Supervised Learning

CS 586
Machine Learning

Prepared by Jugal Kalita

With help from Alpaydin's *Introduction to Machine Learning*,
Chapter 2.

Learning a Class from Examples

- *Task*: Learn class C of a “family car”.
- *Training examples*: A group of people label examples of cars shown as “family car” or “not family car”. Some are positive examples and some are negative examples.
- *Class learning*: To learn a class, the learner has to find a description that is consistent with all the positive examples and none of the negative examples.
- *Prediction:* Once the class description is learned, the learner can can predict the class of a previously unseen car.

Features and Training Examples

- A car may have many features.
- Examples of features: year, make, model, color, seating capacity, price, engine power, type of transmission, miles/gallon, etc.
- Based on expert knowledge or some other technique, we decide that we will use only two of the features: price and engine power.
- Among all the features, these two explain best the difference between a family car and other types of cars.
- Let $x_1 = price$, $x_2 = engine\ power$ be the two input features.

- Training Examples: $X = \{\mathbf{x}^t, r^t\}, t = 1 \cdots N$. Here

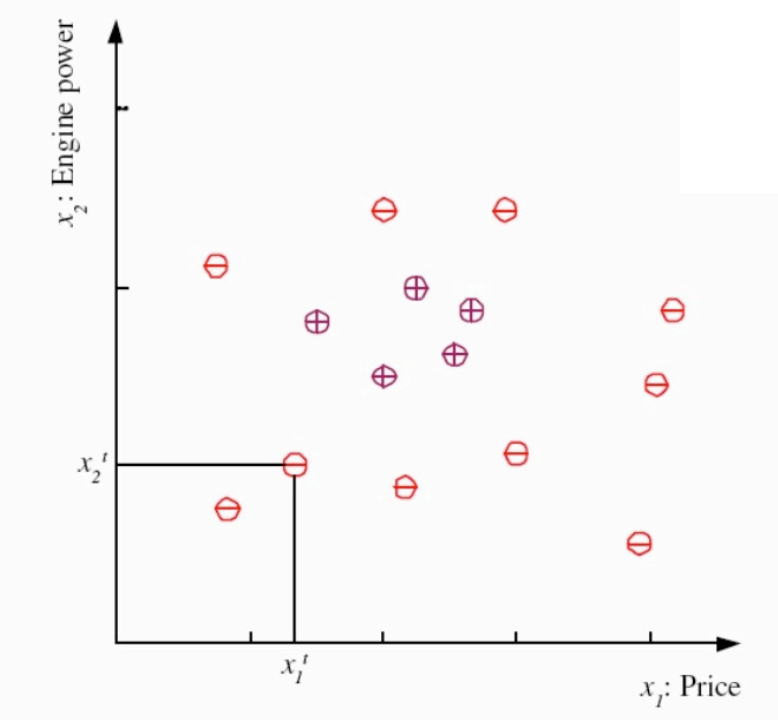
$$\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} price \\ engine\ power \end{bmatrix}$$

and

$$r = \begin{cases} 1\ if\ \mathbf{X}\ is\ positive \\ 0\ if\ \mathbf{X}\ is\ negative \end{cases}$$

- Thus, we have N training examples and each training example is a 2-tuple.
- We draw the training examples in a graph in the next page. + means a positive example and - means a negative example.

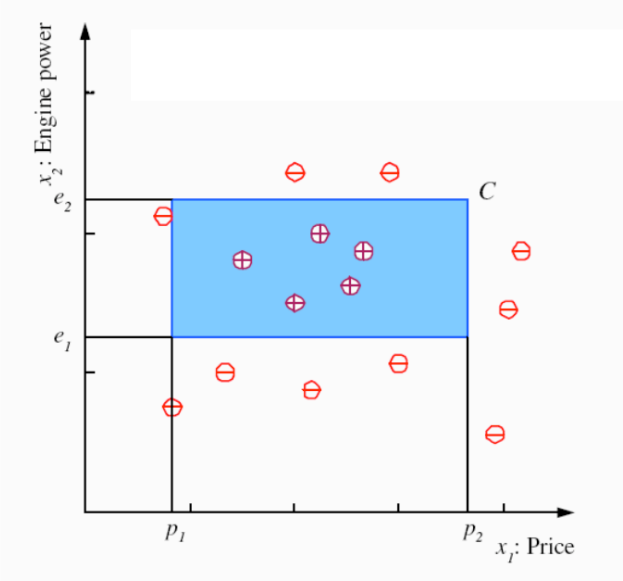
Training Set for Class Car



- Taken from Alpaydin 2010, Introduction to Machine Learning, page 22.

Learning Hypothesis for Class Car

- The ideal hypothesis learned should cover all of the positive examples and none of the negative examples.
- Assume that the hypothesis class is a set of rectangles.



Taken from Alpaydin 2010, Introduction to Machine Learning, page 23.

The Hypothesis Learned

- Since the hypothesis learned is a rectangle, we can express it as a simple rule:

$$h = (p_1 \leq price \leq p_2) \wedge (e_1 \leq engine\ power \leq e_2)$$

  for suitable values of p_1, p_2, e_1 and e_2.
- We can say the hypothesis class $\mathcal{H}$ learned is a set of rectangles.
- The specific hypothesis learned is $h \in \mathcal{H}$.
- Once we have chosen the hypothesis class as rectangle, it amounts to choosing four parameters that define the specific rectangle learned.
- The hypothesis learned should be as close to the real class definition $\mathcal{C}$ as possible.

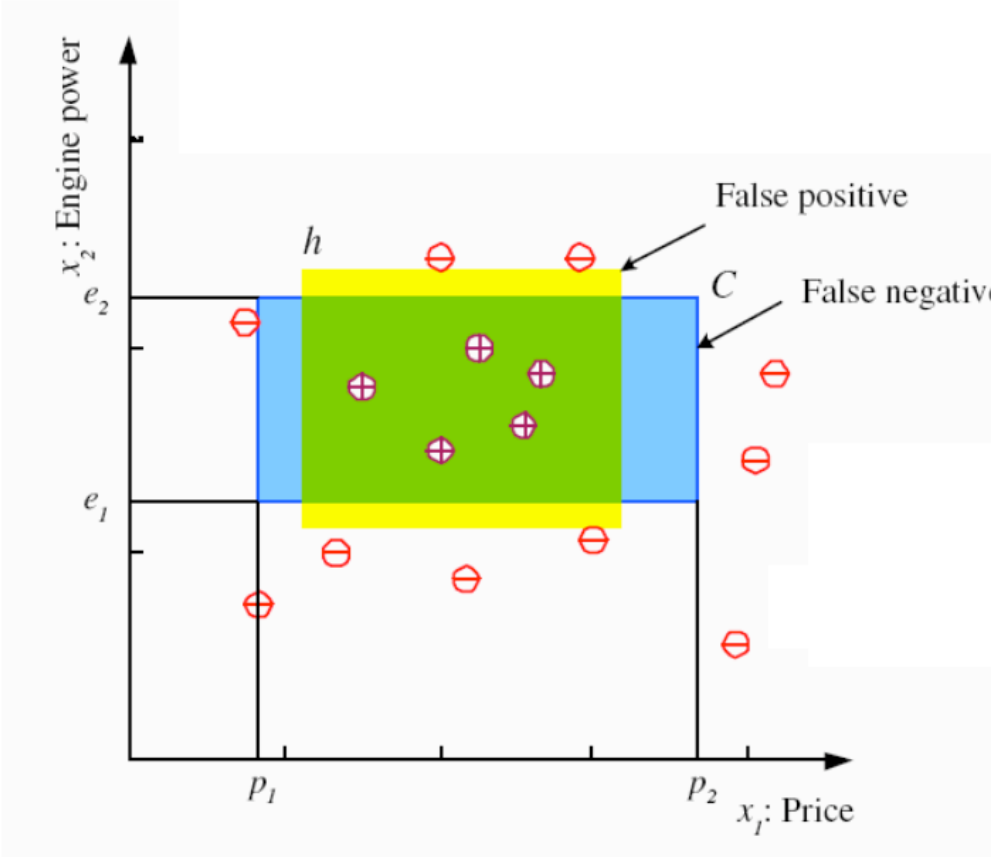
Measuring Error in the Hypothesis Learned

- In real life, we do not know the hypothesis $\mathcal{C}$. So, we cannot evaluate how close $\mathcal{C}$ is to h.
- Empirical Error: It is the proportion of training instances that are classified wrongly by the hypothesis h learned.
- The error of hypothesis h given a training set $\mathcal{X}$ is

$$E(h \mid \mathcal{X}) = \sum_{t=1}^{N} 1\left(h(\mathbf{x}^t) \neq r^t\right)$$

- Here $1(a \neq b)$ is 1 if $a \neq b$ and is 0 if $a = b$.

Actual Class $\mathcal{C}$ and Learned Hypotheis h



Taken from Taken from Alpaydin 2010, Introduction to Machine Learning, page 25.

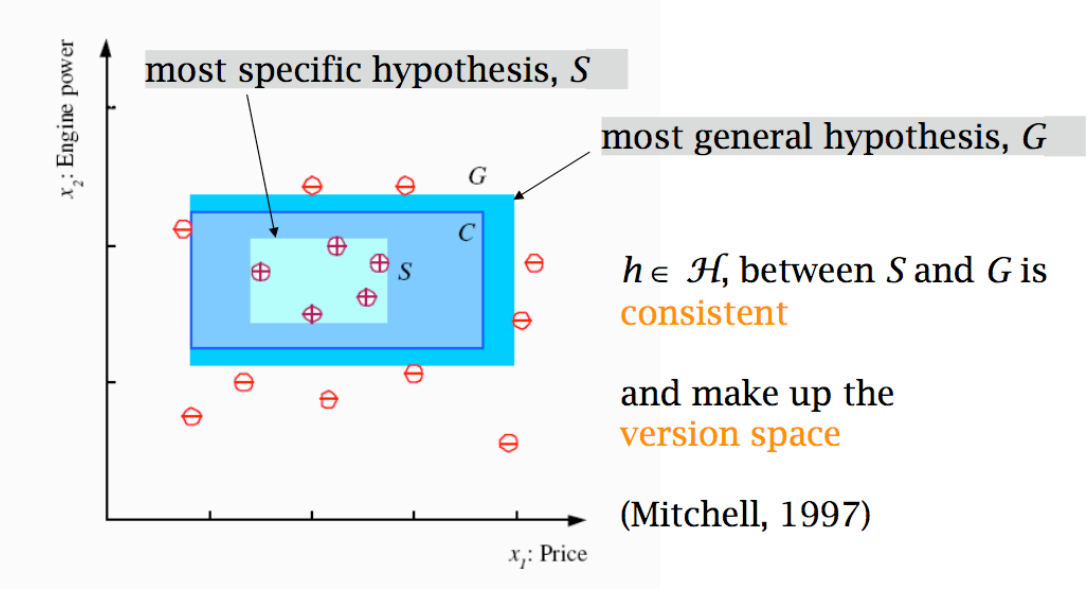
The Problem of Generalization

- Since the hypothesis class $\mathcal{H}$ is the set of all rectangles, we need four points to define a hypothesis h.
- We can find infinitely many rectangles that are consistent with the training examples, i.e, the error E is 0.
- However, different hypotheses that are consistent with the training examples may behave differently with future examples when the system is asked to classify.
- Generalization is the problem of how well the learned classifier will classify future unseen examples. A good learned hypothesis will make fewer mistakes in the future.

Most Specific and Most General Hypotheses

- The most specific hypothesis S is the tightest rectangle drawn around all the positive examples without including any negative examples.
- The most general hypothesis G is the largest rectangle we can draw including all positive examples and no negative examples.
- Any hypothesis $h \in \mathcal{H}$ between S and G is a valid hypothesis with no errors, and thus consistent with the training set.
- All such hypotheses h make up the *Version Space* of hypotheses.
- In fact, there may be a set of most general hypotheses and another set of most specific hypotheses. These two make up the boundary of the version space.

Most Specific and Most General Hypotheses

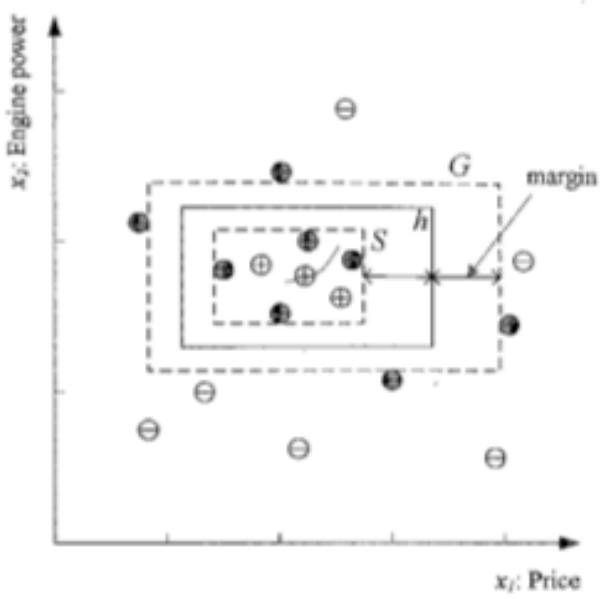


Taken from Taken from Alpaydin 2010, Introduction to Machine Learning, page 26.

Choose Hypothesis with Largest Margin

- To choose the ”best” hypothesis, we should choose h halfway between S and G..
- In other words. we should increase the *margin*, the distance between the boundary and the instance closest to it.
- For the error function to have a minimum value at h with the maximum margin, we should use an error (loss) function which not only checks whether an instance is on the correct side of the boundary but also how far away it is.

Choose Hypothesis with Largest Margin for Best Separation

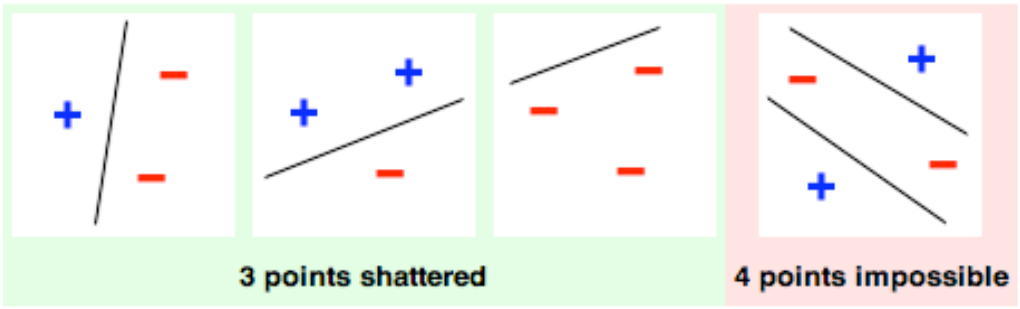


Taken from Taken from Alpaydin 2010, Introduction to Machine Learning, page 27.

Vapnik-Chervonenkis (VC) Dimension

- It gives a pessimistic bound on the number of items a classification hypothesis class can classify without any error.
- Assume we have N 2-D points in a dataset. If we label the points in this dataset arbitrarily as +ve and -ve, we can label them in 2^N ways. Therefore, 2^N different learning problems can be defined with N data points.
- If for each of these 2^N labelings of the dataset, we can find a hypothesis $h \in \mathcal{H}$ that separates the +ve examples from the -ve examples, we say that h *shatters* N points.
- The maximum number of points that can be shattered by $\mathcal{H}$ is called the *Vapnik-Chervonenkis dimension* of $\mathcal{H}$.
- $VC(\mathcal{H})$ is measures the *capacity* of $\mathcal{H}$.

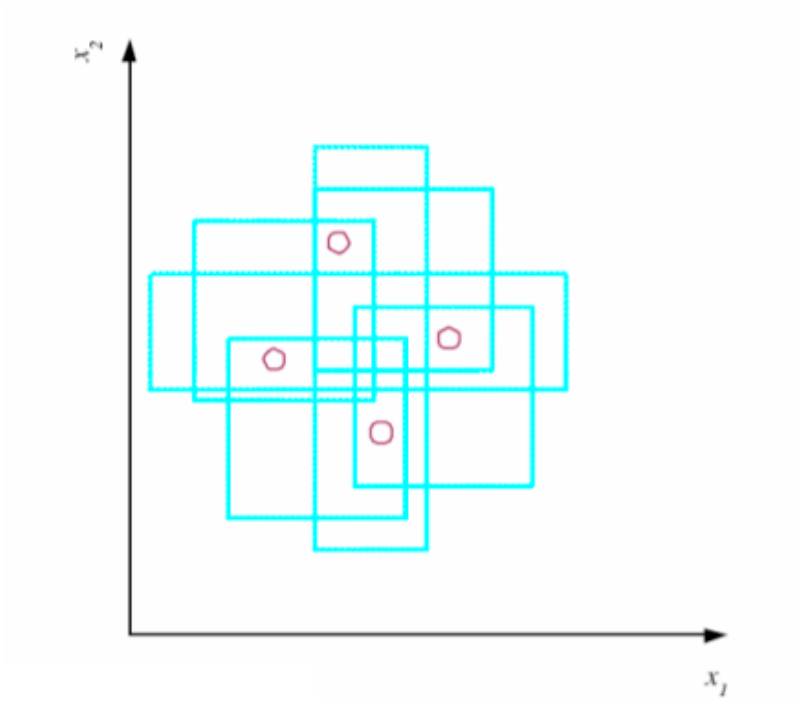
VC Dimension Example



A hypothesis $h \in \mathcal{H}$ where $\mathcal{H}$ is the set of straight lines in 2-D can shatter 3 points.

Taken from http://en.wikipedia.org/wiki/Vapnik_chervonenkis_dimension.

VC Dimension Example



A hypothesis $h \in \mathcal{H}$ where $\mathcal{H}$ is the set of axis-aligned rectangles in 2-D can shatter 4 points.

Taken from Taken from Alpaydin 2010, Introduction to Machine Learning, page 28.

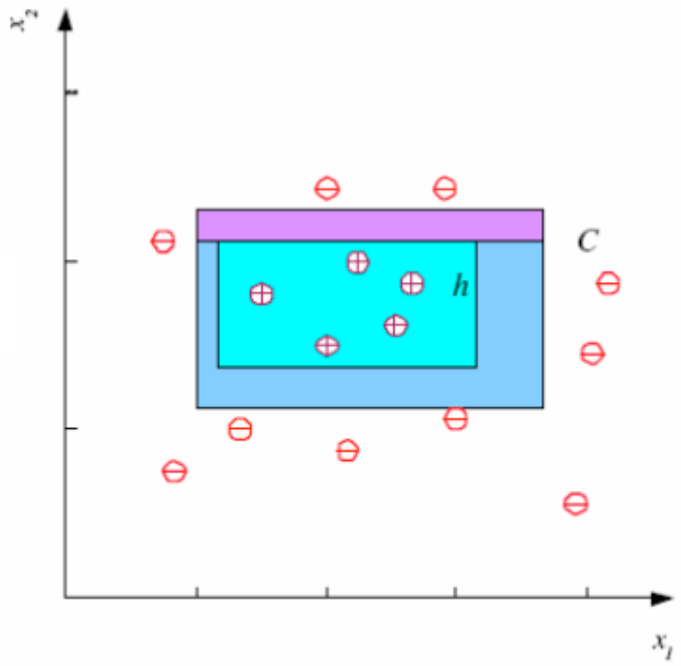
VC Dimension: Discussion

- VC Dimension gives a very pessimistic estimate of the classification capacity of a hypothesis class.
- For example, it says that we can correctly classify only three points using a straight line hypothesis, and only 4 points using a rectangle hypothesis.
- What's Missing: VC Dimension does not take into account the probability distribution from which instances are drawn.
- In real life, the world usually changes smoothly. Instances that are close to each other usually share the same label.
- Thus, the classification capacity of a hypothesis class is usually much more than its VC Dimension.

Probably Approximately Correct (PAC) Learning

- When we learn a hypothesis, we want it to be approximately correct, i.e., the error probability is bounded by a small value.
- PAC Learning: Given a class C, and examples drawn from some unknown but fixed probability distribution $p(x)$, we want to find the number of examples N that the learner must see so that it can learn a hypothesis h with error at most $\epsilon > 0$ with probability at least $1 - \delta$, where $\delta \leq \frac{1}{2}$.
- Alpyadin gives an example where he derives the value of N for the tightest rectangle hypothesis.
- Russell and Norvig (3rd Edition) gives an example where he derives the value of N for decision list learning. A decision list is like a decision tree, but at each node a new condition is learned starting from an empty node.

PAC Learning for the Tightest Rectangle Hypothesis



Taken from Taken from Alpaydin 2010, Introduction to Machine Learning, page 30.

We want to learn the tightest possible rectangle h around a set of positive training examples.

PAC Learning for the Tightest Rectangle Hypothesis (continued)

- The error region between the actual class description C and the hypothesis learned $h = S$ is sum of 4 rectangular strips.
- Let the probability of a positive example falling in any one of the strips be $\epsilon/4$ for a total error of ϵ.
- We count the overlaps in the corners twice, the total error is actually less that $4 \times \epsilon/4$.
- The probability that a randomly drawn +ve example misses a strip is $1 - \epsilon/4$.
- The probability that N independent draws of positive examples miss a strip is $(1 - \epsilon/4)^N$.
- The probability that all N independent draws of positive examples miss all four strips is $4(1 - \epsilon/4)^N$.

PAC Learning for the Tightest Rectangle Hypothesis (continued)

- Using Taylor series expansion of e^{-x}, we can get the inequality $(1 - x) \leq exp(-x)$. We are going to use to find a value for N.
- Using this inequality and a little algebra, we can get

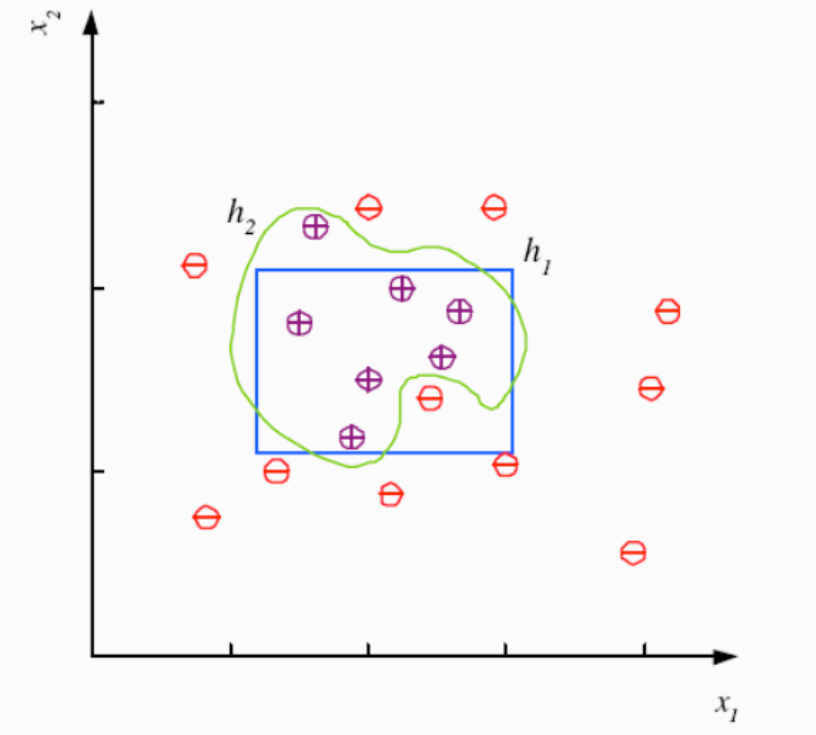
$$N \geq \frac{4}{\epsilon} log(\delta/4)$$

- Thus, provided that we take at least $N \geq \frac{4}{\epsilon} log(\delta/4)$ independent examples from C and use the tightest rectangle as our hypothesis h, with *confidence probability* of at least $1-\delta$, a given point will be misclassified with *error probability* at most ϵ.

Noise

- There may be noise in the training examples due to several reasons.
  - Input attributes may be recorded imprecisely. E.g., we could measure something that should have a value of 2, to be 3 instead.
  - There may be errors in labeling data points. E.g.. a teacher can train a positive example negative and vice versa.
  - There may be relevant other attributes that we missed. E.g., the color attribute may be important in classifying a car as a family car. We are not considering this attribute.

Noise Example



Taken from Alpaydin 2010, Introduction to Machine Learning, page 31.

Noise (continued)

- When we have noise, there is no simple boundary between positive and negative examples.
- With noise, one needs a complicated hypothesis that corresponds to a hypothesis class with larger capacity.
- A rectangle needs 4 parameters, but a complex hypothesis needs more parameters to obtain 0 error.

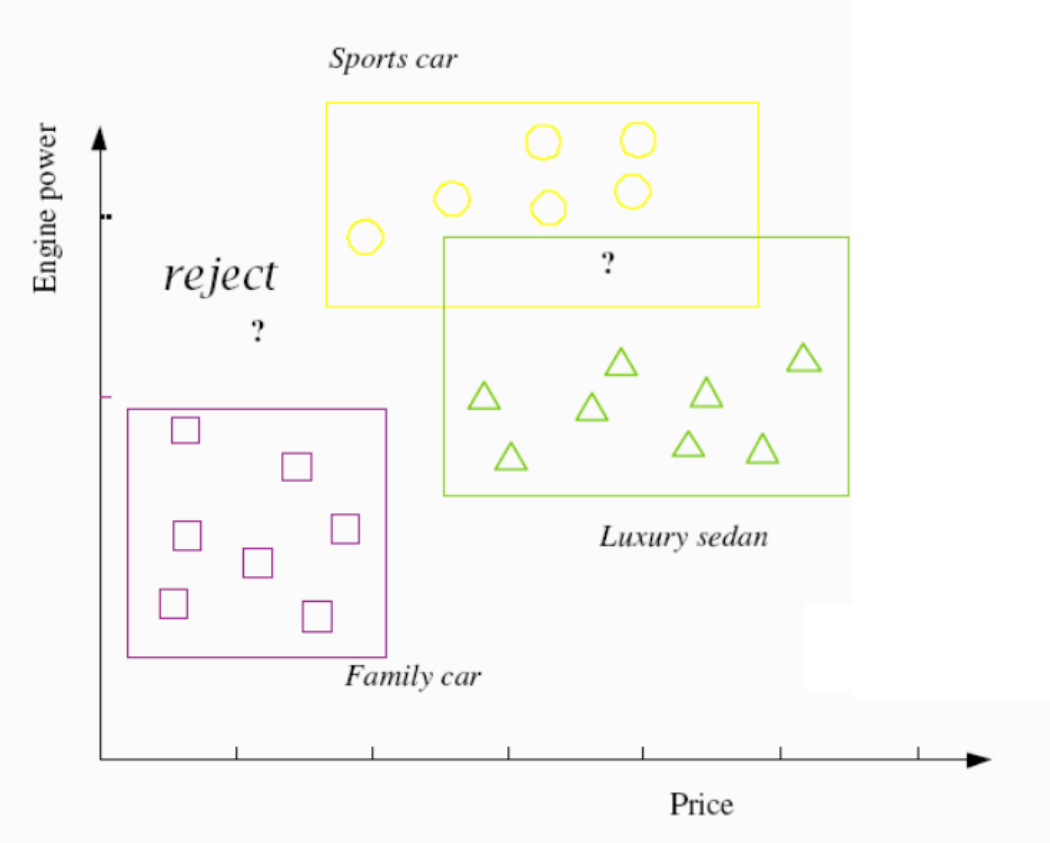
Choose a Simple Hypothesis

- A simple hypothesis is still preferred because
  - It is simple to use. E.g., we can check whether a point is inside a rectangle more easily than other shapes.
  - it is simple to train and has fewer parameters. Thus, it needs fewer training examples.
  - It is a simple model to explain.
  - if there is error in the input training data, a simple hypothesis may generalize better, being able to classify unseen examples better in the future.

Learning Multiple Classes

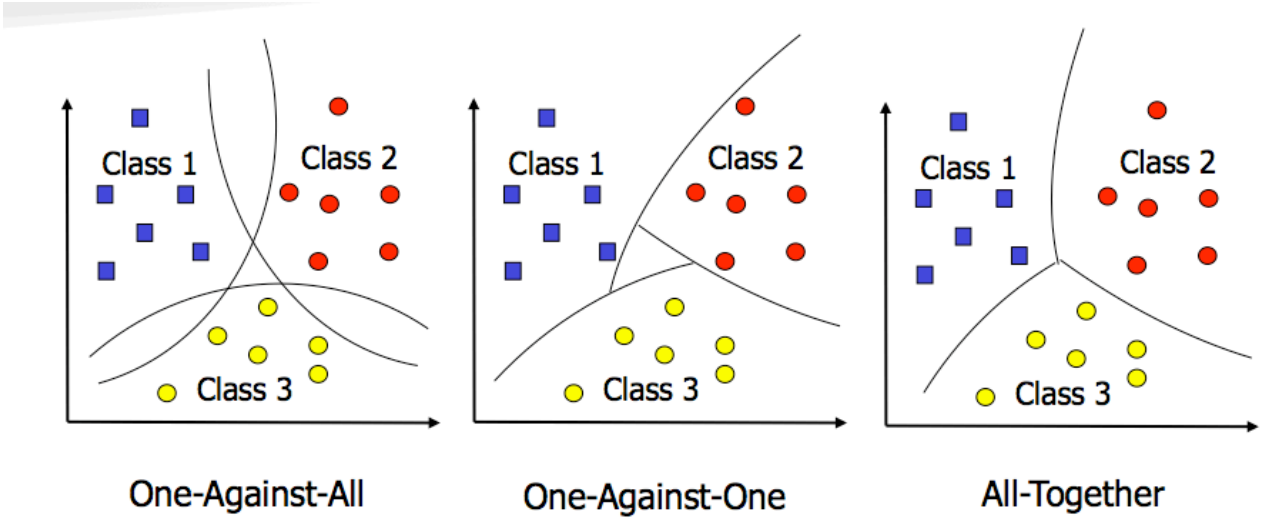
- The problems we have seen so far are 2-class problems: family car, not family car; 0/1, etc.
- A multi-class classifier can be implemented in several ways.
  - One-against-all: Train K classifiers. An individual classifier answers the question if an item belongs to a specific class or not.
  - One-against-one: Construct a binary classifier for each pair of classes. We need $\frac{1}{2}K(K-1)$ classifiers.
  - All-together Classifier: Most accurate, but most difficult. Learn to classify into the K classes at the same time in one classifier.

Multi-class Classifiers



Taken from Alpaydin 2010, Introduction to Machine Learning, page 33.

Comparing Multi-class Classifiers



Taken from Habib and Kalita, 2009.

Model Selection and Generalization

- Learning is an ill-defined problem. It is not possible to find a unique solution.
- Inductive Bias: We need an assumption regarding the nature of $\mathcal{H}$, whether it is a straight line, a rectangle, a circle, etc. If the inductive bias is not well-suited learning will not take place well.
- Our goal is to train a learner that can generalize well, or perform well on unseen data.
- Overfitting or underfitting may occur.
- Overfitting occurs when $\mathcal{H}$ is more complex than C.
- Underfitting occurs when $\mathcal{H}$ is less complex than C.

Trade-offs in Learning

- There is a trade-off among three factors:
  - Complexity of $\mathcal{H}$
  - Amount N of training data available
  - Amount of generalization error E we want to incur.

Cross Validation

- To run a supervised learning experiment, we need to split data into three sub-sets
  - Training Set (50%, say)
  - Validation Set (25%, say)
  - Test Set (25%, say)

Running a Supervised Learning Experiment

- We need training data of size N
- We need to choose a hypothesis class.
- We need describe the hypothesis class in terms of parameters that define the hypothesis class: 4 points for rectangle, intercept and slope for line, center and radius for circle, etc. The parameters will be learned.
- We need to define an error or loss function that computes the difference between what the current model predicts as output and the desired output as given in training.
- We need a learning algorithm or optimization procedure that gives us the learned parameters that minimize the error.