6.891 Fall 1999: Problem Set Grading Guide

We are trying out a cross-grading system where each student will grade another student's problem. An answer key will be provided (see below following this guide). The goal of this is to reduce the grading burden on any one person and to allow you to see possibly alternative solutions to the problem. This can help you to better understand the concepts we are trying to convey to the class. We realize this may be the first time you are asked to do this, and we expect questions, so feel free to let us know if you have any comments. You will assign one of four marks to each part of each problem according to the guidelines below. Thank you for helping us out with this system.

For each part of each problem assign one of the following grades:

- $\surd+$: got correct answer and properly justified solution (i.e., "nailed the problem and answer completely correct").
- $\surd$: showed significant work and insight but may not have completely vsolved the part, or the answer was slightly off (i.e., "got pretty much everything right except made a small error or left off some justification").
- $\surd-$: showed significant work and insight but had a some gaps in justification or answer (i.e., "had the right idea, but got lost somewhere or had a major error").
- 0 : did not show significant work or insight, or answer was not applicable to the question.

Note that there may be more than one way of obtaining a reasonable solution. Also, the exact numbers for estimations such as the parameters estimated by the EM algorithm will vary based on initial conditions and how many iterations were run. So we should be reasonable when grading those parts. The important thing is to see that work was put into the solution and that the logic of the solution makes sense.

6.891 Fall 1999: Problem Set #2 Solutions

Problem 1: Review of Bayesian Decision Theory

A:

We want to minimize the risk $R(\alpha_i|x)$ by deciding some ω_i:

$$\begin{aligned} R(\alpha_i|x) &= \sum_{j=1}^{c} \lambda(\alpha_i|\omega_j) P(\omega_j|x) && (1) \\ &= \sum_{j\neq i}^{c} \lambda_s P(\omega_j|x) && (2) \\ &= \lambda_s[1 - P(\omega_i|x)] && (3) \end{aligned}$$

So to minimize the risk we should choose ω_i such that $P(\omega_i|x)$ is maximum. We also need the condition that:

$$\begin{aligned} \lambda_s[1 - P(\omega_i|x)] &\leq \lambda_r && (4) \\ P(\omega_i|x) &\geq 1 - \frac{\lambda_r)}{\lambda_s} && (5) \end{aligned}$$

since otherwise we could choose to reject. If $\lambda_r = 0$, then we should always reject since this minimizes the risk. If $\lambda_r > \lambda_s$, then we should never reject since it's cheaper to misclassify.

B:

(a)-(d) From first principles we can see that the density of the sum of independent random variables is the convolution of the densities of the random variables:

$$\begin{aligned} p(x_3) &= \int_{-\infty}^{\infty} p_{x_1}(x_1) p_{x_2}(x_3 - x_1) dx_1 && (6) \\ &= p_{x_1}(x_1) * p_{x_2}(x_2) && (7) \end{aligned}$$

One way to do this is to work through the integral and find that the "mean of the sum" is the "sum of the means" and that the "variance of the sum" is the "sum of the variances". In fact, one can show this using linearity of expectations, but this doesn't directly show that the distribution is Gaussian. So we will use the trick of characteristic functions for Gaussians (which is like moving into the Fourier domain of the pdf so that convolution becomes multiplication). Treating x_i as vectors, and sphering (whitening) x_i so that $y_i = \sum_i^{-1/2}(x_i - \mu_i)$ and simplifying linear algebra terms:

$$
\begin{aligned}
\Phi_{y_i}(jv) &= E[e^{jv^T y_i}] && (8)\\
&= \int_{-\infty}^{\infty} \frac{1}{(2\pi)^{d/2}|\sum_{yi}|^{1/2}} e^{jv^T y_i} e^{-1/2 y_i^T \sum_{yi}^{-1} y_i} && (9)\\
&= \frac{1}{(2\pi)^{d/2}|\sum_i|^{1/2}} (2\pi)^{d/2} |\sum_i|^{1/2} e^{jv^T \mu_i} e^{-1/2 v^T \sum_i v} && (10)\\
&= e^{jv^T \mu_i - 1/2 v^T \sum_i v} && (11)\\
& && (12)
\end{aligned}
$$

then:

$$
\begin{aligned}
\Phi_{x_3}(jv) &= \Phi_{x_1}(jv)\Phi_{x_2}(jv) && (13)\\
&= e^{jv^T(\mu_1+\mu_2) - 1/2 v^T (\sum_1 + \sum_2) v} && (14)\\
& && (15)
\end{aligned}
$$

. We can get the unique inverse transform of this and see that it will be a Gaussian and by inspection, μ_3=$\mu_1 + \mu_2$ and $\sum_3 = \sum_1 + \sum_2$. This is the general case so you can show it for the 1-D case as well using the same idea.

Problem 2: Bayesian Classification

A:

(a) Since we have iid samples, we can write the joint probability as a product and simplify:

$$
\begin{aligned}
P(D|\theta) &= \prod_{j=1}^{n} P(x_n|\theta) && (16)\\
&= \prod_{j=1}^{n}\prod_{i=1}^{d} \theta_i^{x_{ji}} (1-\theta_i)^{1-x_{ji}} && (17)\\
&= \prod_{i=1}^{d}\prod_{j=1}^{n} \theta_i^{x_{ji}} (1-\theta_i)^{1-x_{ji}} && (18)\\
&= \prod_{i=1}^{d} \theta_i^{s_i} (1-\theta_i)^{n-s_i} && (19)\\
& && (20)
\end{aligned}
$$

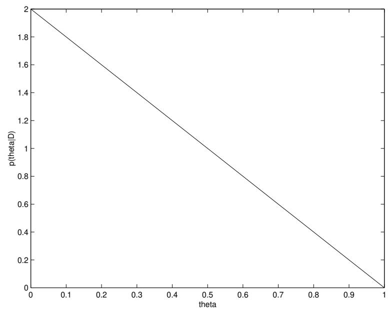

Figure 1: Plot for $p(\theta|D)$ for $s_1 = 0$ in (2c).

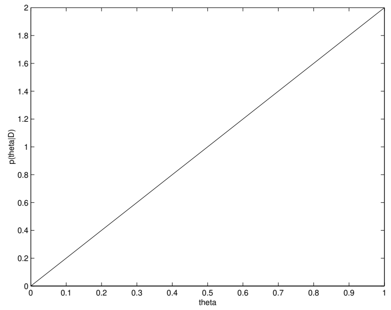

Figure 2: Plot for $p(\theta|D)$ for $s_1 = 1$ in (2c).

where $s_i = \sum_{j=1}^{n} x_{ji}$.

(b) Assuming a uniform prior on θ and using Bayes' rule and the integral identity given:

$$\begin{aligned}
p(\theta|D) &= \frac{P(D|\theta)p(\theta)}{P(D)} && (21)\\
&= \frac{\frac{1}{T}\prod_{i=1}^{d}\theta_i^{s_i}(1-\theta_i)^{n-s_i}}{\frac{1}{T}\int_0^1\prod_{i=1}^{d}\theta_i^{s_i}(1-\theta_i)^{n-s_i}d\theta} && (22)\\
&= \frac{\prod_{i=1}^{d}\theta_i^{s_i}(1-\theta_i)^{n-s_i}}{\prod_{i=1}^{d}\frac{s_i!(n-s_i)!}{(s_i+n-s_i+1)!}} && (23)\\
&= \prod_{i=1}^{d}\frac{(n+1)!}{s_i!(n-s_i)!}\theta_i^{s_i}(1-\theta_i)^{n-s_i} && (24)
\end{aligned}$$

(c) Substituting into part (b), for $s_1 = 0$, $p(\theta|D) = (1-\theta)$ and for $s_1 = 1$, $p(\theta|D) = 2\theta$.

(d) For this part we need to do a few tricks with the integration, use the integral identity in part (b), and note that x_i will either be 0 or 1:

$$\begin{aligned}
p(x|\theta) &= \int_0^1 P(x|\theta)p(\theta|D)d\theta && (25)\\
&= \int_0^1 \cdots \int_0^1 \prod_{i=1}^{d} \theta_i^{x_i}(1-\theta_i)^{1-x_i} \prod_{j=1} \frac{(n+1)!}{s_j!(n-s_j)!}\theta_j^{s_j}(1-\theta_j)^{n-s_j} d\theta && (26)\\
&= \prod_{i=1}^{d} \frac{(n+1)!}{s_i!(n-s_i)!} \int_0^1 \theta_i^{x_i+s_i}(1-\theta_i)^{n+1-x_i-s_i} d\theta_i && (27)\\
&= \prod_{i=1}^{d} \frac{(n+1)!}{s_i!(n-s_i)!} \frac{(x+s_i)!(n+1-s_i-x_i)!}{(n+2)!} && (28)\\
&= \prod_{i=1}^{d} \frac{(s_i+1)^{x_i}(n-s_i+1)^{1-x_i}}{n+2} && (29)\\
&= \prod_{i=1}^{d} (\frac{(s_i+1)}{(n+2)})^{x_i}(\frac{n-s_i+1}{n+2})^{1-x_i} && (30)\\
&= \prod_{i=1}^{d} (\frac{(s_i+1)}{(n+2)})^{x_i}(1-\frac{s_i+1}{n+2})^{1-x_i} && (31)\\
& && (32)
\end{aligned}$$

(e) Recall that the likelihood function is:

$$\begin{aligned}
L(\theta) &= P(D|\theta) && (33)\\
&= \prod_{i=1}^{n} P(x_i|\theta) && (34)
\end{aligned}$$

The Bayesian estimate is:

$$\begin{aligned}
p(\theta|D) &= \frac{P(D|\theta)p(\theta)}{P(D)} && (35)\\
&= \frac{p(\theta)}{P(D)} \prod_{i=1}^{n} p(x_i|\theta) && (36)
\end{aligned}$$

So:

$$p(\theta|D) \propto L(\theta)P(\theta) \tag{37}$$

since $P(D)$ is independent of θ. By definition, the maximum likelihood (ML) function peaks at the value $\hat{\theta}$. In the limit as we get more observations, the Bayesian estimate will peak around $\hat{\theta}$ so the integral will be dominated by the region around $\hat{\theta}$:

$$\begin{aligned} p(x|D) &= \int P(x|\theta)p(\theta|D)d\theta && (38) \\ &\simeq p(x|\hat{\theta}) \int p(\theta|D)d\theta && (39) \\ &= p(x|\hat{\theta}) && (40) \end{aligned}$$

So in the limit as the number of observations n increases, the effective Bayesian estimate approaches the ML estimate.

Problem 3: Density Estimation

Parameters of part C:

```
ps1a 2 Gaussian:
priors:
    0.0893    0.9107

means:
    5.5895    0.6958

variances:
   13.4282    9.9552

ps1a 3 Gaussian:
priors:
    0.5362    0.3163    0.1475

means:
    1.9689   -2.8151    6.5603

variances:
    0.9398    3.7391    5.9443
```

```
ps1a 10 Gaussian:
priors:
  Columns 1 through 7

    0.0263    0.2819    0.0179    0.0156    0.2718    0.0880    0.0141

  Columns 8 through 10

    0.0091    0.0267    0.2485

means:
  Columns 1 through 7

    7.5719    1.9360   -5.4957    9.2463    1.5628    3.6146    7.9215

  Columns 8 through 10

   11.3614    6.5582   -3.1837

variances:
  Columns 1 through 7

    0.1774    0.5729    1.5545    0.5349    1.9464    2.0829    2.0911

  Columns 8 through 10

    0.7200    0.8011    2.1723

ps1b 2 Gaussian:
priors:
    0.7498    0.2502

means:
    5.2376   12.0564

variances:
    7.8913   18.9337

ps1b 3 Gaussian:
priors:
    0.0362    0.7770    0.1869

means:
   15.5000    5.2587   12.2936

variances:
```

```
   30.1922    7.8292   10.7064

ps1b 10 Gaussian:
priors:
  Columns 1 through 7

    0.1124    0.2183    0.0040    0.0658    0.1247    0.0951    0.0934

  Columns 8 through 10

    0.0568    0.1361    0.0935

means:
  Columns 1 through 7

    2.4129   10.9640   25.5777    1.1710    3.3335    5.6947    5.5924

  Columns 8 through 10

   17.1304    8.2622    5.5950

variances:
  Columns 1 through 7

    0.7539    3.9926    2.1855    0.3229    0.7806    1.0639    1.1973

  Columns 8 through 10

    3.1851    0.9384    1.1937
```

See the plots for the estimated densities.

Problem 4: Gaussian Classifier

You didn't have to do much here except look at `demo.m`and run the test code since we already did coding to get you guys started.

Problem 5: Spherically-Symmetric Gaussians

A:

The maximum estimate for the mean with a spherical covariances is the same as with full covariances. See the derivation below for the ML estimates of the mean and covariance.

$$
\begin{aligned}
\ln \mathcal{L} &= \ln \prod_n \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(x^n-\mu)^T \Sigma^{-1} (x^n-\mu)} \\
&= \ln \prod_n \frac{1}{(2\pi)^{\frac{d}{2}} (\sigma^{2d})^{\frac{1}{2}}} e^{-\frac{(x^n-\mu)^T (x^n-\mu)}{2\sigma^2}} \\
&= \sum_n -\frac{d}{2} \ln 2\pi - \frac{2d}{2} \ln \sigma - \frac{(x^n-\mu)^T (x^n-\mu)}{2\sigma^2} \qquad (41) \\
&\Rightarrow \\
\frac{\partial \mathcal{L}}{\partial \sigma} &= 0 \\
\sum_n -\frac{d}{\sigma} + \frac{(x^n-\mu)^T (x^n-\mu)}{\sigma^3} &= 0 \\
N d \sigma^2 &= \sum_n (x^n-\mu)^T (x^n-\mu) \\
\sigma^2 &= \frac{1}{Nd} \sum_n (x^n-\mu)^T (x^n-\mu) \qquad (42) \\
\frac{\partial \mathcal{L}}{\partial \mu} &= 0 \\
\sum_n \frac{(x^n-\mu)}{2\sigma^2} &= 0 \\
\sum_n (x^n-\mu) &= 0 \\
\mu &= \frac{1}{N} \sum_n x^n \qquad (43)
\end{aligned}
$$

B:

See the code.

C:

See the plots.

Problem 6: Parzen Classifier

A:

See the matlab code for the classifier.

B:

The ML estimate for σ for a Parzan density estimator is 0. When σ goes to zero, the density estimate becomes infinite at each of the training samples and zero elsewhere. If we then evaluate the product of all of the density values at the training samples (the likelihood), we get the product of infinite probabilities. This is larger than any finite likelihood that will result from a positive σ and is thus the ML estimate. Unfortunately, this estimator does not generalize well.

C:

One way to choose σ is the start out with an initial value and use a technique called cross-validation to search for a σ with the minimum average error. We cross-validate by splitting the training data into N disjoint subsets of equal size s_i. The classifer is trained on N training sets $t_i = D - s_i$ where we train on all the data except one of the subsets (the hold-out set). The bias of the classifer is the average over the different

D:

See the plots for the decision boundaries.

Problem 7: Nearest Neighbor Classifier

A:

See the plots for the decision boundaries.

B:

The density estimator which is consistent with a nearest neighbor estimator is a Parzan density estimator with spherical covariances as above where σ has been set to 0. This can be seen by considering the decision function that results from a Parzan density estimator. Namely, if we let x_j^i be the i'th example of the j'th class, then the decision rule amounts to:

$$\arg\max_j \sum_i e^{\frac{-1}{2\sigma^2}\left(x - x_j^i\right)^2} \tag{44}$$

where I have dropped the normalization in front of the Gaussian since it is the same across all examples and all classes (well, the normalization due to the prior might not be, but in the limit, such constant terms will drop out as seen below). Furthermore, since I am taking the j which maximizes the expression, such a j value does not change if I divide the entire expression by the same quantity. I carefully pick this constant to be the double sum in the expression below. Thus, the problem becomes

$$\arg\max_j \frac{\sum_i e^{\frac{-1}{2\sigma^2}\left(x-x_j^i\right)^2}}{\sum_k \sum_l e^{\frac{-1}{2\sigma^2}\left(x-x_l^k\right)^2}} = \arg\max_j \sum_i \frac{e^{\frac{-1}{2\sigma^2}\left(x-x_j^i\right)^2}}{\sum_k \sum_l e^{\frac{-1}{2\sigma^2}\left(x-x_l^k\right)^2}} \tag{45}$$

On the right hand side, we notice that each term in the sum is actually part of a softmin function. Namely, each of these is the result of taking the softmin of the squared distance from x to a given data point. By lowering σ, we are making the softmin more and more "hard." In the limit, as σ approaches 0, the softmin becomes a true min which evaluates to 1 at the true min and 0 for all other values. Thus, the sum over i will be equal to 1 for the class that contains the closest point and 0 for all others. This rule will then clearly choose the class associated with the closest point.

The Code and Plots:

The code is included separately. The plots are shown in the remaining figures.

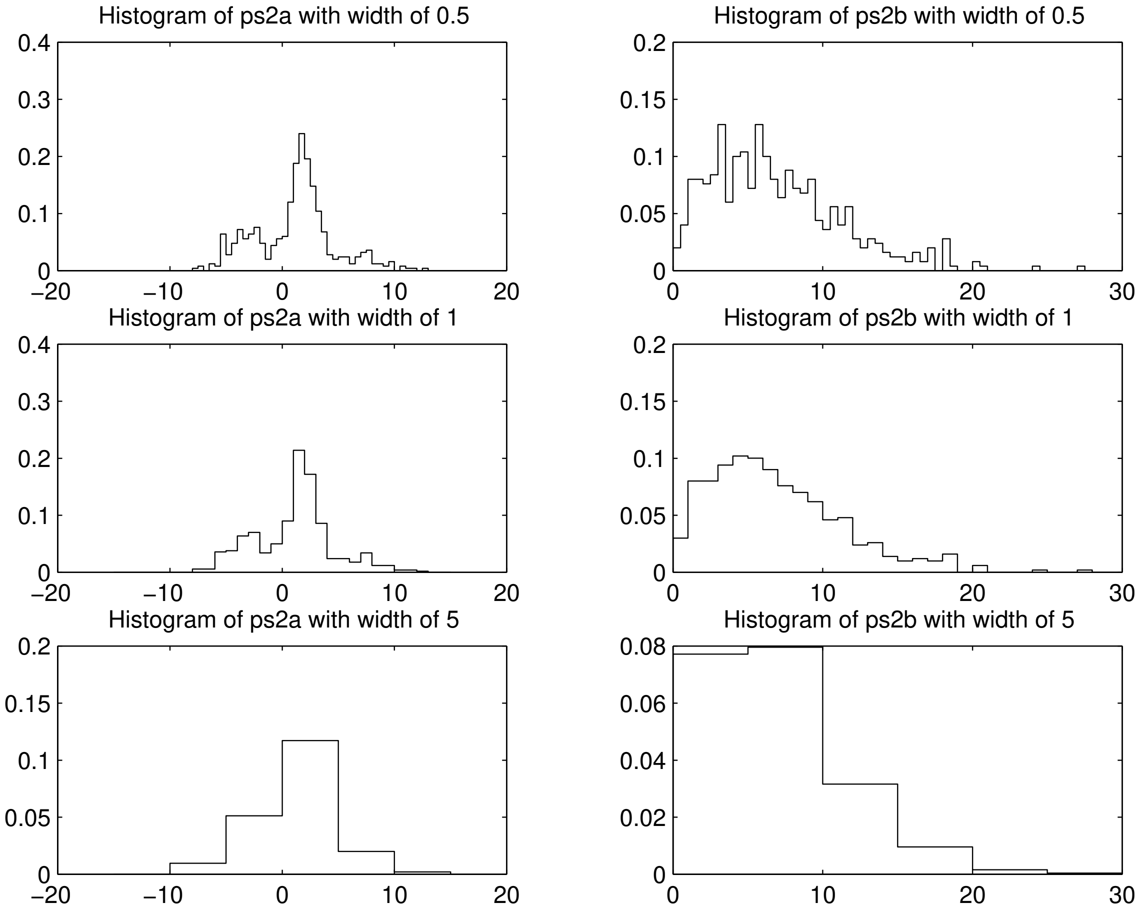

Figure 3: Output from Histogram Estimate (3A)

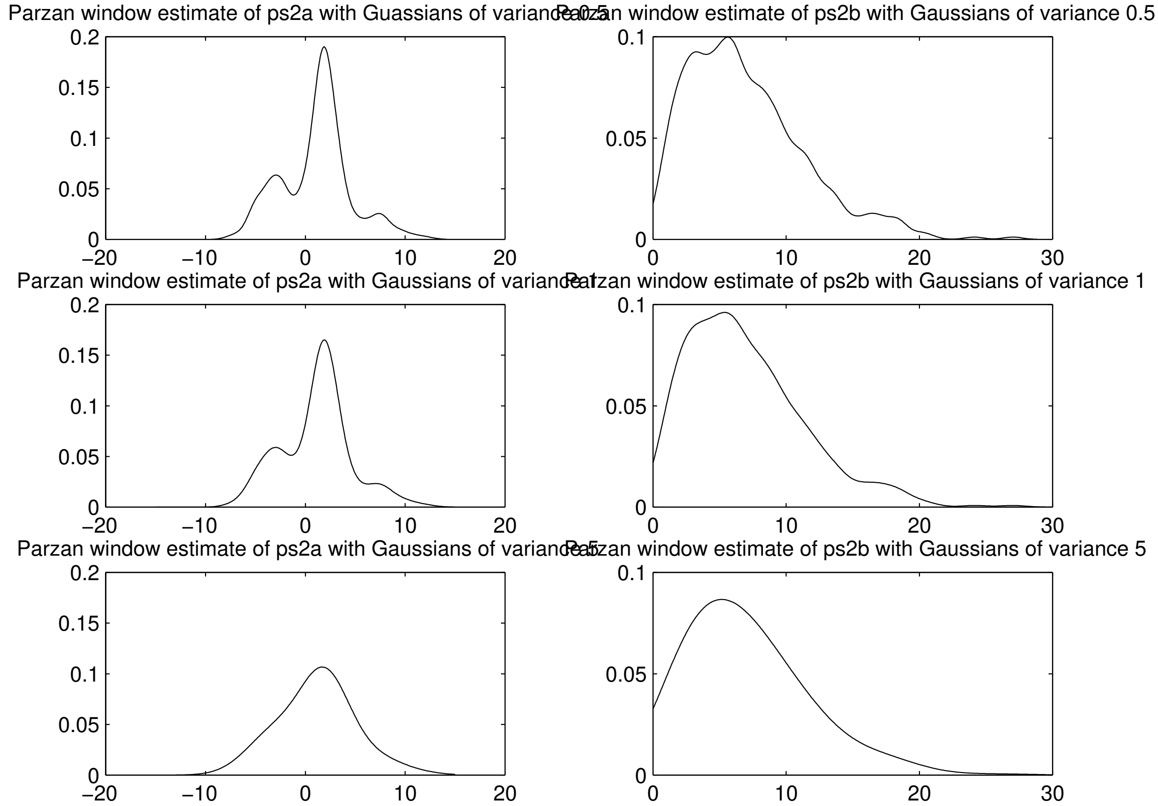

Figure 4: Output from Parzen Window Estimate with Gaussians (3B-Gaussians)

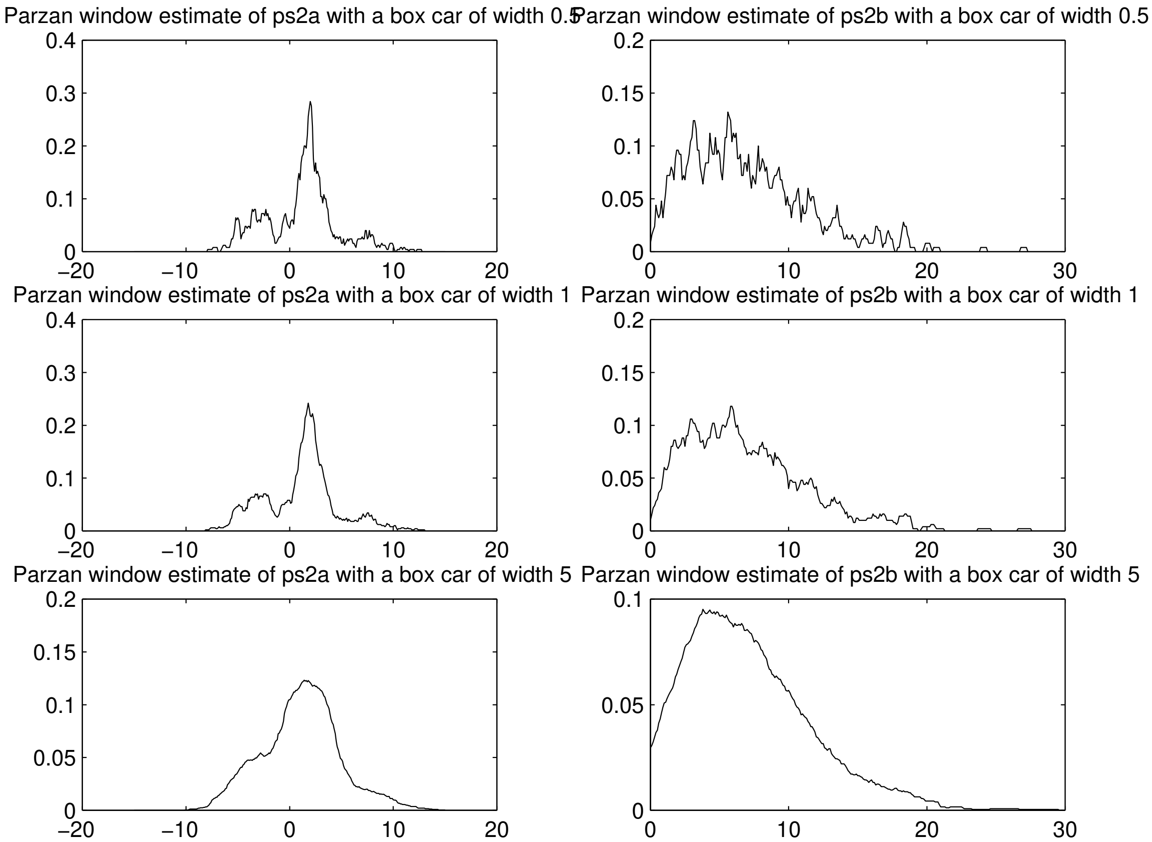

Figure 5: Output from Parzan Window Estimate with box car functions (3B-Boxcar)

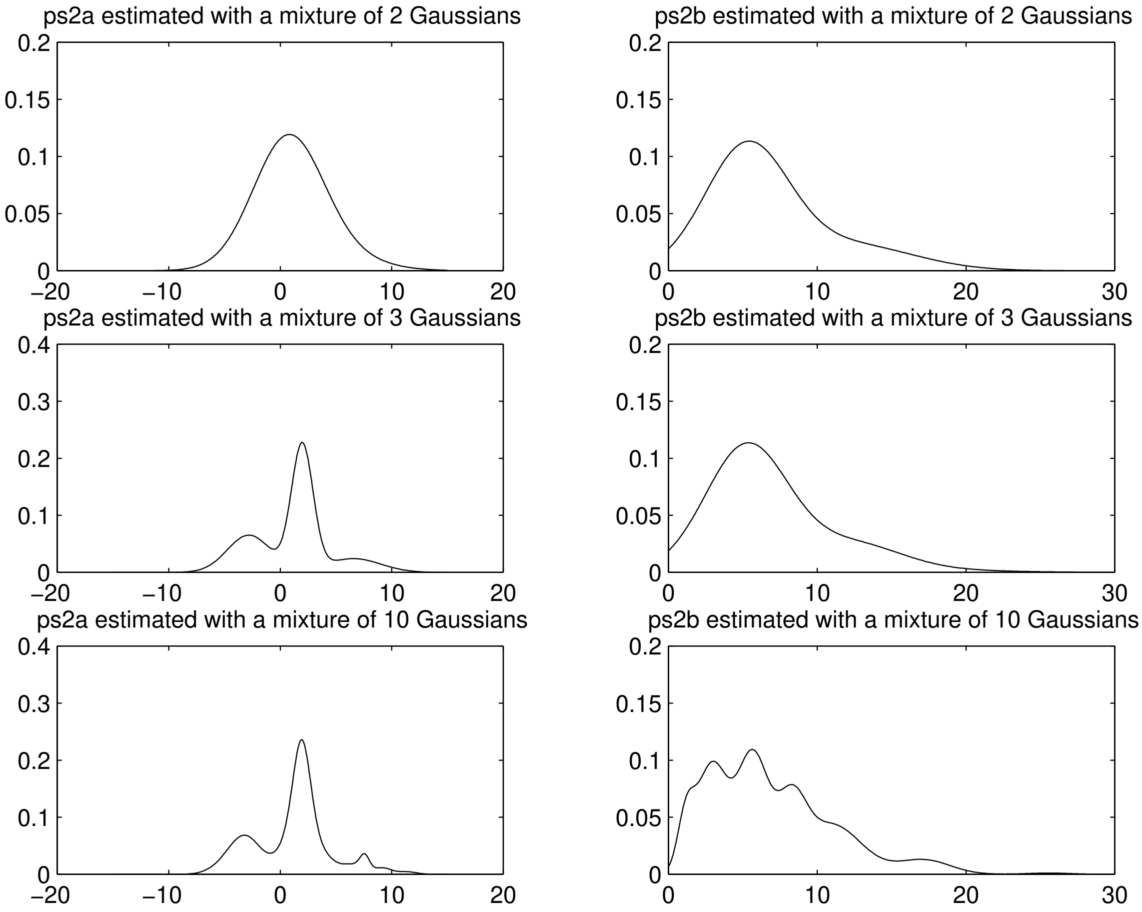

Figure 6: Output from mixture of Gaussians Estimate (3C)

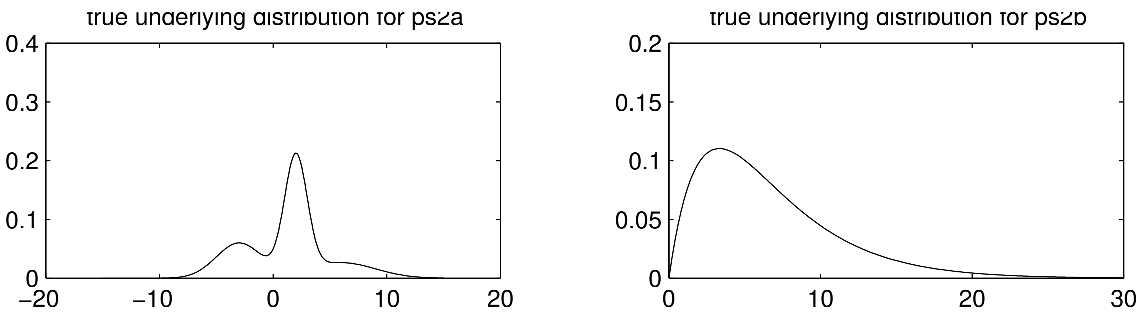

Figure 7: True distributions

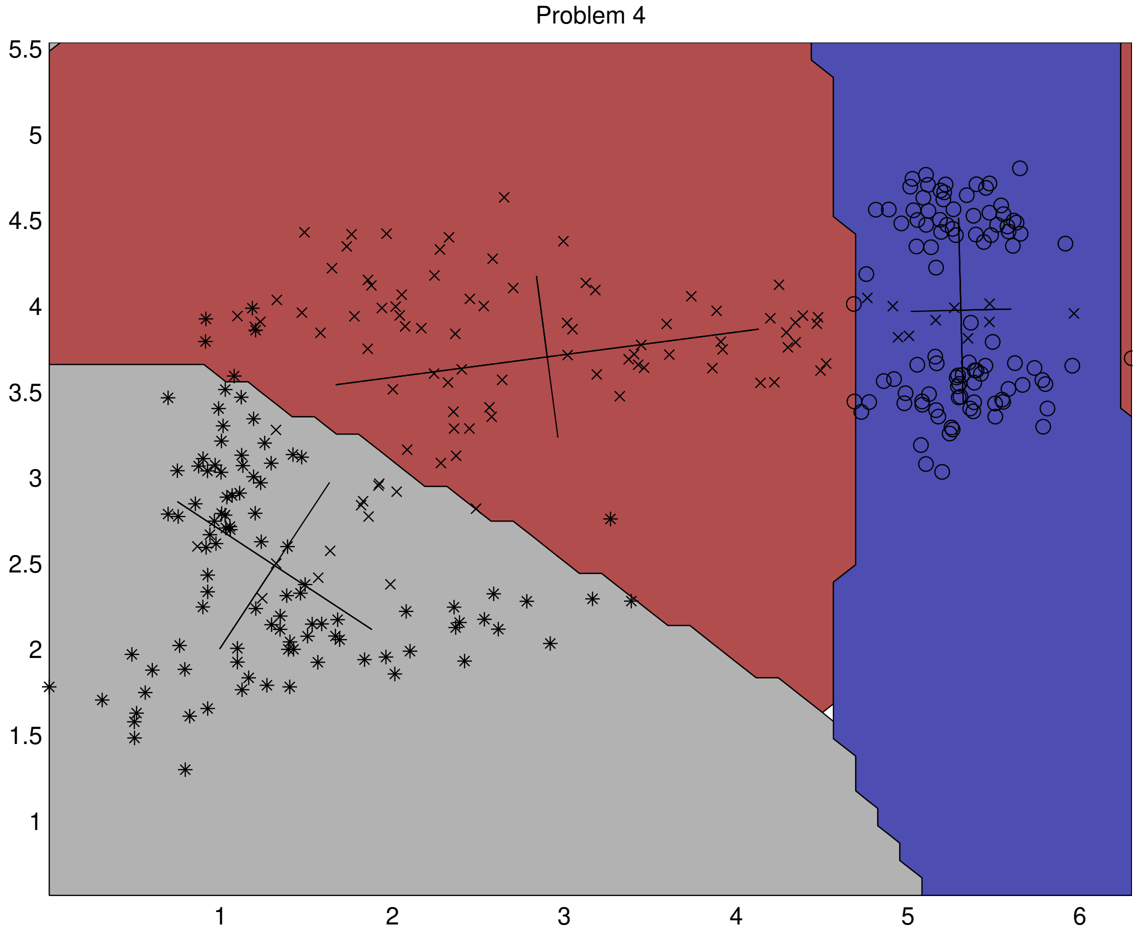

Figure 8: General Gaussian Classifier

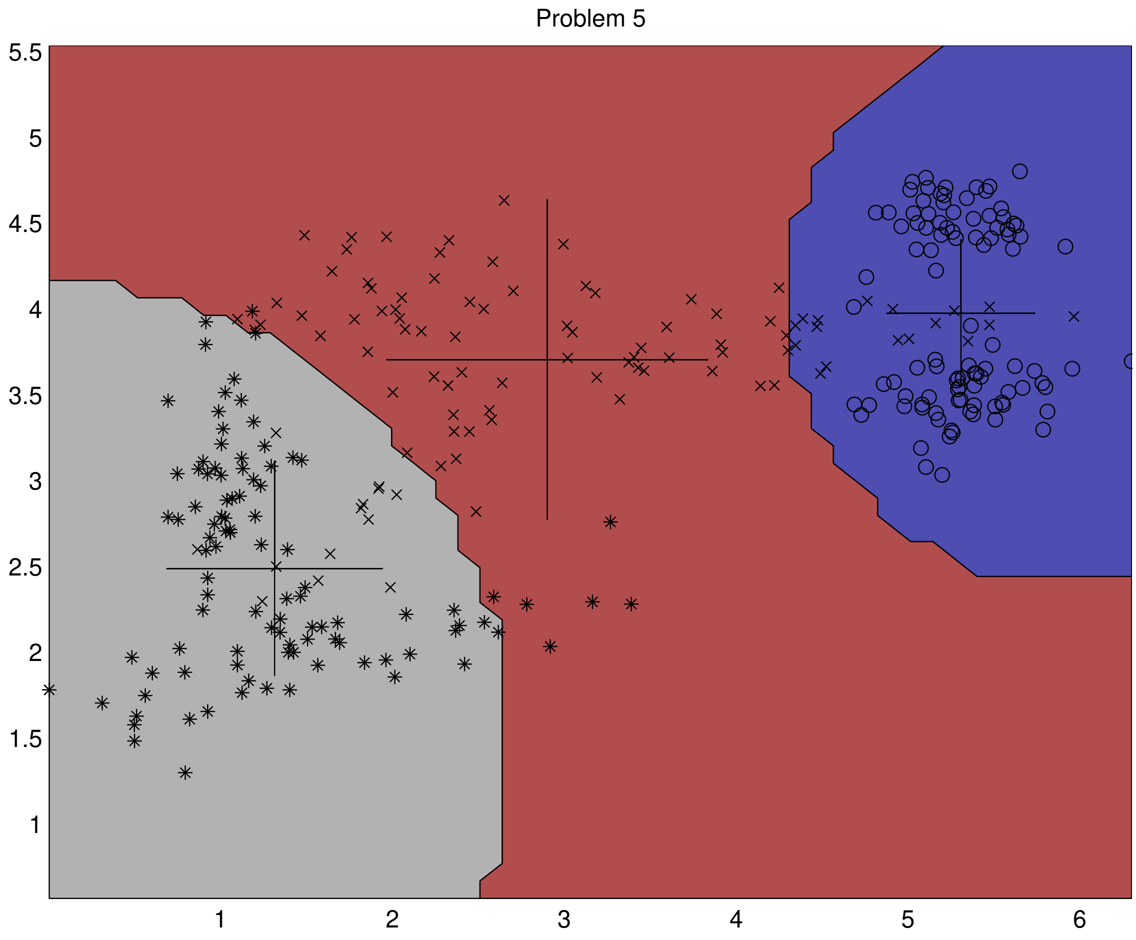

Figure 9: Spherical Gaussian Classifier

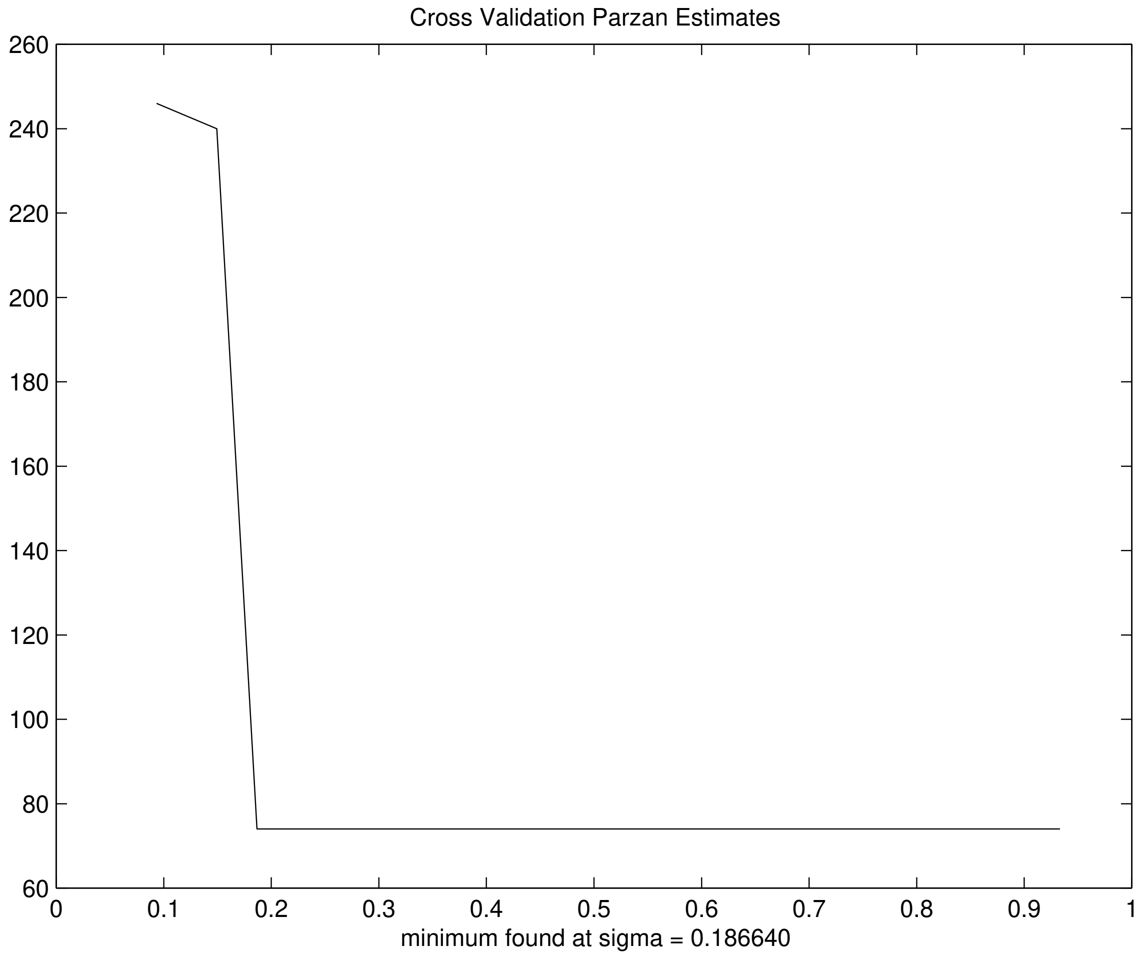

Figure 10: Parzen Cross-Validation Plot

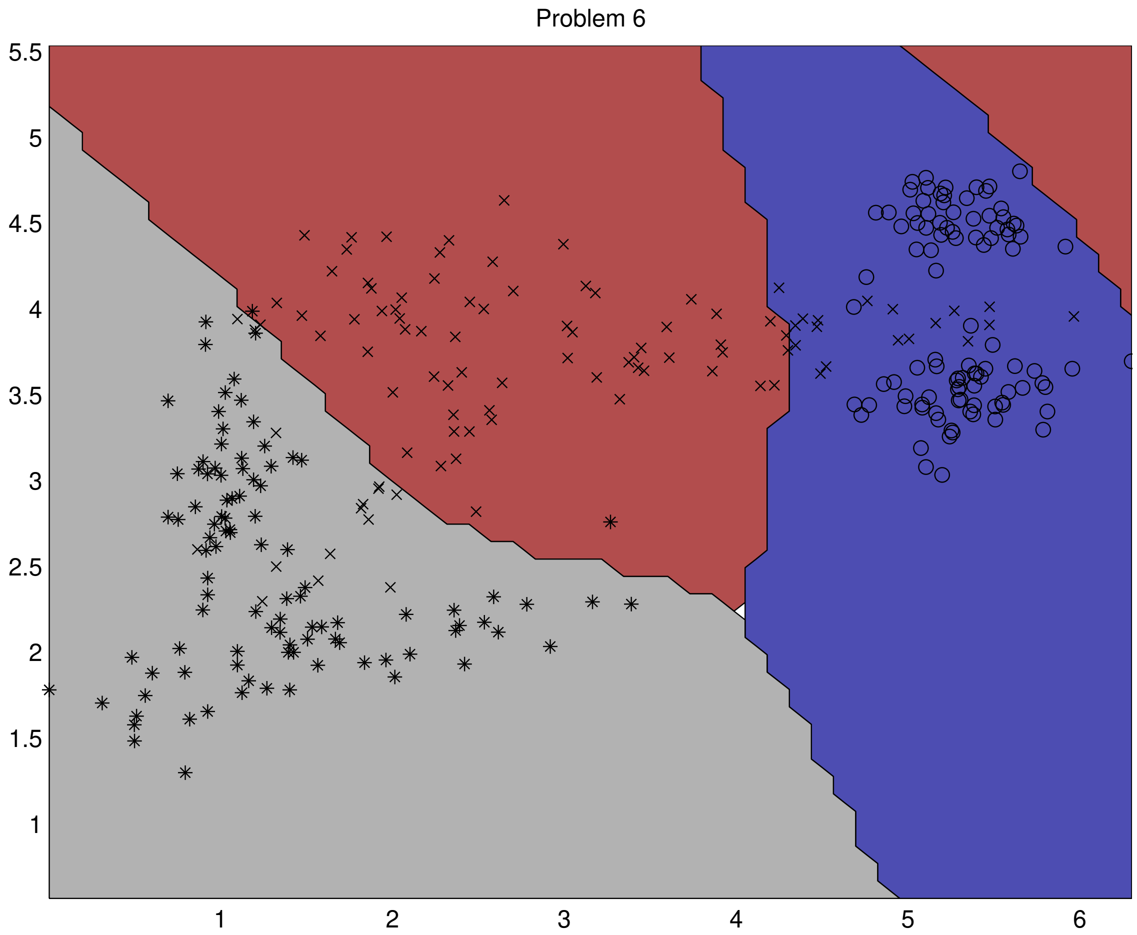

Figure 11: Parzen Classifier

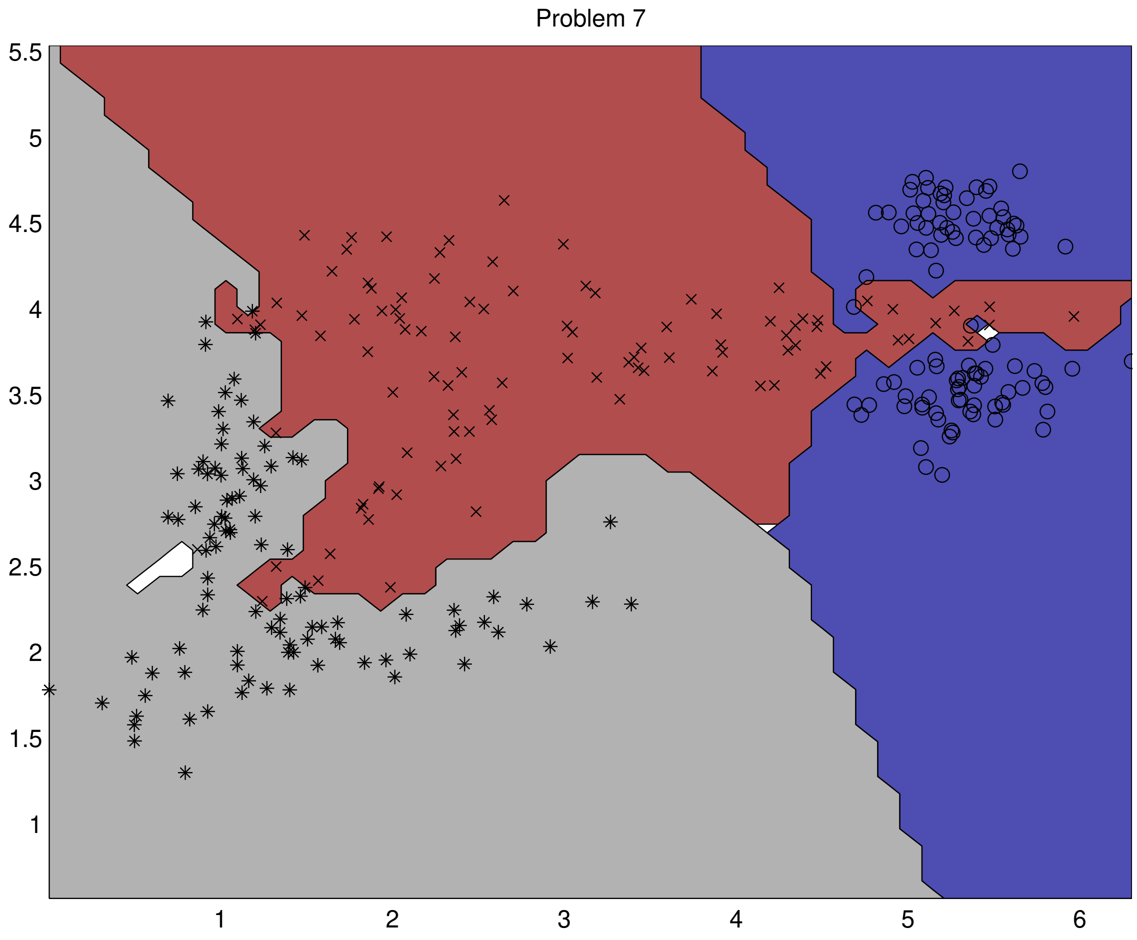

Figure 12: Nearest Neighbor Classifier