha_0(i)a_{ij}b_j(\mathcal{O}_1)\beta_1(j)}{P(\mathcal{O}\,|\,\lambda)}$$

where

$$P(\mathcal{O}\,|\,\lambda) = \sum_{j=0}^{N-1} \alpha_{T-1}(j).$$

Substituting $\widehat{\alpha}_t(i)$ and $\widehat{\beta}_t(j)$, the term in the numerator is

$$\begin{aligned}\widehat{\alpha}_0(i)a_{ij}b_j(\mathcal{O}_1)\widehat{\beta}_1(j) &= (c_0c_1\cdots c_t)\alpha_t(i)a_{ij}b_j(\mathcal{O}_{t+1})(c_{t+1}c_{t+2}\cdots c_{T-1})\beta_{t+1}(j)\\ &= (c_0c_1\cdots c_{T-1})\alpha_t(i)a_{ij}b_j(\mathcal{O}_{t+1})\beta_{t+1}(j)\end{aligned}$$

while the denominator is

$$\sum_{j=0}^{N-1} \widehat{\alpha}_{T-1}(j) = \sum_{j=0}^{N-1}(c_0c_1\cdots c_{T-1})\alpha_{T-1}(j).$$

It follows that we can compute the exact value of π_i using the same formula, with the scaled values in place of the unscaled values.

7. a) Derivation

b) We can use either

$$\log\bigl(P(\mathcal{O}\,|\,\lambda)\bigr) = -\sum_{j=0}^{T-1} \log c_j$$

or

$$\log\bigl(P(\mathcal{O}\,|\,\lambda)\bigr) = -\sum_{j=0}^{T-1} \log d_j.$$

8. Plug these values into the re-estimation formulas and show that the same values as given in the book are obtained as the re-estimates.

9. a) Similar to the pseudo-code given for the α-pass in the book, but use matrices B_t in place of B.

b) Similar to the pseudo-code given for the α-pass in the book, but the equivalent of the B matrix will depend on the current state X_t and the previous state X_{t-1}, instead of just depending on X_t. Although not required for the solution to this problem, note that the re-estimation process becomes fairly complex in this case. Of course, for higher order HMMs, things become progressively more complex.

10. a) Programming problem. Must be clear that the model converged—see Section 9.2 for an example.

    b) Programming problem.

    c) Programming problem.

    d) Programming problem.

11. a) Programming problem.

    b) Closely related to the English text example given in Section 9.2 and in problem 10, above.

    c) Programming problem. Note that an alternative (and perhaps better) approach here would be to simply generate an HMM using a large number of characters of English text and use the resulting A matrix.

    d) Programming problem. An alternative (and pehaps better) approach to determine the key would be to generate an HMM using a large number of characters of English text, then try to match rows of the B^T obtained for this ciphertext problem to the rows of B^T from the English text version of the problem. The only issue with such an approach is that the hidden states do not need to match in both cases.

12. a) Programming problem.

    b) Programming problem.

    c) Programming problem. This will be particularly challenging, even if restricted to just a couple of thousand characters, as a very large amount of text will be needed (and a correspondingly large amount of computation) to ensure a reasonable chance of convergence.

13. Programming problem. This paper discusses an HMM analysis of Hamptonese: E. Le and M. Stamp, Hamptonese and hidden Markov models, *Lecture Notes in Control and Information Sciences*, Vol. 321, New Directions and Applications in Control Theory, Springer 2005, W. P. Dayawansa, A. Lindquist, and Y. Zhou, editors, pp. 367–378.

14. a) Programming problem.

    b) Programming problem. Results can vary, depending on the initialization strategy used, but in general we expect to obtain better results with more data and/or more random restarts. Also, note that if we first do $n = 1000$ random restarts, we can average each of the 1000 results and consider the result as the solution for the $n = 1$

case (averaged over 1000 trials), while the very best solution over those 1000 restarts would be the answer for the $n = 1000$ case. Similarly, we can obtain averages for the $n = 10$ (over 100 trials) and $n = 100$ (over 10 trials) cases, all from a single experiment with 1000 random restarts.

c) Programming problem.

d) Programming problem.

15. a) Programming problem. Note that the program will need to be very efficient to test 1,000,000 random restarts—a program that takes only 1 second per restart will require more than 11.5 days to complete 1,000,000 random restarts. Also, another issue that is not addressed here is that the distribution on the random restarts can matter. That is, it might make a difference if the restarts are initialized close together (i.e., with a small variance) or farther apart (larger variance). It is interesting to experiment with this aspect as well.

b) Programming problem.

c) Programming problem.

d) Programming problem.

e) Programming problem.

f) Programming problem.

g) Programming problem.

h) Programming problem.

16. a) Programming problem. Anyone who can solve the Zodiac 340 would certainly get an A+ in my class!

b) Programming problem.

Chapter 3

1. Easy.

2. a) Not sure that there is a nice closed form expression for any of these.

   b) TBD

   c) TBD

   d) TBD

3. For example, column 1 is a match state and the emission probabilities are given by $P(\texttt{A}) = P(\texttt{C}) = 5/28$, $P(\texttt{2}) = 2/28$, and $P(x) = 1/28$ for all symbols $x \notin \{\texttt{A}, \texttt{C}, \texttt{2}\}$.

4. TBD

5. a) These plots nicely highlight local alignments.

   b) See part a).

   c) See part a).

   6. TBD

   TBD

7. a) Use the same dynamic program used to generate the pairwise alignments, but modify the gap penalty function g for the (partial) MSA under consideration.

   b) The obvious disadvantage is that this alternative approach is much slower.

8. TBD

9. Verification.

10. a) TBD

    b) TBD

    c) TBD

    d) TBD

11. a) There are 63 states, one of which, for example, is (I_0, I_3, M_2). To get credit (at least in my class), you need to list all of these states.

    b) The answer here is 25. For example, one of the 25 sequences is (I_0, M_1, M_2).

    c) There are also 25 states here, including (I_3, M_1)

12. a) TBD

    b) Yes.

Chapter 4

1. a) Easy.

   b) Easy.

   c) Easy.

   d) Easy.

   e) Undefined.

   f) Undefined

2. a) Amongst other things, reducing the dimensionality serves to concentrate the relevant statistical information, which might otherwise be sparse, and hence difficult to separate from the noise.

   b) The key point is that in PCA we obtain linear combinations of input features, which are in a lower dimensional space (assuming we eliminate some directions corresponding to small eigenvalues). However, we do not directly reduce the number of input features (more about this topic below).

3. If a student studies for a test, they might focus on a few major topics, as these are likely to be the source of most of the test questions. So, a student might be able to do very well, even though they have ignored various aspects of the material. However, this strategy is not advisable in my class.

4. a)

$$C = \begin{bmatrix} \boxed{} & 2.5 & \boxed{} \\ 2.5 & \boxed{} & \boxed{} \\ 4.0 & \boxed{} & \boxed{} \end{bmatrix}$$

   b) $\lambda_1 = 1.5$ and $\lambda_2 = \boxed{}$.

   c) Unit eigenvector for λ_1 is

$$\begin{bmatrix} \boxed{} \\ -0.816497 \\ \boxed{} \end{bmatrix}$$

   Unit eigenvector for λ_2 is

$$\begin{bmatrix} \boxed{} \\ \boxed{} \\ 0.707107 \end{bmatrix}$$

5. a) If x is an eigenvalue of $\widehat{C}$, then $\widehat{C}x = \lambda x$, and $C = 1/n\widehat{C}$. The answer is straightforward from here.

   b) This is clear from the solution to part a).

6. That the score is 0 is clear from the definition. This is also clearly desirable.

7. a) TBD

   b) TBD

   c) Verify that all of the eigenvectors u_i are unit vectors. TBD

8. a) Multiply by A on both sides to obtain $AA^TAv_i = \lambda_i Av_i$. Thus the eigenvectors of C are Av_i with eigenvalues λ_i/n.

   b) The matrix A^TA is $n \times n$ matrix while C is $m \times m$.

   c) TBD

9. a) The variances appear on the main diagonal of the covariance matrix. Summing these elements, in this case we obtain a total variance of $6.\boxed{}6$, to 2 decimal places.

   b) Sum the eigenvalues.

   c) Using only the first eigenvector accounts for a fraction of λ_1/s of the total variance, where s is the sum of the eigenvalues, and so on.

10. Let s_i be the score of Y_i. Then $s_1 = 0.\boxed{}0$, $s_2 = 3.7\boxed{}$, $s_3 = 0.9\boxed{}$, and $s_4 = 0.\boxed{}8$.

11. a)

$$\Delta = \begin{bmatrix} \boxed{} & -4.38 & -\boxed{} & 3.74 \\ -1.16 & \boxed{} & -\boxed{} & 1.44 \\ 1.53 & \boxed{} & -0.54 & -\boxed{} \end{bmatrix}$$

   b) Let s_i be the score of Y_i. Then $s_1 = 3.\boxed{}7$, $s_2 = 0.0\boxed{}$, $s_3 = 1.\boxed{}3$, and $s_4 = 3.2\boxed{}$.

12. a) TBD

    b) TBD

13. a)

$$\Delta = \begin{bmatrix} -\boxed{} & -0.82 & \boxed{} & -0.50 \\ -1.15 & -\boxed{} & -\boxed{} & 0.81 \end{bmatrix}$$

    b)

$$\nabla = \begin{bmatrix} -\boxed{} & -1.82 & \boxed{} & -0.74 \\ 0.26 & -\boxed{} & \boxed{} & -1.22 \end{bmatrix}$$

    c) Easy calculation.

14. a) TBD

    b) TBD

    c) TBD

    d) TBD

15. a) TBD
    b) TBD
    c) TBD

16. a) Yes, since we are, in effect, throwing away low-variance parts of the data.
    b) We convert $m \times n$ numbers into $\ell \times n$, so $(\ell n)/(mn) = \ell/m$.
    c) TBD

17. a) The second feature has the highest positive correlation, while the $\boxed{}$ feature has the highest negative correlation.
    b) Let L_i be the component loading vector corresponding to u_i. Then

$$\begin{aligned}
L_1 &= \big(\, \boxed{} \quad 1.269 \quad -\boxed{} \quad -1.089 \quad \boxed{} \quad 0.152 \,\big) \\
L_2 &= \big(\, 0.272 \quad \boxed{} \quad -0.891 \quad \boxed{} \quad -0.153 \quad -\boxed{} \,\big) \\
L_3 &= \big(\, -\boxed{} \quad 0.253 \quad \boxed{} \quad 0.296 \quad \boxed{} \quad -0.610 \,\big)
\end{aligned}$$

    c) Feature 3 is most important, while feature 6 is least important.

Chapter 5

1. Derivation.

2. TBD

3. Combining these equations, we have $n/e\, 2^{-\lambda} = 1$ which implies $\lambda = \log_2(n/e)$. It is straightforward to complete the problem from this point.

4. The Langrangian can be written as

$$L(w_1, w_2, b, \lambda) = \frac{w_1^2 + w_2^2}{2} + \sum_{i=1}^{n} \lambda_i (1 - z_i(w_1 x_i + w_2 y_i + b)).$$

   Compute partial derivatives with respect to w_1, w_2, b, and each λ_i, then substitute and simplify. It is tedious, but it does work out.

5. TBD

6. We have $a_1 x + a_2 y = \alpha$ and $a_1 x + a_2 y = \beta$ and $a_1 x + a_2 y - (\beta + \alpha)/2 = 0$. Let

$$w_1 = \frac{\boxed{}}{\alpha - \beta}, \quad w_2 = \frac{\boxed{}}{\alpha - \beta}, \quad \text{and} \quad b = -\frac{\boxed{}}{\alpha - \beta}$$

   and the desired result follows.

7. TBD

8. a) TBD

   b) TBD

9. a) The scoring function is $f(X) = b + \sum_{i=1}^{n} \lambda_i z_i (X_i \bullet X)$ Expand this expression for $f(X)$ to show that the weight associated with X_i is $\lambda_i z_i(\boxed{} + y_i)$

   b) Bigger is better (or at least "more important", in some sense).

   c) We can reduce dimensionality by getting rid of features that have small weights in a linear SVM. In PCA, the situation (at least with respect to the individual input features) is more complex—explain why this is the case.

10. a) Accuracy is 97%.

    b) Accuracy is $\boxed{}$%.

    c) Accuracy is 92%.

    d) Accuracy is $\boxed{}$%.

11. TBD

12. a) HMM feature weight is 0.1969, SSD feature weight is -□, OGS feature weight is -0.7049, The biggest weight belongs to the □ score, while the smallest weight belongs to the HMM score.

    b) Remove the score with smallest weight and recompute linear SVM, then again remove the score with the smallest weight.

13. Since $f(X) = \sum_{i=1}^{s} \lambda_i z_i K(X_i, X) + b$, we can let $X = X_j$ for any $j \in \{1, 2, \ldots, n\}$ and solve for b. To solve for s we just need to find the vectors for which equality holds, which correspond to the non-zero λ_i.

14. a) TBD

    b) TBD

    c) TBD

15. a) We find that $\lambda = (0.00, \Box, 2.50, 0.00, \Box, 1.25)$ and $b = -\Box$. For these λ and b values, we have $f(X_0) = \Box$ and $z_0 = 1$ $f(X_1) = 2.75$ and $z_1 = \Box$ $f(X_2) = \Box$ and $z_2 = 1$ $f(X_3) = -3.50$ and $z_3 = -\Box$ $f(X_4) = -\Box$ and $z_4 = -1$ $f(X_5) = -1.00$ and $z_5 = -\Box$ where we have listed the z_i from the training data for comparison. The support vectors correspond to the non-zero λ_i.

    b) We find that $\lambda = (0.20, -\Box, 1.79, 0.00, \Box, 0.92)$ and $b = -4.83$.

    c) For part b), for example, the equation of the hyperplane is $x + y = -4.83$.

16. a) Verification required.

    b) Easy calculation.

    c) The first sum is independent of x, so to minimize $\text{MSE}(x)$, we must maximize the sum $\frac{1}{n}\sum(\widetilde{V}_i \bullet x)^2$, which is the mean of the squares of the scalar projections. By the hint, we have $\mu_{x^2} = \mu_x^2 + \sigma_x^2$, which in this particular case tells us that

$$\frac{1}{n}\sum(\widetilde{V}_i \bullet x)^2 = \frac{1}{n}\sum \widetilde{V}_i \bullet x + \sigma_x^2$$

    From part a), the projected means are 0, which implies

$$\sigma_x^2 = \frac{1}{n}\sum(\Box \bullet x)^2$$

    d) This follows from the fact that the covariances in the projection space are all 0.

    e) We find $\partial L(x, \lambda)/\partial\lambda = x \bullet x - 1$ and $\partial L(x, \lambda)/\partial x = 2Cx - 2\lambda x$. The first of these recovers the constraint, $x \bullet x = 1$ and the second yields $Cx = \Box$. Together, these imply that by selecting x to be a unit eigenvector of the covariance matrix C, we will maximize the variance in the projection space.

Chapter 6

1. a) Calculus is your friend.

   b) Similar to a), just a bit more notation to deal with.

2. a) The real question here is, why is -1 the good case, instead of +1?

   b) TBD

3. a) Varies.

   b) Varies, depending on your answer to part a).

   c) Also varies, depending on your answer to part a).

4. Clustering 1, entropy.

$$\begin{aligned}
E_1 &= -\left(\frac{6}{8}\ln\frac{6}{8} + \frac{1}{8}\ln\frac{1}{8} + \frac{1}{8}\ln\frac{1}{8}\right) = 0.736\\
E_2 &= -\left(\boxed{}\ln\boxed{} + \boxed{}\ln\boxed{} + \boxed{}\ln\boxed{}\right) = \boxed{}\\
E_3 &= -\left(\boxed{}\ln\boxed{} + \boxed{}\ln\boxed{} + \boxed{}\ln\boxed{}\right) = \boxed{}\\
E &= \frac{8(0.736) + \boxed{}(\boxed{}) + \boxed{}(\boxed{})}{22} = \boxed{}
\end{aligned}$$

   Clustering 1, purity.

$$\begin{aligned}
U_1 &= 6/8 = 0.750\\
U_2 &= \boxed{} = \boxed{}\\
U_3 &= \boxed{} = \boxed{}\\
U &= \frac{8(0.750) + \boxed{}(\boxed{}) + \boxed{}(\boxed{})}{22} = \boxed{}
\end{aligned}$$

   Clustering 2, entropy.

$$\begin{aligned}
E_1 &= -\left(\frac{3}{8}\ln\frac{3}{8} + \frac{3}{8}\ln\frac{3}{8} + \frac{2}{8}\ln\frac{2}{8}\right) = 1.082\\
E_2 &= -\left(\boxed{}\ln\boxed{} + \boxed{}\ln\boxed{} + \boxed{}\ln\boxed{}\right) = \boxed{}\\
E_3 &= -\left(\boxed{}\ln\boxed{} + \boxed{}\ln\boxed{} + \boxed{}\ln\boxed{}\right) = \boxed{}\\
E &= \frac{8(1.082) + \boxed{}(\boxed{}) + \boxed{}(\boxed{})}{22} = \boxed{}
\end{aligned}$$

Clustering 2, purity.

$$\begin{aligned} U_1 &= 3/8 = 0.375 \\ U_2 &= \boxed{} = \boxed{} \\ U_3 &= \boxed{} = \boxed{} \\ U &= \frac{8(0.375) + \boxed{}(\boxed{}) + \boxed{}(\boxed{})}{22} = \boxed{} \end{aligned}$$

5. a) Programming problem.
   b) Programming problem.

6. a) It's clear that $0 \le p_i \le 1$ for each i, so only need to show that $\sum p_i = 1$.
   b) Programming problem.
   c) Seems to be quite similar to K-means.

7. The results are given by

$$\begin{aligned} p_{1,1} &= \boxed{}, & p_{2,1} &= \boxed{} \\ p_{1,2} &= 0.6047, & p_{2,2} &= \boxed{} \\ p_{1,3} &= \boxed{}, & p_{2,3} &= \boxed{} \\ p_{1,4} &= \boxed{}, & p_{2,4} &= 0.5357 \\ p_{1,5} &= \boxed{}, & p_{2,5} &= \boxed{} \end{aligned}$$

8. a) Programming problem.
   b) Results may vary depending on initial values selected.

9. TBD

10. TBD

11. a) TBD
    b) TBD

12. TBD

13. a) Programming problem.
    b) May depend on initial values selected.
    c) May also depend on initial values selected.
    d) May depend on your powers of observation.

14. a) If there really are 2 distributions, then the mean and (especially) the variance for each will be likely be much smaller than the overall mean and variance.

    b) Follows from part a).

15. One advantage of K-means is simplicity, while an advantage of EM is that it allows for more general "shapes" of cluster, depending on the type of distributions.

16. a) Suppose that we initially select a reachable non-core point, which is then marked as visited. Complete the explanation.

    b) Suppose that X is not a core point, and $d(X,Y) < \varepsilon$, $d(X,Z) < \varepsilon$, and $d(Y,Z) > \varepsilon$. Complete the explanation.

    c) Pretty pictures.

Chapter 7

1. Pretty picture.

2. a) TBD

   b) TBD

   c) TBD

3. See the errata for a new-and-improved version of this problem.

   a) TBD

   b) TBD

4. TBD

5. TBD

6. a) To relate $\widehat{S}_B$ to S_B, we note that

   $$(\widehat{\mu}_x - \widehat{\mu}_y)^2 = (w^T\mu_x - w^T\mu_y)^2 = \big(w^T(\mu_x - \mu_y)\big)\big(w^T(\mu_x - \mu_y)\big).$$

   Then use the fact that the transpose of a matrix product is the product of the transposes in reverse order.[1]

   b) TBD

7. a) This depends on the fact that S_B and S_W are symmetric matrices.

   b) Use the fact that S is symmetric.

   c) TBD

8. a) TBD

   b) TBD

9. TBD

10. a) TBD

    b) We have $w_1 = 1/(\sqrt{1+m^2})$ and $w_2 = \boxed{}$.

11. TBD

12. a) TBD

    b) TBD

[1] Try saying "the transpose of a matrix product is the product of the transposes in reverse order" fast, five times.

c) In addition to swapping elements, also test the cases where the selected elements are moved to the other cluster (one at a time). Then there will be 3 cases to test for each pair selected.

13. TBD

14. Good question.

15. Completely analogous calculation.

16. a) PCA is based on the length of orthogonal bisectors, while least squares is based on □.

    b) Linear regression is simpler, but PCA is probably more "accurate" in some sense.

17. Easy.

Chapter 8

1. a) In each fold, we reserve M/n match cases for testing, so this is the number of match scores per fold. We do this n times, the number of match scores is $n \cdot M/n = M$.

   b) We score all N nomatch samples in each fold, for nN total nomatch scores.

   c) A larger n will do a better job of smoothing out any bias in the data, while a smaller values requires fewer models be constructed. In practice 5-fold is often sufficient.

2. a) TBD

   b) TBD

3. a) A similar example can be found in the PowerPoint slides.

   b) Again, see the slides for a worked example.

4. a) Easy.

   b) Easy.

   c) Easy.

5. Same process as given in the example.

6. a) Straightforward calculation.

   b) Straightforward calculation.

7. The first few points for the ROC curve are given by

	FPR	TPR
1	1.0	1.0
2	0.8	1.0
3	0.8	0.8
4	0.6	0.8
⋮	⋮	⋮

8. The first few points for the PR curve are given by

	Recall	Precision
1	1.0	0.5
2	1.0	0.55
3	0.8	0.5
4	0.8	0.57
⋮	⋮	⋮

9. a) We have $\mathrm{TPR} = \mathrm{TP}\,/(\mathrm{TP} + \mathrm{FN})$ and $\mathrm{FPR} = \mathrm{FP}\,/(\mathrm{FP} + \mathrm{TN})$. Now, if we duplicate each nomatch score n times, the TPR is unaffacted, since it only involved match scores, and the FPR is now calculated as $n\,\mathrm{FP}\,/(n\,\mathrm{FP} + n\,\mathrm{TN}) = \mathrm{FP}\,/(\mathrm{FP} + \mathrm{TN})$ and we see that the FPR is also unchanged. For PR curves, recall is the same as TPR, so it is not affected by changes in the nomatch cases. But, precision is computed as $\mathrm{TP}\,/(\mathrm{TP} + \mathrm{FP})$ and if we duplicate each nomatch score n times, the precision becomes $\mathrm{TP}\,/(\mathrm{TP} + n\,\mathrm{FP}) \neq \mathrm{TP}\,/(\mathrm{TP} + \mathrm{FP})$. Thus, from the perspective of ROC analysis, upsampling is of no use in analyzing an imbalance, but PR curves might be useful.

   b) See part a).

10. a) Similar to problem worked in the book.

    b) Similar to problem worked in the book.

    c) An advantage of changing the threshold is that it can reduce the FPR. An advantage of a secondary test is that we can reduce the number of misclassifications.

11. a) See http://ftp.cs.wisc.edu/machine-learning/shavlik-group/davis.icml06.pdf.

    b) See a).

    c) See a).

12. Depends on example selected.