Machine Learning 2007: Lecture 3

Instructor: Tim van Erven (Tim.van.Erven@cwi.nl)
Website: www.cwi.nl/~erven/teaching/0708/ml/

September 20, 2007

Overview

- **Organisational Matters**
- Hypothesis Spaces
- Method: Least Squares Linear Regression
- Being Informal about Feature Vectors
- Method: LIST-THEN-ELIMINATE for Concept Learning
  - A Biased Hypothesis Space
  - An Unbiased Hypothesis Space?

Organisational Matters

Course Organisation:

- Intermediate exam: October 25, 11.00 – 13.00 in 04A05.
- Biweekly exercises

This Lecture versus Mitchell

- All of it is in the book (Chapters 1 and 2), except for “Being Informal About Feature Vectors”.
- The presentation is different though: We recognise methods from Mitchell as methods to deal with regression and classification.

Overview

- Organisational Matters
- **Hypothesis Spaces**
- Method: Least Squares Linear Regression
- Being Informal about Feature Vectors
- Method: LIST-THEN-ELIMINATE for Concept Learning
  - A Biased Hypothesis Space
  - An Unbiased Hypothesis Space?

Reminder of Machine Learning Categories

Prediction: Given data $D = y_1, \ldots, y_n$, predict how the sequence continues with y_{n+1}.

Regression: Given data $D = \begin{pmatrix} y_1 \\ \mathbf{x}_1 \end{pmatrix}, \ldots, \begin{pmatrix} y_n \\ \mathbf{x}_n \end{pmatrix}$, learn to predict the value of the label y for any new feature vector $\mathbf{x}$. Typically y can take infinitely many values. Acceptable if your prediction is close to the correct y.

Classification: Given data $D = \begin{pmatrix} y_1 \\ \mathbf{x}_1 \end{pmatrix}, \ldots, \begin{pmatrix} y_n \\ \mathbf{x}_n \end{pmatrix}$, learn to predict the class label y for any new feature vector $\mathbf{x}$. Only finitely many categories. Your prediction is either correct or wrong.

Hypotheses and Hypothesis Spaces

Definition of a Hypothesis:

A hypothesis h is a candidate description of the regularity or patterns in your data.

- Prediction example: $y_{n+1} = h(y_1, \ldots, y_n) = y_{n-1} + y_n$
- Regression example: $y = h(\mathbf{x}) = 5x_1$
- Classification example: $y = h(\mathbf{x}) = \begin{cases} +1 & \text{if } 3x_1 - 20 > 0; \\ -1 & \text{otherwise.} \end{cases}$

Definition of a Hypothesis Space:

A hypothesis space $\mathcal{H}$ is the set $\{h\}$ of hypotheses that are being considered.

- Regression example: $\{h_a(\mathbf{x}) = a \cdot x_1 | a \in \mathbb{R}\}$

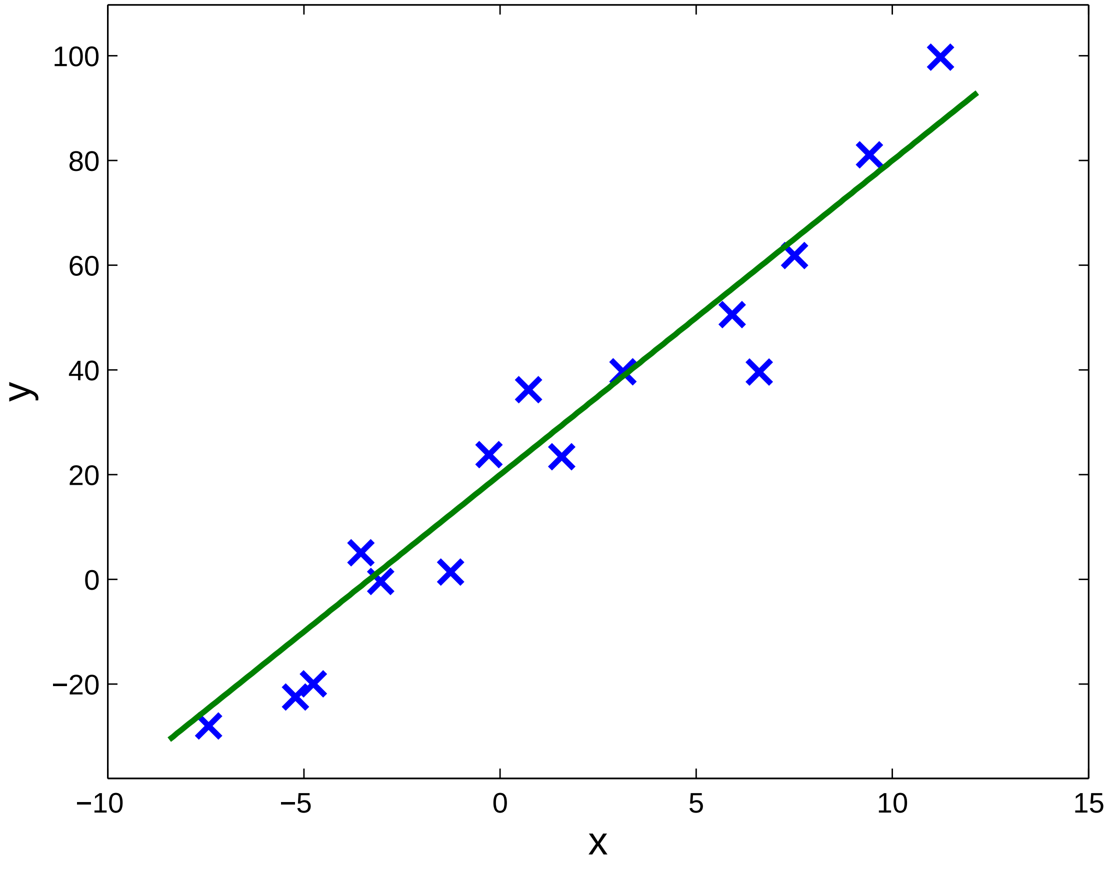
Overview

- Organisational Matters
- Hypothesis Spaces
- **Method: Least Squares Linear Regression**
- Being Informal about Feature Vectors
- Method: LIST-THEN-ELIMINATE for Concept Learning
    - A Biased Hypothesis Space
    - An Unbiased Hypothesis Space?

Linear Regression

Linear Regression:

In linear regression the goal is to select a linear hypothesis that best captures the regularity in the data.



Hypothesis Space of Linear Hypotheses

Linear Function:

$$y = h_{\mathbf{w}}(\mathbf{x}) = w_0 + w_1 x_1 + \ldots + w_d x_d$$

- $\mathbf{x} = (x_1, \ldots, x_d)^\top$ is a d-dimensional feature vector.
- $\mathbf{w} = (w_0, w_1, \ldots, w_d)^\top$ are called the **weights**.

Examples:

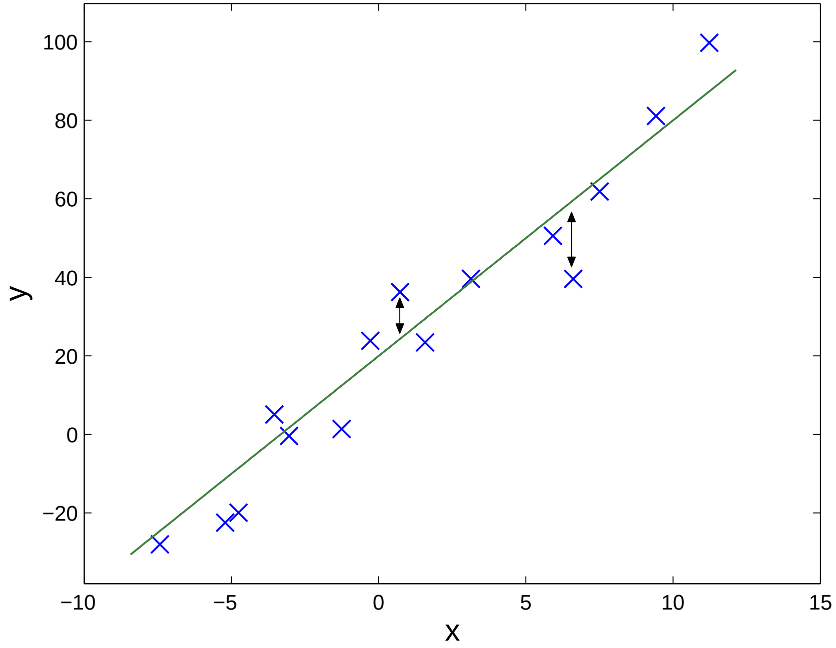
$$h_{\mathbf{w}}(\mathbf{x}) = 2 + 9x_1 \qquad (w_0 = 2, w_1 = 9)$$

$$h_{\mathbf{w}}(\mathbf{x}) = 3 + 16x_1 - 2x_3 \qquad (w_0 = 3, w_1 = 16, w_2 = 0, w_3 = -2)$$

Hypothesis Space of All Linear Hypotheses:

$$\mathcal{H} = \{h_{\mathbf{w}} \mid \mathbf{w} \in \mathbb{R}^{d+1}\}.$$

Example: A Linear Function with Noise



Data generated by a linear function

$$y = 6x + 20 + \epsilon,$$

where ϵ is noise with distribution $\mathcal{N}(0, 10)$. Can we recover this function from the data alone?

Determining Weights from the Data

Squared Error:

For given $\mathbf{w}$, we may evaluate the squared error of $h_{\mathbf{w}}$ on a single data-item $\begin{pmatrix} y_i \\ \mathbf{x}_i \end{pmatrix}$:

$$\text{Squared Error} = (y_i - h_{\mathbf{w}}(\mathbf{x}_i))^2$$

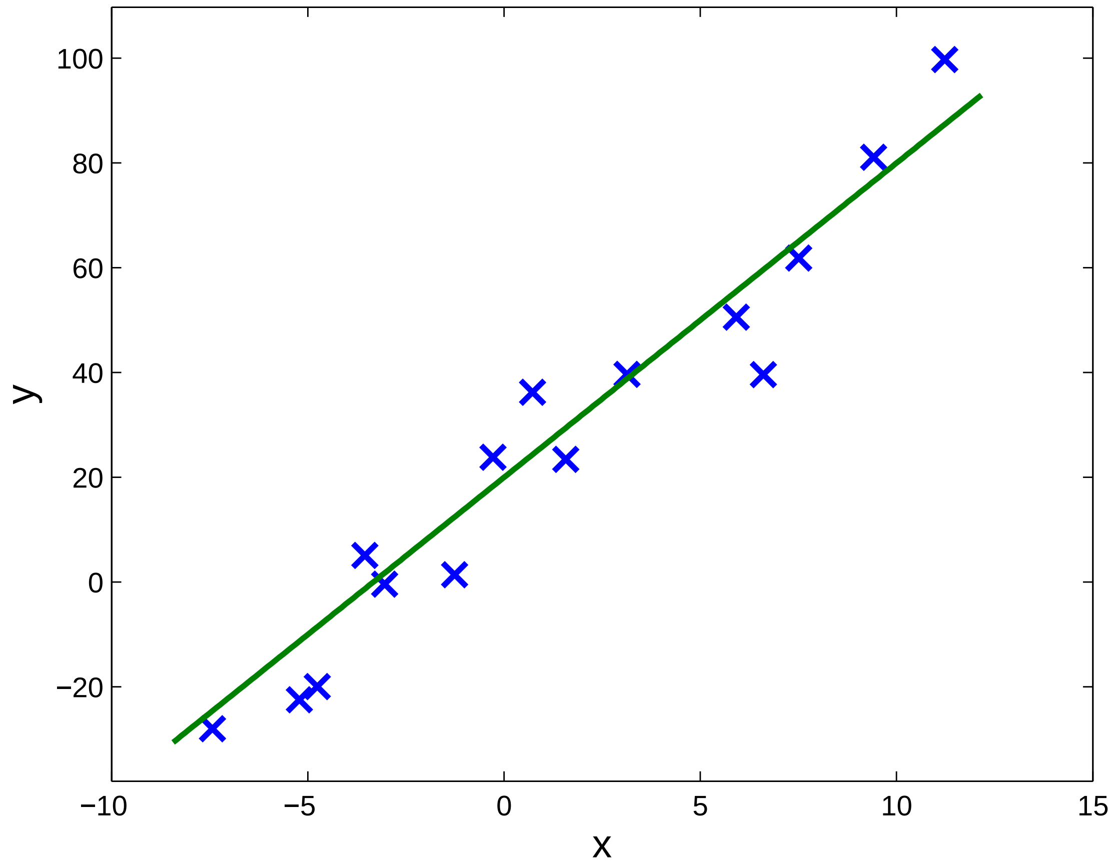
Least Squares Linear Regression:

Given data $D = \begin{pmatrix} y_1 \\ \mathbf{x}_1 \end{pmatrix}, \ldots, \begin{pmatrix} y_n \\ \mathbf{x}_n \end{pmatrix}$, select $\mathbf{w}$ to minimize the sum of squared errors $\text{SSE}(D)$ on all data:

$$\min_{\mathbf{w}} \text{SSE}(D) = \min_{\mathbf{w}} \sum_{i=1}^{n} (y_i - h_{\mathbf{w}}(\mathbf{x}_i))^2.$$

Linear Regression Example

The previous example again:

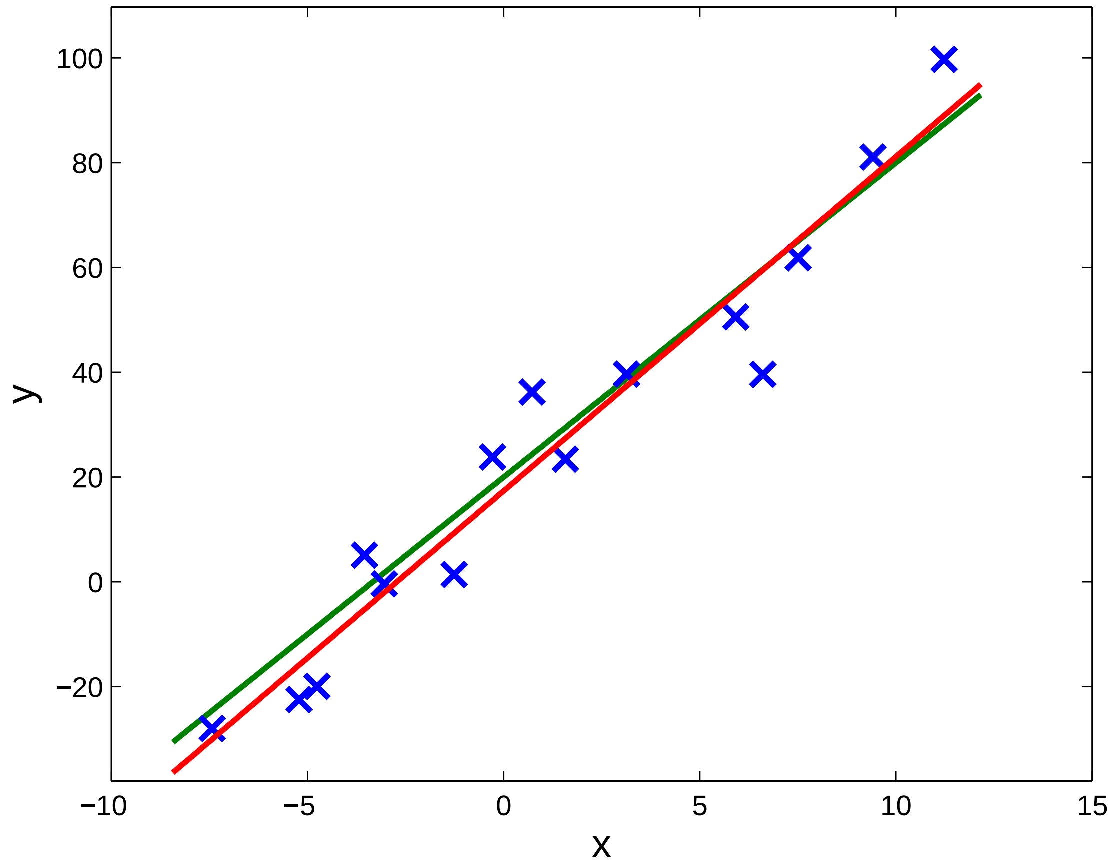


Original Function

$$y = 6x + 20 + \epsilon$$

Linear Regression Example

The previous example again:



Original Function	Least Squares
$y = 6x + 20 + \epsilon$	$y = 6.38x + 17.37$

Least Squares Linear Regression:

- Only looks for linear patterns in the data.
  - For example, it cannot discover $y = x_1^2$ even if it gets an infinite amount of data.
- Minimizes the sum of squared errors.
  - Why not something else, like for example the sum of absolute errors?

$$\min_{\mathbf{w}} \sum_{i=1}^{n} |y_i - h_{\mathbf{w}}(\mathbf{x}_i)|$$

Overview

- Organisational Matters
- Hypothesis Spaces
- Method: Least Squares Linear Regression
- **Being Informal about Feature Vectors**
- Method: List-Then-Eliminate for Concept Learning
  - A Biased Hypothesis Space
  - An Unbiased Hypothesis Space?

EnjoySport Representation 1

Numbering Attribute Values:

Attribute	Sky			AirTemp		EnjoySport	
Value	Sunny	Cloudy	Rainy	Warm	Cold	No	Yes
Encoding	1	2	3	1	2	1	2

EnjoySport Representation 1

Numbering Attribute Values:

Attribute	Sky			AirTemp		EnjoySport	
Value	Sunny	Cloudy	Rainy	Warm	Cold	No	Yes
Encoding	1	2	3	1	2	1	2

Example:

Sky, AirTemp	**EnjoySport**	**Representation**
Sunny, Warm	Yes	$\mathbf{x} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, y = 2$
Rainy, Cold	No	$\mathbf{x} = \begin{pmatrix} 3 \\ 2 \end{pmatrix}, y = 1$
Sunny, Cold	Yes	$\mathbf{x} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, y = 2$

- The difference between feature vectors has no clear meaning. For example $\begin{pmatrix} 3 \\ 2 \end{pmatrix} - \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$.

EnjoySport Representation 2

Another Way to Do It:

Attribute	Sky			AirTemp		EnjoySport	
Value	Sunny	Cloudy	Rainy	Warm	Cold	No	Yes
Encoding	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\\0\end{pmatrix}$	$\begin{pmatrix}0\\0\\1\end{pmatrix}$	$\begin{pmatrix}1\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\end{pmatrix}$	1	2

EnjoySport Representation 2

Another Way to Do It:

Attribute	Sky			AirTemp		EnjoySport	
Value	Sunny	Cloudy	Rainy	Warm	Cold	No	Yes
Encoding	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\\0\end{pmatrix}$	$\begin{pmatrix}0\\0\\1\end{pmatrix}$	$\begin{pmatrix}1\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\end{pmatrix}$	1	2

Example (table is on its side to fit vectors):

Sky, AirTemp	Sunny, Warm	Rainy, Cold	Sunny, Cold
EnjoySport	Yes	No	Yes
Representation	$\mathbf{x} = \begin{pmatrix}1\\0\\0\\1\\0\end{pmatrix}, y = 2$	$\mathbf{x} = \begin{pmatrix}0\\0\\1\\0\\1\end{pmatrix}, y = 1$	$\mathbf{x} = \begin{pmatrix}1\\0\\0\\0\\1\end{pmatrix}, y = 2$

- The number of non-zero entries in the difference between two vectors is twice the number of attributes that differ.

Being Informal about Feature Vectors

- (Feature) vectors $\mathbf{x}$ and labels y contain numbers.
- But sometimes it will be convenient to be **informal** (mathematically imprecise):

Formal		**Informal**
$\mathbf{x} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$	$\Leftrightarrow$	$\mathbf{x} = \begin{pmatrix} \text{Sunny} \\ \text{Warm} \end{pmatrix}$
$y = 2$	$\Leftrightarrow$	$y =$ Yes

- Why?
  - Reason 1: Don't care about details of representation.
  - Reason 2: Emphasize meaning of features and labels.
- Don't forget what's really going on!

Overview

- Organisational Matters
- Hypothesis Spaces
- Method: Least Squares Linear Regression
- Being Informal about Feature Vectors
- **Method: LIST-THEN-ELIMINATE for Concept Learning**
  - A Biased Hypothesis Space
  - An Unbiased Hypothesis Space?

Hypothesis Space for EnjoySport

A hypothesis h is specified by a list of constraints on the attributes: Sky, AirTemp, Humidity, Wind, Water, Forecast.

$$h(\mathbf{x}) = \begin{cases} \text{yes} & \text{if } \mathbf{x} \text{ satisfies all constraints,} \\ \text{no} & \text{otherwise.} \end{cases}$$

Hypothesis Space for EnjoySport

A hypothesis h is specified by a list of constraints on the attributes: Sky, AirTemp, Humidity, Wind, Water, Forecast.

$$h(\mathbf{x}) = \begin{cases} \text{yes} & \text{if } \mathbf{x} \text{ satisfies all constraints,} \\ \text{no} & \text{otherwise.} \end{cases}$$

List of constraints looks like: $\langle ?, \text{Cold}, \text{High}, ?, ?, ? \rangle$

Attribute	Description
$?$	Any value is acceptable for the attribute.
$\emptyset$	No value is acceptable.
Warm	Single required value for attribute (e.g. Warm)

Hypothesis Space for EnjoySport

A hypothesis h is specified by a list of constraints on the attributes: Sky, AirTemp, Humidity, Wind, Water, Forecast.

$$h(\mathbf{x}) = \begin{cases} \text{yes} & \text{if } \mathbf{x} \text{ satisfies all constraints,} \\ \text{no} & \text{otherwise.} \end{cases}$$

List of constraints looks like: $\langle ?, \text{Cold}, \text{High}, ?, ?, ? \rangle$

Attribute	Description
$?$	Any value is acceptable for the attribute.
$\emptyset$	No value is acceptable.
Warm	Single required value for attribute (e.g. Warm)

Hypothesis Space:

$$\mathcal{H} = \{h\} = \{\langle ?, ?, ?, ?, ?, ? \rangle, \langle \text{Sunny}, ?, ?, ?, ?, ? \rangle, \langle \text{Cloudy}, ?, ?, ?, ?, ? \rangle, \ldots, \langle \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset \rangle\}$$

LIST-THEN-ELIMINATE *Algorithm*

- Given: data $D = \begin{pmatrix} y_1 \\ \mathbf{x}_1 \end{pmatrix}, \ldots, \begin{pmatrix} y_n \\ \mathbf{x}_n \end{pmatrix}$.
- A hypothesis h is **consistent** with example $\begin{pmatrix} y_i \\ \mathbf{x}_i \end{pmatrix}$ if it assigns the right label to $\mathbf{x}_i$: $h(\mathbf{x}_i) = y_i$.
- LIST-THEN-ELIMINATE finds the set, VersionSpace, of all hypotheses that are consistent with the training data.

List-Then-Eliminate *Algorithm*

- Given: data $D = \begin{pmatrix} y_1 \\ \mathbf{x}_1 \end{pmatrix}, \ldots, \begin{pmatrix} y_n \\ \mathbf{x}_n \end{pmatrix}$.
- A hypothesis h is **consistent** with example $\begin{pmatrix} y_i \\ \mathbf{x}_i \end{pmatrix}$ if it assigns the right label to $\mathbf{x}_i$: $h(\mathbf{x}_i) = y_i$.
- List-Then-Eliminate finds the set, VersionSpace, of all hypotheses that are consistent with the training data.

List-Then-Eliminate Algorithm:

1: VersionSpace $\leftarrow \mathcal{H}$
2: **for** $i = 1, \ldots, n$ **do**
3: Remove from VersionSpace any h such that $h(\mathbf{x}_i) \neq y_i$.
4: **end for**
5: **return** VersionSpace
6:

List-Then-Eliminate *Example Run*

Simplified Hypothesis Space:

Suppose for the moment that $\mathcal{H} = \{\langle ?, ? \rangle, \langle \text{Sunny}, ? \rangle, \langle \emptyset, ? \rangle\}$.

Example Run:

	$\mathbf{x}_1 = \begin{pmatrix} \text{Sunny} \\ \text{Warm} \end{pmatrix}, y_1 = \text{Yes}$	$\mathbf{x}_2 = \begin{pmatrix} \text{Rainy} \\ \text{Cold} \end{pmatrix}, y_2 = \text{No}$
$\langle ?, ? \rangle$	$+$	$-$
$\langle \text{Sunny}, ? \rangle$	$+$	$+$
$\langle \emptyset, ? \rangle$	$-$	$+$

- $+ =$ consistent, $- =$ inconsistent

Resulting VersionSpace:

VersionSpace = $\{\langle \text{Sunny}, ? \rangle\}$

Classifying New Instances

LIST-THEN-ELIMINATE:

- Given: data $D = \begin{pmatrix} y_1 \\ \mathbf{x}_1 \end{pmatrix}, \ldots, \begin{pmatrix} y_n \\ \mathbf{x}_n \end{pmatrix}$.
- LIST-THEN-ELIMINATE finds the set, VersionSpace, of all hypotheses that are consistent with the training data.

Classifying New Instances:

- Suppose we get $\mathbf{x}_{n+1}$, how should we classify it?

Classifying New Instances

List-Then-Eliminate:

- Given: data $D = \begin{pmatrix} y_1 \\ \mathbf{x}_1 \end{pmatrix}, \ldots, \begin{pmatrix} y_n \\ \mathbf{x}_n \end{pmatrix}$.
- List-Then-Eliminate finds the set, VersionSpace, of all hypotheses that are consistent with the training data.

Classifying New Instances:

- Suppose we get $\mathbf{x}_{n+1}$, how should we classify it?
- If all hypotheses in VersionSpace agree on the label of $\mathbf{x}_{n+1}$, then it's easy; Otherwise we don't know:

$$y_{n+1} = \begin{cases} z & \text{if } h(\mathbf{x}_{n+1}) = z \text{ for all } h \in \text{VersionSpace}, \\ \text{don't know} & \text{otherwise.} \end{cases}$$

Inductive Bias and Practical Issues

Inductive Bias:

- Can only learn target concepts that are contained in the hypothesis space $\mathcal{H}$.
- Not robust if the target concept is not in $\mathcal{H}$.
- Sensitive to noise/errors in the training data: might accidentally remove the best hypothesis.
- Doesn’t have any preference between consistent hypotheses. (Strength or weakness?)

Inductive Bias and Practical Issues

Inductive Bias:

- Can only learn target concepts that are contained in the hypothesis space $\mathcal{H}$.
- Not robust if the target concept is not in $\mathcal{H}$.
- Sensitive to noise/errors in the training data: might accidentally remove the best hypothesis.
- Doesn't have any preference between consistent hypotheses. (Strength or weakness?)

Practical Issue:

- Uses too much memory (to store VersionSpace). The book discusses the CANDIDATE-ELIMINATION algorithm, which does the same thing using less memory.

Overview

- Organisational Matters
- Hypothesis Spaces
- Method: Least Squares Linear Regression
- Being Informal about Feature Vectors
- Method: LIST-THEN-ELIMINATE for Concept Learning
  - **A Biased Hypothesis Space**
  - An Unbiased Hypothesis Space?

Some Notation: The Sets $\mathcal{X}$ and $\mathcal{Y}$

$\mathcal{X}$ and $\mathcal{Y}$:

- $\mathcal{X} = \{\mathbf{x}\}$ is the set of all possible feature vectors.
- $\mathcal{Y} = \{y\}$ is the set of all possible labels.

Some Notation: The Sets $\mathcal{X}$ and $\mathcal{Y}$

$\mathcal{X}$ and $\mathcal{Y}$:

- $\mathcal{X} = \{\mathbf{x}\}$ is the set of all possible feature vectors.
- $\mathcal{Y} = \{y\}$ is the set of all possible labels.

The Number of Elements in a Set:

For any set A, we let $|A|$ denote the number of elements in A. For example, $|\{a, b, c\}| = 3$.

Some Notation: The Sets $\mathcal{X}$ and $\mathcal{Y}$

$\mathcal{X}$ and $\mathcal{Y}$:

- $\mathcal{X} = \{\mathbf{x}\}$ is the set of all possible feature vectors.
- $\mathcal{Y} = \{y\}$ is the set of all possible labels.

The Number of Elements in a Set:

For any set A, we let $|A|$ denote the number of elements in A. For example, $|\{a, b, c\}| = 3$.

EnjoySport Example:

Attribute	Sky	AirTemp	Humidity	Wind	Water	Forecast
# Values	3	2	2	2	2	2

- The number of possible feature vectors:
- The number of possible labels:

Some Notation: The Sets $\mathcal{X}$ and $\mathcal{Y}$

$\mathcal{X}$ and $\mathcal{Y}$:

- $\mathcal{X} = \{\mathbf{x}\}$ is the set of all possible feature vectors.
- $\mathcal{Y} = \{y\}$ is the set of all possible labels.

The Number of Elements in a Set:

For any set A, we let $|A|$ denote the number of elements in A. For example, $|\{a, b, c\}| = 3$.

EnjoySport Example:

Attribute	Sky	AirTemp	Humidity	Wind	Water	Forecast
# Values	3	2	2	2	2	2

- The number of possible feature vectors: $|\mathcal{X}| = 3 \cdot 2^5 = 96$
- The number of possible labels:

Some Notation: The Sets $\mathcal{X}$ and $\mathcal{Y}$

$\mathcal{X}$ and $\mathcal{Y}$:

- $\mathcal{X} = \{\mathbf{x}\}$ is the set of all possible feature vectors.
- $\mathcal{Y} = \{y\}$ is the set of all possible labels.

The Number of Elements in a Set:

For any set A, we let $|A|$ denote the number of elements in A. For example, $|\{a, b, c\}| = 3$.

EnjoySport Example:

Attribute	Sky	AirTemp	Humidity	Wind	Water	Forecast
# Values	3	2	2	2	2	2

- The number of possible feature vectors: $|\mathcal{X}| = 3 \cdot 2^5 = 96$
- The number of possible labels: $|\mathcal{Y}| = 2$

Counting Hypotheses

LIST-THEN-ELIMINATE:

- Syntactically distinct hypotheses: $5 \cdot 4^5 = 5120$
- But $\langle \text{Cloudy}, ?, ?, \emptyset, ?, \text{Change} \rangle = \langle \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset \rangle$ and the same holds for any hypothesis containing at least one $\emptyset$.
- Semantically distinct hypotheses: $1 + 4 \cdot 3^5 = 973$

Counting Hypotheses

LIST-THEN-ELIMINATE:

- Syntactically distinct hypotheses: $5 \cdot 4^5 = 5120$
- But $\langle$Cloudy$, ?, ?, \emptyset, ?,$ Change$\rangle = \langle\emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset\rangle$ and the same holds for any hypothesis containing at least one $\emptyset$.
- Semantically distinct hypotheses: $1 + 4 \cdot 3^5 = 973$

All possible hypotheses:

- A hypothesis h can be any function from $\mathcal{X}$ to $\mathcal{Y}$.
- To each feature vector in $\mathcal{X}$ it might assign any label from $\mathcal{Y}$.
- Semantically distinct hypotheses: $|\mathcal{Y}|^{|\mathcal{X}|} = 2^{96} \approx 10^{29}$

Conclusion:

LIST-THEN-ELIMINATE has a very strong **representation bias**.

Overview

- Organisational Matters
- Hypothesis Spaces
- Method: Least Squares Linear Regression
- Being Informal about Feature Vectors
- Method: LIST-THEN-ELIMINATE for Concept Learning
  - A Biased Hypothesis Space
  - **An Unbiased Hypothesis Space?**

An Unbiased Hypothesis Space

All Possible Hypotheses:

- Why not take all possible hypotheses as a hypothesis space for LIST-THEN-ELIMINATE?

$$\mathcal{H} = \{h | h \text{ is a function from } \mathcal{X} \text{ to } \mathcal{Y}\}$$

An Unbiased Hypothesis Space

All Possible Hypotheses:

- Why not take all possible hypotheses as a hypothesis space for List-Then-Eliminate?

$$\mathcal{H} = \{h | h \text{ is a function from } \mathcal{X} \text{ to } \mathcal{Y}\}$$

List-Then-Eliminate:

- Given: data $D = \begin{pmatrix} y_1 \\ \mathbf{x}_1 \end{pmatrix}, \ldots, \begin{pmatrix} y_n \\ \mathbf{x}_n \end{pmatrix}$.
- What happens if we try to classify a new feature vector $\mathbf{x}_{n+1}$?

Classifying New Instances

- For any hypothesis $h \in \mathcal{H}$, there exists a $h' \in \mathcal{H}$ such that

$$h(\mathbf{x}) \neq h'(\mathbf{x}) \quad \text{if } \mathbf{x} = \mathbf{x_{n+1}},$$
$$h(\mathbf{x}) = h'(\mathbf{x}) \quad \text{for any other } \mathbf{x}.$$

Classifying New Instances

- For any hypothesis $h \in \mathcal{H}$, there exists a $h' \in \mathcal{H}$ such that

$$h(\mathbf{x}) \neq h'(\mathbf{x}) \quad \text{if } \mathbf{x} = \mathbf{x_{n+1}},$$
$$h(\mathbf{x}) = h'(\mathbf{x}) \quad \text{for any other } \mathbf{x}.$$

Consequence:

- Suppose $\mathbf{x}_{n+1}$ does not occur in D.
- Then for every $h \in$ VersionSpace, there exists an alternative $h' \in$ VersionSpace that disagrees on the label of $\mathbf{x}_{n+1}$:

$$h(\mathbf{x}_{n+1}) \neq h'(\mathbf{x}_{n+1})$$

Conclusion:

In an unbiased hypothesis space, the LIST-THEN-ELIMINATE algorithm **cannot generalise** at all. Bias is unavoidable!

Summary

- Hypothesis h: candidate description of regularity in the data
- Hypothesis space $\mathcal{H}$: set of hypotheses being considered

Summary

- Hypothesis h: candidate description of regularity in the data
- Hypothesis space $\mathcal{H}$: set of hypotheses being considered
- Least squares linear regression:
  - Method for regression
  - Selects the linear hypothesis that minimizes the sum of squared errors on the data.

Summary

- Hypothesis h: candidate description of regularity in the data
- Hypothesis space $\mathcal{H}$: set of hypotheses being considered
- Least squares linear regression:
  - Method for regression
  - Selects the linear hypothesis that minimizes the sum of squared errors on the data.
- The LIST-THEN-ELIMINATE algorithm:
  - Method for classification/concept learning
  - Finds the set, VersionSpace, of hypotheses in $\mathcal{H}$ that are consistent with the data.
  - With $\mathcal{H}$ containing a list of constraints on attributes, it has a strong representation bias.
  - With $\mathcal{H}$ containing all possible hypotheses it cannot generalise: bias is unavoidable!