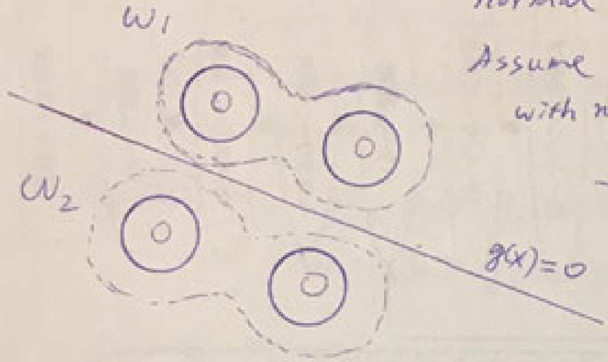
CSE 569 Homework #4 Solutions

Question 1 Solution.

(a) Look at the following sketch for bimodal normal densities for ω_1 & ω_2.



Assume the densities are symmetric with respect to the line $g(x)=0$. Then $g(x)=0$ will also give the optimal decision boundary.

(b) An example is given below.

(The contours illustrate the equiprobability contours of the distributions.)

Note that the distributions are unimodal, but otherwise the "shape" (as illustrated by the contours of equiprobability) can be arbitrary, and thus any straight-line decision boundary would give poor performance.

(c) Since there is no constraint on μ_i and σ_i, we could let $\mu_1 = \mu_2$, and in this case, the optimal decision cannot be a line (it will in general be a circle). This is illustrated below:

ω_1: the shaded region.

ω_2: the rest.

Note: the above example is for some $P(\omega_i)$ & σ_i. In general, depending on $P(\omega_i)$ & σ_i, the decision circle will change (and whether the inner disk is for ω_1 or for ω_2 will also change).

Question 2 Solution.

(Note: the problem description is missing a subscript i for w^t in

$$g_i(x) = w_i^t x + w_{i0}$$

But this should be clear since we have the correct version in the lecture slides.)

By definition,
$$x_1 \in R_i \iff g_i(x_1) \geq g_k(x_1),\ \forall k$$
$$x_2 \in R_i \iff g_i(x_2) \geq g_k(x_2),\ \forall k \quad \Bigg\}\ \star$$

Consider any $Y = \lambda x_1 + (1-\lambda) x_2$, for $0 \leq \lambda \leq 1$

we have
$$\begin{aligned} g_k(Y) &= w_k^t Y + w_{k0} = w_k^t(\lambda x_1 + (1-\lambda)x_2) + w_{k0} \\ &= \lambda[w_k^t x_1 + w_{k0}] + (1-\lambda)[w_k^t x_2 + w_{k0}] \\ &\leq \lambda[w_i^t x_1 + w_{k0}] + (1-\lambda)[w_i^t x_2 + w_{k0}] \quad \text{because of } \star \\ &= w_i^t[\lambda x_1 + (1-\lambda)x_2] + w_{i0} \\ &= g_i(Y) \end{aligned}$$

This says $g_k(Y) \leq g_i(Y)$, $\forall k$,

and thus $\Rightarrow Y \in R_i$.

This is true for any $x_1 \in R_i$, $x_2 \in R_i$, $0 \leq \lambda \leq 1$, and any i,

Hence all R_i is convex.

(For showing a region R_i is convex, we only need to show $\forall x_1, x_2 \in R_i$, $0 \leq \lambda \leq 1$, $\lambda x_1 + (1-\lambda)x_2$ is still in R_i.)

Question 3 Solution.

Proof by contradiction.

Let's assume we have two sets of vectors, $S_1 = \{X_1, X_2, \ldots, X_N\}$ and $S_2 = \{Y_1, Y_2, \cdots, Y_M\}$, which are linearly separable AND whose convex hulls intersect. We will show this will lead to a contradiction.

<1> S_1 & S_2 linearly separable $\Rightarrow$ there exists a linear discriminant function $g(X) = W^t X + w_0$ such that

$$\begin{cases} g(X) > 0, \ \forall X \in S_1 \\ \text{AND} \\ g(X) < 0, \ \forall X \in S_2 \end{cases}$$

<2> "The convex hulls of S_1 & S_2 intersect" $\Rightarrow$ there exists a point Z, which is in both the convex hull of S_1 and the convex hull of S_2, i.e., we can write

$$Z = \sum_{i=1}^{N} \alpha_i X_i \quad \text{AND} \quad Z = \sum_{j=1}^{M} \beta_j Y_j$$

for some non-negative α_i, β_j, with $\sum_{i=1}^{N} \alpha_i = 1$, $\sum_{j=1}^{M} \beta_j = 1$.

Thus we can have

$$\begin{aligned} g(Z) &= W^t Z + w_0 \\ &= W^t \sum_{i=1}^{N} \alpha_i X_i + w_0 \\ &= \sum_{i=1}^{N} \alpha_i \underbrace{(W^t X_i + w_0)}_{>0} \\ &> 0 \end{aligned} \qquad \text{AND} \qquad \begin{aligned} g(Z) &= W^t Z + w_0 \\ &= W^t \sum_{j=1}^{M} \beta_j Y_j + w_0 \\ &= \sum \beta_j \underbrace{(W^t Y_j + w_0)}_{<0} \\ &< 0 \end{aligned}$$

(In the last step, we also used the fact $\alpha_i \geq 0$, $\beta_j \geq 0$, $\sum \alpha_i = 1$, $\sum \beta_j = 1$).

This is a contradiction.

Question 4 Solution.

There are different but correct ways of writing down the pseudocode for the new algorithm. For example, we may look for worst-classified samples for each class separately and then use them to update the solution vector, *or* we may look for such samples considering both classes jointly (thus, for example, if one class contains misclassified samples while the other class is already error-free, the correction will be done only by few samples from the first class). The following sample code is one for the latter case (considering both classes jointly).

Also, we assume that the samples have been "normalized" according to the protocol we learned in the lectures, so that for any sample $\mathbf{y}_i$, if $\mathbf{a}^t\mathbf{y}_i > 0$, it is correctly classified by **a**.

The new algorithm can be written as follows.

New Algorithm: Updating based on the current worst sample(s)

```
initialize  a //  randomly chosen
Repeat
   worst_sample =  999999999999; // A large positive number
   for k = 1:Nsample  // check for every sample
        if  a^t y_k  < worst_sample    // A full solution should store all such samples & their
                 worst_sample  =  a^t y_k     //  corresponding k, if they are equally "worst".
                 worst_k = k;
        endif
   endfor
   a = a + y_worst_k    // A full solution should update with all samples found above.
until the following stop criterion is met:
        {All patterns properly classified (all a^t y_k >0) &
        the worst-classified samples for both classes are away from the decision boundary by
        at least a margin b (a^t y_worst_k.>= b)} or {A preset maximum # of epochs is reached.)
 return a
```

Note: In the above algorithm, if we do keep a list of (equally) worst samples in each iteration, we may eventually find the support vectors (if linearly separable).

Question 5 Solution.

(1) Plotting the samples in the 2-d space, we will see that it is easy to draw a line to separate the samples from the two classes. So they are linearly separable.

(2) The big data matrix Y is (remember: we need to "augment" the data by adding 1 and add "-" to the samples from the second class in forming this big Y)

```
Y = [ 1 1 1
      1 2 2
      1 2 0
     -1 -0 -0
     -1 -1 -0
     -1 -0 -1 ]
```

With this, we can compute the pseudoinverse $Y^+ = (Y^tY)^{-1}Y^t$, which is something like,

$$\begin{bmatrix} -0.1216 & 0.2297 & -0.0405 & 0.4730 & 0.2568 & 0.3378 \\ 0.0270 & -0.1622 & -0.3243 & -0.2162 & 0.0541 & -0.2973 \\ -0.1081 & -0.3514 & 0.2973 & -0.1351 & -0.2162 & 0.1892 \end{bmatrix}$$

With Y^+, we can compute the corresponding solution vector **a** under different **b**.

For $\mathbf{b}_1 = (1, 1, 1, 1, 1, 1)^t$, we have $\mathbf{a}_1 = Y^+\mathbf{b}_1 = (-1.1351 \quad 0.9189 \quad 0.3243)^t$, which corresponds to the following decision boundary:

$$g_1(x_1,x_2) = 0.9189x_1 + 0.3243x_2 - 1.1351 = 0$$

For $\mathbf{b}_2 = (1, 1, 1, 1, 1, 2)^t$, we have $\mathbf{a}_2 = Y^+\mathbf{b}_2 = (-1.4730 \quad 1.2162 \quad 0.1351)^t$, which corresponds to the following decision boundary:

$$g_2(x_1,x_2) = 1.2162x_1 + 0.1351x_2 - 1.4730 = 0$$

(3) For the decision boundary given by $g_1(x_1,x_2)$, we can easily verify $g_1(x_1,x_2) > 0$ for samples in class 1, and $g_1(x_1,x_2) < 0$ for samples in class 2, and thus $g_1(x_1,x_2)$ classifies all samples correctly.

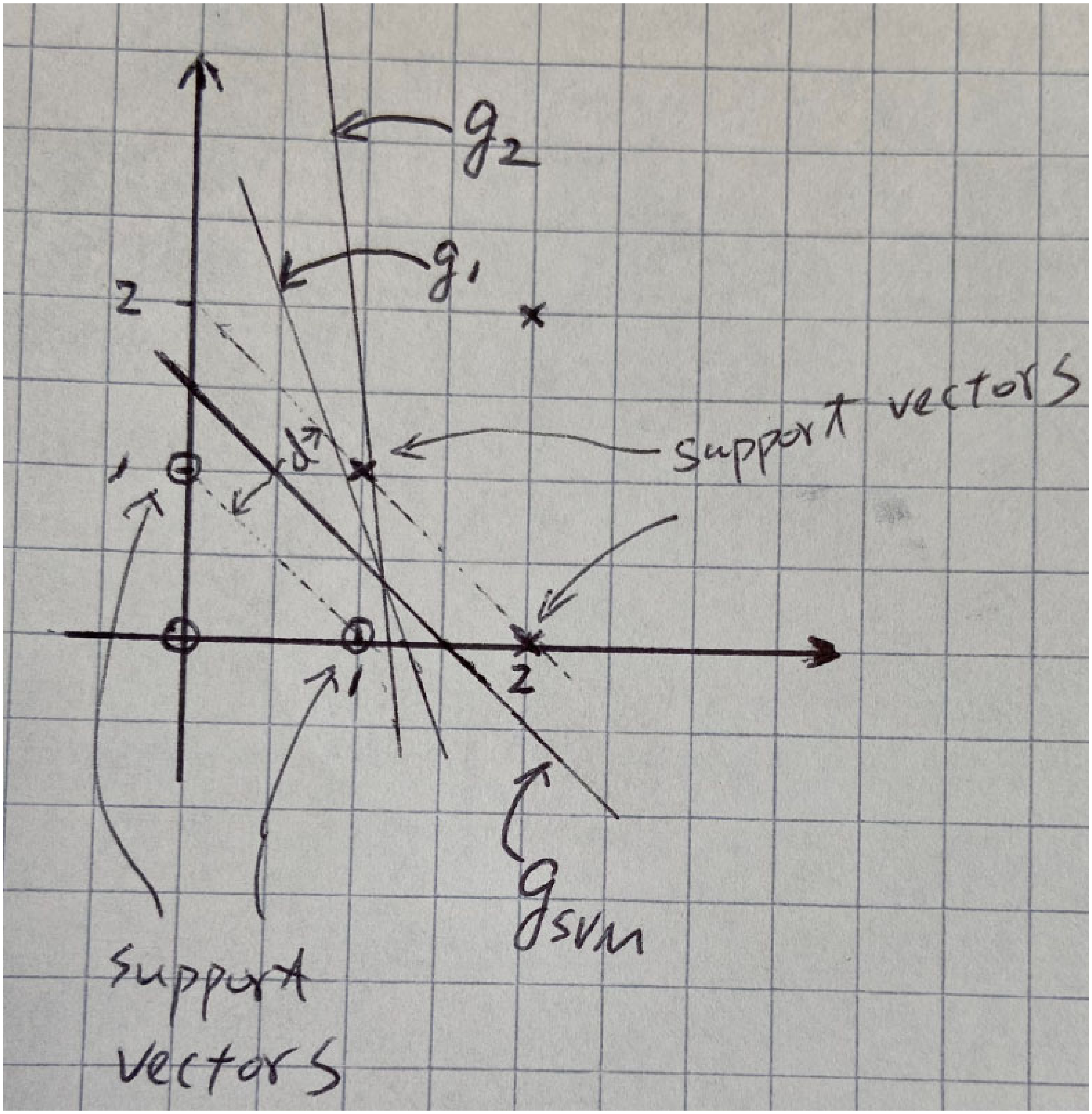
For the decision boundary given by $g_2(x_1,x_2)$, we can find that it will misclassify $(1,1)^t$ from class 1, since $g_2(1,1) < 0$. But otherwise, $g_2(x_1,x_2)$ can correctly classify other samples.

Question 6 Solution.

(1) See the scanned image on the next page. Samples are illustrated in X and O. g_{SVM} gives us the max margin linear classifier.

(2) From the illustration on the next page, we can easily figure out the margin d (using the definition from Slides 37-38 in Notes 07) is sqrt(2)/2 since this happens to be half of length from (0,0) to (1,1) in the figure. (In general, you can get the equations for the H_1 and H_2 planes and the compute the distance between them; See Slide 38, Notes 07.)

(3) This is given in the scanned image: g_1, g_2, g_{SVM}.



g_2
g_1
2
support vectors
$\vec{d}$
1
1
2
g_{svm}
support vectors