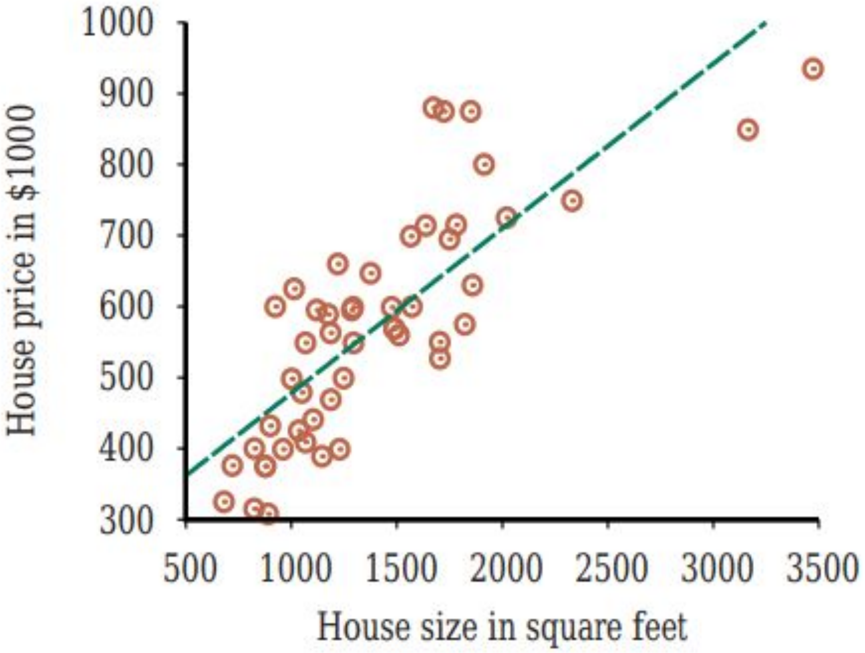
CS 471: Introduction to AI

Module 6 Part III: Machine Learning

Nearest Neighbor Classification

Parametric Models

- Linear regression uses the training data to estimate a fixed set of parameters w.
- A learning model that summarizes data with a set of parameters is called a parametric model.



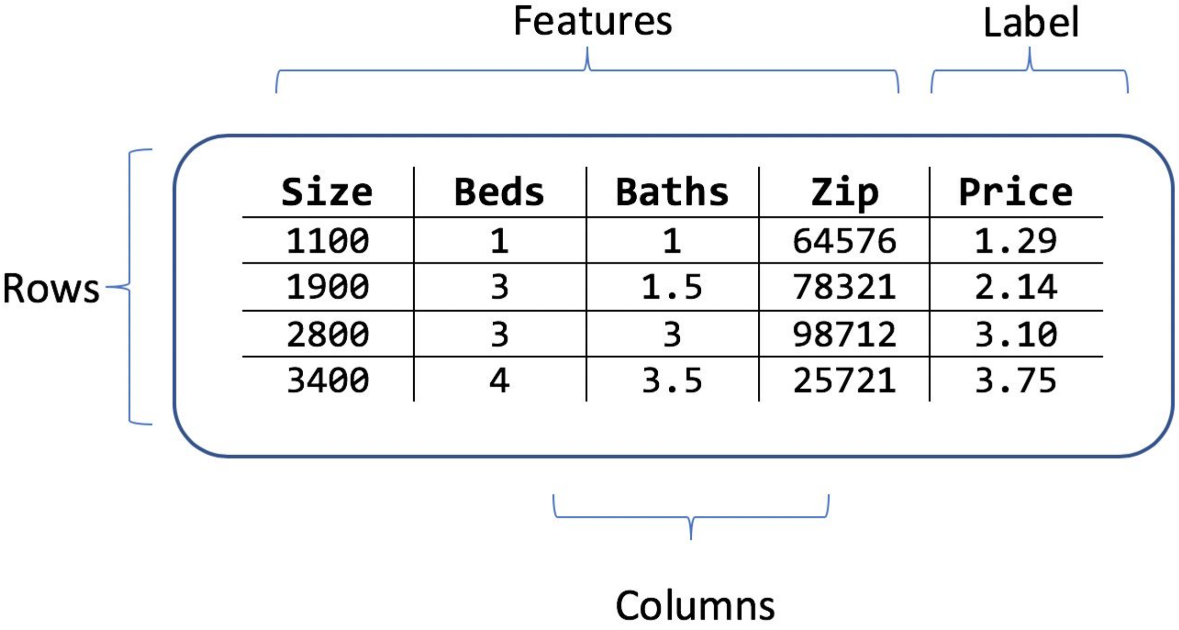
$y = 246 + 0.232x$ ($y = w_0 + w_1 x$)

Nonparametric Models

- Cannot be characterized by a set of parameters.
- Simplest learning method is table lookup: take all the training examples, put them in a lookup table.

  When given a new x, sees if x is in the table; if it is, return the corresponding y.

  Does not generalize well: when x is not in the table we have no information about a plausible value.



Features

Label

Rows

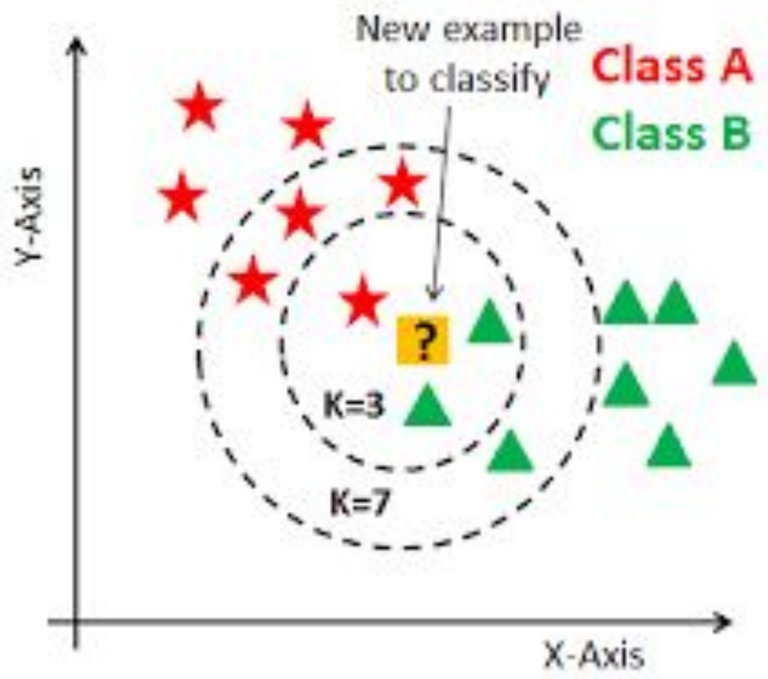
Size	Beds	Baths	Zip	Price
1100	1	1	64576	1.29
1900	3	1.5	78321	2.14
2800	3	3	98712	3.10
3400	4	3.5	25721	3.75

Columns

Nearest-neighbor Models

We can improve on table lookup with a slight variation:

Given a query x_q, instead of finding an example that is equal to x_q, find the k examples that are nearest to x_q. This is called k-nearest-neighbors lookup.

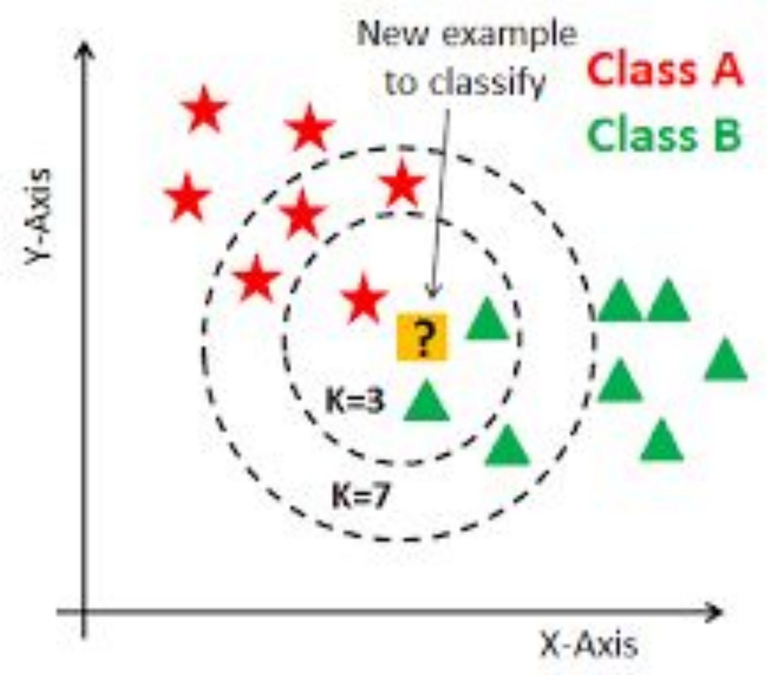


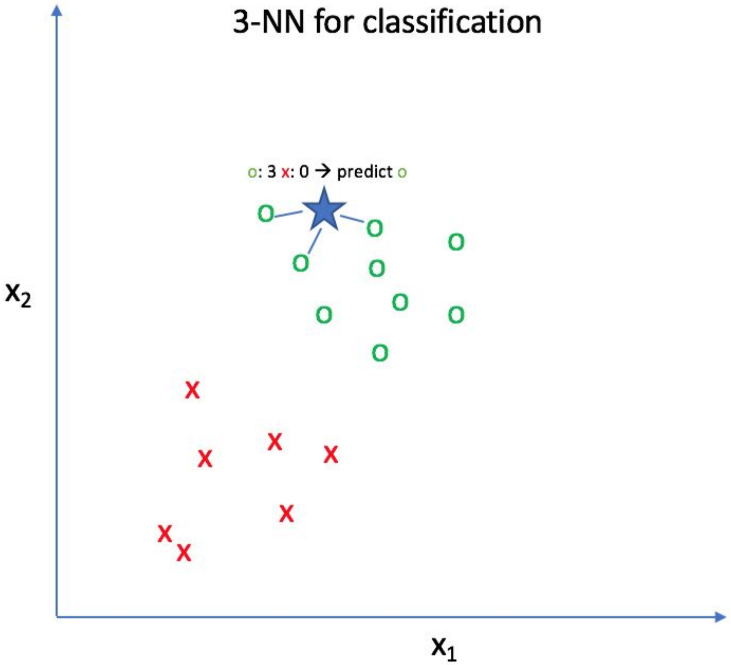
Nearest-neighbor Classification

To do classification, find the set of neighbors and take the most common output value;

If k=3 and the output values are <Yes, No, Yes>, then the classification will be Yes.

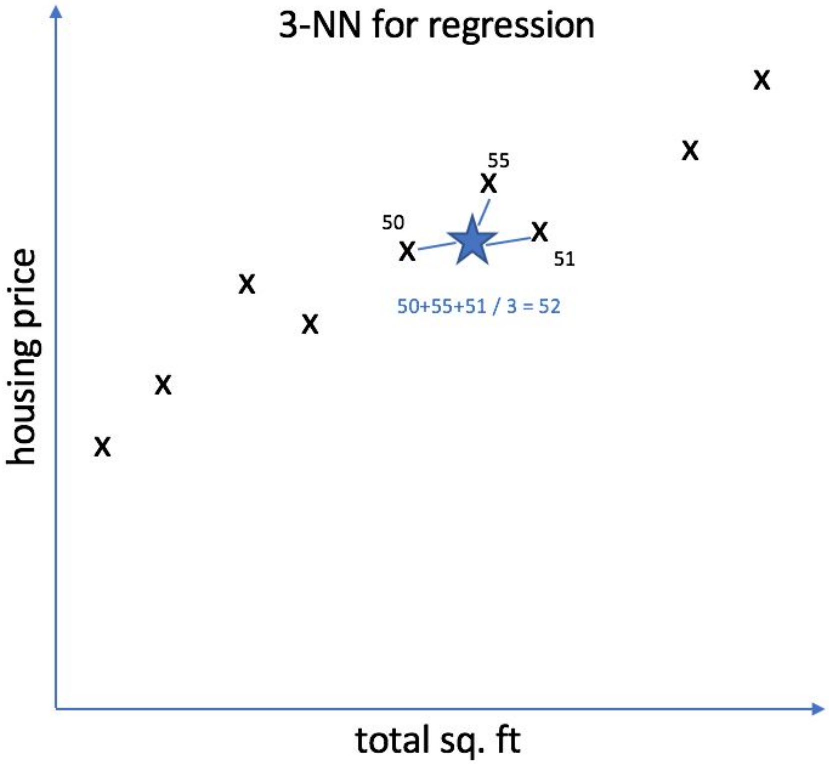
To avoid ties on binary classification, k is usually chosen to be an odd number.



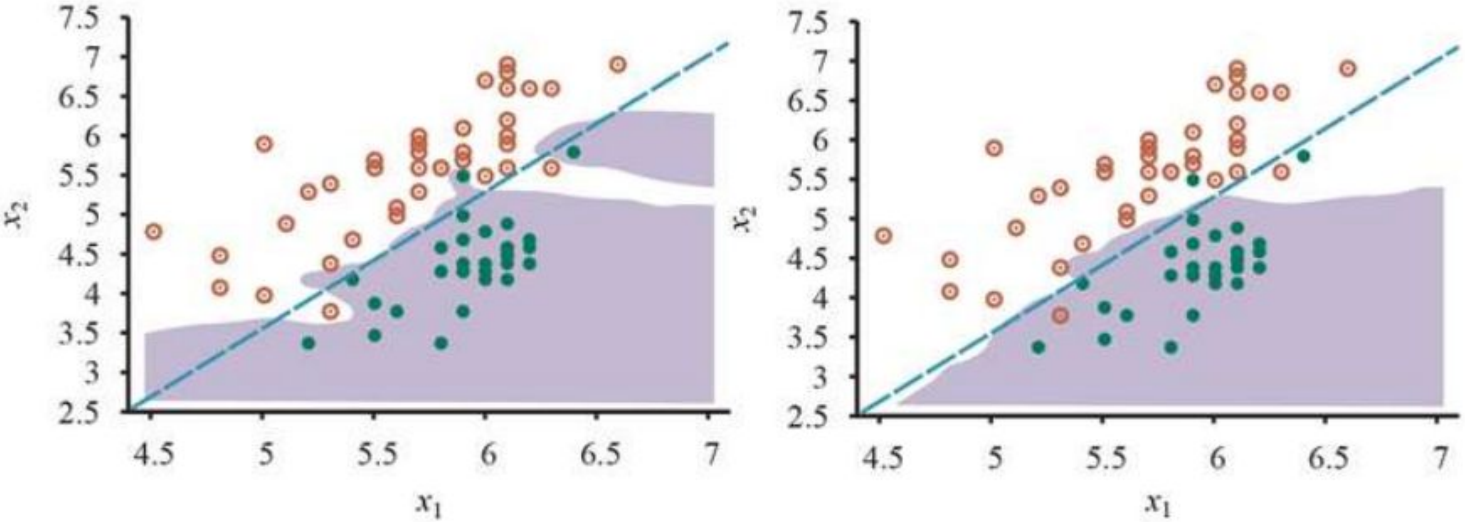


Nearest-neighbor Regression

To do regression, we can take the mean or median of the k neighbors.



How to choose k?



$(k=1)$ $(k=5)$

k is a hyperparameter;

k=1 overfitting; k=5 good fit

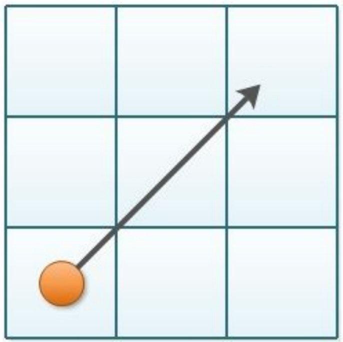
Validation dataset or k-fold cross-validation can be used to select the best value of k.

Distance Metric

How do we measure the distance from a query point to an example point?

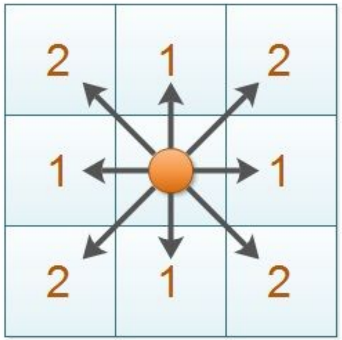
Typically, distances are measured using Euclidean distance or Manhattan distance:

Euclidean Distance

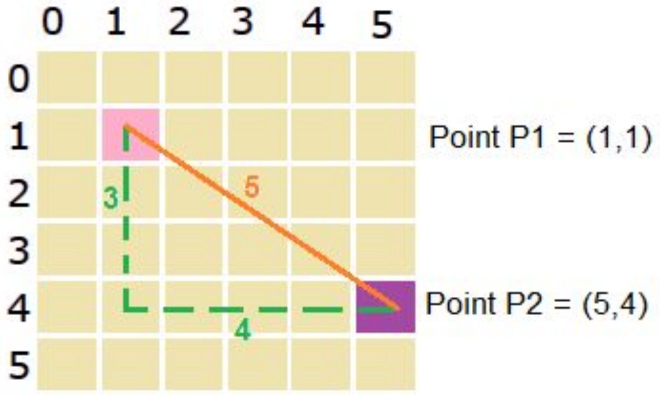


$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Manhattan Distance



$$|x_1 - x_2| + |y_1 - y_2|$$



Euclidean distance = $\sqrt{(5-1)^2 + (4-1)^2}$ = 5

Manhattan distance = |5-1| + |4-1| = 7

Exercise

Consider a dataset with inputs being Acid durability and strength. Goal is to classify if a paper tissue is good or bad based on the inputs. Here are 4 training examples:

x1	x2	Y (output)
7	7	Bad
7	4	Bad
3	4	Good
1	4	Good

Given a test tissue paper with x1 = 3 and x2 = 7, find out the quality of tissue paper?

Assume k = 3 and Euclidean distance can be used as a distance metric.

Exercise

Given a test tissue paper with x1 = 3 and x2 = 7, find out the quality of tissue paper?

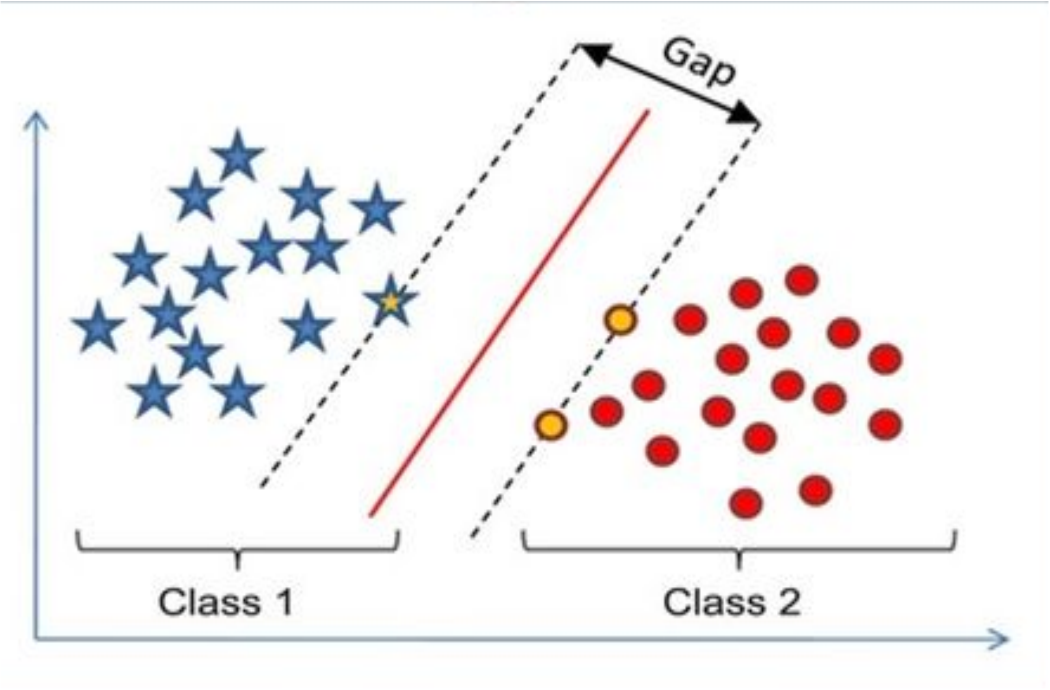
x1	x2	Y (output)	Distance
7	7	Bad	4
7	4	Bad	5
3	4	Good	3
1	4	Good	sqrt(13)

Answer is good since 2 of the 3 neighbors have output good

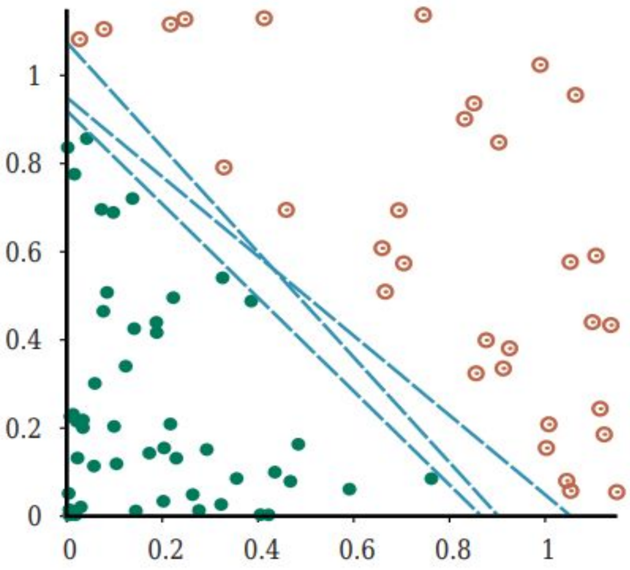
Support Vector Machines

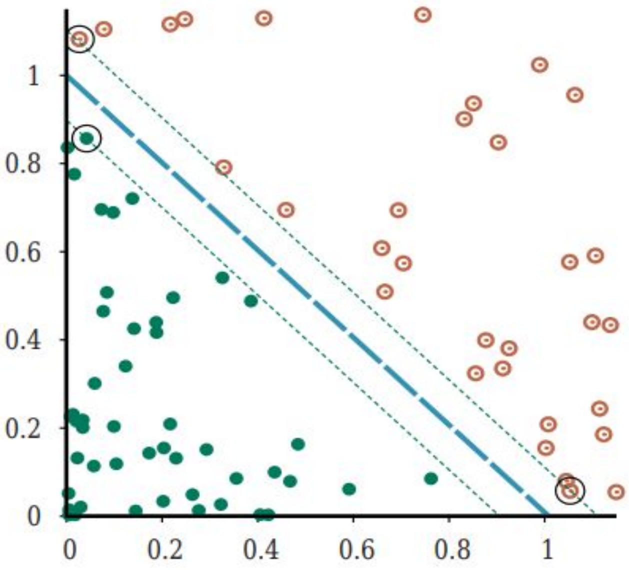
Support Vector Machines

- One of the most popular supervised learning approach
- SVMs construct a maximum margin separator: a decision boundary with the largest possible distance to example points.



Support Vector Machines



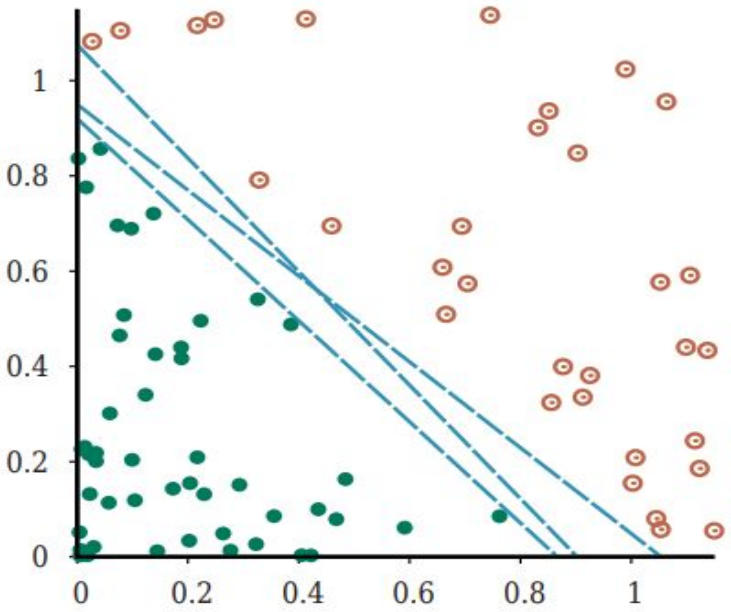


Two classes of points (orange open and green filled circles) and three possible linear separators.

The maximum margin separator (heavy line), is at the midpoint of the margin (area between dashed lines). The support vectors (points with large black circles) are the examples closest to the separator; here there are three.

Support Vector Machines

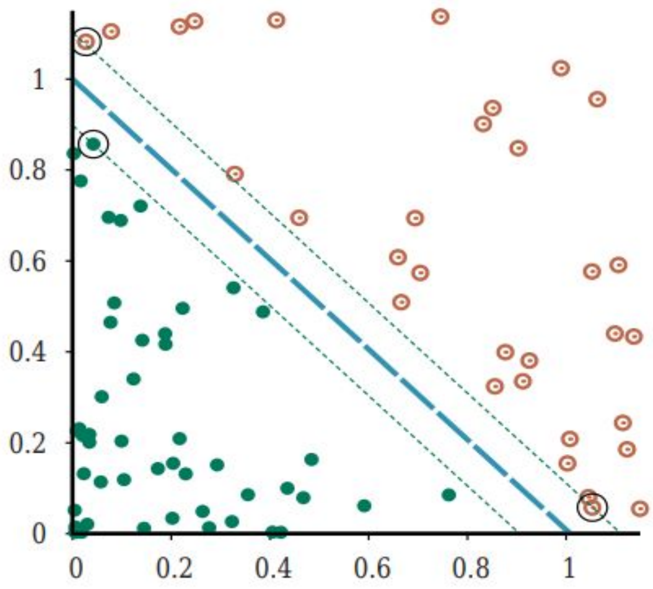
- Consider the lowest of the three separating lines. It comes very close to five of the black examples.
- Although it classifies all the examples correctly, and thus minimizes loss, it is possible that other black examples will turn out to fall on the wrong side of the line.
- Key insight of SVM is to create a decision boundary with the largest possible distance to example points.



Support Vector Machines

- Goal of the SVM is to find the maximum margin separator.
- Now, how do we find this separator?

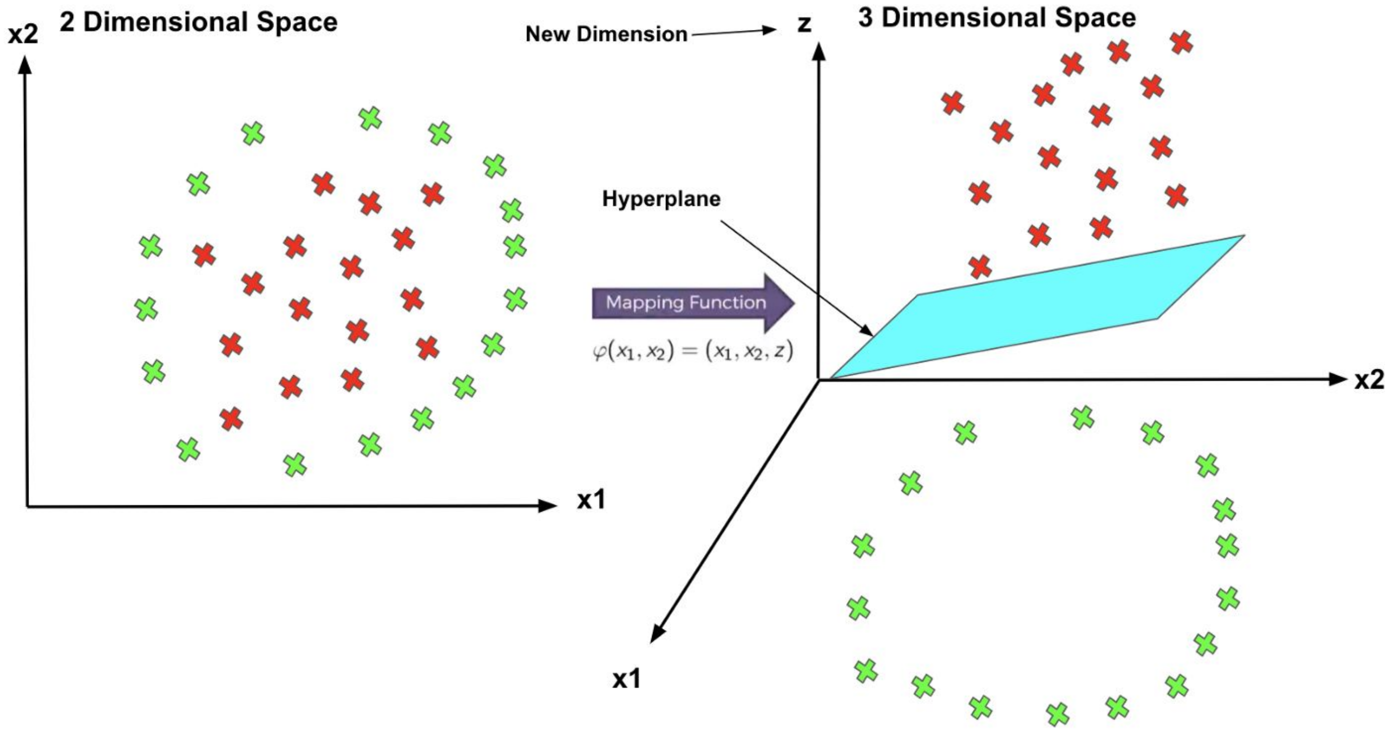
  Separator is defined as the set of points {x : w · x + b=0}. We could search the space of w and b with gradient descent to find the parameters that maximize the margin while correctly classifying all the examples.



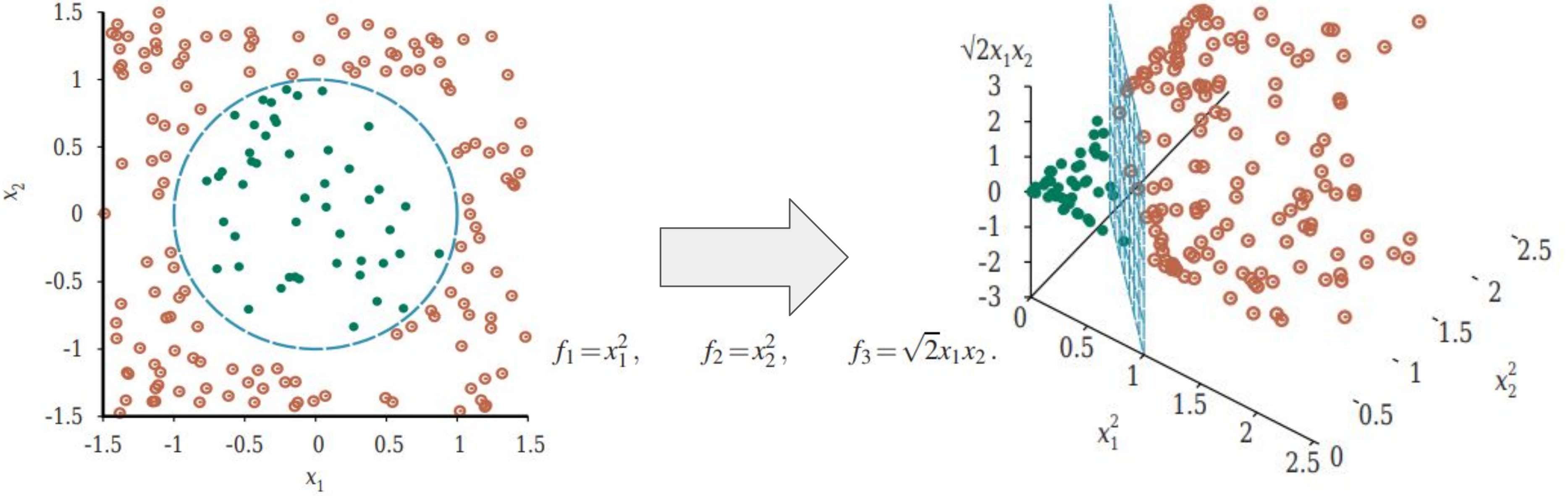
Support Vector Machines

What if the examples are not linearly separable?

Data is mapped into a space of sufficiently high dimension, to make it linearly separable.



Support Vector Machines



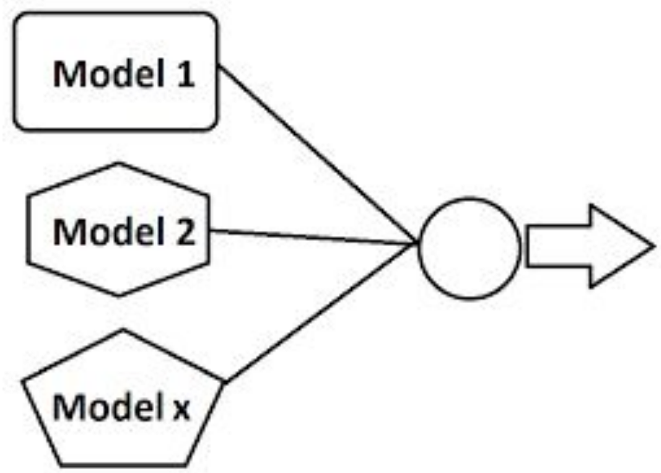
$f_1 = x_1^2, \qquad f_2 = x_2^2, \qquad f_3 = \sqrt{2}x_1x_2.$

Circular decision boundary in becomes a linear decision boundary in three dimensions

Ensemble Learning

Ensemble Learning

- So far we have looked at learning methods in which a single model is used to make predictions.
- The idea of ensemble learning is to select a collection, or ensemble, of models, h_1,h_2,...,h_n, and combine their predictions by averaging or voting.
- Ensemble model: Combination of individual base models.



Advantage of Ensemble Learning

Less Bias

- Base model may be too restrictive, imposing a strong bias (such as linear decision boundary in logistic regression).
- An ensemble can be more expressive, and thus have less bias, than the base models.

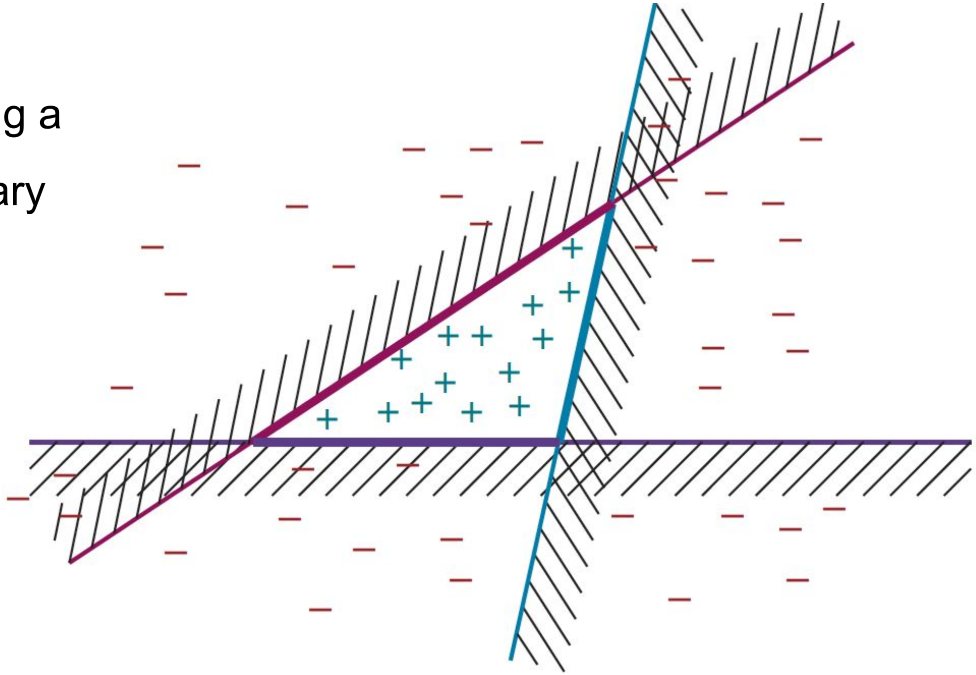


Figure shows that an ensemble of three linear classifiers can represent a triangular region that could not be represented by a single linear classifier.

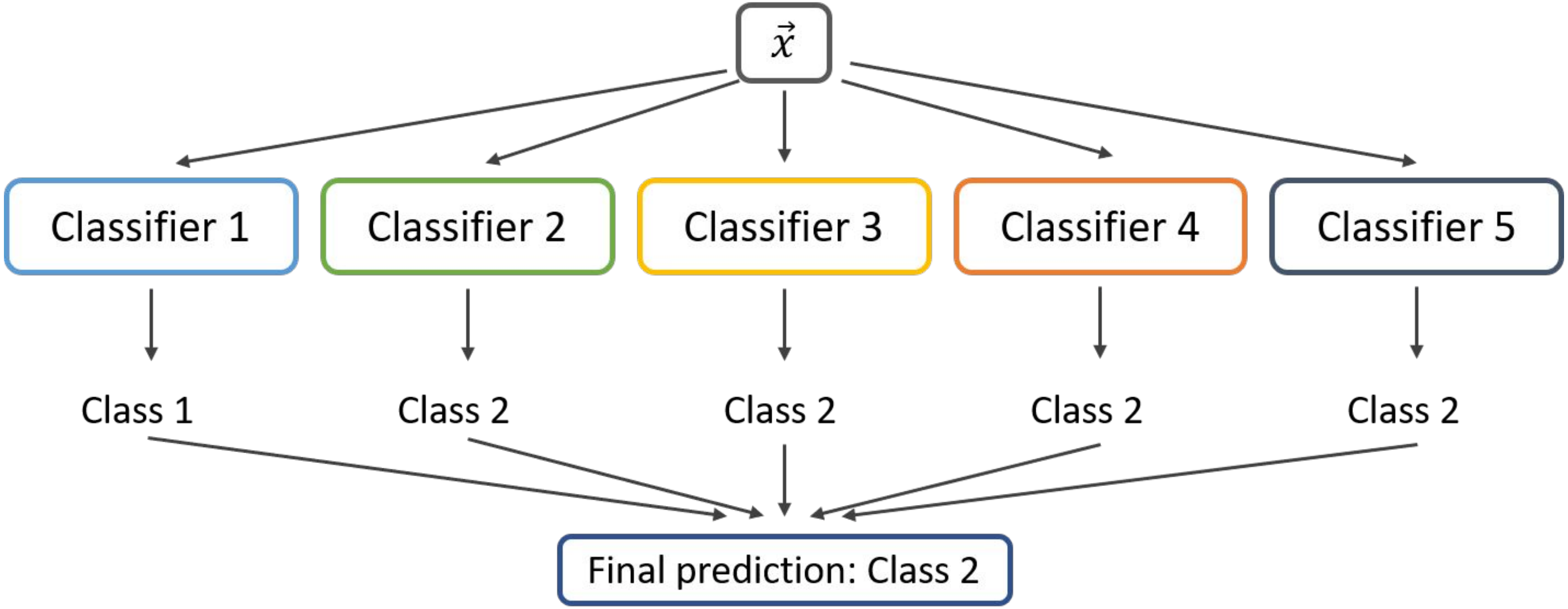
Advantage of Ensemble Learning

Less Variance

Consider an ensemble of 5 binary classifiers that we combine using majority voting.

For the ensemble to misclassify a new example, at least three of the five classifiers have to misclassify it.

The hope is that this is less likely than a single misclassification by a single classifier.



Disadvantage of Ensemble Learning

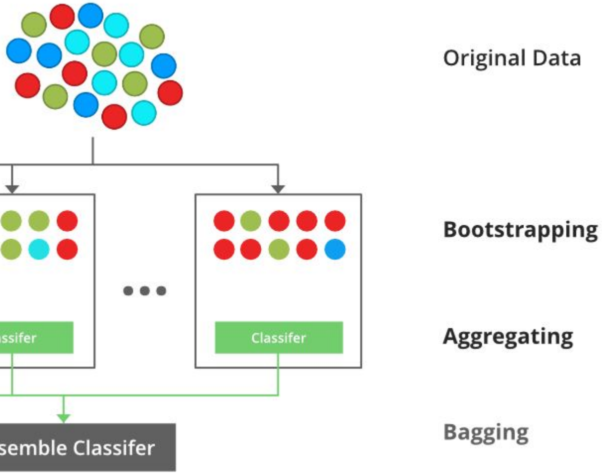
- Ensemble learning is n times more computationally expensive
- Correlated models will share some of the same errors.

  Need to choose independent models:

Two ways of creating ensembles: bagging and boosting.

Bagging

1. Randomly pick N examples from the training set.
2. Run ML algorithm on the N examples to get a model
3. Repeat this process K times, getting K different models
4. Aggregate the predictions from all K models.
    a. For classification problems, take the majority vote
    b. For regression problems, take the average: $h(\mathbf{x}) = \frac{1}{K}\sum_{i=1}^{K} h_i(\mathbf{x})$

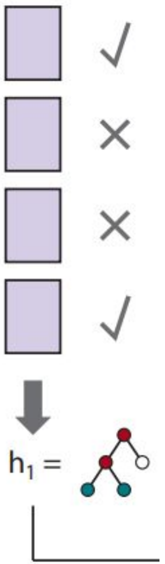


Bagging

- Most commonly used with decision trees: because decision trees are unstable: a slightly different set of examples can lead to a different tree.
- Models can be computed in parallel

Boosting

- Most popular ensemble method
- Weighted training set: each example has an associated weight $w_j \geq 0$ that describes the importance of each example during training.
- Boosting starts with equal weights w_j =1 for all the examples.



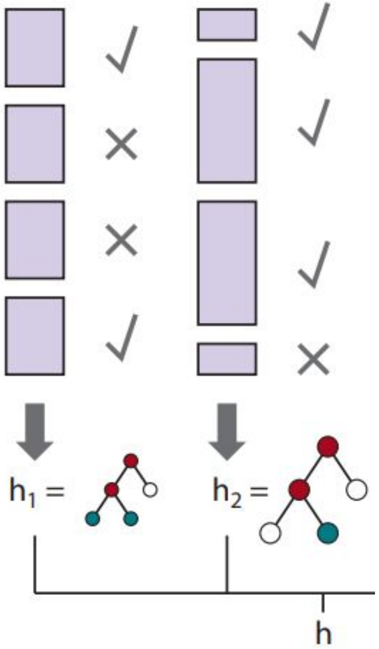
Each shaded rectangle corresponds to an example; the height of the rectangle corresponds to the weight.

Boosting

Generates the first model, h_1, which will classify some of the training examples correctly and some incorrectly.

We would like the next model to do better on the misclassified examples, so we increase their weights while decreasing the weights of the correctly classified examples.

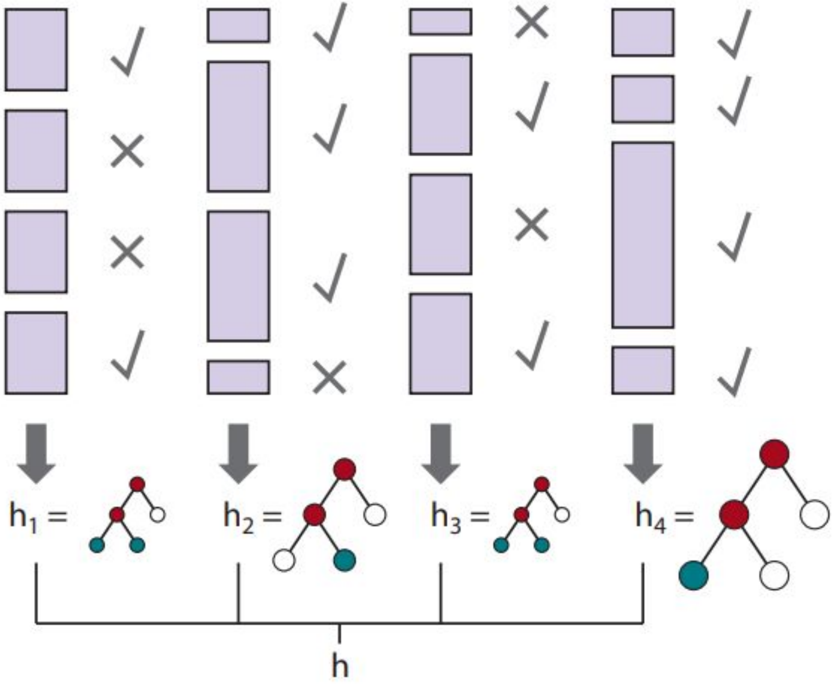
From this new weighted training set, we generate model h_2.



Each shaded rectangle corresponds to an example; the height of the rectangle corresponds to the weight. The checks and crosses indicate whether the example was classified correctly by the current hypothesis.

Boosting

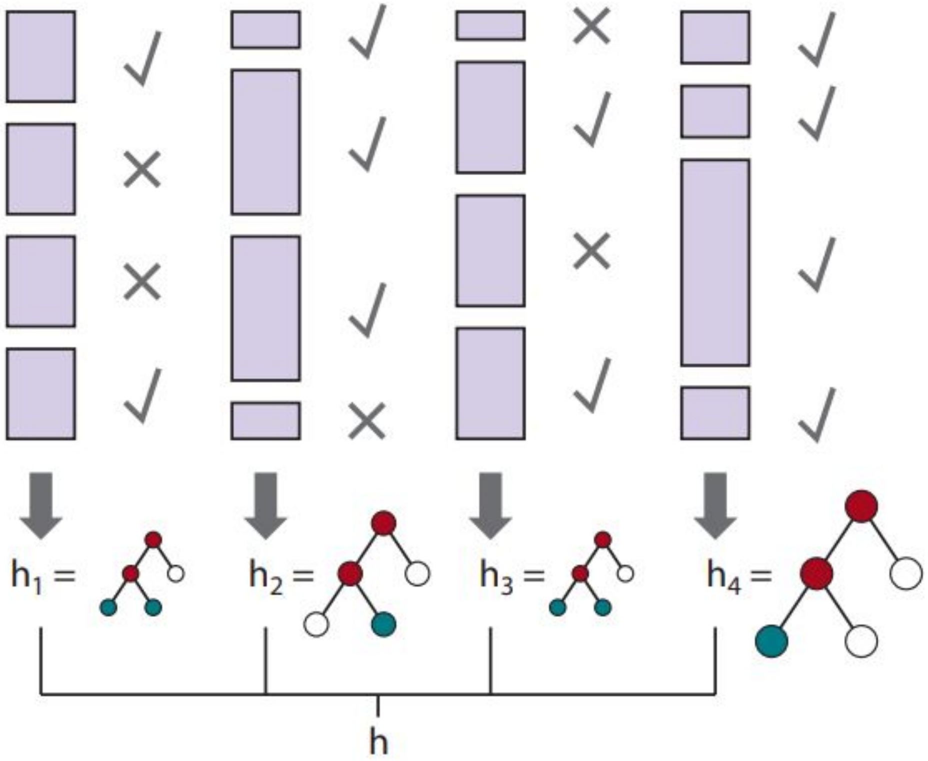
- Process continues until we have generated K hypotheses, where K is an input to the boosting algorithm.
- Examples that are difficult to classify will get increasingly larger weights until the algorithm is forced to create a hypothesis that classifies them correctly.



The size of the decision tree indicates the weight of that hypothesis in the final ensemble.

Boosting

- The final ensemble lets each hypothesis vote: $$h(\mathbf{x}) = \sum_{i=1}^{K} z_i h_i(\mathbf{x})$$ where z_i is the weight of the ith hypothesis.



The size of the decision tree indicates the weight of that hypothesis in the final ensemble.

Boosting

- Sequential algorithm, so we can’t compute all the models in parallel as we could with bagging.
- Many variants of the boosting idea, with different ways of adjusting the example weights and combining the models.
  - All share the idea that difficult examples get more weight as we move from one model to the next.

THANK YOU!