Supervised Learning

Hsing-Kuo Pao (鮑興國)
National Taiwan University of Science & Technology (Taiwan Tech)

Outline

- Learning from data: an example for supervised learning
- Classification problem and various issues
  - Deciding hypothesis space
  - Handling noise
  - Multiple class classification
- PAC learning
- VC dimension
- Regression problem

Learning from Data:
A Supervised Learning Example

— Supervised Learning

Learning a Class from Examples

Suppose we want to learn a class C

- Example: “sports car”
- Given a collection of cars, have people label them as sports car (positive example) or non-sports car (negative example)
- Task: find a description that is shared by all of the positive examples and none of the negative examples
- Once we have this definition for C, we can
  - Predict – given a new unseen car, predict whether or not it is a sports car
  - Describe/compress – understand what people expect in a car

Choosing an Input Representation

- Suppose that of all the features describing cars, we choose price and engine power. Choosing just two features
  - Making things simpler
  - Allowing us to ignore irrelevant attributes
- Let
  - x_1 represent the price (in USD)
  - x_2 represent the engine volume (in cm3)
- Then each car is represented

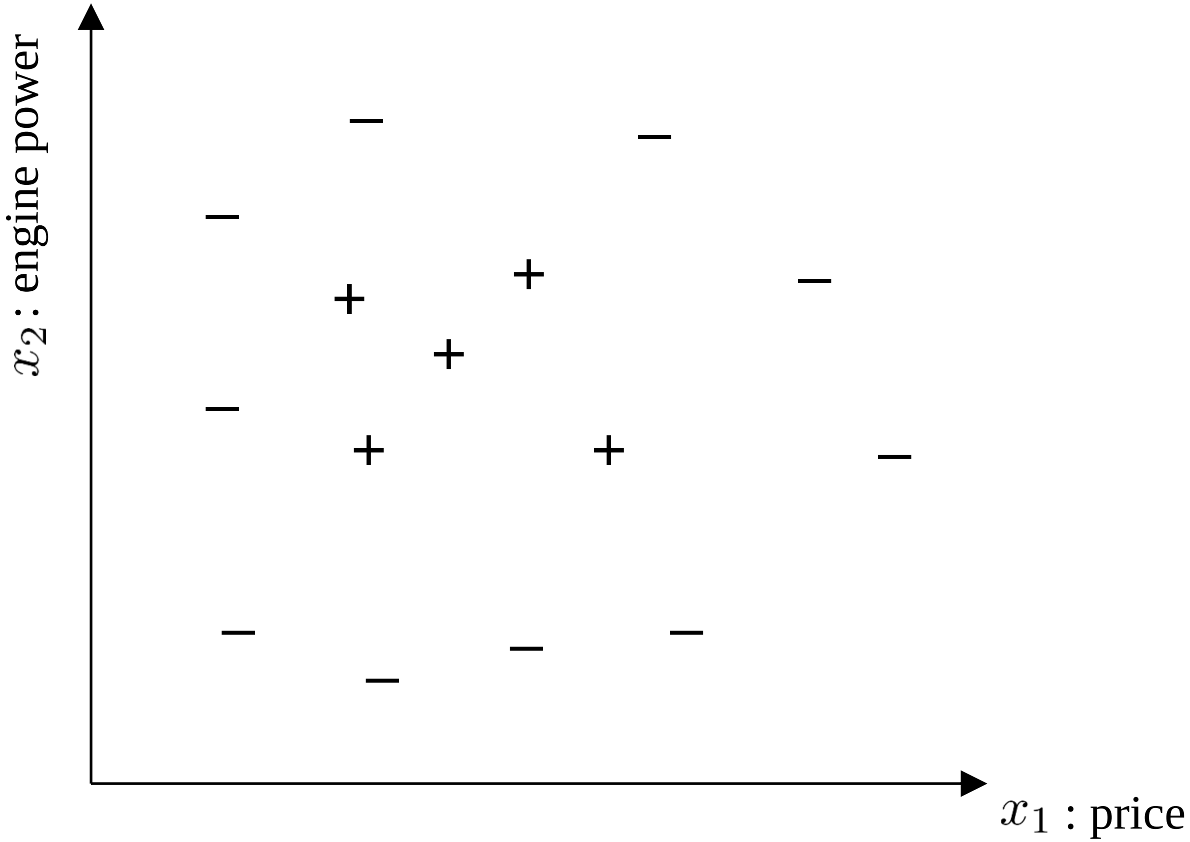
$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

and its label y denotes its type $y = \begin{cases} 1 & \text{if } \mathbf{x} \text{ is a positive example} \\ 0 & \text{if } \mathbf{x} \text{ is a negative example} \end{cases}$

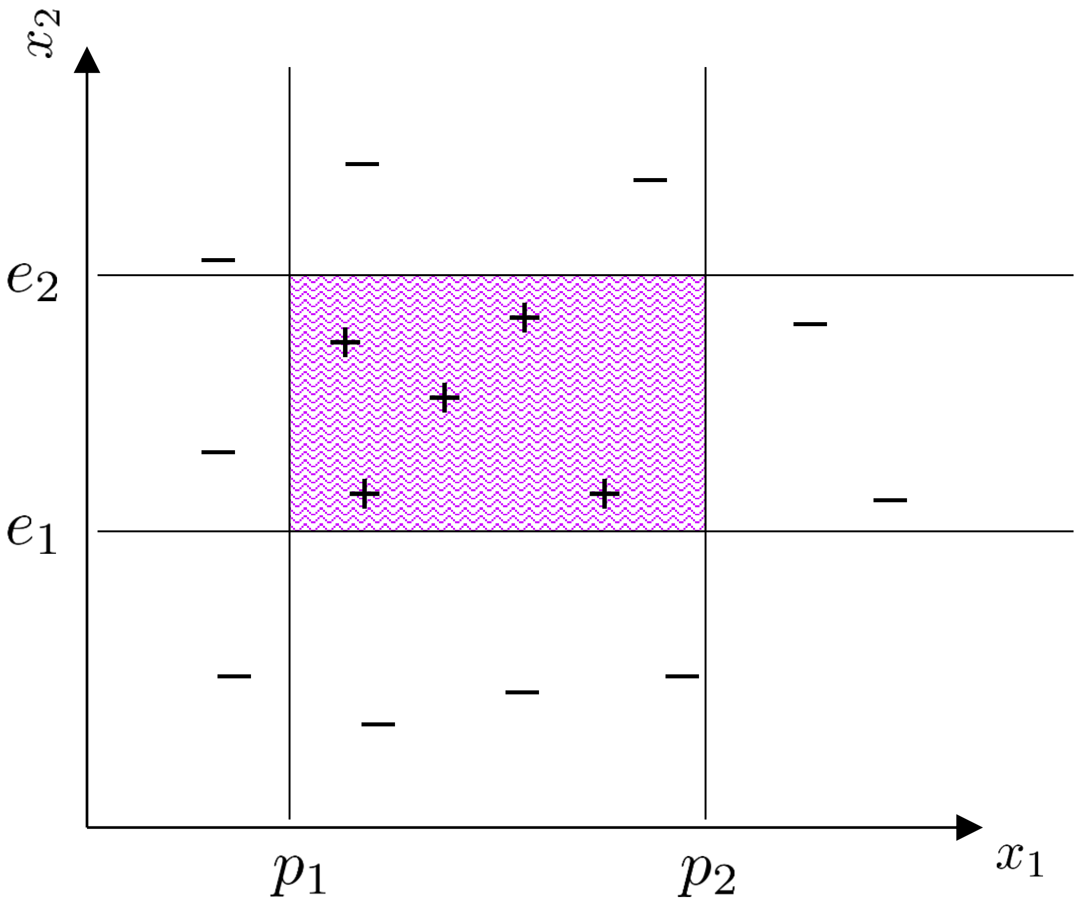
- Each example represented by the pair $(\mathbf{x}, y)$
and a training set containing m examples represented by

$$\mathcal{D} = \{(\mathbf{x}^{(i)}, y_i), i = 1, \ldots, m\}$$

Plotting the Training Data



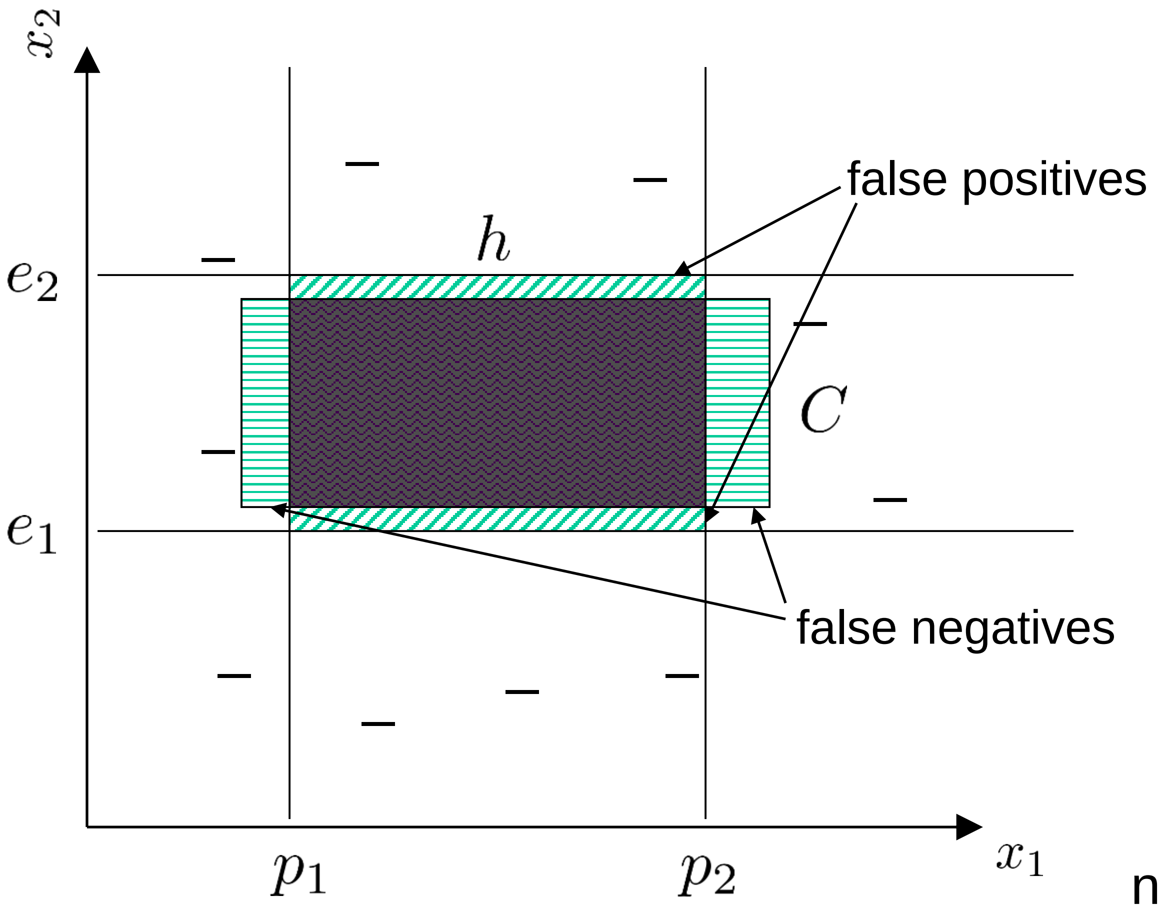
Hypothesis Class



Suppose that we think that for a car to be a sports car, its price and its engine power should be in a certain range:

$$(p_1 \leq \text{price} \leq p_2) \text{ AND } (e_1 \leq \text{engine} \leq e_2)$$

Concept Class



no training errors

Suppose that the actual class is C

task: find $h \in \mathcal{H}$ that is **consistent** with $\mathcal{D}$

Choosing a Hypothesis

(carefully review what we have done…)

- Each (p_1, p_2, e_1, e_2) defines a hypothesis $h \in \mathcal{H}$
- Empirical Error: proportion of training instances where predictions of h do not match the training set

$$E(h \mid \mathcal{D}) = \frac{1}{m} \sum_{i=1}^{m} \mathbf{1}(h(\mathbf{x}^{(i)}) \neq y_i)$$

- We need to find the best one (possibly with the minimum empirical error)…

The Complete Story of Learning

- Issue #1: choosing the hypothesis space
  - Is $C \in \mathcal{H}$ or not?
  - Usually the hypothesis can not be too complicated!
- Issue #2: if $C \in \mathcal{H}$, how to find it?
  - e.g., parametric methods: learning problem can be reduced to parameter estimation if the hypothesis space is fixed

```
00000000 55 push ebp                 0000000A E805000000 call 0x14
00000001 89E5 mov ebp,esp            0000000F 83C404 add esp,byte +0x4
```

a complete prediction program?

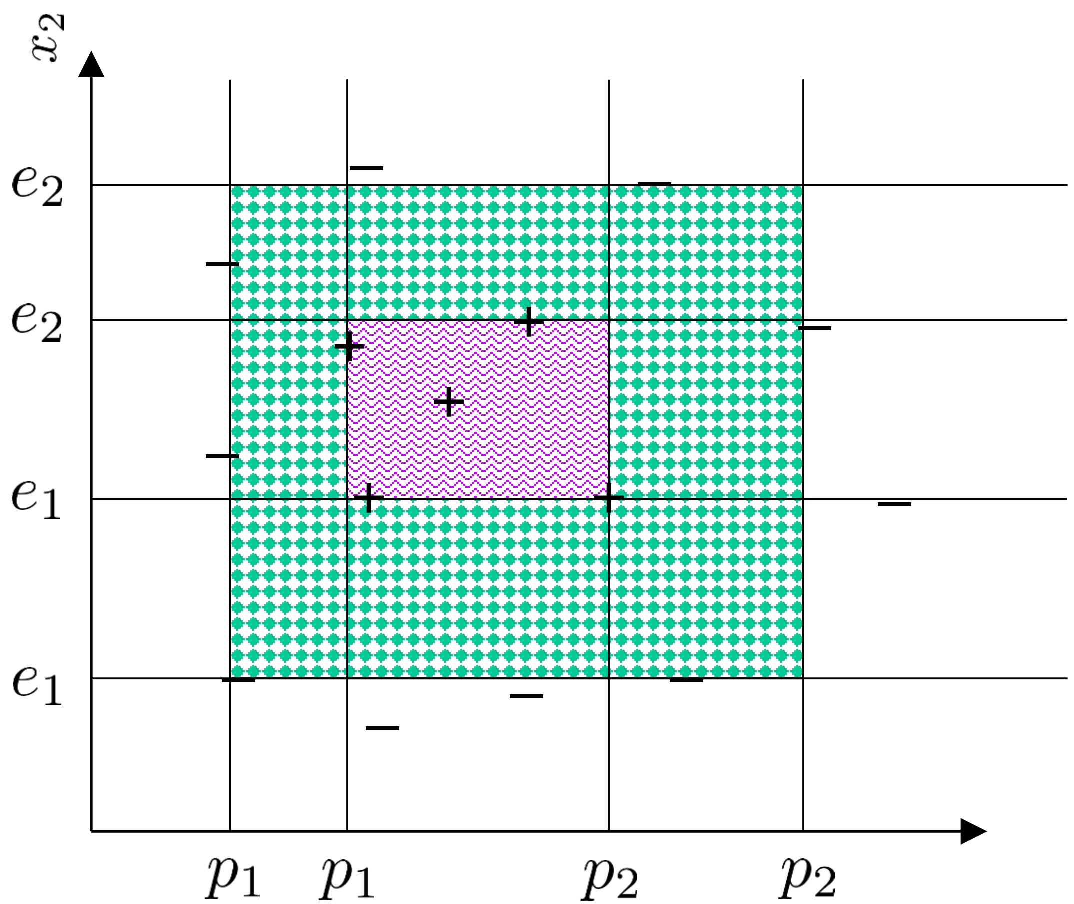
program 1: skeleton **program 2: parameter estimation**

```
0000000A 89F6 mov esi,esi            00000015 89E5 mov ebp,esp
0000000C 55 push ebp                 00000017 83EC38 sub esp,byte +0x38
0000000D 89E5 mov ebp,esp            0000001A 57 push edi
0000000F C9 leave                    0000001B 56 push esi
00000010 C3 ret                      0000001C 8B4508 mov eax,[ebp+0x8]
```

The Complete Story of Learning (cont'd)

- Deciding the hypothesis space is hard (or an art) in general!
  - Inductive bias, Occam's razor, …
- Once the hypothesis space is decided, the rest is relatively easy
  - Model evaluation (based on error rate, F-measure, or other cost sensitive measures, …)
  - Parameter estimation: relatively straight-forward
    - Parametric approach
    - Nonparametric approach
- Focusing on the second part most of the time!

Hypothesis Choice



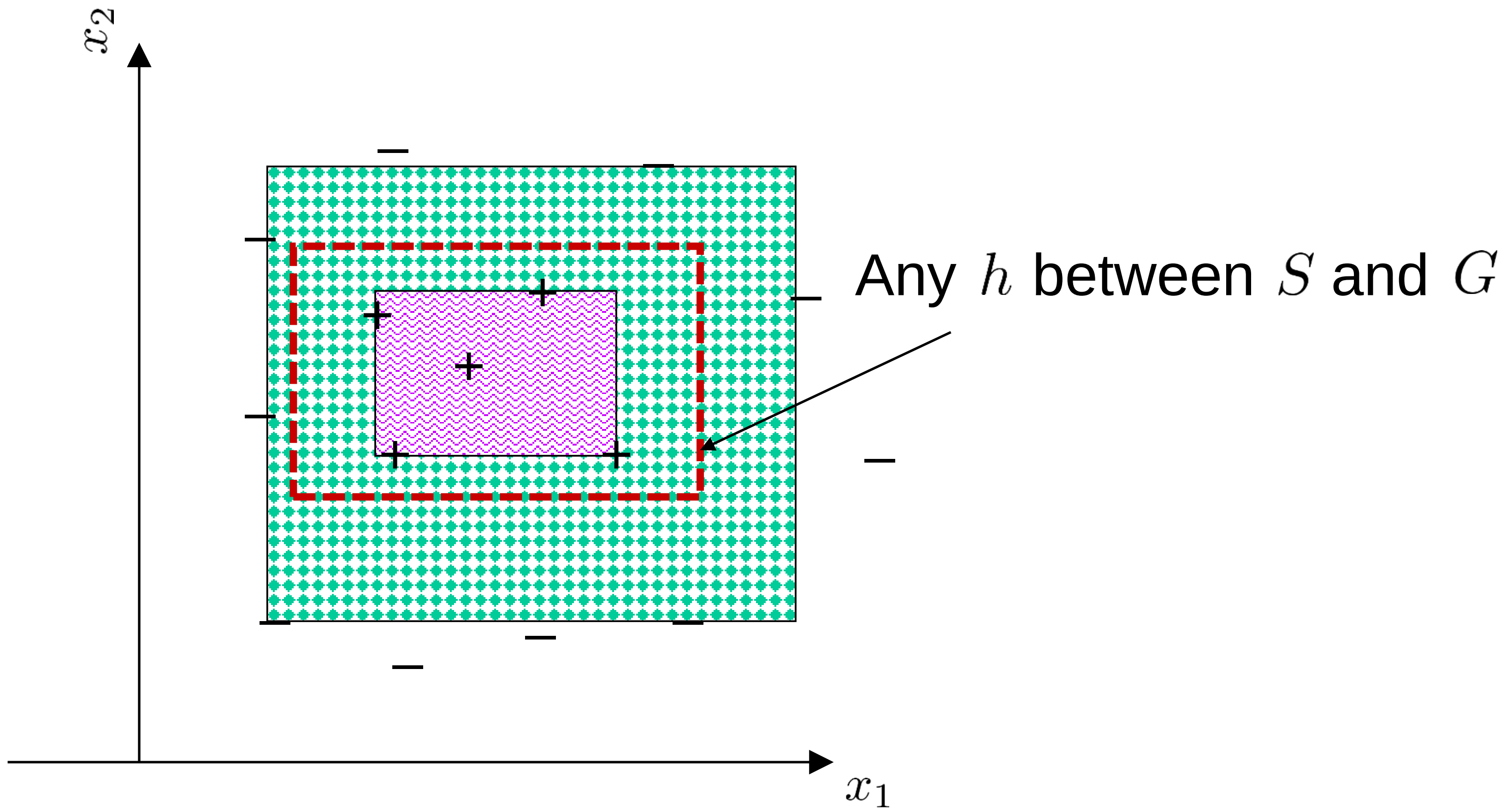
Most specific?

Most general?

Most specific hypothesis S

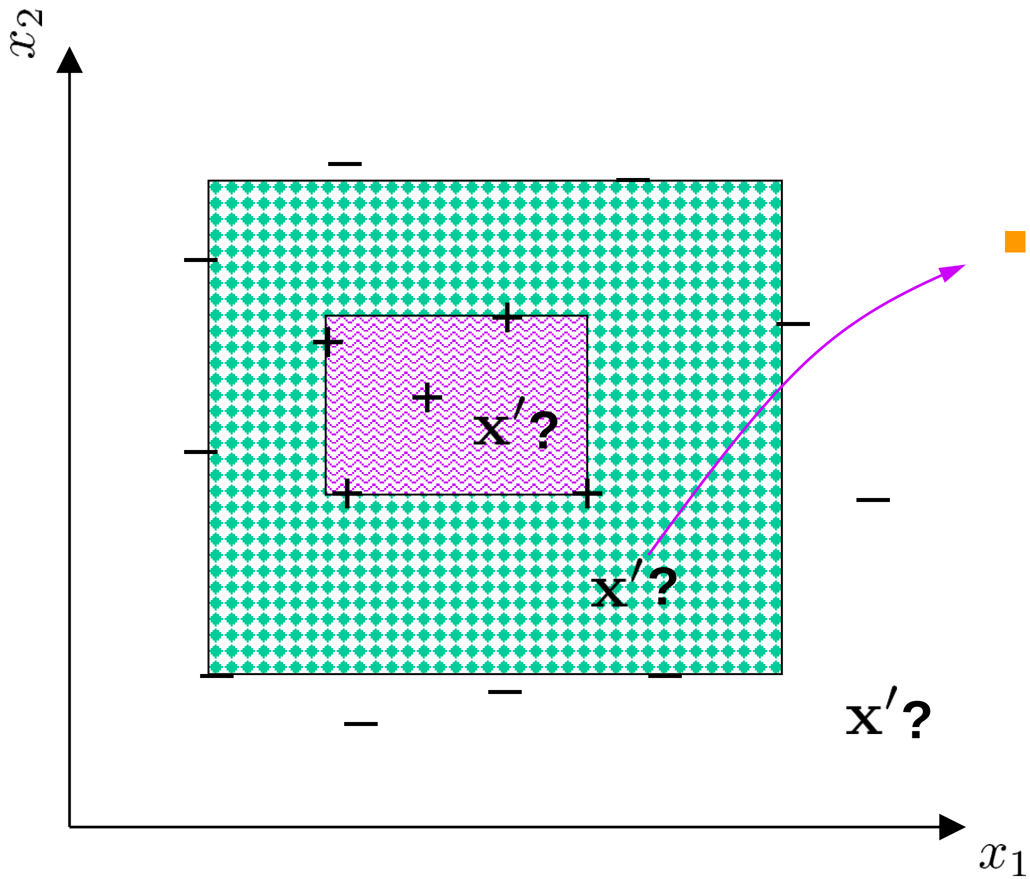
Most general hypothesis G

Consistent Hypothesis



G and S define the boundaries of the Version Space.
The set of hypotheses more general than S and more specific than G forms the **Version Space**, the set of consistent hypotheses

Now what?



- *Using the average of S and G or just rejecting it to experts?*

How do we make prediction for a new $\mathbf{x}'$?

Version Space on another Example

example #	x_1	x_2	x_3	x_4	y
1	1	1	0	0	1
2	1	0	0	0	1
3	0	1	1	1	0

- $\mathcal{H} =$ conjunctive rules

$$S = x_1 \wedge (\neg x_3) \wedge (\neg x_4)$$

$$G = x_1, \neg x_3, \neg x_4$$

Issues

- Hypothesis space must be flexible enough to represent concept
- Making sure that the gap of S and G sets do not get too large
- Assumes no noise!
  - Inconsistently labeled examples will cause the version space to **collapse**
  - There have been extensions to handle this…

PAC Learning

— Supervised Learning

PAC Learning

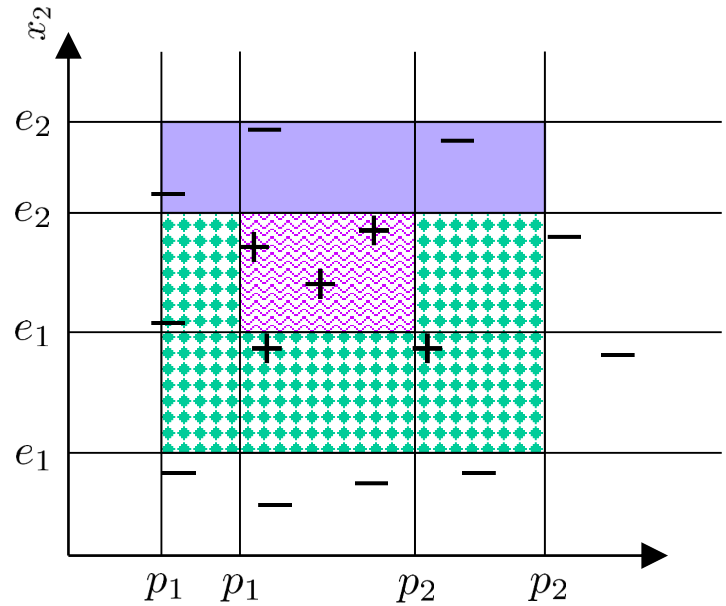
- Probably Approximately Correct learning: a framework for machine learning theory
- **Given:** a class C, and examples drawn from some unknown but fixed probability distribution $p(\mathbf{x})$
- **Find:** the number of examples m, such that with probability at least $1-\delta$, the hypothesis h has error at most ϵ, for arbitrary $\delta \leq 1/2$ and $\epsilon > 0$

$$P(C \Delta h \leq \epsilon) \geq 1 - \delta$$

A PAC Learner from Sports Car

- The probability that a randomly drawn example misses the strip is at least

  $$1 - 4\epsilon$$



- For m independently draws

  $$(1 - 4\epsilon)^m$$

- Choose m and δ such that

  $$4(1 - \epsilon m/4) \leq 4\exp(-\epsilon m/4) \leq \delta\,, \quad (1 - x) \leq \exp(-x)$$

- That is,

  $$m \geq (4/\epsilon)\log(4/\delta)$$

- Collecting more data to reduce error!

VC Dimension

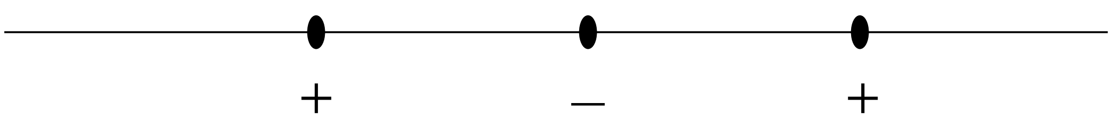
— Supervised Learning

Vapnik-Chervonenkis (VC) Dimension

- Suppose we have a training set with m points
- The number of ways of labeling m points as positive or negative?
  - A different labeling is a different problem!
- If, for **some** set of m points and for **any** of labeling of the m points, we can find a consistent hypothesis $h \in \mathcal{H}$, we say that $\mathcal{H}$ **shatters** m points
- The maximum number of points that can be shattered by $\mathcal{H}$ is the VC dimension of $\mathcal{H}$, $\mathrm{VC}(\mathcal{H})$
- $\mathrm{VC}(\mathcal{H})$ is a measure of the capacity of the hypothesis class $\mathcal{H}$

Example I

- $\mathbf{x} \in \mathbb{R}$, $\mathcal{H} =$ interval on line
  - There exists two points that can be shattered
  - No set of three points can be shattered
  - $\mathrm{VC}(\mathcal{H}) = 2$

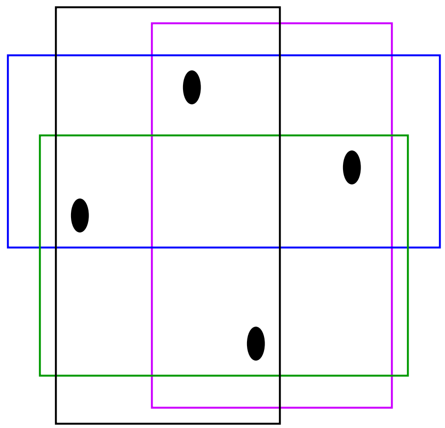


  - An example of three points (and a labeling) that cannot be shattered

Example II

- $\mathbf{x} \in \mathbb{R}^2, \mathcal{H} =$ Axis parallel rectangles
  - There exist four points that can be shattered
  - No set of five points can be shattered

$$\mathrm{VC}(\mathcal{H}) = 4$$



- Hypotheses consistent with all ways of labeling three positive;
- Check that there hypothesis for all ways of labeling one, two or four points positive

Example III

- A lookup table has infinite VC dimension!

no error in **training** no generalization

some error in **training** some generalization

- A hypothesis space with low VC dimension

Comments

- VC dimension is distribution-free; it is independent of the probability distribution from which the instances are drawn
- In this sense, it gives us a worse case complexity (pessimistic)
  - In real life, the world is smoothly changing, instances close by most of the time have the same labels, no worry about all possible labelings
- However, this is still useful for providing bounds, such as the sample complexity of a hypothesis class.
- In general, we will see that there is a connection between the VC dimension (which we would like to minimize) and the error on the training set (empirical risk)

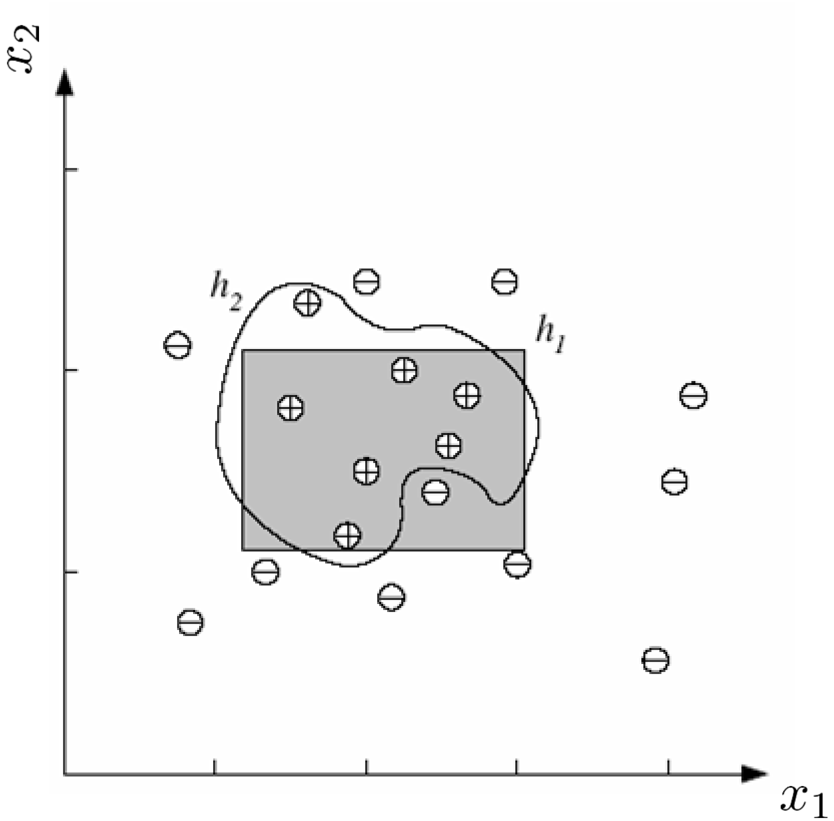
Handling Noise and Multiple Classes

— Supervised Learning

Noise

- Noise: unwanted anomaly in the data
- Another reason we can’t always have a perfect hypothesis
  - error in sensor readings for input
  - teacher noise: error in labeling the data
  - additional attributes which we have not taken into account. These are called **hidden** or **latent** because they are unobserved.

When there is noise…



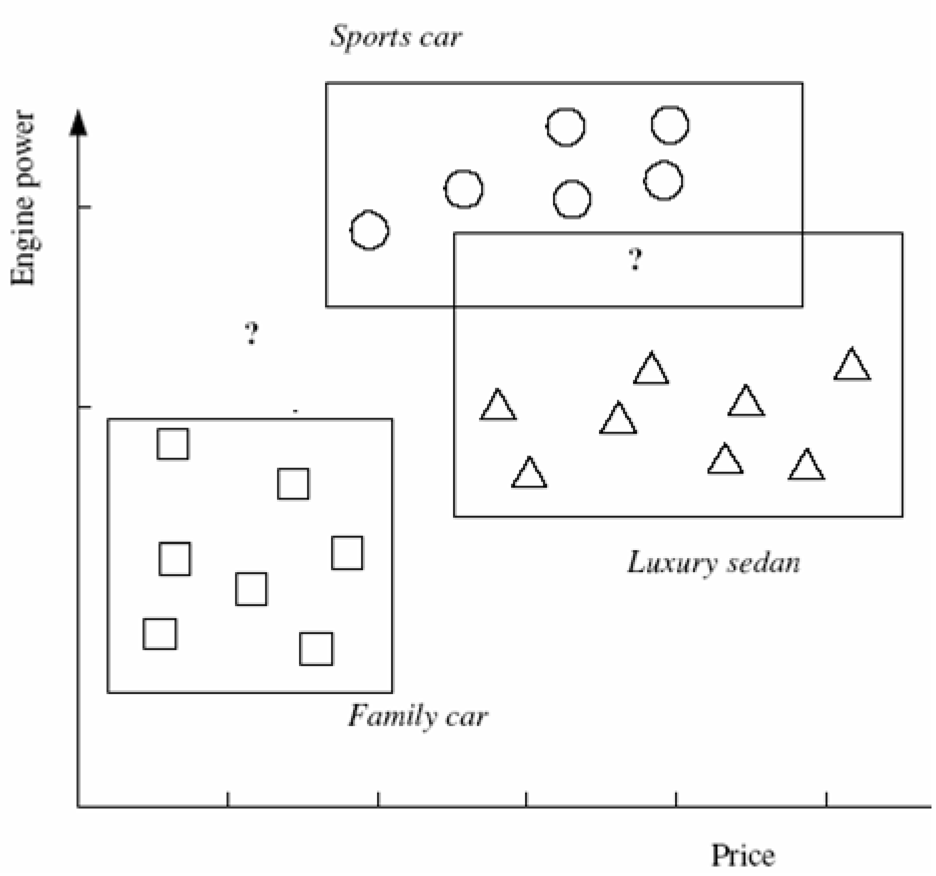
- There may not have a simple boundary between the positive and negative instances
- Zero (training) misclassification error may not be possible

Something about Simple Models

- Easier to classify a new instance
- Easier to explain
- Fewer parameters, means it is easier to train. The sample complexity is lower.
- Lower variance. A small change in the training samples will not result in a wildly different hypothesis
- High bias. A simple model makes strong assumptions about the domain; great if we're right, a disaster if we are wrong.

  optimality?: *min* (variance + bias)
- May have better generalization performance, especially if there is noise.
- Occam's razor: simpler explanations are more plausible

Learning Multiple Classes



- K-class classification

$\Rightarrow$ K two-class problems (one against all)

$\Rightarrow$ could introduce *doubt*

$\Rightarrow$ could have unbalance data

Regression Problems

— Supervised Learning

Regression

- Supervised learning where the output is not a classification (e.g. 0/1, true/false, yes/no), but the output is a real number.

$$\mathcal{D} = \{(\mathbf{x}^{(i)}, y_i), i = 1, \ldots, m, y_i \in \mathbb{R}\}$$

Interpolation

- Assuming no noise, we want to find a function $f(\mathbf{x})$ that passes through these points such that $y_i = f(\mathbf{x}^{(i)})$
  - Polynomial interpolation, given m points, we can find the $(m-1)$st degree polynomial which can predict the output for any $\mathbf{x}$
  - Example: time series prediction, given data up to the present, predict future data (extrapolation)

Regression

- Suppose that the true function is f

  $y_i = f(\mathbf{x}^{(i)}) + \epsilon_i$, where ϵ_i is random noise

- Suppose that we learn $h(\mathbf{x})$ as our model. The empirical error on the training set is

$$E(h \mid \mathcal{D}) = \frac{1}{m} \sum_{i=1}^{m} L(h(\mathbf{x}^{(i)}), y_i)$$

$\Rightarrow$ Because y_i and $h(\mathbf{x}^{(i)})$ are numeric, it makes sense for L to be the distance between them.

$\Rightarrow$ Common distance measures:

  - mean squared error

$$E = \frac{1}{m} \sum_{i=1}^{m} \left(y_i - h(\mathbf{x}^{(i)}) \right)^2$$

  - absolute value of difference
  - etc.

Linear Regression

- Assume $h(\mathbf{x})$ is linear, with $\mathbf{x} = (x_1, x_2, \ldots, x_n)$

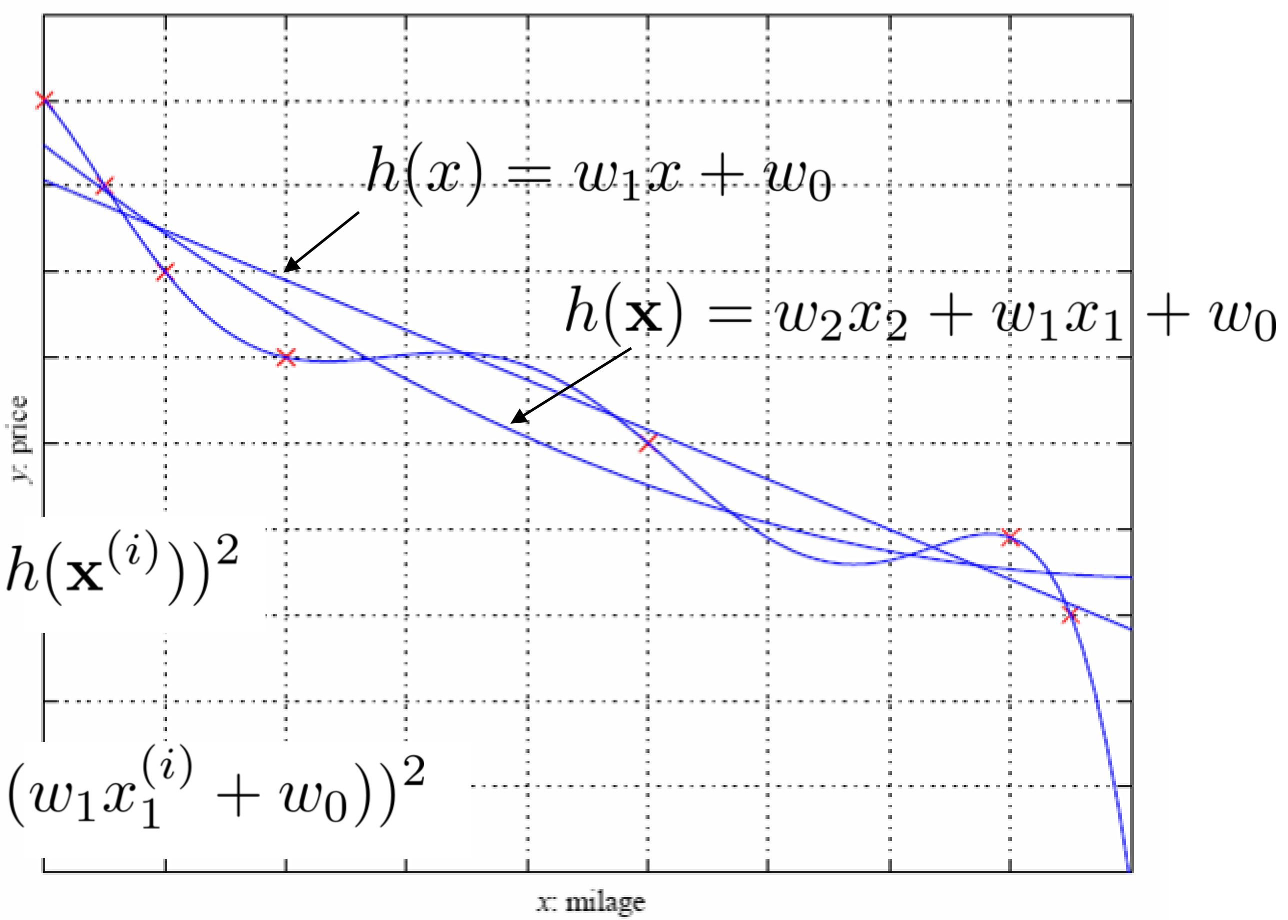
$$h(\mathbf{x}) = w_0 + w_1 x_1 + \cdots w_n x_n = w_0 + \sum_{j=1}^{n} w_j x_j$$

and we want to minimize the mean squared error

$$E = \frac{1}{m} \sum_{i=1}^{m} \left(y_i - h(\mathbf{x}^{(i)}) \right)^2 = \frac{1}{m} \sum_{i=1}^{m} \left(y_i - \left(w_0 + \sum_{j=1}^{n} w_j x_j^{(i)} \right) \right)^2$$

- We can solve this for the w_j that minimizes the error

With Different Complexity



$$E = \frac{1}{m}\sum_{i=1}^{m}(y_i - h(\mathbf{x}^{(i)}))^2$$

$$E = \frac{1}{m}\sum_{i=1}^{m}(y_i - (w_1 x_1^{(i)} + w_0))^2$$

Model Selection

- Learning problem is ill-posed
- Need **inductive bias**
  - Assuming a hypothesis class
  - Example: sports car problem, assuming most specific rectangle
  - But different hypothesis classes will have different capacities
    - Higher capacity, better able to fit the data
    - But goal is not to fit the data, it's to generalize
    - How do we measure? **cross-validation**: Split data into training and validation set; use training set to find hypothesis and validation set to test generalization. With enough data, the hypothesis that is most accurate on validation set is the best.
    - Choosing the right bias: **model selection**

Underfitting and Overfitting

- Matching the complexity of the hypothesis with the complexity of the target function
  - if the hypothesis is less complex than the function, we have **underfitting**. In this case, if we increase the complexity of the model, we will reduce both training error and validation error.
  - if the hypothesis is too complex, we may have **overfitting**. In this case, the validation error may go up even the training error goes down. For example, we fit the noise, rather than the target function.

Triple Trade-offs

- T. G. Dietterich. “Machine Learning”. In *Nature Encyclopedia of Cognitive Science*. London: Macmillan, 2003.

- Three trade-off factors:
  - complexity of the hypothesis (capacity of the hypothesis class)
  - amount of training data
  - generalization error on new examples
  - training data ↑, the generalization error ↓
  - complexity of model ↑, the generalization error ↓ first and then starts to ↑
  - validation is the answer!