

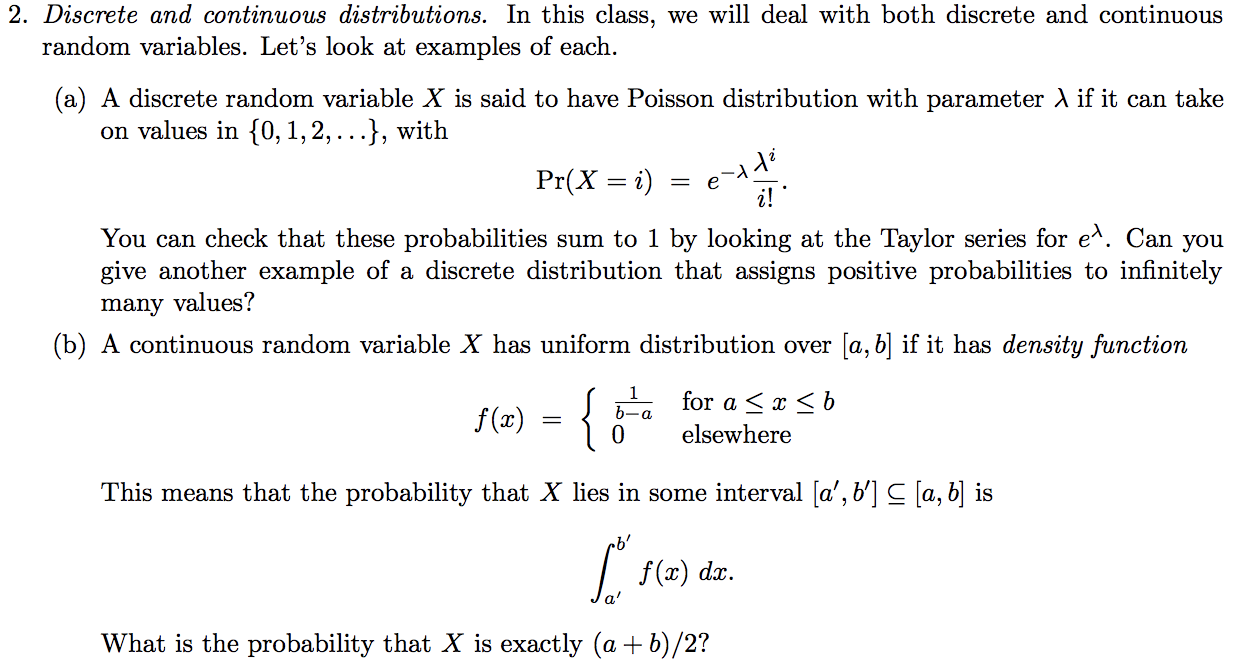
1. (a)

error rate =

(b)

The classifier should return label A.

error rate =



1. (a)

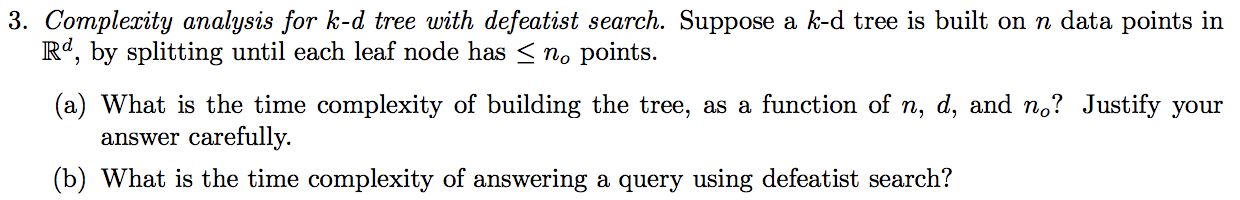
another example of a discrete distribution that assigns positive probabilities to infinitely many values is as the following:

on values in {1, 2, 3, …},

we can check that these probabilities sum to 1:

(b)

the probability that X is exactly (a + b)/2 is



1. (a)

Since each leaf node has less than points, we need at least leaf nodes. If the tree is approximately balanced, then the depth of this tree would be . At each level of the tree, we need to find the median of each group of points so as to split them into more small clusters. One way of finding median is quick selection which is O(n) time complexity where n is the size of inputs. Hence if we use quick selection here, at each level, we would need O(n) time to find groups’ medians.

To sum up, the time complexity would be .

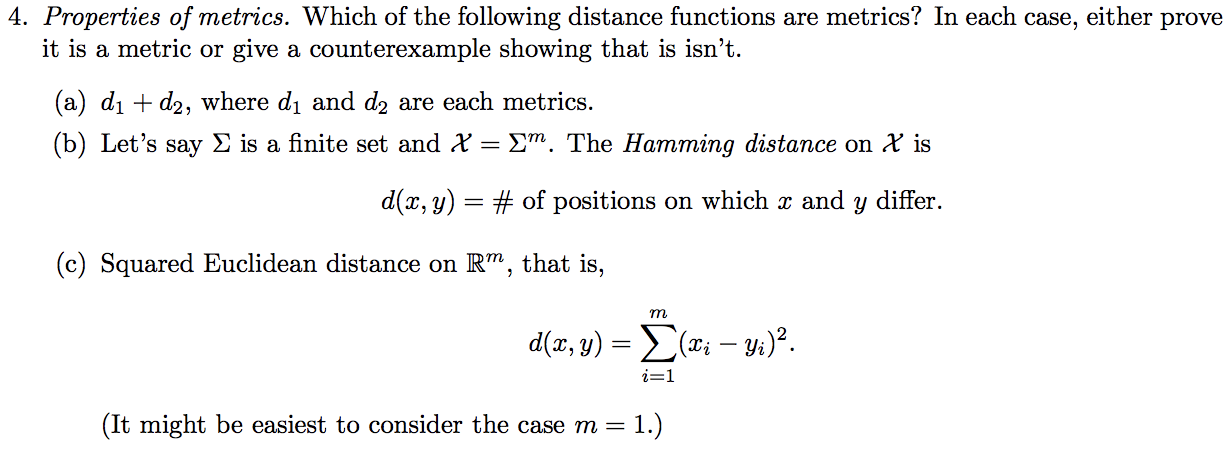
(b)

The time complexity of answering a query using defeatist search is .

The first step is to find a leaf node where the query node resides. For this step, we would go deep down and search for the node. Therefore, the time complexity is the depth of the tree, .

The second step is to find the nearest point. Among these at most points, we need to calculate the distance, and generally speaking, the complexity of distance is proportional to the dimension of the node, d. Hence for this step, the time complexity would be .

To sum up, the time complexity would be .



1. (a)

is a matric.

Let’s denote .

since are metrics, we have , therefore .

1. iff .

First, prove

Since and , we know that .

Owing to the property of , we know .

Second, prove

Owing to the property of , when , we have , therefore

This is because .

1. .

This is because

(b)

The hamming distance defined in the problem is a matric.

By definition, is the number of positions, hence this value cannot be negative.

1. iff .

First, prove

Since then we know that there is 0 position on which x and y differ, meaning that x and y are identical, namely .

Second, prove

Since x and y are equal, then there is no position on which they differ, namely

According to definition, we can know that if x and y differ on some position, we can also say that y and x differ on that position. That’s to say, the definition is symmetric. Hence,

1. .

Let us define and where , and .

We define a function

therefore, we can derive the following

We first prove

There are 5 different cases in terms of

in this case, , hence .

in this case, , hence .

in this case, hence .

in this case, , hence .

in this case, , hence .

In all cases, we have , hence this inequality stands for all cases.

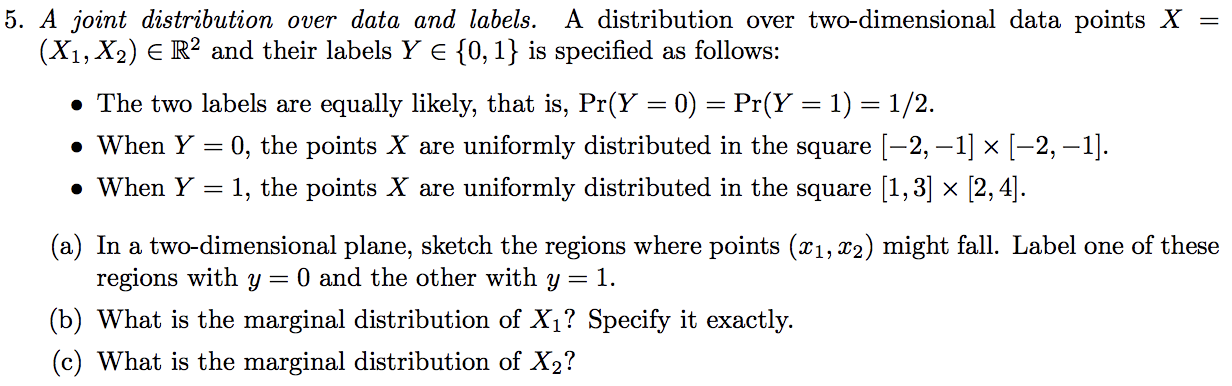
Therefore,

namely,

(c)

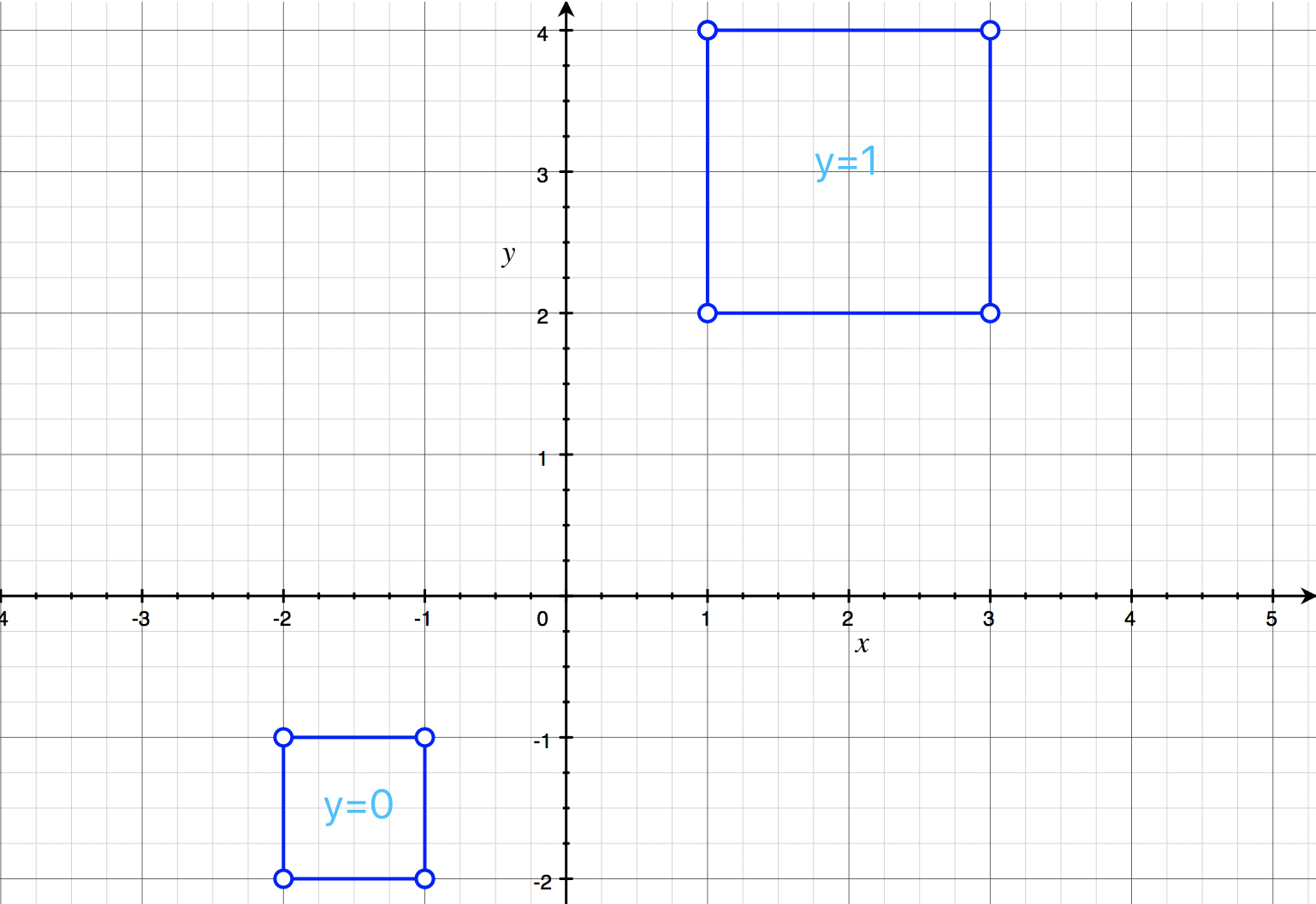
The squared Euclidean distance defined above is not a matric.

Counterexample: In this case, we have .



1. (a)

the figure is as follows:



(b)

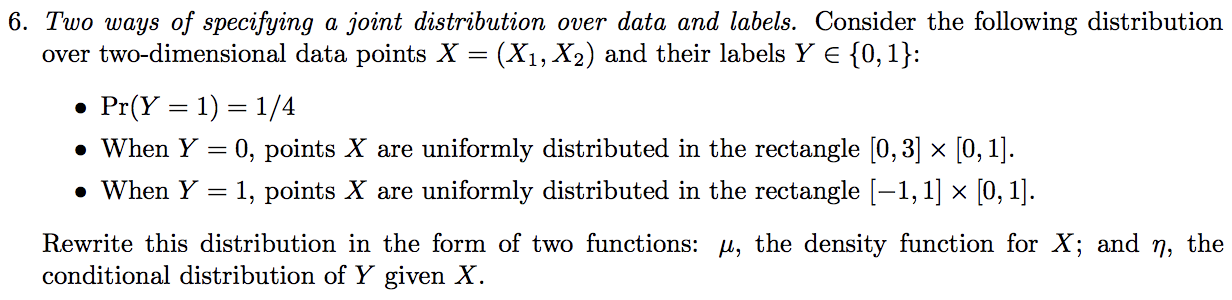
when ,

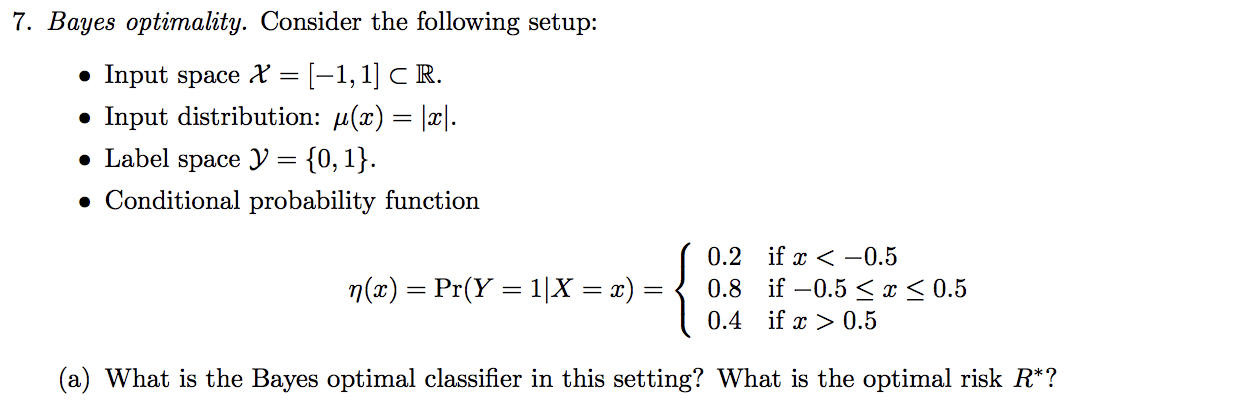
when .

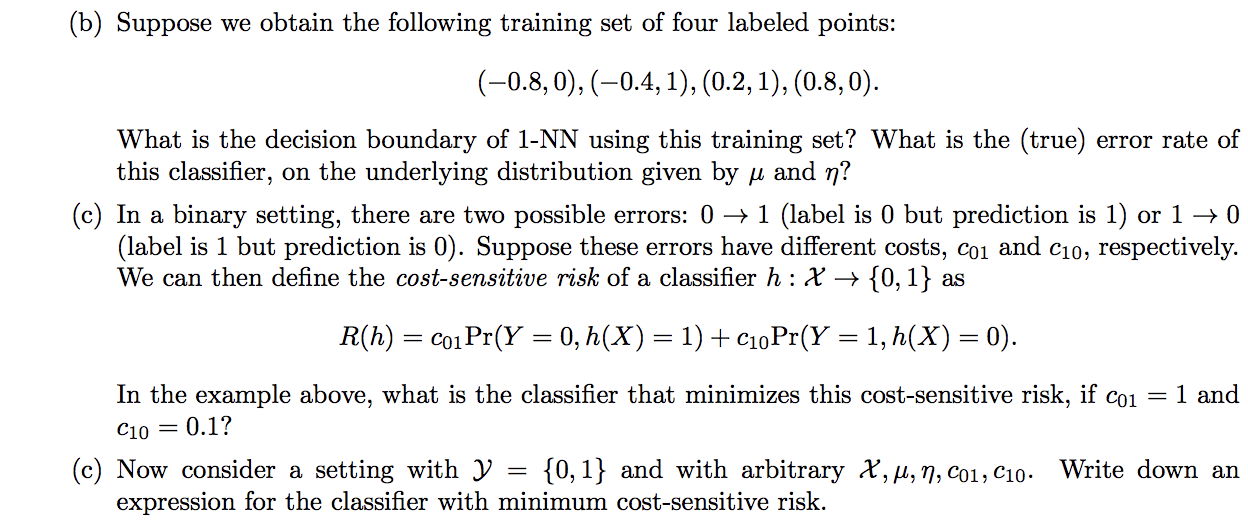
Therefore,

(c)

Similar to (b), we can calculate the marginal distribution of :







1. (a)

(b)

the decision boundaries are and .

Hence, this classifier would perform as follows:

error rate

(c)

for , if we predict 1, the error rate would be if we predict 0, the error rate would be Hence, for this part, we should predict 0.

For , if we predict 1, the error rate would be if we predict 0, the error rate would be

Hence, for this part, we also should predict 0.

For , if we predict 1, the error rate would be if we predict 0, the error rate would be

Hence, for this part, we also should predict 0.

To sum up, the classifier would always predict 0, namely

(d)

For a certain x, let’s check the cost of predicting 1 and 0 respectively.

If we predict 1, it would contribute to the error rate; if we predict 0, it would contribute to the error rate.

Therefore, the classifier would perform in the following way: