Chapter ML:II

II. Machine Learning Basics

- On Data
- Regression
- Concept Learning: Search in Hypothesis Space
- Concept Learning: Search in Version Space
- Measuring Performance

On Data [Tan et al. 2005]

- An object $o \in O$ is described by a set of attributes.
  An object is also known as record, point, case, sample, entity, or instance.

- An attribute A is a property of an object.
  An attribute is also known as variable, field, characteristic, or feature.

- A measurement scale is a system (often a convention) of assigning a numerical or symbolic value to an attribute of an object.

Attributes

Objects

ID	Check	Status	Income	Risk
1	+	single	125 000	No
2	-	married	100 000	No
3	-	single	70 000	No
4	+	married	120 000	No
5	-	divorced	95 000	Yes
6	-	married	60 000	No
7	+	divorced	220 000	No
8	-	single	85 000	Yes
9	-	married	75 000	No
10	-	single	90 000	Yes

On Data [Tan et al. 2005]

- An object $o \in O$ is described by a set of attributes.
  An object is also known as record, point, case, sample, entity, or instance.
- An attribute A is a property of an object.
  An attribute is also known as variable, field, characteristic, or feature.
- A measurement scale is a system (often a convention) of assigning a numerical or symbolic value to an attribute of an object.

Attributes (columns) / Objects (rows)

ID	Check	Status	Income	Risk
1	+	single	125 000	No
2	-	married	100 000	No
3	-	single	70 000	No
4	+	married	120 000	No
5	-	divorced	95 000	Yes
6	-	married	60 000	No
7	+	divorced	220 000	No
8	-	single	85 000	Yes
9	-	married	75 000	No
10	-	single	90 000	Yes

On Data [Tan et al. 2005]

- An object $o \in O$ is described by a set of attributes.
  An object is also known as record, point, case, sample, entity, or instance.

- An attribute A is a property of an object.
  An attribute is also known as variable, field, characteristic, or feature.

- A measurement scale is a system (often a convention) of assigning a numerical or symbolic value to an attribute of an object.

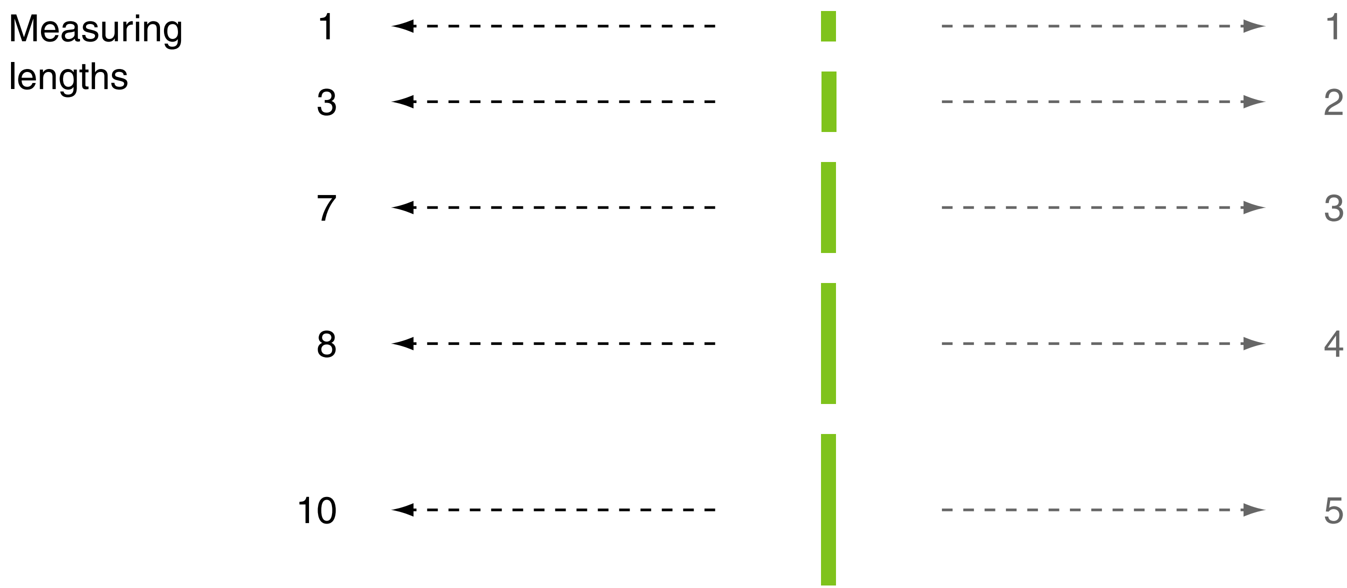
Attributes (columns) / Objects (rows)

ID	Check	Status	Income	Risk
1	+	single	125 000	No
2	-	married	100 000	No
3	-	single	70 000	No
4	+	married	120 000	No
5	-	divorced	95 000	Yes
6	-	married	60 000	No
7	+	divorced	220 000	No
8	-	single	85 000	Yes
9	-	married	75 000	No
10	-	single	90 000	Yes

On Data [Tan et al. 2005]

- Attribute values may vary from one object to another or one time to another.
- The same attribute can be mapped to different attribute values.
  Example: height can be measured in feet or meters.
- Different attributes can be mapped to the same set of values.
  Example: attribute values for ID and age are integers.

The way an attribute is measured may not match the attribute's properties:



On Data [Tan et al. 2005]

Types of Attributes

Type		Comparison	Statistics	Examples
categorical (qualitative)	nominal	values are names, only information to distinguish objects $= \quad \neq$	mode, entropy, contingency, correlation, χ^2 test	zip codes, employee IDs, eye color, gender: {male, female}

On Data [Tan et al. 2005]

Types of Attributes

Type		Comparison	Statistics	Examples
categorical (qualitative)	nominal	values are names, only information to distinguish objects $= \quad \neq$	mode, entropy, contingency, correlation, χ^2 test	zip codes, employee IDs, eye color, gender: {male, female}
	ordinal	enough information to order objects $< \quad > \quad \leq \quad \geq$	median, percentiles, rank correlation, run tests, sign tests	hardness of minerals, grades, street numbers, quality: {good, better, best}

On Data [Tan et al. 2005]

Types of Attributes

Type		Comparison	Statistics	Examples
categorical (qualitative)	nominal	values are names, only information to distinguish objects $= \quad \neq$	mode, entropy, contingency, correlation, χ^2 test	zip codes, employee IDs, eye color, gender: {male, female}
	ordinal	enough information to order objects $< \quad > \quad \leq \quad \geq$	median, percentiles, rank correlation, run tests, sign tests	hardness of minerals, grades, street numbers, quality: {good, better, best}
numeric (quantitative)	interval	differences are meaningful, a unit of measurement exists $+ \quad -$	mean, standard deviation, Pearson's correlation, t-test, F-test	calendar dates, temperature in Celsius, temperature in Fahrenheit

On Data [Tan et al. 2005]

Types of Attributes

Type		Comparison	Statistics	Examples
categorical (qualitative)	nominal	values are names, only information to distinguish objects $= \quad \neq$	mode, entropy, contingency, correlation, χ^2 test	zip codes, employee IDs, eye color, gender: {male, female}
	ordinal	enough information to order objects $< \quad > \quad \leq \quad \geq$	median, percentiles, rank correlation, run tests, sign tests	hardness of minerals, grades, street numbers, quality: {good, better, best}
numeric (quantitative)	interval	differences are meaningful, a unit of measurement exists $+ \quad -$	mean, standard deviation, Pearson's correlation, t-test, F-test	calendar dates, temperature in Celsius, temperature in Fahrenheit
	ratio	differences and ratios are meaningful $* \quad /$	geometric mean, harmonic mean, percent variation	temperature in Kelvin, monetary quantities, counts, age, length, electrical current

On Data [Tan et al. 2005]

Types of Attributes

Type		Permissible transformation	Comment
categorical (qualitative)	nominal	any one-to-one mapping, permutation of values	A reassignment of employee ID numbers will not make any difference.

On Data [Tan et al. 2005]

Types of Attributes

Type		Permissible transformation	Comment
categorical (qualitative)	nominal	any one-to-one mapping, permutation of values	A reassignment of employee ID numbers will not make any difference.
	ordinal	any order-preserving change of values: $x \rightarrow f(x)$, where f is a monotonic	An attribute encompassing the notion of "{good, better, best}" can be represented equally well by the values $\{1, 2, 3\}$.

On Data [Tan et al. 2005]

Types of Attributes

Type		Permissible transformation	Comment
categorical (qualitative)	nominal	any one-to-one mapping, permutation of values	A reassignment of employee ID numbers will not make any difference.
	ordinal	any order-preserving change of values: $x \rightarrow f(x)$, where f is a monotonic	An attribute encompassing the notion of “{good, better, best}” can be represented equally well by the values $\{1, 2, 3\}$.
numeric (quantitative)	interval	$x \rightarrow a \cdot x + b$, where a and b are constants	Thus, the Fahrenheit and Celsius temperature scales differ in terms of where their zero value is and the size of a unit (degree).

On Data [Tan et al. 2005]

Types of Attributes

Type		Permissible transformation	Comment
categorical (qualitative)	nominal	any one-to-one mapping, permutation of values	A reassignment of employee ID numbers will not make any difference.
	ordinal	any order-preserving change of values: $x \rightarrow f(x)$, where f is a monotonic	An attribute encompassing the notion of "{good, better, best}" can be represented equally well by the values $\{1, 2, 3\}$.
numeric (quantitative)	interval	$x \rightarrow a \cdot x + b$, where a and b are constants	Thus, the Fahrenheit and Celsius temperature scales differ in terms of where their zero value is and the size of a unit (degree).
	ratio	$x \rightarrow a \cdot x$, where a is a constant	Length can be measured in meters or feet.

Remarks:

- Identifying, considering, and measuring an attribute A of an object O is the heart of model formation and always goes along with a sort of abstraction. Formally, this abstraction is operationalized by a model formation function $\alpha : O \rightarrow X$. [ML Introduction]

- The terms "attribute" and "feature" can be used synonymously. However, a slight distinction is the following: attributes are often associated with objects, O, while features usually designate the dimensions of the feature space, X.

- The type of an attribute is also referred to as the type of a *measurement scale* or *level of measurement*.

- We call a transformation of an attribute *permissible* if its meaning is unchanged after the transformation.

- Distinguish between *discrete* attributes and *continuous* attributes. The former can only take a finite or countably infinite set of values, the latter can be measured in infinitely small units. Be careful when deriving from this distinction an attribute's type.

- We will encode attributes of interval type or ratio type by real numbers. Note that attributes of nominal type and ordinal type can also be encoded by real numbers.

- Particular learning methods require particular attribute types.

On Data [Tan et al. 2005]

Types of Data Sets

Data sets may not be a homogeneous collection of objects but come along with differently intricate characteristics:

1. Inhomogeneity of attributes:

2. Inhomogeneity of objects:

3. Inhomogeneity of distributions:

4. Curse of dimensionality:

5. Resolution:

On Data [Tan et al. 2005]

Types of Data Sets

Data sets may not be a homogeneous collection of objects but come along with differently intricate characteristics:

1. Inhomogeneity of attributes:
   Consider the combination of different attribute types within a single object.

2. Inhomogeneity of objects:
   Consider the combination of different objects in a single data set.

3. Inhomogeneity of distributions:
   The correlation between attributes varies in the instance space.

4. Curse of dimensionality:
   Attribute number and object density stand in exponential relation.

5. Resolution:
   The number of objects or attributes may be given at different resolutions.

On Data [Tan et al. 2005]

Types of Data Sets: Record Data

Collection of records, each of which consists of a fixed set of attributes:

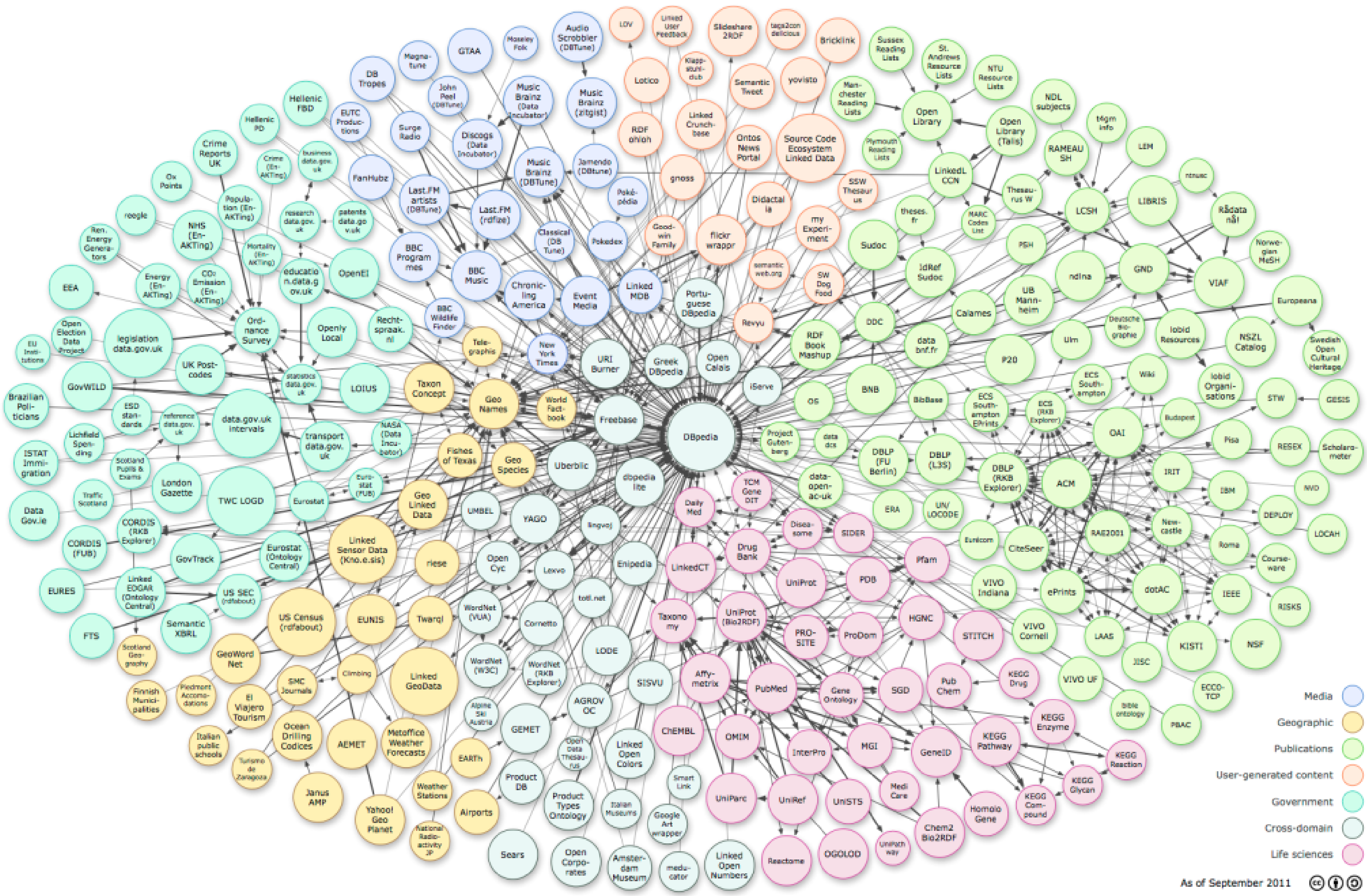
ID	Check	Status	Income	Risk
1	+	single	125 000	No
2	-	married	100 000	No
3	-	single	70 000	No
4	+	married	120 000	No
5	-	divorced	95 000	Yes
6	-	married	60 000	No
7	+	divorced	220 000	No
8	-	single	85 000	Yes
9	-	married	75 000	No
10	-	single	90 000	Yes

- If all elements in a data set have the same fixed set of numeric attributes, they can be thought of as points in a multi-dimensional space.
- Such data can be represented by a matrix, where each row stores an object and each column stores an attribute.
  Example: term-document matrices in information retrieval.

On Data [Tan et al. 2005]

Types of Data Sets: Graph Data

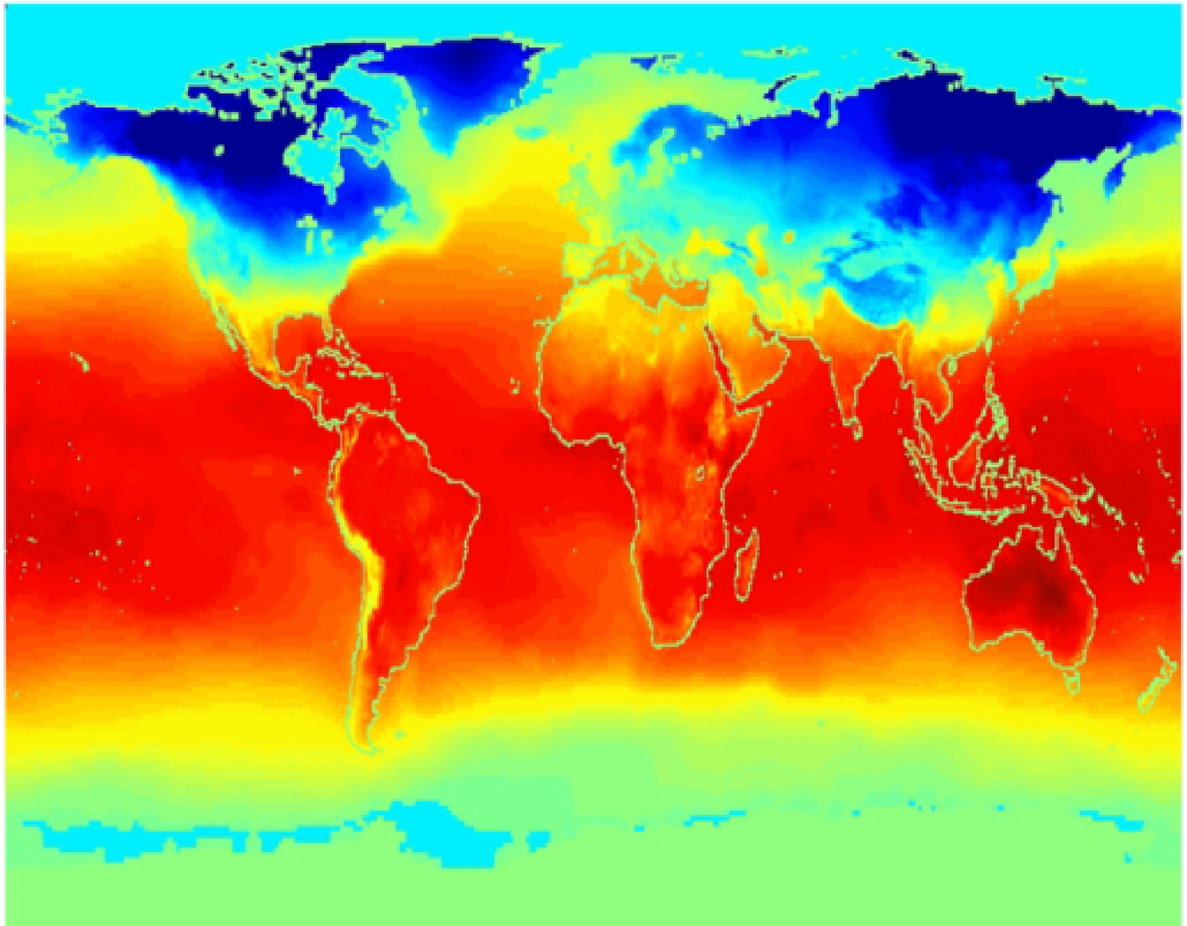
Graph of the Linked Open Data cloud [richard.cyganiak.de] :



On Data [Tan et al. 2005]

Types of Data Sets: Ordered Data

Average monthly temperature of land and ocean (= spatio-temporal data) :



On Data [Tan et al. 2005]

Data Quality

When repeating measurements of a quantity, measurement errors and data collection errors may occur during the measurement process. Questions:

1. What kinds of data quality problems exist?
2. How to detect data quality problems?
3. How to address data quality problems?

On Data [Tan et al. 2005]

Data Quality

When repeating measurements of a quantity, measurement errors and data collection errors may occur during the measurement process. Questions:

1. What kinds of data quality problems exist?
2. How to detect data quality problems?
3. How to address data quality problems?

Definition 1 (Precision, Bias, Accuracy)

Given a set of repeated measurements of the same quantity. Then, the closeness of the measurements to one another is called *precision*, a possible systematic variation is called *bias*, and the closeness to the true value is called *accuracy*.

On Data [Tan et al. 2005]

Data Quality

When repeating measurements of a quantity, measurement errors and data collection errors may occur during the measurement process. Questions:

1. What kinds of data quality problems exist?
2. How to detect data quality problems?
3. How to address data quality problems?

Definition 1 (Precision, Bias, Accuracy)
Given a set of repeated measurements of the same quantity. Then, the closeness of the measurements to one another is called *precision*, a possible systematic variation is called *bias*, and the closeness to the true value is called *accuracy*.

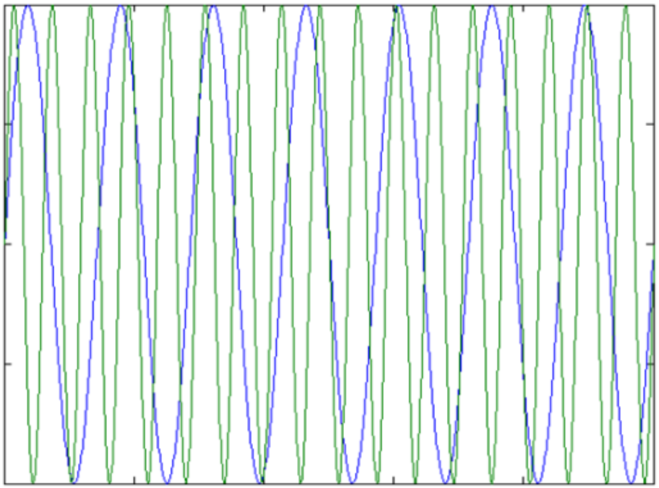
Examples for data quality problems:

- noise, artifacts, outliers
- missing values
- duplicate data

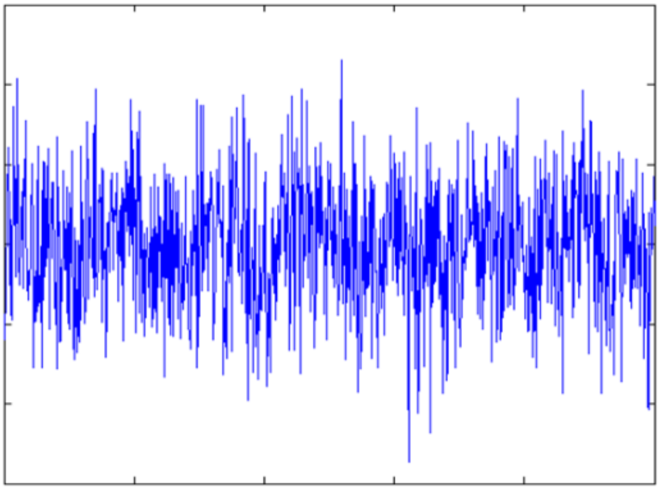
On Data [Tan et al. 2005]

Data Quality: Noise

Noise refers to random modifications of attributes that often have a spatial or temporal characteristics:



sin waves



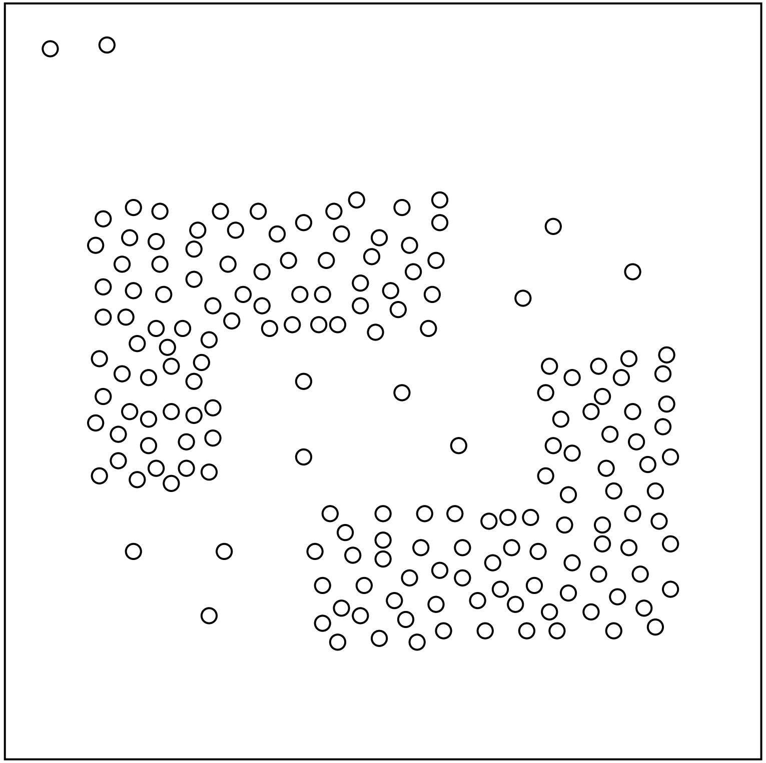
sin waves with noise

Artifacts refer to more deterministic distortions of a measurement process.

On Data [Tan et al. 2005]

Data Quality: Outliers

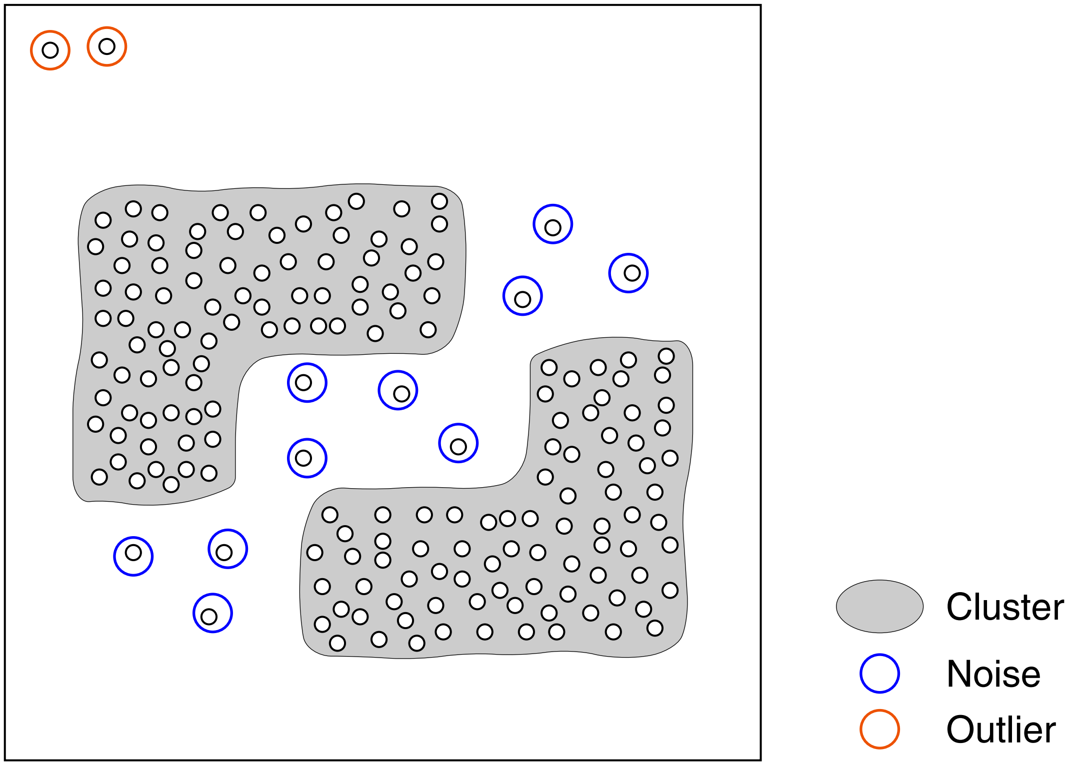
Outliers are members in the data set with characteristics that are considerably different than most of the other elements:



On Data [Tan et al. 2005]

Data Quality: Outliers

Outliers are members in the data set with characteristics that are considerably different than most of the other elements:



On Data [Tan et al. 2005]

Data Quality: Missing Values

Main reasons for missing values:

1. Information is not collected.
   Example: people decline to give their age or weight.

2. Attributes may not be applicable to all elements in O.
   Example: annual income is not applicable to children.

Strategies for handling missing values:

- eliminate members of the data
- estimate missing values
- ignore the missing value during analysis
- replace with all possible values weighted by their probabilities

On Data [Tan et al. 2005]

Data Preprocessing

- aggregation of objects in O
- sampling of object set O
- sampling of feature space X
- selection of attributes (features) [attributes versus features]
- transformation of attributes (features)
- discretization and binarization of attributes (features)
- dimensionality reduction of feature space X

Chapter ML:II

II. Machine Learning Basics

- On Data
- Regression
- Concept Learning: Search in Hypothesis Space
- Concept Learning: Search in Version Space
- Measuring Performance

Regression

Classification versus Regression

- X is a p-dimensional feature space or input space.

Customer 1	
house owner	yes
income (p.a.)	51 000 EUR
repayment (p.m.)	1 000 EUR
credit period	7 years
SCHUFA entry	no
age	37
married	yes
...	

...

Customer n	
house owner	no
income (p.a.)	55 000 EUR
repayment (p.m.)	1 200 EUR
credit period	8 years
SCHUFA entry	no
age	?
married	yes
...	

Classification:

- $C = \{-1, 1\}$ is a set of classes. (similarly: $C = \{0, 1\}$, $C = \{\text{no}, \text{yes}\}$)
- $c : X \to C$ is the ideal classifier for X.
- $D = \{(\mathbf{x}_1, c(\mathbf{x}_1)), \ldots, (\mathbf{x}_n, c(\mathbf{x}_n))\} \subseteq X \times C$ is a set of examples.

Regression

Classification versus Regression

- X is a p-dimensional feature space or input space.

Customer 1			Customer n	
house owner	yes		house owner	no
income (p.a.)	51 000 EUR		income (p.a.)	55 000 EUR
repayment (p.m.)	1 000 EUR		repayment (p.m.)	1 200 EUR
credit period	7 years	. . .	credit period	8 years
SCHUFA entry	no		SCHUFA entry	no
age	37		age	?
married	yes		married	yes
. . .			. . .	

Classification:

- $C = \{-1, 1\}$ is a set of classes. (similarly: $C = \{0, 1\}$, $C = \{\text{no}, \text{yes}\}$)
- $c : X \to C$ is the ideal classifier for X.
- $D = \{(\mathbf{x}_1, c(\mathbf{x}_1)), \ldots, (\mathbf{x}_n, c(\mathbf{x}_n))\} \subseteq X \times C$ is a set of examples.

Regression:

- $Y \subseteq \mathbf{R}$ is the output space.
- y_i is an observed credit line value for an $\mathbf{x}_i \in X$.
- $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\} \subseteq X \times Y$ is a set of examples.

Regression

The Linear Regression Model

- Given $\mathbf{x}$ predict a real-valued output under a linear model:

$$y(\mathbf{x}) = w_0 + \sum_{j=1}^{p} w_j \cdot x_j$$

- Vector notation with $x_0 = 1$ and $\mathbf{w} = (w_0, w_1, \ldots, w_p)^T$:

$$y(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$$

- Assess goodness of fit as residual sum of squares:

$$\text{RSS}(\mathbf{w}) = \sum_{i=1}^{n} (y_i - \mathbf{w}^T\mathbf{x}_i)^2 \qquad (1)$$

Regression

The Linear Regression Model

- Given $\mathbf{x}$ predict a real-valued output under a linear model:

$$y(\mathbf{x}) = w_0 + \sum_{j=1}^{p} w_j \cdot x_j$$

- Vector notation with $x_0 = 1$ and $\mathbf{w} = (w_0, w_1, \ldots, w_p)^T$:

$$y(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$$

- Assess goodness of fit as residual sum of squares:

$$\text{RSS}(\mathbf{w}) = \sum_{i=1}^{n} (y_i - \mathbf{w}^T\mathbf{x}_i)^2 \tag{1}$$

- Estimate $\mathbf{w}$ by the least squares method:

$$\hat{\mathbf{w}} = \underset{\mathbf{w} \in \mathbf{R}^{p+1}}{\text{argmin}} \ \text{RSS}(\mathbf{w}) \tag{2}$$

Remarks:

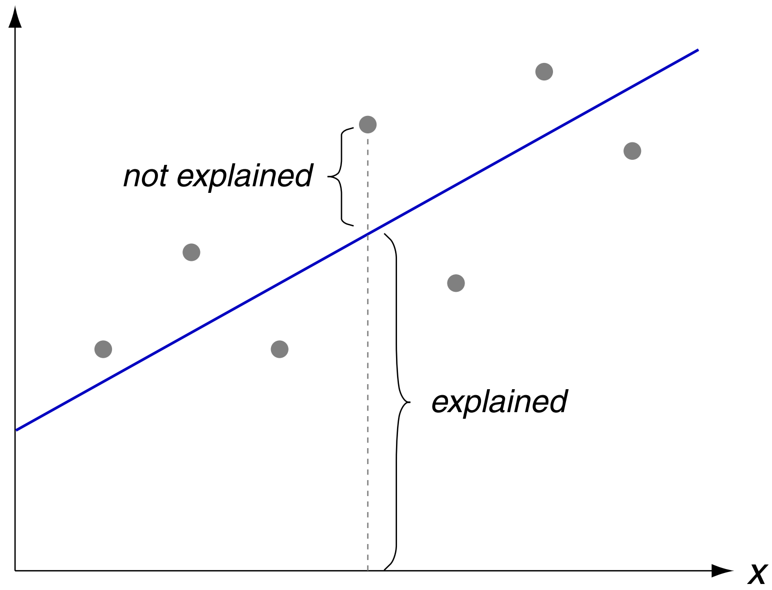
- From a statistical viewpoint, $\mathbf{x} = x_1, \ldots, x_p$ and y form random variables (vectorial and scalar respectively). Each feature vector, $\mathbf{x}_i$, and outcome, y_i, is the result of a random experiment and hence governed by a—usually unknown—probability distribution.
- The distributions of y_i and $(y_i - y(\mathbf{x}_i))$ are identical.
- Estimating $\mathbf{w}$ via RSS minimization is based on the following assumptions:
  1. The random variables y_i are statistically independent. Actually, the conditional independence of the y_i under $\mathbf{x}_i$ is sufficient.
  2. The means $E(y_i)$ lie on a straight line, known as the true (population) regression line: $E(y_i) = \mathbf{w}^T\mathbf{x}_i$
  3. The probability distributions $P(y_i \mid \mathbf{x}_i)$ have the same variance.
- The three assumptions above are called the *weak set* (of assumptions). Along with a fourth assumption about the distribution shape of y_i they become the *strong set* of assumptions.
- In order to avoid cluttered notation, we won't use different symbols to distinguish random variables from ordinary variables. I.e., if $\mathbf{x}, x, y$ denote a (vectorial or scalar) random variable this will become clear from the context.

Regression

One-Dimensional Feature Space

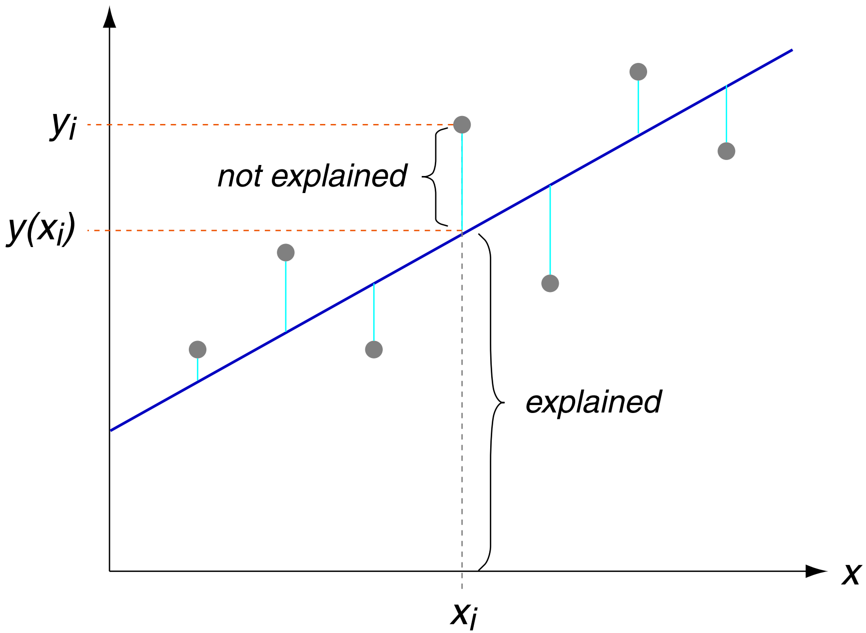
Regression

One-Dimensional Feature Space



Regression

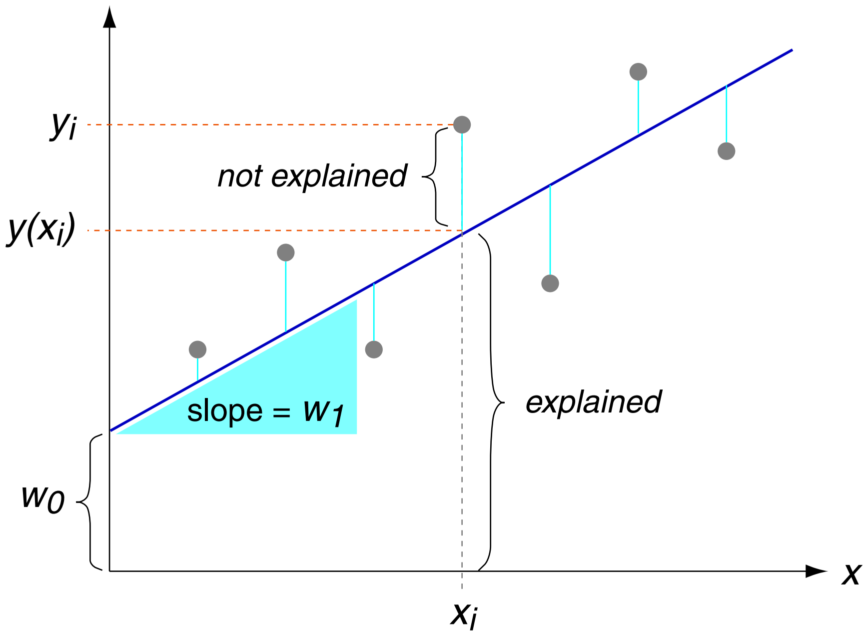
One-Dimensional Feature Space



$$\text{RSS} = \sum_{i=1}^{n} (y_i - y(x_i))^2$$

Regression

One-Dimensional Feature Space



$$y(x) = w_0 + w_1 \cdot x, \qquad \text{RSS}(w_0, w_1) = \sum_{i=1}^{n} (y_i - w_0 - w_1 \cdot x_i)^2$$

Regression

One-Dimensional Feature Space

Minimize RSS(w_0, w_1) :

1. $$\frac{\partial}{\partial w_0} \sum_{i=1}^{n} (y_i - w_0 - w_1 \cdot x_i)^2 \; = \; 0$$

$$\rightsquigarrow \; \ldots \; \rightsquigarrow \quad \hat{w}_0 \; = \; \frac{1}{n} \sum_{i=1}^{n} y_i \; - \; \frac{w_1}{n} \sum_{i=1}^{n} x_i \; = \; \bar{y} - \hat{w}_1 \cdot \bar{x}$$

Regression

One-Dimensional Feature Space

Minimize RSS(w_0, w_1) :

1. $$\frac{\partial}{\partial w_0} \sum_{i=1}^{n} (y_i - w_0 - w_1 \cdot x_i)^2 \; = \; 0$$

$$\rightsquigarrow \; \ldots \; \rightsquigarrow \quad \hat{w}_0 \; = \; \frac{1}{n} \sum_{i=1}^{n} y_i \; - \; \frac{w_1}{n} \sum_{i=1}^{n} x_i \; = \; \bar{y} - \hat{w}_1 \cdot \bar{x}$$

2. $$\frac{\partial}{\partial w_1} \sum_{i=1}^{n} (y_i - w_0 - w_1 \cdot x_i)^2 \; = \; 0$$

$$\rightsquigarrow \; \ldots \; \rightsquigarrow \quad \hat{w}_1 \; = \; \frac{\sum_{i=1}^{n} (x_i - \bar{x}) \cdot \sum_{i=1}^{n} (y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

Regression

Higher-Dimensional Feature Space

- Recall Equation (1) : $\text{RSS}(\mathbf{w}) = \sum_{\mathbf{x}_i \in D} (y(\mathbf{x}_i) - \mathbf{w}^T \mathbf{x}_i)^2$

- Let $\mathbf{X}$ denote the $n \times (p+1)$ matrix
  where row i is the extended input vector $(1 \;\; \mathbf{x}_i^T)$, $\mathbf{x}_i \in D$.

- Let $\mathbf{y}$ denote the n-vector of outputs in the training set D.

Regression

Higher-Dimensional Feature Space

- Recall Equation (1) : $\text{RSS}(\mathbf{w}) = \sum_{\mathbf{x}_i \in D} (y(\mathbf{x}_i) - \mathbf{w}^T\mathbf{x}_i)^2$

- Let $\mathbf{X}$ denote the $n \times (p+1)$ matrix
  where row i is the extended input vector $(1 \;\; \mathbf{x}_i^T)$, $\mathbf{x}_i \in D$.

- Let $\mathbf{y}$ denote the n-vector of outputs in the training set D.

$\leadsto$ $\text{RSS}(\mathbf{w}) = (\mathbf{y} - \mathbf{X}\mathbf{w})^T(\mathbf{y} - \mathbf{X}\mathbf{w})$

RSS$(\mathbf{w})$ is a quadratic function in $p+1$ parameters.

Regression

Higher-Dimensional Feature Space

- Recall Equation (1): $\text{RSS}(\mathbf{w}) = \sum_{\mathbf{x}_i \in D} (y(\mathbf{x}_i) - \mathbf{w}^T \mathbf{x}_i)^2$
- Let $\mathbf{X}$ denote the $n \times (p+1)$ matrix
  where row i is the extended input vector $(1 \;\; \mathbf{x}_i^T)$, $\mathbf{x}_i \in D$.
- Let $\mathbf{y}$ denote the n-vector of outputs in the training set D.

$\leadsto$ $\text{RSS}(\mathbf{w}) = (\mathbf{y} - \mathbf{X}\mathbf{w})^T (\mathbf{y} - \mathbf{X}\mathbf{w})$

RSS($\mathbf{w}$) is a quadratic function in $p+1$ parameters.

Minimize RSS($\mathbf{w}$):

$$\frac{\partial\, \text{RSS}}{\partial \mathbf{w}} = -2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\mathbf{w}) = 0$$

$$\frac{\partial^2\, \text{RSS}}{\partial \mathbf{w} \partial \mathbf{w}^T} = -2\mathbf{X}^T\mathbf{X}$$

Regression

Higher-Dimensional Feature Space

Minimize RSS($\mathbf{w}$) : (continued)

$$\begin{aligned} \mathbf{X}^T(\mathbf{y} - \mathbf{X}\mathbf{w}) &= 0 \\ \Leftrightarrow \quad \mathbf{X}^T\mathbf{X}\mathbf{w} &= \mathbf{X}^T\mathbf{y} \\ \leadsto \quad \hat{\mathbf{w}} &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T \ \mathbf{y} \end{aligned}$$

Regression

Higher-Dimensional Feature Space

Minimize RSS$(\mathbf{w})$: (continued)

$$\begin{aligned} \mathbf{X}^T(\mathbf{y} - \mathbf{X}\mathbf{w}) &= 0 \\ \Leftrightarrow \qquad \mathbf{X}^T\mathbf{X}\mathbf{w} &= \mathbf{X}^T\mathbf{y} \\ \leadsto \qquad \hat{\mathbf{w}} &= \underbrace{(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T}_{\text{pseudo inverse of } \mathbf{X}} \mathbf{y} \end{aligned}$$

If $\mathbf{X}$ has full column rank $p+1$.

Regression

Higher-Dimensional Feature Space

Minimize RSS$(\mathbf{w})$: (continued)

$$\begin{aligned} \mathbf{X}^T(\mathbf{y} - \mathbf{X}\mathbf{w}) &= 0 \\ \Leftrightarrow \quad \mathbf{X}^T\mathbf{X}\mathbf{w} &= \mathbf{X}^T\mathbf{y} \\ \leadsto \quad \hat{\mathbf{w}} &= \underbrace{(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T}_{\text{pseudo inverse of } \mathbf{X}} \mathbf{y} \end{aligned}$$

If $\mathbf{X}$ has full column rank $p+1$.

$$\hat{y}(\mathbf{x}_i) = \mathbf{x}_i^T \hat{\mathbf{w}}$$

Regression function with least squares estimator $\hat{\mathbf{w}}$.

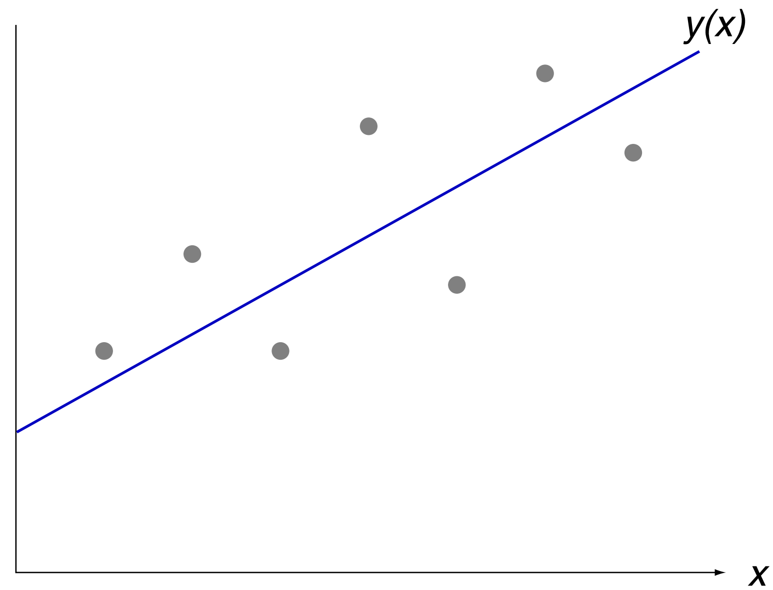
$$\begin{aligned} \hat{\mathbf{y}} &= \mathbf{X}\,\hat{\mathbf{w}} \\ &= \mathbf{X}\,(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \end{aligned}$$

The n-vector of fitted values at the training input.

Regression

Linear Regression for Classification (illustrated for $p = 1$)

Regression learns a real-valued function given as $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$.



$$y(x) = (w_0 \;\; w_1) \begin{pmatrix} 1 \\ x \end{pmatrix}$$

Regression

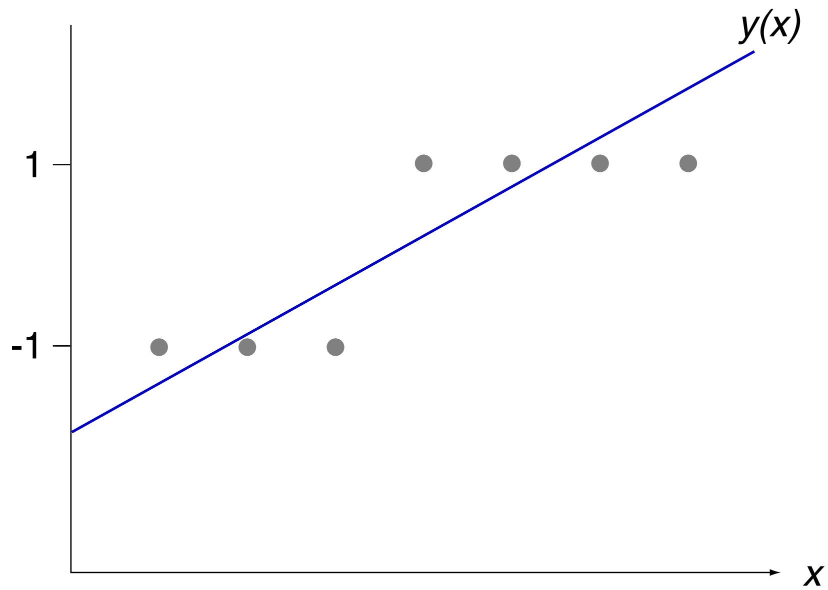
Linear Regression for Classification (illustrated for $p = 1$)

Binary-valued (± 1) functions are also real-valued.

Regression

Linear Regression for Classification (illustrated for $p = 1$)

Use linear regression to learn $\mathbf{w}$ from D, where $y_i = \pm 1 \;\approx\; y(\mathbf{x}_i) = \mathbf{w}^T\mathbf{x}_i$.

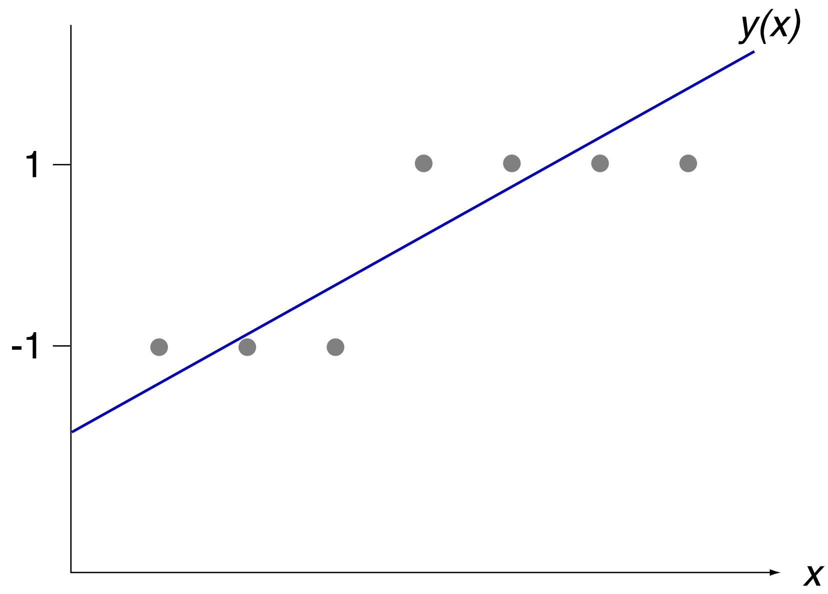


$$y(x) = (w_0 \;\; w_1) \begin{pmatrix} 1 \\ x \end{pmatrix}$$

Regression

Linear Regression for Classification (illustrated for $p = 1$)

Use linear regression to learn $\mathbf{w}$ from D, where $y_i = \pm 1 \ \approx \ y(\mathbf{x}_i) = \mathbf{w}^T \mathbf{x}_i$.



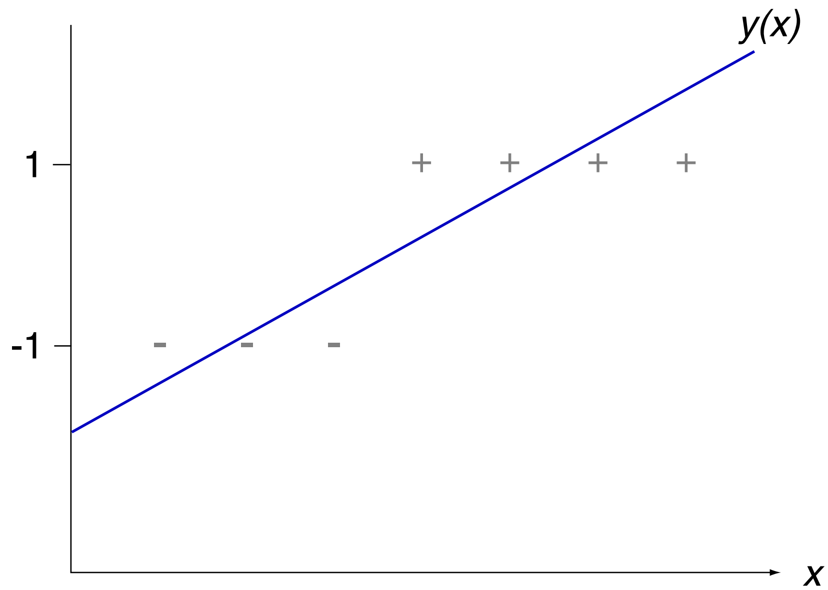
The function $\text{sign}(\mathbf{w}^T \mathbf{x}_i)$ is likely to agree with $y_i = \pm 1$.

- Regression: $y(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$
- Classification: $y(\mathbf{x}) = \text{sign}(\mathbf{w}^T \mathbf{x})$

Regression

Linear Regression for Classification (illustrated for $p = 1$)

Use linear regression to learn $\mathbf{w}$ from D, where $y_i = \pm 1 \approx y(\mathbf{x}_i) = \mathbf{w}^T \mathbf{x}_i$.



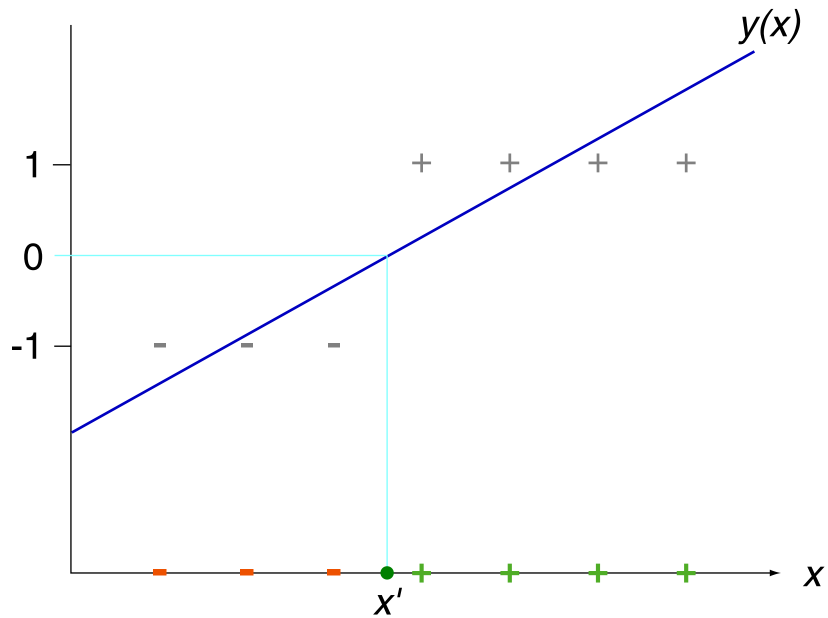
The function $\text{sign}(\mathbf{w}^T \mathbf{x}_i)$ is likely to agree with $y_i = \pm 1$.

- Regression: $y(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$
- Classification: $y(\mathbf{x}) = \text{sign}(\mathbf{w}^T \mathbf{x})$

Regression

Linear Regression for Classification (illustrated for $p = 1$)

Use linear regression to learn $\mathbf{w}$ from D, where $y_i = \pm 1 \;\approx\; y(\mathbf{x}_i) = \mathbf{w}^T\mathbf{x}_i$.



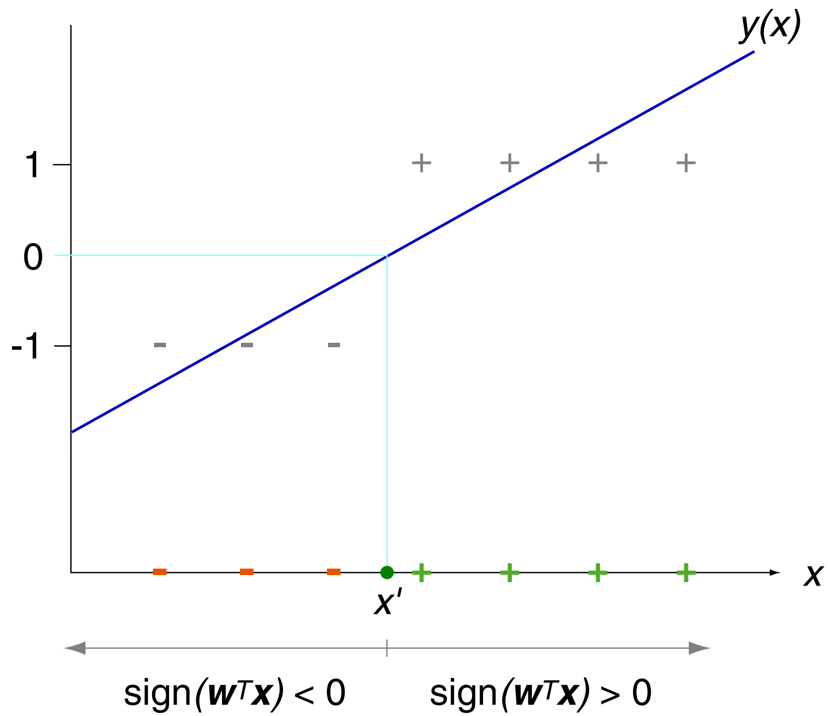
The function $\text{sign}(\mathbf{w}^T\mathbf{x}_i)$ is likely to agree with $y_i = \pm 1$.

- Regression: $y(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$
- Classification: $y(\mathbf{x}) = \text{sign}(\mathbf{w}^T\mathbf{x})$

Regression

Linear Regression for Classification (illustrated for $p = 1$)

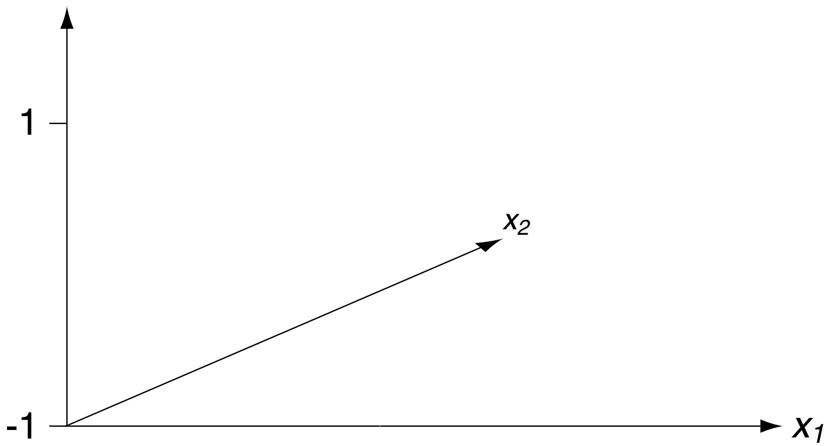
Use linear regression to learn $\mathbf{w}$ from D, where $y_i = \pm 1 \;\approx\; y(\mathbf{x}_i) = \mathbf{w}^T\mathbf{x}_i$.



- The discrimination point, ●, is defined by $w_0 + w_1 \cdot x' = 0$.
- For $p = 2$ we are given a discrimination *line*.

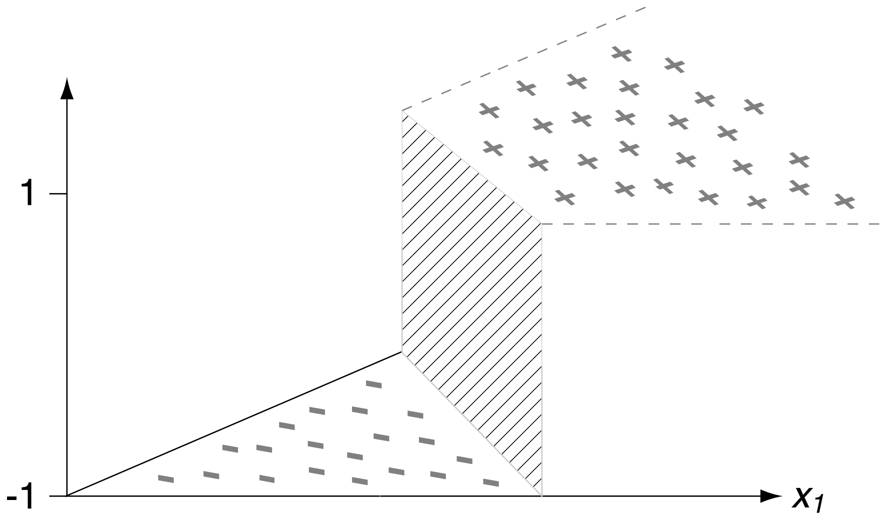
Regression

Linear Regression for Classification (illustrated for $p = 2$)



Regression

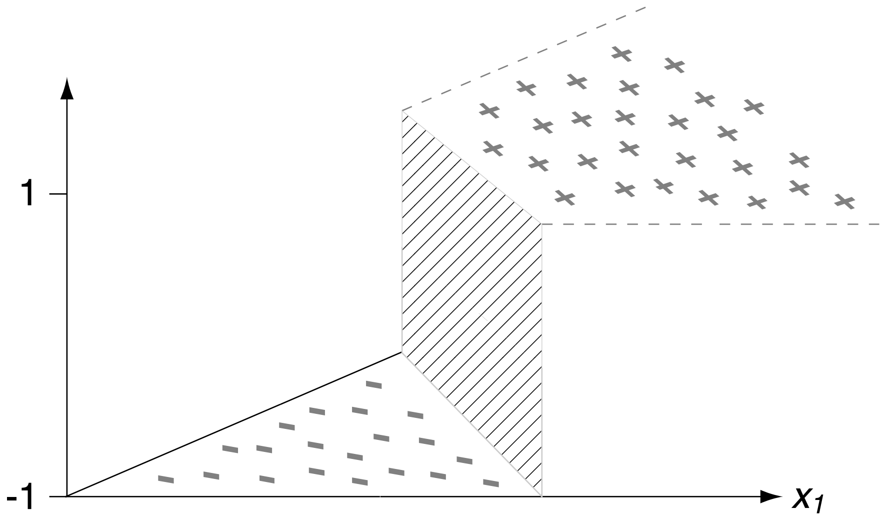
Linear Regression for Classification (illustrated for $p = 2$)



Regression

Linear Regression for Classification (illustrated for $p = 2$)

Use linear regression to learn $\mathbf{w}$ from D, where $y_i = \pm 1 \;\approx\; y(\mathbf{x}_i) = \mathbf{w}^T\mathbf{x}_i$.

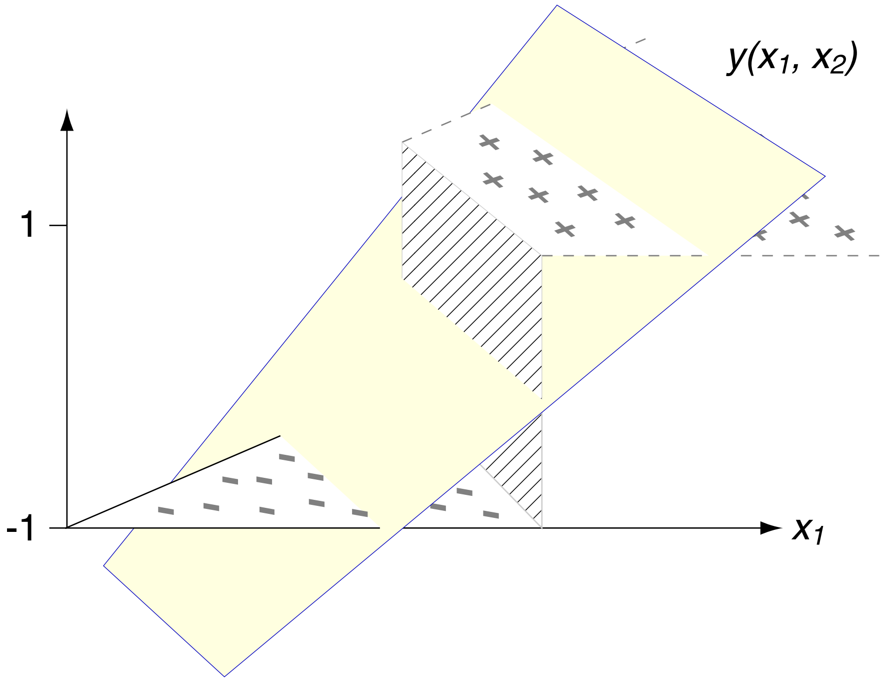


$$y(x) = (w_0 \;\; w_1 \;\; w_2) \begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix}$$

Regression

Linear Regression for Classification (illustrated for $p = 2$)

Use linear regression to learn $\mathbf{w}$ from D, where $y_i = \pm 1 \;\approx\; y(\mathbf{x}_i) = \mathbf{w}^T \mathbf{x}_i$.



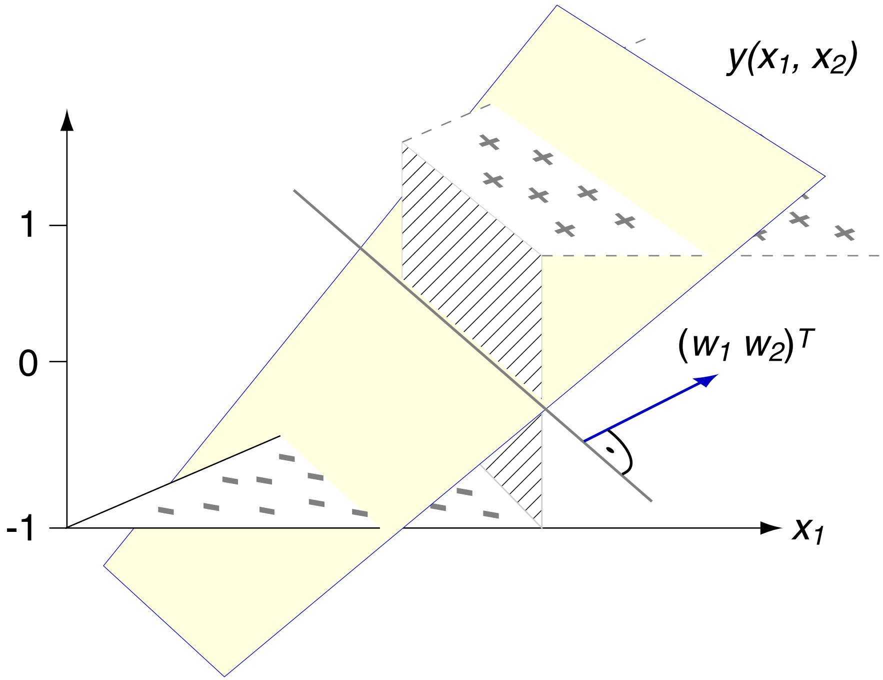
The function sign$(\mathbf{w}^T \mathbf{x}_i)$ is likely to agree with $y_i = \pm 1$.

- Regression: $y(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$
- Classification: $y(\mathbf{x}) = \text{sign}(\mathbf{w}^T \mathbf{x})$

Regression

Linear Regression for Classification (illustrated for $p = 2$)

Use linear regression to learn $\mathbf{w}$ from D, where $y_i = \pm 1 \;\approx\; y(\mathbf{x}_i) = \mathbf{w}^T \mathbf{x}_i$.



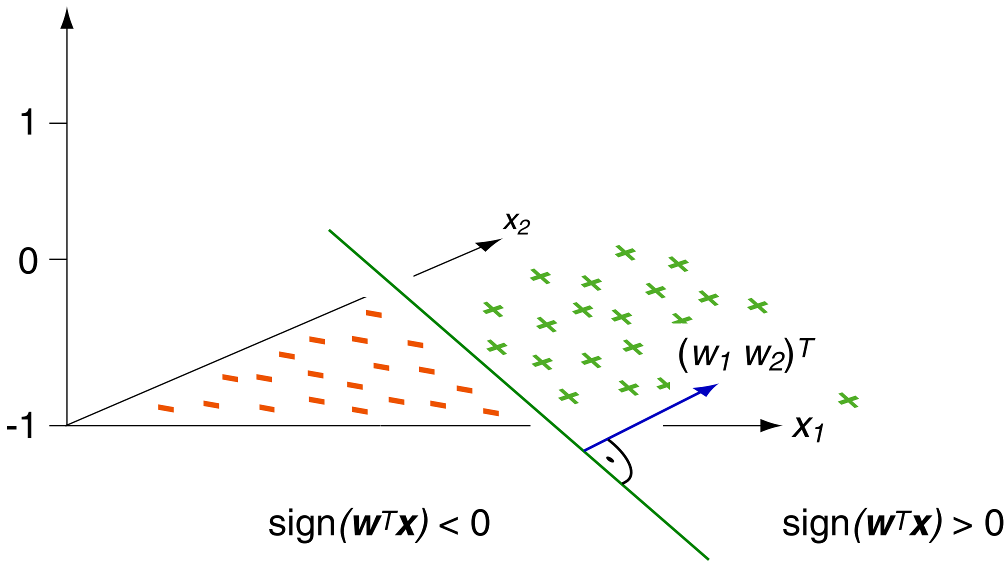
The function $\text{sign}(\mathbf{w}^T\mathbf{x}_i)$ is likely to agree with $y_i = \pm 1$.

- Regression: $y(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$
- Classification: $y(\mathbf{x}) = \text{sign}(\mathbf{w}^T\mathbf{x})$

Regression

Linear Regression for Classification (illustrated for $p = 2$)

Use linear regression to learn $\mathbf{w}$ from D, where $y_i = \pm 1 \;\approx\; y(\mathbf{x}_i) = \mathbf{w}^T\mathbf{x}_i$.

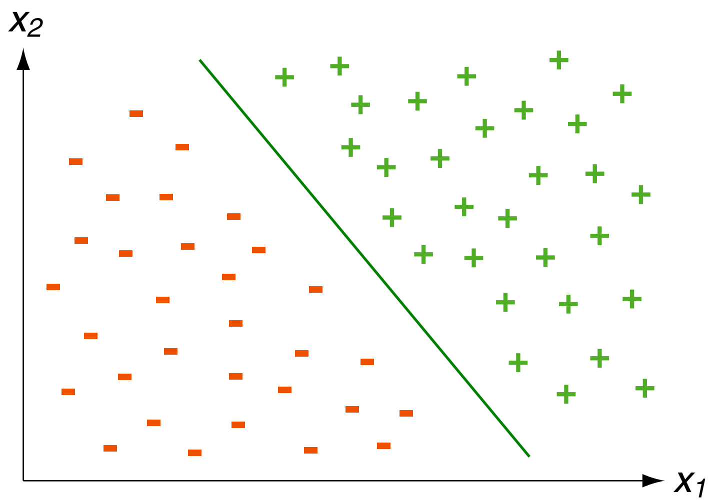


- The discrimination line, —, is defined by $w_0 + w_1 \cdot x_1 + w_2 \cdot x_2 = 0$.
- For $p = 3$ ($p > 3$) we are given a discriminating *(hyper)plane*.

Regression

Linear Regression for Classification (illustrated for $p = 2$)

Use linear regression to learn $\mathbf{w}$ from D, where $y_i = \pm 1 \;\approx\; y(\mathbf{x}_i) = \mathbf{w}^T \mathbf{x}_i$.



- The discrimination line, —, is defined by $w_0 + w_1 \cdot x_1 + w_2 \cdot x_2 = 0$.
- For $p = 3$ ($p > 3$) we are given a discriminating *(hyper)plane*.

Regression

The Linear Model Structure

The components (variables, random variables) of the input vector $\mathbf{x} = (x_1, \ldots, x_p)$ can come from different sources [Hastie et al. 2001] :

1. quantitative inputs
2. transformations of quantitative inputs, such as $\log x_j$, $\sqrt{x_j}$
3. basis expansions, such as $x_j = (x_1)^j$
4. encoding of a qualitative variable g, $g \in \{1, \ldots, p\}$, as $x_j = I(g = j)$
5. interactions between variables, such as $x_3 = x_1 \cdot x_2$

Regression

The Linear Model Structure

The components (variables, random variables) of the input vector $\mathbf{x} = (x_1, \ldots, x_p)$ can come from different sources [Hastie et al. 2001] :

1. quantitative inputs
2. transformations of quantitative inputs, such as $\log x_j$, $\sqrt{x_j}$
3. basis expansions, such as $x_j = (x_1)^j$
4. encoding of a qualitative variable g, $g \in \{1, \ldots, p\}$, as $x_j = I(g = j)$
5. interactions between variables, such as $x_3 = x_1 \cdot x_2$

No matter the source of the x_j, the model is still linear in the parameters $\mathbf{w}$:

$$y(\mathbf{x}) = w_0 + \sum_{j=1}^{p} w_j \cdot \phi_j(x_j)$$

Regression

The Linear Model Structure

The components (variables, random variables) of the input vector $\mathbf{x} = (x_1, \ldots, x_p)$ can come from different sources [Hastie et al. 2001] :

1. quantitative inputs
2. transformations of quantitative inputs, such as $\log x_j$, $\sqrt{x_j}$
3. basis expansions, such as $x_j = (x_1)^j$
4. encoding of a qualitative variable g, $g \in \{1, \ldots, p\}$, as $x_j = I(g = j)$
5. interactions between variables, such as $x_3 = x_1 \cdot x_2$

No matter the source of the x_j, the model is still linear in the parameters $\mathbf{w}$:

$$y(\mathbf{x}) = w_0 + \sum_{j=1}^{p} w_j \cdot \phi_j(x_j)$$

- linear in the parameters: constant w_j and additive combination

Regression

The Linear Model Structure

The components (variables, random variables) of the input vector $\mathbf{x} = (x_1, \ldots, x_p)$ can come from different sources [Hastie et al. 2001] :

1. quantitative inputs
2. transformations of quantitative inputs, such as $\log x_j$, $\sqrt{x_j}$
3. basis expansions, such as $x_j = (x_1)^j$
4. encoding of a qualitative variable g, $g \in \{1, \ldots, p\}$, as $x_j = I(g = j)$
5. interactions between variables, such as $x_3 = x_1 \cdot x_2$

No matter the source of the x_j, the model is still linear in the parameters $\mathbf{w}$:

$$y(\mathbf{x}) = w_0 + \sum_{j=1}^{p} w_j \cdot \phi_j(x_j)$$

- linear in the parameters: constant w_j and additive combination
- basis functions: input variables (space) become feature variables (space)

Regression

Theoretical Properties of the Solution

Theorem 1 (Gauss-Markov)

Let $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ be a set of examples to be fitted with a linear model as $y(\mathbf{x}) = \mathbf{x}^T\mathbf{w}$. Within the class of linear unbiased estimators for $\mathbf{w}$, the least squares estimator $\hat{\mathbf{w}}$ has minimum variance, i.e., is most efficient.

Regression

Theoretical Properties of the Solution

Theorem 1 (Gauss-Markov)
Let $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ be a set of examples to be fitted with a linear model as $y(\mathbf{x}) = \mathbf{x}^T\mathbf{w}$. Within the class of linear unbiased estimators for $\mathbf{w}$, the least squares estimator $\hat{\mathbf{w}}$ has minimum variance, i.e., is most efficient.

Related followup issues:

- mean and variance of $\hat{\mathbf{w}}$
- proof of the Gauss-Markov theorem
- weak set and strong set of assumptions
- efficiency and consistency of unbiased estimators
- rank deficiencies, where the feature number p exceeds $|D| = n$
- relation of mean least squares and the maximum likelihood principle

Chapter ML:II (continued)

II. Machine Learning Basics

- On Data
- Regression
- Concept Learning: Search in Hypothesis Space
- Concept Learning: Search in Version Space
- Measuring Performance

Concept Learning: Search in Hypothesis Space

A Learning Task

Given is a set D of examples: days that are characterized by the six features "Sky", "Temperature", "Humidity", "Wind", "Water", and "Forecast", along with a statement (in fact: a feature) whether or not our friend will enjoy her favorite sport.

Example	Sky	Temperature	Humidity	Wind	Water	Forecast	EnjoySport
1	sunny	warm	normal	strong	warm	same	yes
2	sunny	warm	high	strong	warm	same	yes
3	rainy	cold	high	strong	warm	change	no
4	sunny	warm	high	strong	cool	change	yes

- What is the concept behind "EnjoySport" ?
- What are possible hypotheses to formalize the concept "EnjoySport" ?
  Similarly: What are the elements of the set or class "EnjoySport" ?

Remarks:

- Domains of the features in the learning task:

Sky	Temperature	Humidity	Wind	Water	Forecast
sunny	warm	normal	strong	warm	same
rainy	cold	high	weak	cool	change
cloudy					

- A hypothesis is a finding or an insight gained by inductive reasoning. A hypothesis cannot be inferred or proved by deductive reasoning.

- Within concept learning tasks, hypotheses are used to capture the target concept.
  A hypothesis is justified inductively, by its means to represent a given set of observations, which are called examples here.

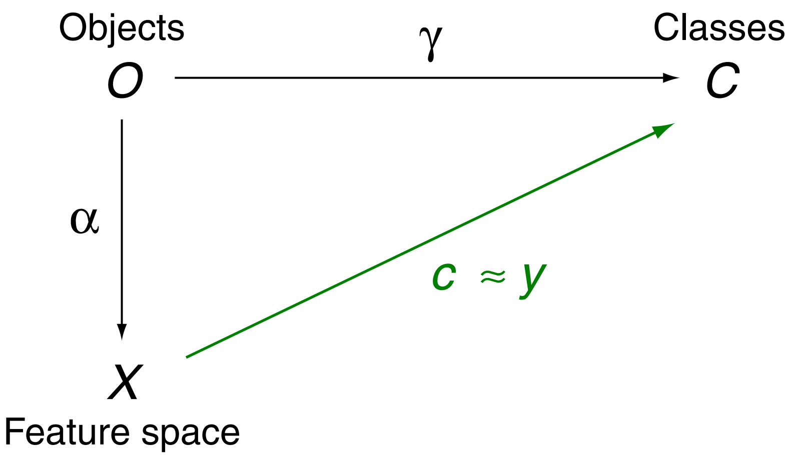
Concept Learning: Search in Hypothesis Space

Definition 1 (Concept, Hypothesis, Hypothesis Space)

A concept is a subset of an object set O and hence determines a subset of the feature space $X = \alpha(O)$. Concept learning is the approximation of the ideal classifier $c : X \rightarrow \{0, 1\}$ by a function y, where c is defined as follows:

$$c(\mathbf{x}) = \begin{cases} 1 & \text{if } \alpha^{-1}(\mathbf{x}) \text{ belongs to the concept} \\ 0 & \text{otherwise} \end{cases}$$

A function $h : X \rightarrow \{0, 1\}$ is called hypothesis. A set H of hypotheses among which the approximation function y is searched is called hypothesis space.



Concept Learning: Search in Hypothesis Space

Usually, an example set D, $D = \{(\mathbf{x}_1, c(\mathbf{x}_1)), \ldots, (\mathbf{x}_n, c(\mathbf{x}_n))\}$, contains positive ($c(\mathbf{x}) = 1$) and negative ($c(\mathbf{x}) = 0$) examples. [Learning Task]

Definition 2 (Hypothesis-Fulfilling, Consistency)

An example $(\mathbf{x}, c(\mathbf{x}))$ fulfills a hypothesis h iff $h(\mathbf{x}) = 1$. A hypothesis h is consistent with an example $(\mathbf{x}, c(\mathbf{x}))$ iff $h(\mathbf{x}) = c(\mathbf{x})$.

A hypothesis h is consistent with a set D of examples, denoted as *consistent*(h, D), iff:

$$\forall (\mathbf{x}, c(\mathbf{x})) \in D : \; h(\mathbf{x}) = c(\mathbf{x})$$

Remarks:

- The symbol “Iff” or “iff” is an abbreviation for “If and only if”, which means “necessary and sufficient”. [Wolfram]
- The following terms are used synonymously: target concept, target function, classifier, ideal classifier. [ML Introduction]
- The fact that a hypothesis is consistent with an example can also be described the other way round: an example is consistent with a hypothesis.
- Given an example $(\mathbf{x}, c(\mathbf{x}))$, notice the difference between (1) hypothesis-fulfilling and (2) being consistent with a hypothesis. The former asks for $h(\mathbf{x}) = 1$, disregarding the actual target concept value $c(\mathbf{x})$. The latter asks for the equivalence between the target concept $c(\mathbf{x})$ and the hypothesis $h(\mathbf{x})$.
- The consistency of h can be analyzed with respect to a single example or a set D of examples. Given the latter, consistency requires for all elements in D that $h(\mathbf{x}) = 1$ iff $c(\mathbf{x}) = 1$. This is equivalent with the condition that $h(\mathbf{x}) = 0$ iff $c(\mathbf{x}) = 0$ for all $\mathbf{x} \in D$.
- Learning means to determine a hypothesis $h \in H$ that is consistent with D.

Concept Learning: Search in Hypothesis Space

A Learning Task (continued)

Structure of a hypothesis h:

1. conjunction of feature-value pairs
2. three kinds of values: literal, ? (wildcard), $\perp$ (contradiction)

A hypothesis in the example [Learning Task] : $\langle$ *sunny, ?, ?, strong, ?, same* $\rangle$

Concept Learning: Search in Hypothesis Space

A Learning Task (continued)

Structure of a hypothesis h:

1. conjunction of feature-value pairs
2. three kinds of values: literal, ? (wildcard), $\perp$ (contradiction)

A hypothesis in the example [Learning Task] : ⟨ *sunny, ?, ?, strong, ?, same* ⟩

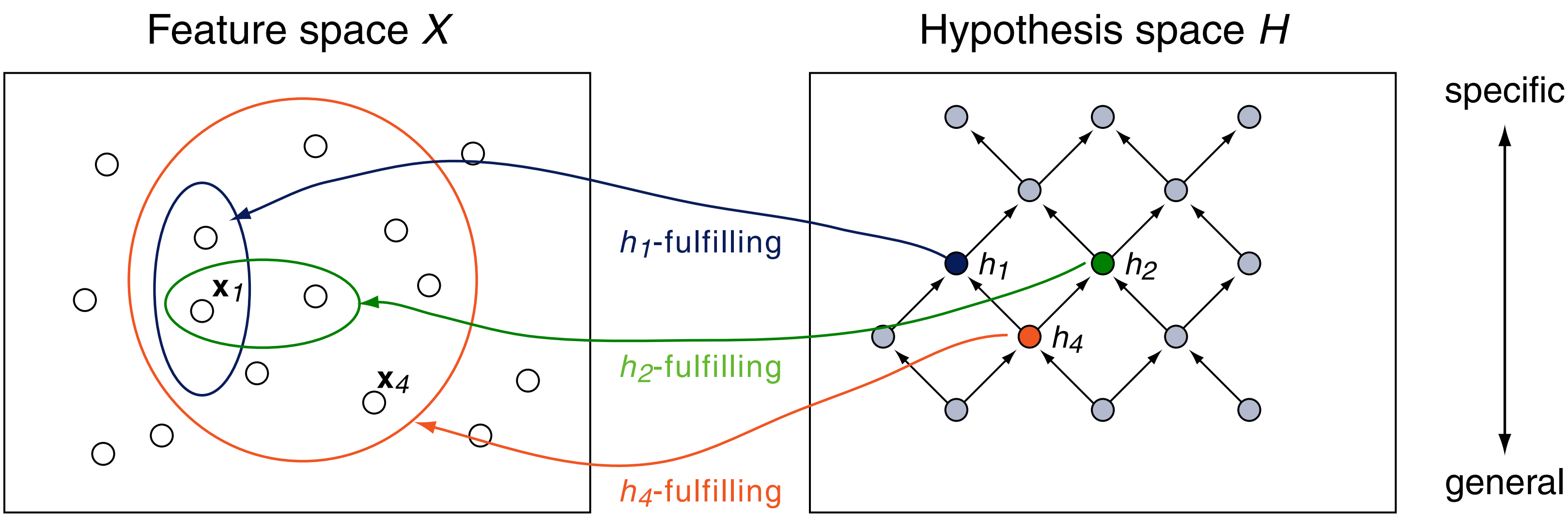
Definition 3 (Maximally Specific / General Hypothesis)
The hypotheses $s_0(\mathbf{x}) \equiv 0$ and $g_0(\mathbf{x}) \equiv 1$ are called maximally specific and maximally general hypothesis respectively. No $\mathbf{x} \in X$ fulfills s_0, and all $\mathbf{x} \in X$ fulfill g_0.

Maximally specific / general hypothesis in the example [Learning Task] :

- $s_0 = \langle\ \perp, \perp, \perp, \perp, \perp, \perp\ \rangle$
- $g_0 = \langle$ *?, ?, ?, ?, ?, ?* $\rangle$

Concept Learning: Search in Hypothesis Space

Ordering of Hypotheses



$\mathbf{x}_1 = ($*sunny, warm, normal, strong, warm, same*$)$

$\mathbf{x}_4 = ($*sunny, warm, high, strong, cool, change*$)$

$h_1 = \langle$ *sunny, ?, normal, ?, ?, ?* $\rangle$

$h_2 = \langle$ *sunny, ?, ?, ?, warm, ?* $\rangle$

$h_4 = \langle$ *sunny, ?, ?, ?, ?, ?* $\rangle$

Concept Learning: Search in Hypothesis Space

Ordering of Hypotheses

Definition 4 (More General Relation)

Let X be a feature space and let h_1 and h_2 be two boolean-valued functions with domain X. Then function h_1 is called more general than function h_2, denoted as $h_1 \geq_g h_2$, iff:

$$\forall \mathbf{x} \in X : \; (\; h_2(\mathbf{x}) = 1 \; \text{ implies } \; h_1(\mathbf{x}) = 1 \;)$$

h_1 is called stricly more general than h_2, denoted as $h_1 >_g h_2$, iff:

$$(h_1 \geq_g h_2) \; \text{ and } \; (h_2 \not\geq_g h_1)$$

Concept Learning: Search in Hypothesis Space

Ordering of Hypotheses

Definition 4 (More General Relation)

Let X be a feature space and let h_1 and h_2 be two boolean-valued functions with domain X. Then function h_1 is called more general than function h_2, denoted as $h_1 \geq_g h_2$, iff:

$$\forall \mathbf{x} \in X : \ (\ h_2(\mathbf{x}) = 1 \ \text{ implies } \ h_1(\mathbf{x}) = 1 \ )$$

h_1 is called stricly more general than h_2, denoted as $h_1 >_g h_2$, iff:

$$(h_1 \geq_g h_2) \ \text{ and } \ (h_2 \not\geq_g h_1)$$

About the maximally specific / general hypothesis:

- s_0 is minimum and g_0 is maximum with regard to $\geq_g$: no hypothesis is more specific wrt. s_0, and no hypothesis is more general wrt. g_0.
- We will consider only hypothesis spaces that contain s_0 and g_0.

Remarks:

- If h_1 is more general than h_2, then h_2 can also be called being more specific than h_1.
- $\geq_g$ and $>_g$ are independent of a target concept c. They depend only on the fact that examples fulfill a hypothesis, i.e., whether $h(\mathbf{x}) = 1$. They require not that $c(\mathbf{x}) = 1$.
- The $\geq_g$-relation defines a partial ordering on the hypothesis space H : $\geq_g$ is reflexive, anti-symmetric, and transitive. The ordering is *partial* since (unlike in a total ordering) not all hypothesis pairs stand in the relation. I.e., we are given hypotheses h_i, h_j, for which neither $h_i \geq_g h_j$ nor $h_j \geq_g h_i$ holds, such as the hypotheses h_1 and h_2 in the hypothesis space.

Remarks: (continued)

- The semantics of the implication, in words "a implies b", denoted as $a \rightarrow b$, is as follows. $a \rightarrow b$ is true if either (1) a is true and b is true, or (2) if a is false and b is true, or (3) if a is false and b is false—in short: "if a is true then b is true as well", or, "the truth of a implies the truth of b". The connective "$\rightarrow$" is the causality connective.

- In particular does the connective "$\rightarrow$" not stand for "entails", which would be denoted as either $\Rightarrow$ or $\models$. Logical entailment (synonymously: logical inference, logical deduction) allows to infer or to proof a fact. From the fact $h_2(\mathbf{x}) = 1$, however, we cannot infer or proof the fact $h_1(\mathbf{x}) = 1$.

- Here, in the definition, the implication specifies a condition that is to be fulfilled by the definiendum (= the thing to be defined). The implication is used to check whether or not a thing falls under the definition: each pair of functions, h_1, h_2, falls under the definition of the $\geq_g$-relation (i.e., stands in the $\geq_g$-relation) if and only if the implication $h_2(\mathbf{x}) = 1 \rightarrow h_1(\mathbf{x}) = 1$ is true for all $\mathbf{x} \in X$.

- In a nutshell: distinguish between "α requires β", denoted as $\alpha \rightarrow \beta$, on the one hand, and "from α follows β", denoted as $\alpha \Rightarrow \beta$, on the other hand. $\alpha \rightarrow \beta$ is considered as a sentence from the *object language* (language of discourse) and stipulates a computing operation, whereas $\alpha \Rightarrow \beta$ is a sentence from the *meta language* and makes an assertion *about* the sentence $\alpha \rightarrow \beta$, namely: "$\alpha \rightarrow \beta$ is a tautology".

- Finally, consider the following sentences from the object language, which are synonymous: "$\alpha \rightarrow \beta$", "α implies β", "if α then β", "α causes β", "α requires β", "β involves α".

Concept Learning: Search in Hypothesis Space

Inductive Learning Hypothesis

> *"Any hypothesis found to approximate the target function well over a sufficiently large set of training examples will also approximate the target function well over other unobserved examples."*
>
> [p.23, Mitchell 1997]

Concept Learning: Search in Hypothesis Space

Find-S Algorithm

```
1. h = s_0   // h is a maximally specific hypothesis in H.
2. FOREACH (x, c(x)) ∈ D DO
     IF c(x) = 1 THEN   // Use only positive examples.
       IF h(x) = 0 DO
         h = min_generalization(h, x)   // Relax hypothesis h wrt. x.
       ENDIF
     ENDIF
   ENDDO
3. return(h)
```

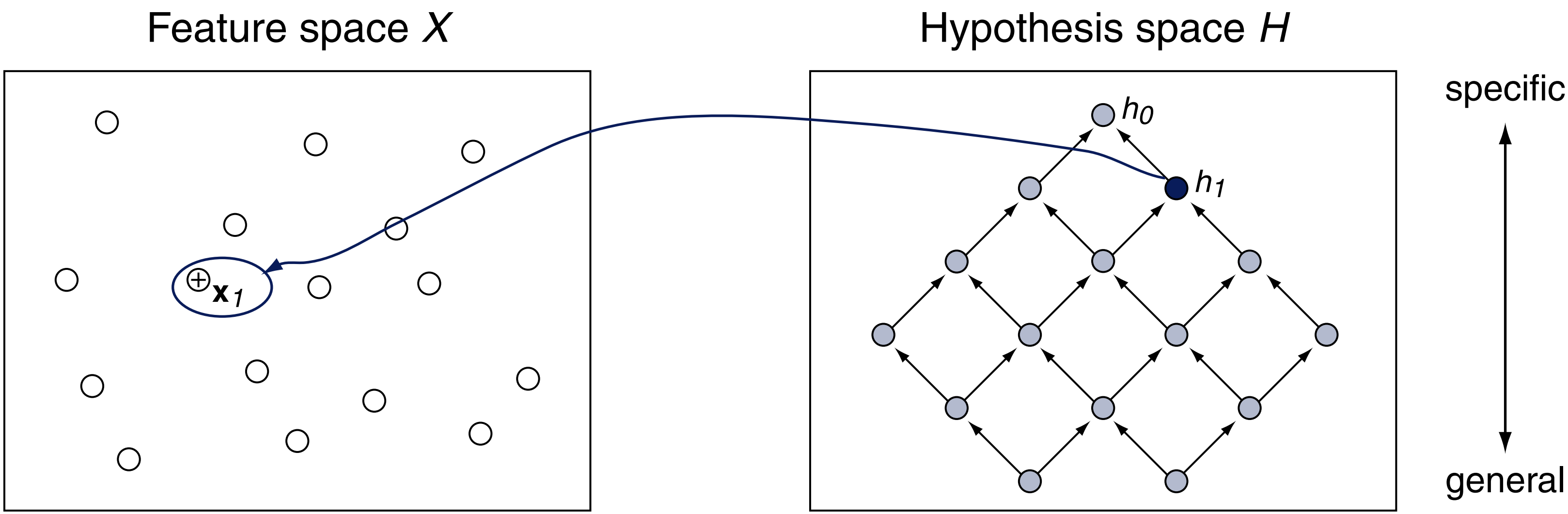
Remarks:

- Another term for “generalization” is “relaxation”.
- The function *min_generalization*$(h, \mathbf{x})$ returns a hypothesis h' that is minimally generalized wrt. h and that is consistent with $(\mathbf{x}, 1)$. Denoted formally: $h' \geq_g h$ and $h'(\mathbf{x}) = 1$ and there is no h'' with $h' >_g h'' \geq_g h$ and $h''(\mathbf{x}) = 1$.
- The relaxation of h given $\mathbf{x}$, *min_generalization*$(h, \mathbf{x})$, may not be deterministic. In such a case, one of the alternatives has to be chosen.
- If a hypothesis h needs to be relaxed towards some h' where $h' \notin H$, the maximally general hypothesis $g_0 \equiv 1$ can be added to H.
- Similarly to *min_generalization*$(h, \mathbf{x})$, a function *min_specialization*$(h, \mathbf{x})$ can be defined, which returns a minimally specialized, consistent hypotheses for negative examples.

Concept Learning: Search in Hypothesis Space

Find-S Algorithm

See the example set D for the concept *EnjoySport*.



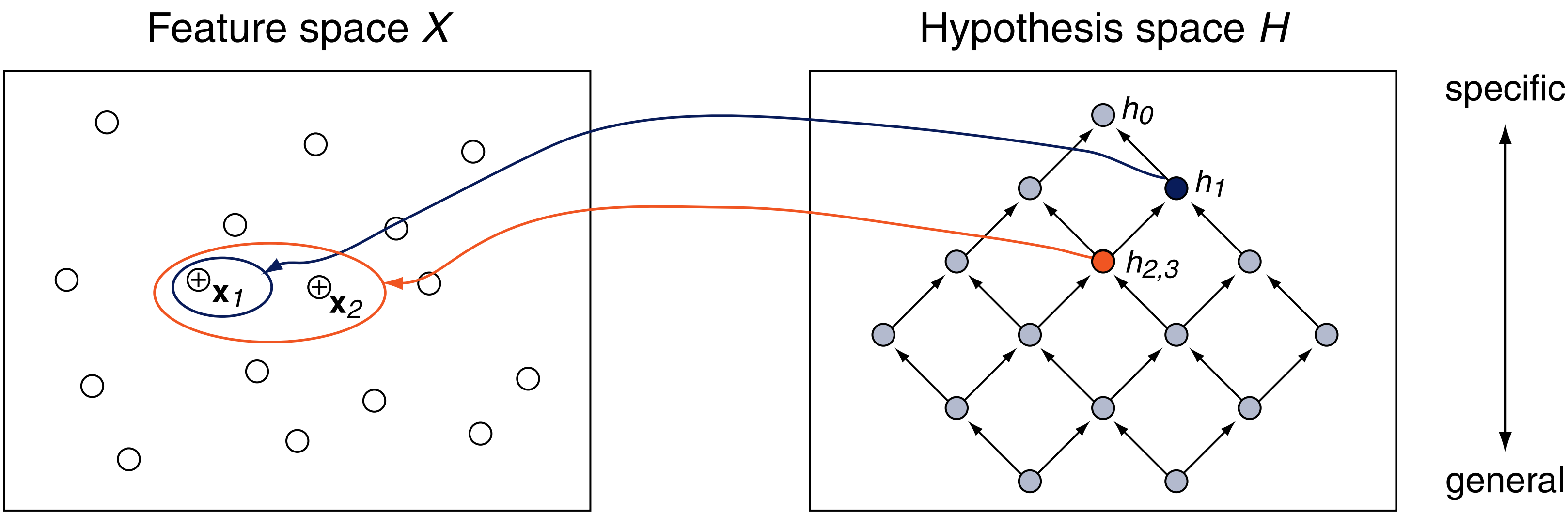
$h_0 = s_0 = \langle \perp, \perp, \perp, \perp, \perp, \perp \rangle$

$\mathbf{x}_1 = ($*sunny, warm, normal, strong, warm, same*$)$ $h_1 = \langle$ *sunny, warm, normal, strong, warm, same* $\rangle$

Concept Learning: Search in Hypothesis Space

Find-S Algorithm

See the example set D for the concept *EnjoySport*.



$h_0 = s_0 = \langle \bot, \bot, \bot, \bot, \bot, \bot \rangle$

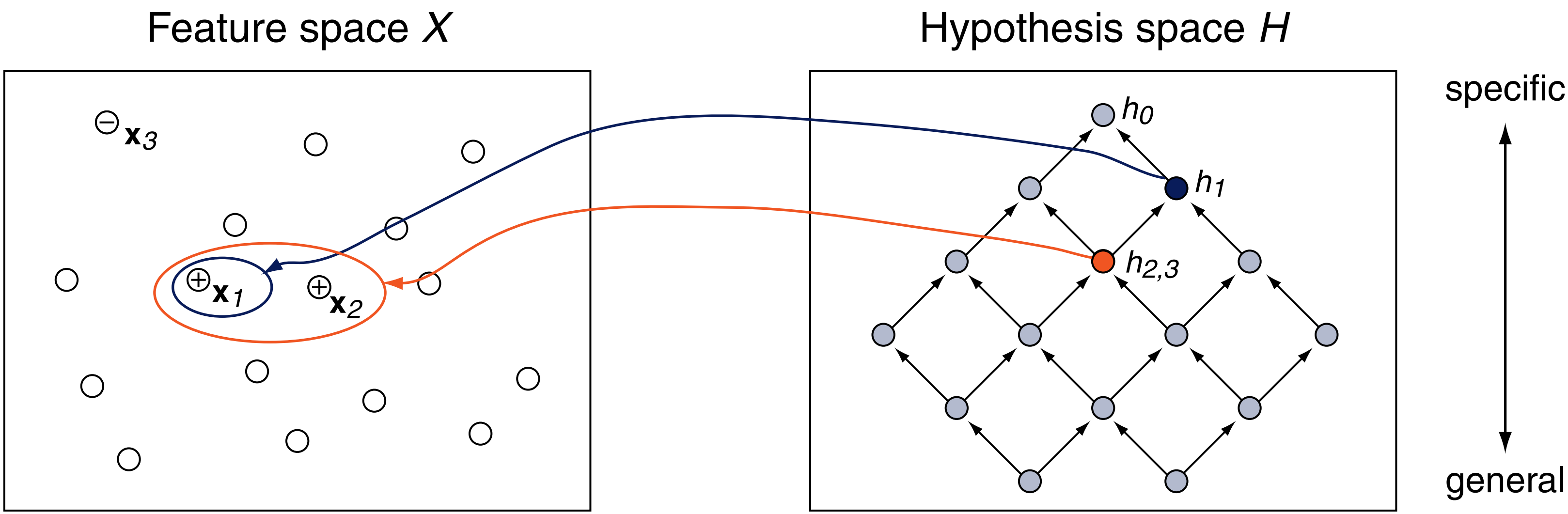
$\mathbf{x}_1 =$ (*sunny, warm, normal, strong, warm, same*) $h_1 = \langle$ *sunny, warm, normal, strong, warm, same* $\rangle$

$\mathbf{x}_2 =$ (*sunny, warm, high, strong, warm, same*) $h_2 = \langle$ *sunny, warm, ?, strong, warm, same* $\rangle$

Concept Learning: Search in Hypothesis Space

Find-S Algorithm

See the example set D for the concept *EnjoySport*.



$h_0 = s_0 = \langle \perp, \perp, \perp, \perp, \perp, \perp \rangle$

$\mathbf{x}_1 =$ (*sunny, warm, normal, strong, warm, same*) $h_1 = \langle$ *sunny, warm, normal, strong, warm, same* $\rangle$

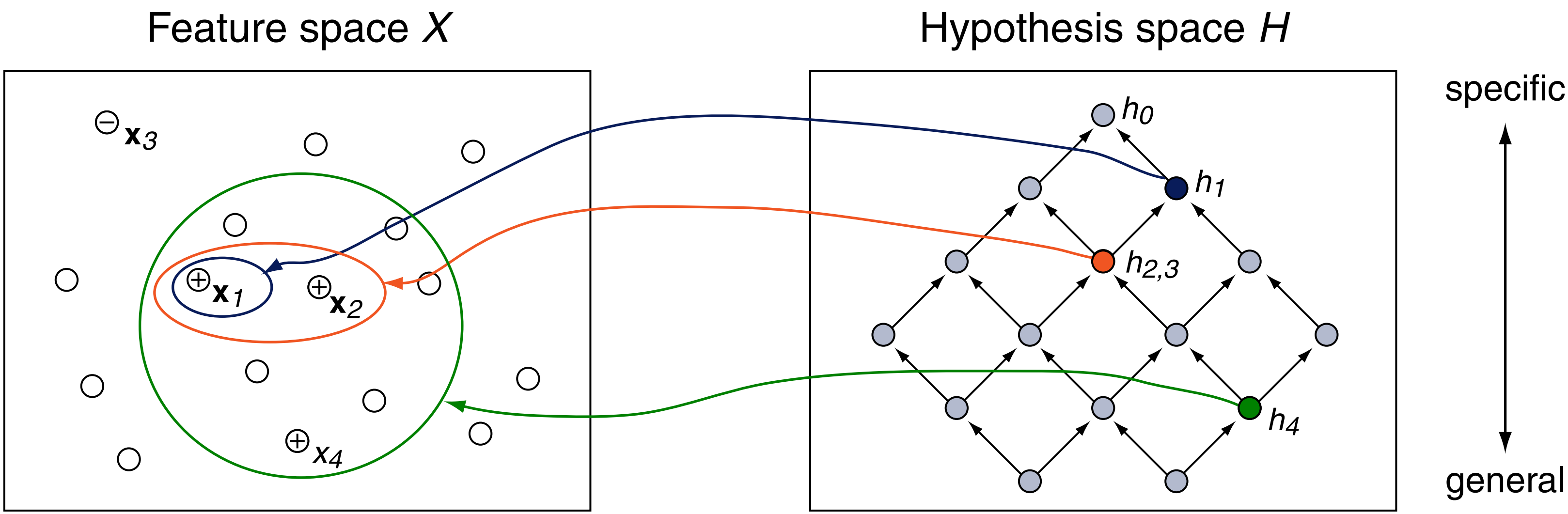
$\mathbf{x}_2 =$ (*sunny, warm, high, strong, warm, same*) $h_2 = \langle$ *sunny, warm, ?, strong, warm, same* $\rangle$

$\mathbf{x}_3 =$ (*rainy, cold, high, strong, warm, change*) $h_3 = \langle$ *sunny, warm, ?, strong, warm, same* $\rangle$

Concept Learning: Search in Hypothesis Space

Find-S Algorithm

See the example set D for the concept *EnjoySport*.



$h_0 = s_0 = \langle \perp, \perp, \perp, \perp, \perp, \perp \rangle$

$\mathbf{x}_1 =$ (*sunny, warm, normal, strong, warm, same*) $h_1 = \langle$ *sunny, warm, normal, strong, warm, same* $\rangle$

$\mathbf{x}_2 =$ (*sunny, warm, high, strong, warm, same*) $h_2 = \langle$ *sunny, warm, ?, strong, warm, same* $\rangle$

$\mathbf{x}_3 =$ (*rainy, cold, high, strong, warm, change*) $h_3 = \langle$ *sunny, warm, ?, strong, warm, same* $\rangle$

$\mathbf{x}_4 =$ (*sunny, warm, high, strong, cool, change*) $h_4 = \langle$ *sunny, warm, ?, strong, ?, ?* $\rangle$

Concept Learning: Search in Hypothesis Space

Discussion of the Find-S Algorithm

1. Did we learn the only concept—or are there others?
2. Why should one pursuit the maximally specific hypothesis?
3. What if several maximally specific hypotheses exist?
4. Inconsistencies in the example set D remain undetected.
5. An inappropriate hypothesis structure or space H remains undetected.

Concept Learning: Search in Version Space

Definition 5 (Version Space)

The version space $V_{H,D}$ of an hypothesis space H and a example set D is comprised of all hypotheses $h \in H$ that are consistent with a set D of examples:

$$V_{H,D} = \{h \mid h \in H \ \wedge \ (\ \forall(\mathbf{x}, c(\mathbf{x})) \in D: \ h(\mathbf{x}) = c(\mathbf{x}) \ ) \ \}$$

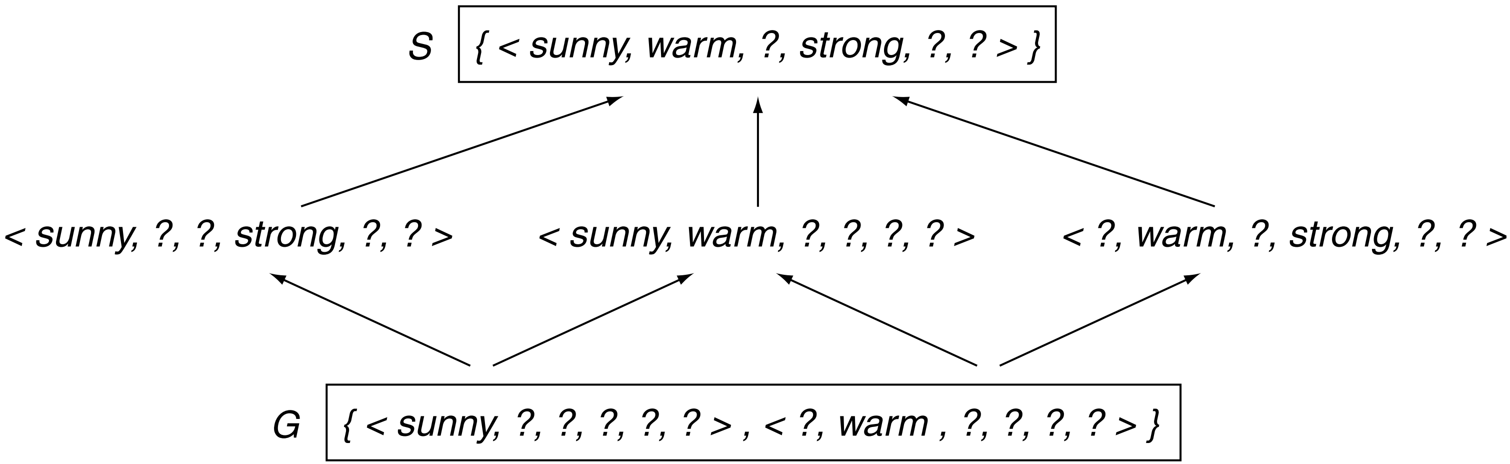
Concept Learning: Search in Version Space

Definition 5 (Version Space)

The version space $V_{H,D}$ of an hypothesis space H and a example set D is comprised of all hypotheses $h \in H$ that are consistent with a set D of examples:

$$V_{H,D} = \{h \mid h \in H \ \wedge \ (\ \forall(\mathbf{x}, c(\mathbf{x})) \in D : \ h(\mathbf{x}) = c(\mathbf{x}) \ ) \ \}$$

Illustration of $V_{H,D}$ for the example set D:



Remarks:

- The term “version space” reflects the fact that $V_{H,D}$ represents the set of all consistent versions of the target concept that is encoded in D.
- A naive approach for the construction of the version space is the following: (1) enumeration of all members of H, and, (2) elimination of those $h \in H$ for which $h(\mathbf{x}) \neq c(\mathbf{x})$ holds. This approach presumes a finite hypothesis space H and is feasible only for toy problems.

Concept Learning: Search in Version Space

Definition 6 (Boundary Sets of a Version Space)

Let H be hypothesis space and let D be set of examples. Then, based on the $\geq_g$-relation, the set of maximally general hypotheses, G, is defined as follows:

$$\{g \mid g \in H \;\wedge\; \textit{consistent}(g, D) \;\wedge\; (\nexists g' : g' \in \; H \;\wedge\; g' >_g \; g \;\wedge\; \textit{consistent}(g', D))\;\}$$

Similarly, the set of maximally specific (i.e., minimally general) hypotheses, S, is defined as follows:

$$\{s \mid s \in H \;\wedge\; \textit{consistent}(s, D) \;\wedge\; (\nexists s' : s' \in \; H \;\wedge\; s >_g \; s' \;\wedge\; \textit{consistent}(s', D))\;\}$$

Concept Learning: Search in Version Space

Definition 6 (Boundary Sets of a Version Space)

Let H be hypothesis space and let D be set of examples. Then, based on the $\geq_g$-relation, the set of maximally general hypotheses, G, is defined as follows:

$$\{g \mid g \in H \; \wedge \; \textit{consistent}(g, D) \; \wedge \; (\nexists g' : g' \in \; H \; \wedge \; g' >_g \; g \; \wedge \; \textit{consistent}(g', D)) \}$$

Similarly, the set of maximally specific (i.e., minimally general) hypotheses, S, is defined as follows:

$$\{s \mid s \in H \; \wedge \; \textit{consistent}(s, D) \; \wedge \; (\nexists s' : s' \in \; H \; \wedge \; s >_g \; s' \; \wedge \; \textit{consistent}(s', D)) \}$$

Theorem 1 (Version Space Representation)

Let X be a feature space and let H be a set of boolean-valued functions with domain X. Moreover, let $c : X \rightarrow \{0, 1\}$ be a target concept and let D be a set of examples of the form $(\mathbf{x}, c(\mathbf{x}))$. Then, based on the $\geq_g$-relation, each member of the version space $V_{H,D}$ lies in between two members of G and S respectively:

$$V_{H,D} = \{h \mid h \in H \; \wedge \; (\exists g \in G \; \exists s \in S : g \geq_g h \geq_g s) \}$$

Concept Learning: Search in Version Space

Candidate Elimination Algorithm [Mitchell 1997]

- Initialization: $G = \{g_0\}, \;\; S = \{s_0\}$
- If $\mathbf{x}$ is a positive example
  - Remove from G any hypothesis that is not consistent with $\mathbf{x}$
  - For each hypothesis s in S that is not consistent with $\mathbf{x}$
    - Remove s from S
    - Add to S all minimal generalizations h of s such that
      1. h is consistent with $\mathbf{x}$ and
      2. some member of G is more general than h
    - Remove from S any hypothesis that is less specific than another hypothesis in S
- If $\mathbf{x}$ is a negative example
  - Remove from S any hypothesis that is not consistent with $\mathbf{x}$
  - For each hypothesis g in G that is not consistent with $\mathbf{x}$
    - Remove g from G
    - Add to G all minimal specializations h of g such that
      1. h is consistent with $\mathbf{x}$ and
      2. some member of S is more specific than h
    - Remove from G any hypothesis that is less general than another hypothesis in G

Remarks:

- The basic idea of Candidate Elimination is as follows.
  - A maximally specific hypothesis $s \in S$ restricts the positive examples in first instance. Hence, s must be relaxed (= generalized) with regard to each positive example that is not consistent with s.
  - Conversely, a maximally general hypothesis $g \in G$ tolerates the negative examples in first instance. Hence, g must be constrained (= specialized) with regard to each negative example that is not consistent with g.

Concept Learning: Search in Version Space

Candidate Elimination Algorithm (pseudo code)

```
1. G = {g_0}  // G is the set of maximally general hypothesis in H.
   S = {s_0}  // S is the set of maximally specific hypothesis in H.

2. FOREACH (x, c(x)) ∈ D DO
     IF c(x) = 1 THEN  // x is a positive example.
       FOREACH g ∈ G DO IF g(x) ≠ 1 THEN G = G \ {g} ENDDO
       FOREACH s ∈ S DO
         IF s(x) ≠ 1 THEN
           S = S \ {s},  S⁺ = min_generalizations(s, x)
           FOREACH s ∈ S⁺ DO IF (∃g ∈ G : g ≥_g s) THEN S = S ∪ {s} ENDDO
           FOREACH s ∈ S DO IF (∃s' ∈ S : s' ≠ s ∧ s' ≥_g s) THEN S = S \ {s} ENDDO
       ENDDO
     ELSE  // x is a negative example.

     ENDIF
   ENDDO

3. return(G, S)
```

Concept Learning: Search in Version Space

Candidate Elimination Algorithm (pseudo code)

```
1. G = {g_0}  // G is the set of maximally general hypothesis in H.
   S = {s_0}  // S is the set of maximally specific hypothesis in H.

2. FOREACH (x, c(x)) ∈ D DO
     IF c(x) = 1 THEN  // x is a positive example.
       FOREACH g ∈ G DO IF g(x) ≠ 1 THEN G = G \ {g} ENDDO
       FOREACH s ∈ S DO
         IF s(x) ≠ 1 THEN
           S = S \ {s},  S+ = min_generalizations(s, x)
           FOREACH s ∈ S+ DO IF (∃g ∈ G : g ≥_g s) THEN S = S ∪ {s} ENDDO
           FOREACH s ∈ S DO IF (∃s' ∈ S : s' ≠ s ∧ s' ≥_g s) THEN S = S \ {s} ENDDO
       ENDDO
     ELSE  // x is a negative example.
       FOREACH s ∈ S DO IF s(x) ≠ 0 THEN S = S \ {s} ENDDO
       FOREACH g ∈ G DO
         IF g(x) ≠ 0 THEN
           G = G \ {g},  G- = min_specializations(g, x)
           FOREACH g ∈ G- DO IF (∃s ∈ S : g ≥_g s) THEN G = G ∪ {g} ENDDO
           FOREACH g ∈ G DO IF (∃g' ∈ G : g' ≠ g ∧ g ≥_g g') THEN G = G \ {g} ENDDO
       ENDDO
     ENDIF
   ENDDO

3. return(G, S)
```

Concept Learning: Search in Version Space

Candidate Elimination Algorithm (illustration)

$\{ < \perp, \perp, \perp, \perp, \perp, \perp > \}$ S_0

$\{ < ?, ?, ?, ?, ?, ? > \}$ G_0,

Concept Learning: Search in Version Space

Candidate Elimination Algorithm (illustration)

$\{ < \bot, \bot, \bot, \bot, \bot, \bot > \}$ S_0

↓

{ < *sunny, warm, normal, strong, warm, same* > } S_1

{ < ?, ?, ?, ?, ?, ? > } G_0, G_1,

$\mathbf{x}_1 =$ (*sunny, warm, normal, strong, warm, same*) *EnjoySport*$(\mathbf{x}_1) = 1$

Concept Learning: Search in Version Space

Candidate Elimination Algorithm (illustration)

{ < ⊥, ⊥, ⊥, ⊥, ⊥, ⊥ > } S_0

↓

{ < *sunny, warm, normal, strong, warm, same* > } S_1

↓

{ < *sunny, warm, ?, strong, warm, same* > } S_2,

{ < ?, ?, ?, ?, ?, ? > } G_0, G_1, G_2

$\mathbf{x}_1 = ($*sunny, warm, normal, strong, warm, same*$)$ *EnjoySport*$(\mathbf{x}_1) = 1$

$\mathbf{x}_2 = ($*sunny, warm, high, strong, warm, same*$)$ *EnjoySport*$(\mathbf{x}_2) = 1$

Concept Learning: Search in Version Space

Candidate Elimination Algorithm (illustration)

{ < $\bot$, $\bot$, $\bot$, $\bot$, $\bot$, $\bot$ > } S_0

↓

{ < *sunny, warm, normal, strong, warm, same* > } S_1

↓

{ < *sunny, warm, ?, strong, warm, same* > } S_2, S_3

{ < *sunny, ?, ?, ?, ?, ?* >, < *?, warm, ?, ?, ?, ?* >, < *?, ?, ?, ?, ?, same* > } G_3

↑

{ < *?, ?, ?, ?, ?, ?* > } G_0, G_1, G_2

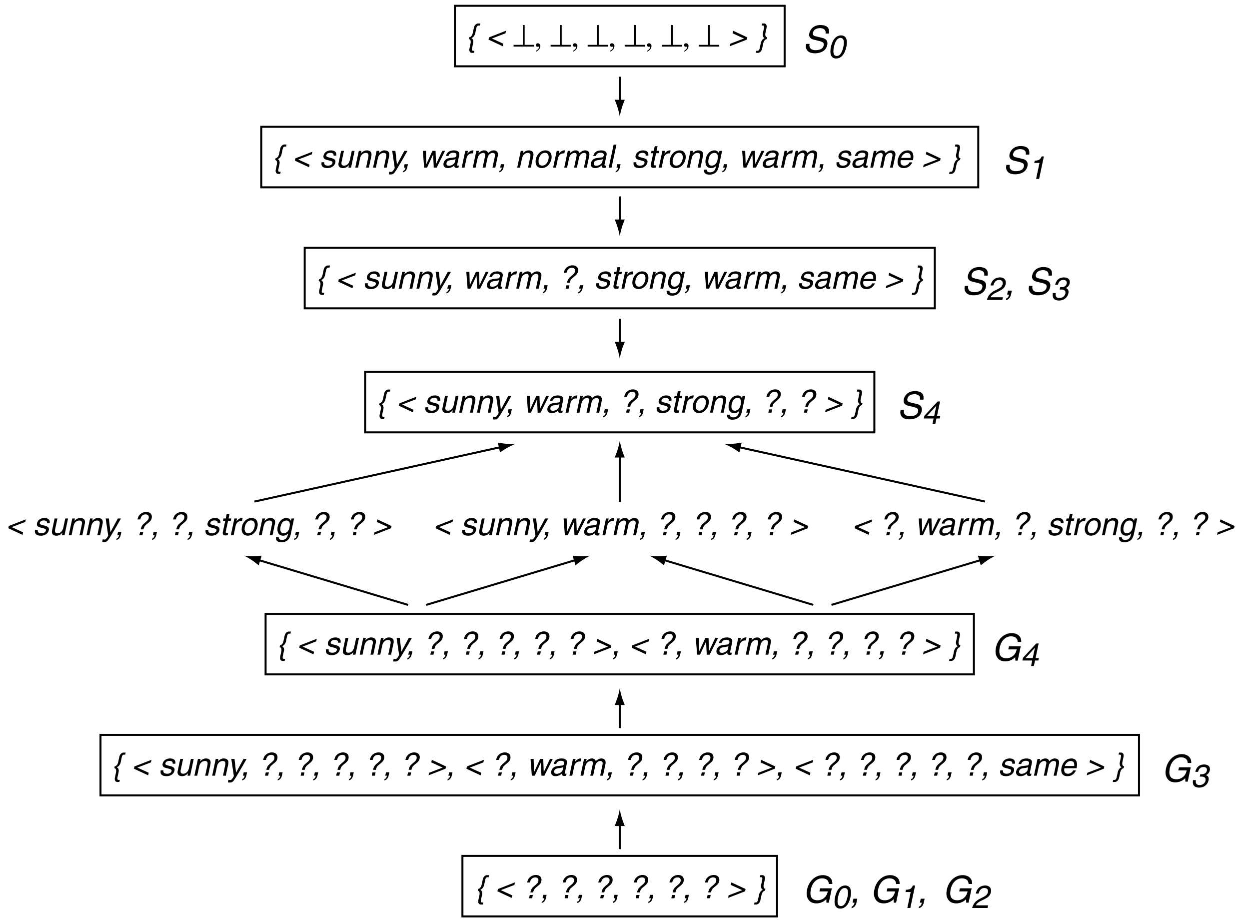
$\mathbf{x}_1 =$ (*sunny, warm, normal, strong, warm, same*) *EnjoySport*$(\mathbf{x}_1) = 1$

$\mathbf{x}_2 =$ (*sunny, warm, high, strong, warm, same*) *EnjoySport*$(\mathbf{x}_2) = 1$

$\mathbf{x}_3 =$ (*rainy, cold, high, strong, warm, change*) *EnjoySport*$(\mathbf{x}_3) = 0$

Concept Learning: Search in Version Space

Candidate Elimination Algorithm (illustration)



$\mathbf{x}_1 = ($*sunny, warm, normal, strong, warm, same*$)$ *EnjoySport*$(\mathbf{x}_1) = 1$

$\mathbf{x}_2 = ($*sunny, warm, high, strong, warm, same*$)$ *EnjoySport*$(\mathbf{x}_2) = 1$

$\mathbf{x}_3 = ($*rainy, cold, high, strong, warm, change*$)$ *EnjoySport*$(\mathbf{x}_3) = 0$

$\mathbf{x}_4 = ($*sunny, warm, high, strong, cool, change*$)$ *EnjoySport*$(\mathbf{x}_4) = 1$

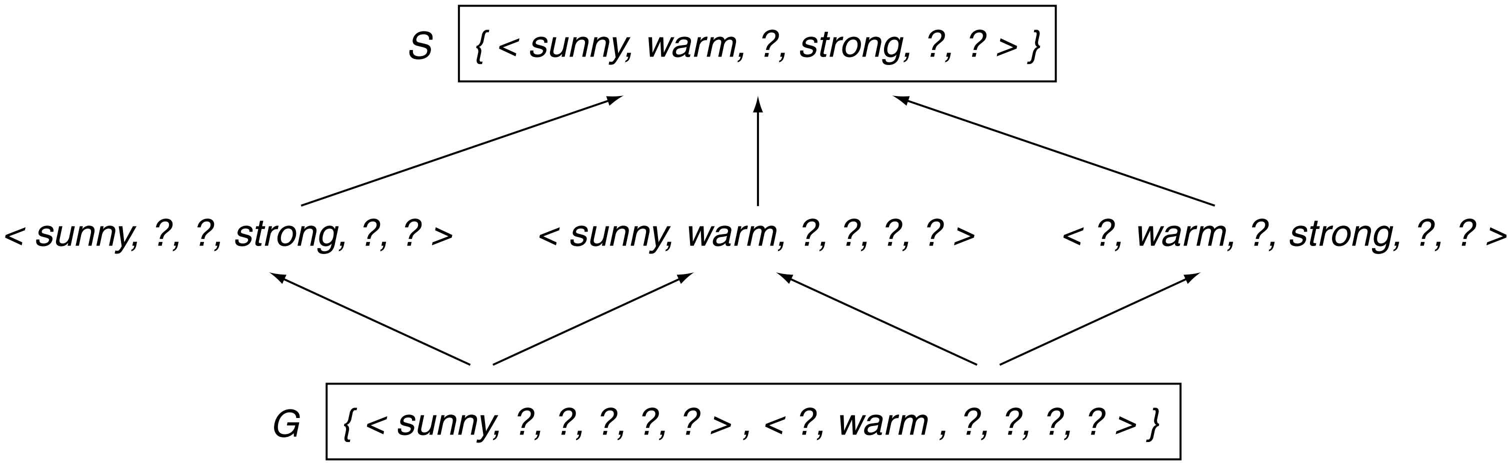
Concept Learning: Search in Version Space

Discussion of the Candidate Elimination Algorithm

1. What about selecting examples from D according to a certain strategy?
   Keyword: Active Learning
2. What are partially learned concepts and how to exploit them?
   Keyword: Ensemble Classification
3. The version space as defined here is “biased”. What does this mean?
4. Will Candidate Elimination converge towards the correct hypothesis?
5. When does one end up with an empty version space?

Concept Learning: Search in Version Space

Question 1: Selecting Examples from D

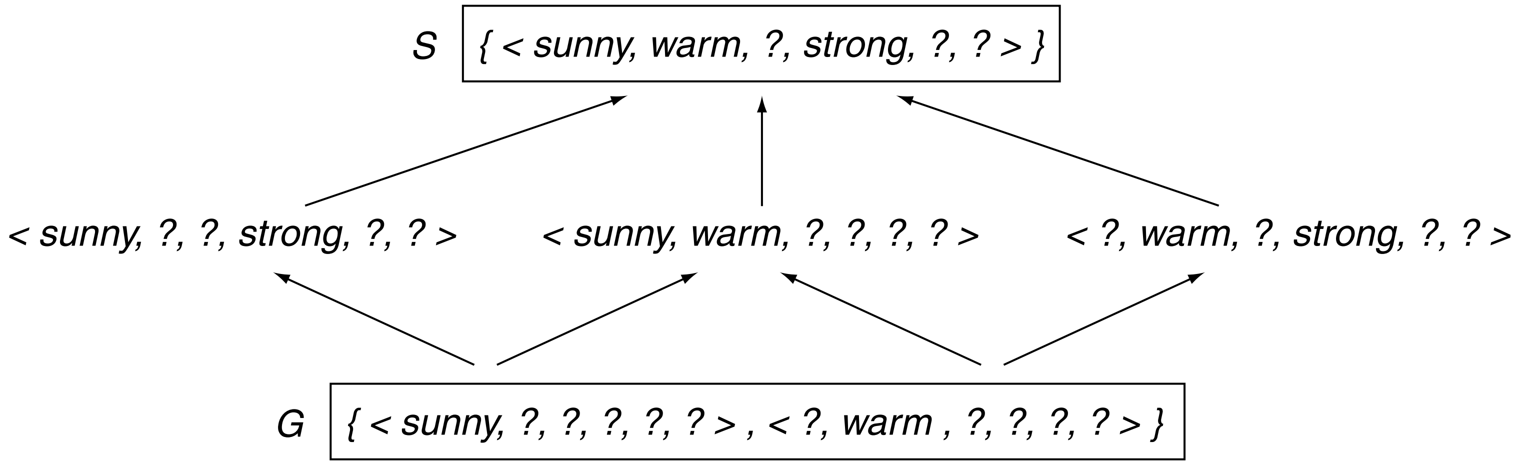


Next Example:

$\mathbf{x}_c = ($*sunny, warm, normal, light, warm, same*$)$

Concept Learning: Search in Version Space

Question 1: Selecting Examples from D



Next Example:

$\mathbf{x}_c = ($*sunny, warm, normal, light, warm, same*$)$

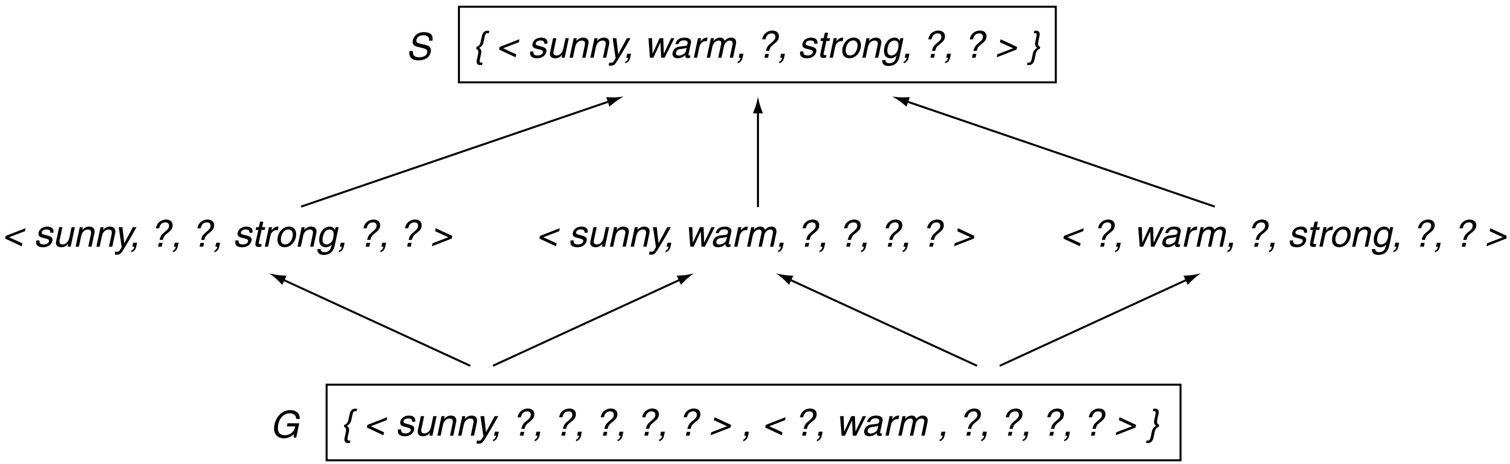
Observation:

Irrespective the value of $c(\mathbf{x}_c)$, the example $(\mathbf{x}_c, c(\mathbf{x}_c))$ will be consistent with three of the six hypotheses. It follows:

- If *EnjoySport*$(\mathbf{x}_c) = 1$ S can be further generalized.
- If *EnjoySport*$(\mathbf{x}_c) = 0$ G can be further specialized.

Concept Learning: Search in Version Space

Question 2: Partially Learned Concepts

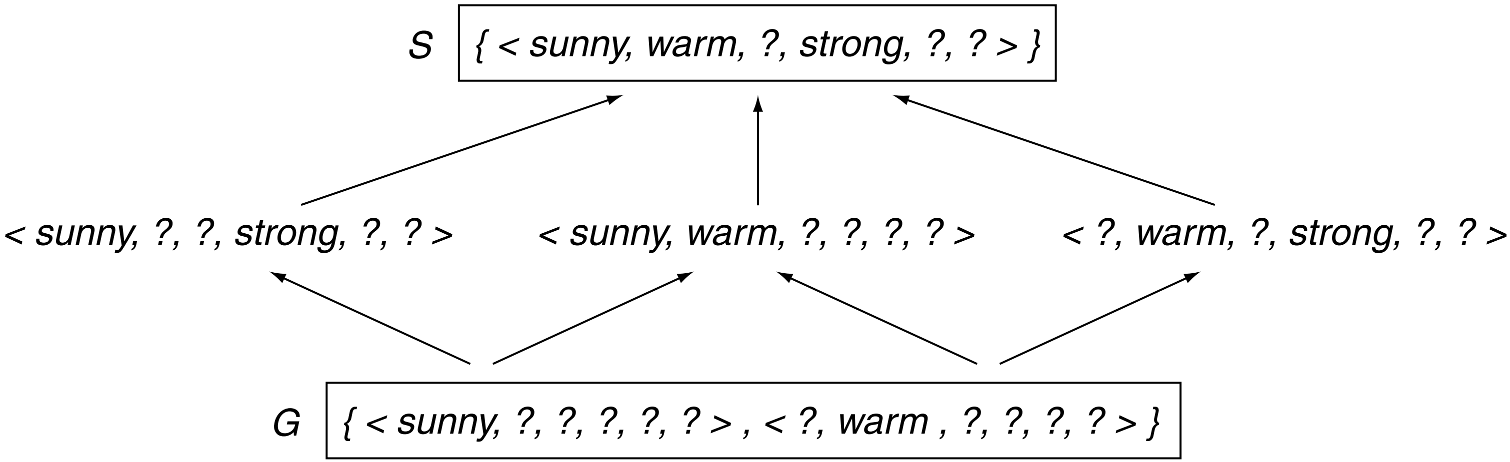


Classify examples using the shown version space:

Example	Sky	Temperature	Humidity	Wind	Water	Forecast	EnjoySport
(a)	sunny	warm	normal	strong	cool	change	
(b)	rainy	cold	normal	light	warm	same	
(c)	sunny	warm	normal	light	warm	same	
(d)	sunny	cold	normal	strong	warm	same	

Concept Learning: Search in Version Space

Question 2: Partially Learned Concepts

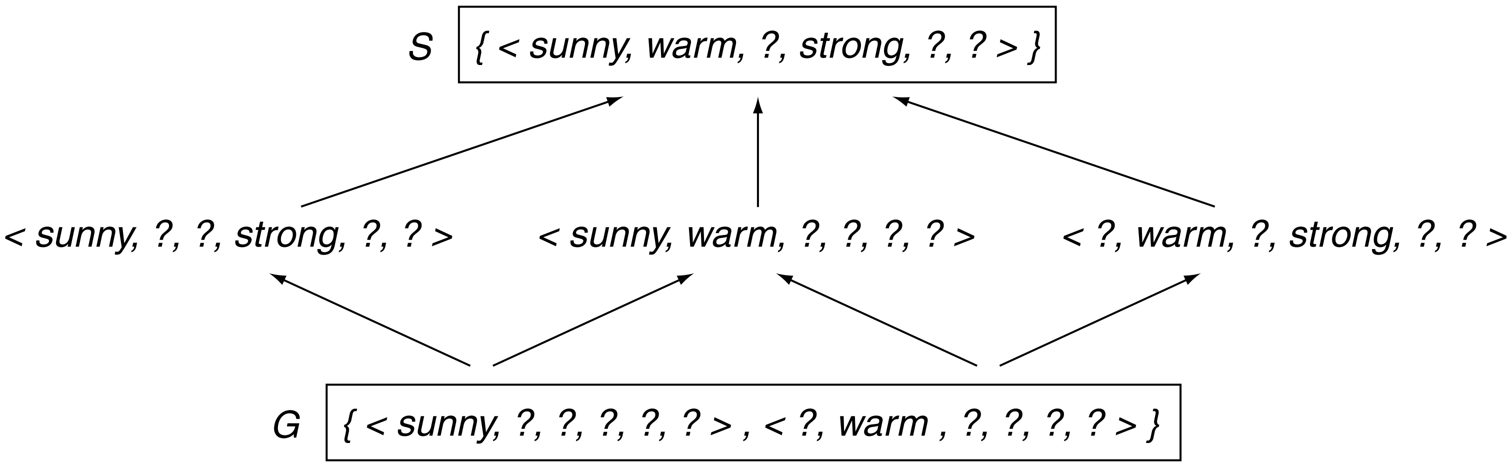


Classify examples using the shown version space:

Example	Sky	Temperature	Humidity	Wind	Water	Forecast	EnjoySport
(a)	sunny	warm	normal	strong	cool	change	6+ : 0–
(b)	rainy	cold	normal	light	warm	same	
(c)	sunny	warm	normal	light	warm	same	
(d)	sunny	cold	normal	strong	warm	same	

Concept Learning: Search in Version Space

Question 2: Partially Learned Concepts

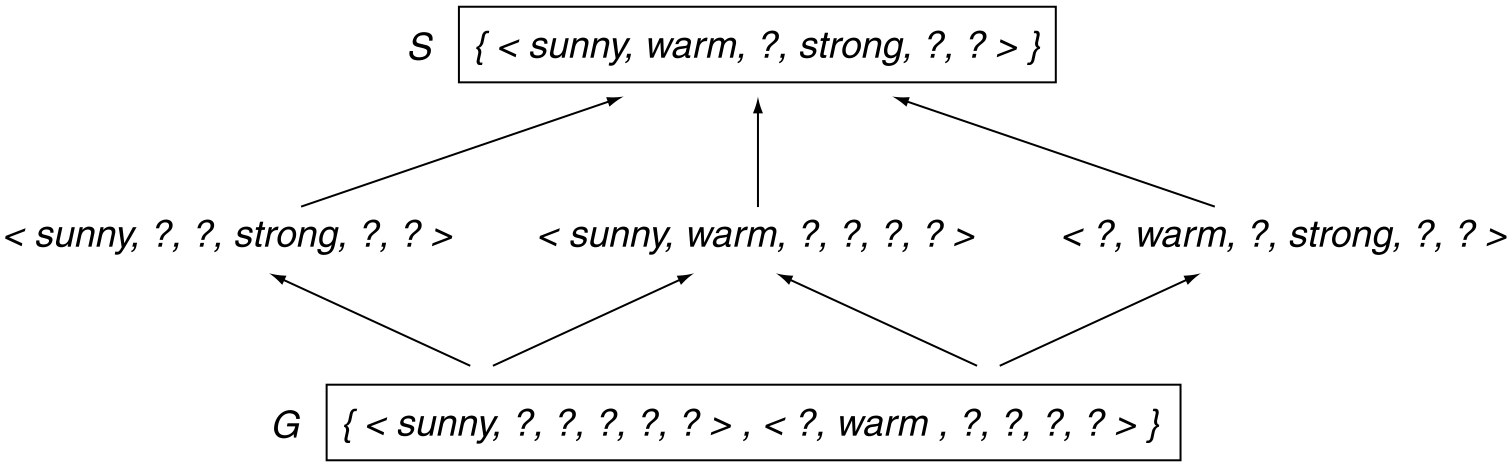


Classify examples using the shown version space:

Example	Sky	Temperature	Humidity	Wind	Water	Forecast	EnjoySport
(a)	sunny	warm	normal	strong	cool	change	6+ : 0–
(b)	rainy	cold	normal	light	warm	same	0+ : 6–
(c)	sunny	warm	normal	light	warm	same	
(d)	sunny	cold	normal	strong	warm	same	

Concept Learning: Search in Version Space

Question 2: Partially Learned Concepts

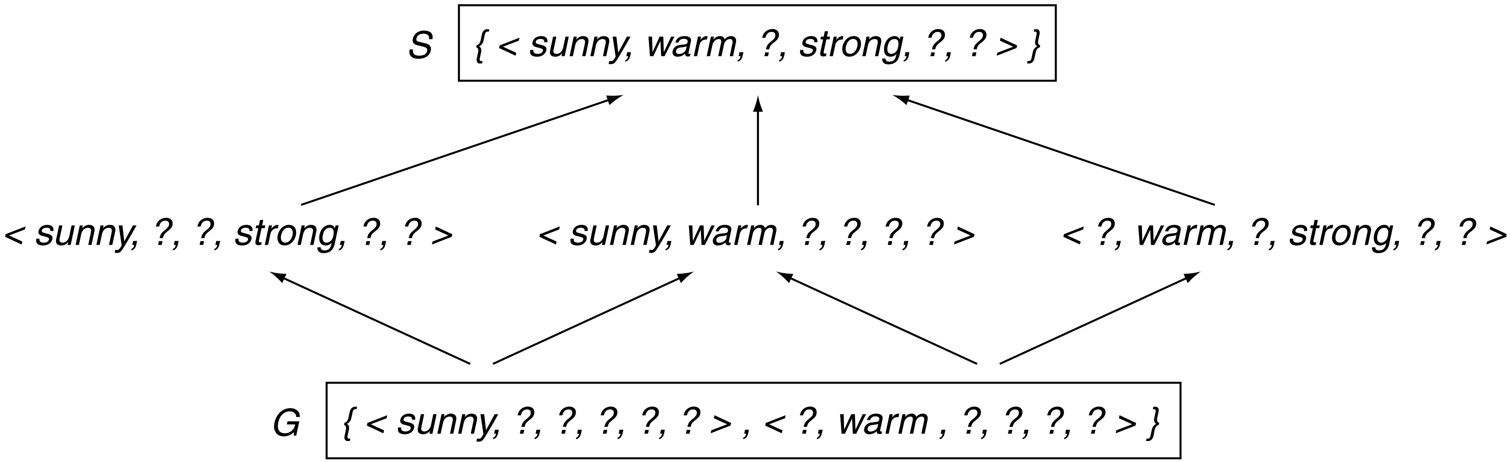


Classify examples using the shown version space:

Example	Sky	Temperature	Humidity	Wind	Water	Forecast	EnjoySport
(a)	sunny	warm	normal	strong	cool	change	6+ : 0–
(b)	rainy	cold	normal	light	warm	same	0+ : 6–
(c)	sunny	warm	normal	light	warm	same	3+ : 3–
(d)	sunny	cold	normal	strong	warm	same	

Concept Learning: Search in Version Space

Question 2: Partially Learned Concepts



Classify examples using the shown version space:

Example	Sky	Temperature	Humidity	Wind	Water	Forecast	EnjoySport
(a)	sunny	warm	normal	strong	cool	change	6+ : 0–
(b)	rainy	cold	normal	light	warm	same	0+ : 6–
(c)	sunny	warm	normal	light	warm	same	3+ : 3–
(d)	sunny	cold	normal	strong	warm	same	2+ : 4–

Concept Learning: Search in Version Space

Question 3: Inductive Bias

A different set of examples D' :

Example	Sky	Temperature	Humidity	Wind	Water	Forecast	EnjoySport
1	sunny	warm	normal	strong	cool	change	yes
2	sunny	warm	normal	light	warm	same	yes

➜ $S = \{ \langle$ *sunny, warm, normal, ?, ?, ?* $\rangle \}$

Concept Learning: Search in Version Space

Question 3: Inductive Bias

A different set of examples D' :

Example	Sky	Temperature	Humidity	Wind	Water	Forecast	EnjoySport
1	sunny	warm	normal	strong	cool	change	yes
2	sunny	warm	normal	light	warm	same	yes

➜ $S = \{ \langle$ *sunny, warm, normal, ?, ?, ?* $\rangle \}$

In particular, the following example is classified as positive:

$\mathbf{x} = ($*sunny, warm, normal, strong, warm, same*$)$

Discussion:

- What if $\mathbf{x}$ were a negative example?
- What assumptions about the target concept are met a-priori by the learner?

Concept Learning: Search in Version Space

Question 3: Inductive Bias (continued)

A different set of examples D'' :

Example	Sky	Temperature	Humidity	Wind	Water	Forecast	EnjoySport
1	sunny	warm	normal	strong	cool	change	yes
2	cloudy	warm	normal	strong	cool	change	yes

➜ $S = \{ \langle$ *?, warm, normal, strong, cool, change* $\rangle \}$

Concept Learning: Search in Version Space

Question 3: Inductive Bias (continued)

A different set of examples D'' :

Example	Sky	Temperature	Humidity	Wind	Water	Forecast	EnjoySport
1	sunny	warm	normal	strong	cool	change	yes
2	cloudy	warm	normal	strong	cool	change	yes

➜ $S = \{ \langle$ *?, warm, normal, strong, cool, change* $\rangle \}$

+

3	rainy	warm	normal	strong	cool	change	no

➜ $S = \{ \}$

Concept Learning: Search in Version Space

Question 3: Inductive Bias (continued)

A different set of examples D'' :

Example	Sky	Temperature	Humidity	Wind	Water	Forecast	EnjoySport
1	sunny	warm	normal	strong	cool	change	yes
2	cloudy	warm	normal	strong	cool	change	yes

➜ $S = \{ \langle$ *?, warm, normal, strong, cool, change* $\rangle \}$

+

3	rainy	warm	normal	strong	cool	change	no

➜ $S = \{ \}$

Observation:

The hypothesis space H should be constructed in a way to contain other possible concepts, e.g. including:

$\langle$ *sunny, ?, ?, ?, ?, ?* $\rangle \vee \langle$ *cloudy, ?, ?, ?, ?, ?* $\rangle$

Concept Learning: Search in Version Space

Question 3: Inductive Bias (continued)

- A learning algorithm that considers all possible hypotheses as equally likely makes no a-priori assumption with regard to the target concept.

- A learning algorithm without a-priori assumptions has no “inductive bias”.

> *“The policy by which an algorithm generalizes from observed training examples to classify unseen instances is its inductive bias. [. . .] Inductive bias is the set of assumptions that, together with the training data, deductively justify the classification by the learner to future instances.”*

[p.63, Mitchell 1997]

Concept Learning: Search in Version Space

Question 3: Inductive Bias (continued)

- A learning algorithm that considers all possible hypotheses as equally likely makes no a-priori assumption with regard to the target concept.
- A learning algorithm without a-priori assumptions has no “inductive bias”.

  “The policy by which an algorithm generalizes from observed training examples to classify unseen instances is its inductive bias. [. . .] Inductive bias is the set of assumptions that, together with the training data, deductively justify the classification by the learner to future instances.”

  [p.63, Mitchell 1997]

➜ A learning algorithm without inductive bias has no directive to classify unseen examples. Put another way: the learner cannot *generalize*.

➜ A learning algorithm without inductive bias will only *memorize*.

Which of the two algorithms Finds-S and Candidate Elimination has a stronger inductive bias?

Chapter ML:II (continued)

II. Machine Learning Basics

- On Data
- Regression
- Concept Learning: Search in Hypothesis Space
- Concept Learning: Search in Version Space
- Measuring Performance

Measuring Performance

Misclassification

Definition 7 (True Misclassification Rate)

Let X be a feature space with a finite number of elements. Moreover, let C be a set of classes, let $y : X \to C$ be a classifier, and let c be the target concept to be learned. Then the true misclassification rate, denoted as $\mathit{Err}^*(y)$, is defined as follows:

$$\mathit{Err}^*(y) = \frac{|\{\mathbf{x} \in X : c(\mathbf{x}) \neq y(\mathbf{x})\}|}{|X|}$$

Measuring Performance

Misclassification

Definition 7 (True Misclassification Rate)

Let X be a feature space with a finite number of elements. Moreover, let C be a set of classes, let $y : X \to C$ be a classifier, and let c be the target concept to be learned. Then the true misclassification rate, denoted as $\mathit{Err}^*(y)$, is defined as follows:

$$\mathit{Err}^*(y) = \frac{|\{\mathbf{x} \in X : c(\mathbf{x}) \neq y(\mathbf{x})\}|}{|X|}$$

Problem:

- Usually c is unknown.

Solution:

- Estimation of $\mathit{Err}^*(y)$, by evaluating y on a test set for whose feature vectors c is known.

Remarks:

- The English word “rate” can be used to denote both the mathematical concept of a flow figure (a change of a quantity per time unit) as well as the mathematical concept of a portion, a percentage, or a ratio, which has a stationary (= time-independent) semantics. This latter semantics is meant here when talking about the misclassification rate.
- Unfortunately, the German word “Rate” is often (mis)used to denote the mathematical concept of a portion, a percentage, or a ratio. Taking a precise mathematical standpoint, the correct German words are “Anteil” or “Quote”. I.e., a semantically correct translation of misclassification rate is “Missklassifikationsanteil”, and not “Missklassifikationsrate”.

Measuring Performance

Misclassification (continued)

Probabilistic foundation:

- For $X' \subseteq X$ and $j \in C$ let $P(X', j)$ denote the probability that a randomly drawn $\mathbf{x} \in X$ is member of X' and belongs to class j.
- P is a probability measure on $X \times C$.
- $D = \{(\mathbf{x}_1, c(\mathbf{x}_1)), \ldots, (\mathbf{x}_n, c(\mathbf{x}_n))\} \subseteq X \times C$ is a set of examples whose elements are drawn independently from each other and according to the (same, identical) distribution P.

Let $(\mathbf{x}_0, c(\mathbf{x}_0))$ be an example drawn according to P and independently of D, and let $y : X \to C$ be a classifier learned on the basis of D. Then we agree on:

1. $P(\mathbf{x}_0 \in X', c(\mathbf{x}_0) = j) \;=\; P(X', j)$
2. *Err*$^*(y) = P(c(\mathbf{x}_0) \neq y(\mathbf{x}_0) \mid D)$

Remarks:

- Let A and B denote two events. Then the following expressions are syntactic variants, i.e., they are semantically equivalent: $P(A, B)$, $P(A$ and $B)$, $P(A \wedge B)$

- The elements in D are considered as random variables that are both independent of each other and identically distributed. This property of a set of random variables is abbreviated with "i.i.d."

Measuring Performance

Misclassification (continued)

Estimation of $\mathit{Err}^*(y)$ based on a finite sample $X' \subseteq X$:

$$\mathit{Err}(y, X') = \frac{|\{\mathbf{x} \in X' : c(\mathbf{x}) \neq y(\mathbf{x})\}|}{|X'|}$$

Requirement:

- $c(\mathbf{x})$ must be known. Hence, $X' \subseteq \{\mathbf{x} \in X : (\mathbf{x}, c(\mathbf{x})) \in D\}$

Measuring Performance

Training Error

- $D = \{(\mathbf{x}_1, c(\mathbf{x}_1)), \ldots, (\mathbf{x}_n, c(\mathbf{x}_n))\} \subseteq X \times C$ is a set of examples.
- $D_{tr} = D$ is the training set.
- $y : X \to C$ is a classifier learned on the basis of D_{tr}.

Training error = misclassification rate with respect to D_{tr}:

$$\mathit{Err}(y, D_{tr}) = \frac{|\{(\mathbf{x}, c(\mathbf{x})) \in D_{tr} : c(\mathbf{x}) \neq y(\mathbf{x})\}|}{|D_{tr}|}$$

Measuring Performance

Training Error

- $D = \{(\mathbf{x}_1, c(\mathbf{x}_1)), \ldots, (\mathbf{x}_n, c(\mathbf{x}_n))\} \subseteq X \times C$ is a set of examples.
- $D_{tr} = D$ is the training set.
- $y : X \to C$ is a classifier learned on the basis of D_{tr}.

Training error = misclassification rate with respect to D_{tr}:

$$\textit{Err}(y, D_{tr}) = \frac{|\{(\mathbf{x}, c(\mathbf{x})) \in D_{tr} : c(\mathbf{x}) \neq y(\mathbf{x})\}|}{|D_{tr}|}$$

Problem:

- $\textit{Err}(y, D_{tr})$ is based on examples that are also exploited to learn y.

→ $\textit{Err}(y, D_{tr})$ quantifies memorization but not generalization capability of y.

→ $\textit{Err}(y, D_{tr})$ is an optimistic estimation, i.e.,
it is constantly lower compared to an application of y in the wild.

Measuring Performance

Holdout Estimation

- $D = \{(\mathbf{x}_1, c(\mathbf{x}_1)), \ldots, (\mathbf{x}_n, c(\mathbf{x}_n))\} \subseteq X \times C$ is a set of examples.
- $D_{tr} \subset D$ is the training set.
- $y : X \to C$ is a classifier learned on the basis of D_{tr}.
- $D_{ts} \subset D$ with $D_{ts} \cap D_{tr} = \emptyset$ is a test set.

Holdout estimation = misclassification rate with respect to D_{ts} :

$$\textit{Err}(y, D_{ts}) = \frac{|\{(\mathbf{x}, c(\mathbf{x})) \in D_{ts} : c(\mathbf{x}) \neq y(\mathbf{x})\}|}{|D_{ts}|}$$

Measuring Performance

Holdout Estimation

- $D = \{(\mathbf{x}_1, c(\mathbf{x}_1)), \ldots, (\mathbf{x}_n, c(\mathbf{x}_n))\} \subseteq X \times C$ is a set of examples.
- $D_{tr} \subset D$ is the training set.
- $y : X \rightarrow C$ is a classifier learned on the basis of D_{tr}.
- $D_{ts} \subset D$ with $D_{ts} \cap D_{tr} = \emptyset$ is a test set.

Holdout estimation = misclassification rate with respect to D_{ts}:

$$\textit{Err}(y, D_{ts}) = \frac{|\{(\mathbf{x}, c(\mathbf{x})) \in D_{ts} : c(\mathbf{x}) \neq y(\mathbf{x})\}|}{|D_{ts}|}$$

Requirements:

- D_{tr} and D_{ts} must be drawn i.i.d.
- D_{tr} and D_{ts} must have similar sizes.

Remarks:

- A typical value for splitting D into training set D_{tr} and test set D_{ts} is 2:1.
- When splitting D into D_{tr} and D_{ts} one has to ensure that the underlying distribution is maintained. Keywords: Stratification, Sample Selection Bias

Measuring Performance

Cross Validation: k-Fold

Improved approach for small sets D :

- Splitting of D into k disjoint sets $D_1, \ldots, D_k$ of similar size.
- For $i = 1, \ldots, k$ do:
  1. $y_i : X \to C$ is a classifier learned on the basis of $D \setminus D_i$
  2. *Err*$(y_i, D_i) = \dfrac{|\{(\mathbf{x}, c(\mathbf{x}) \in D_i : y_i(\mathbf{x}) \neq c(\mathbf{x})\}|}{|D_i|}$
- $y : X \to C$ is a classifier learned on the basis of D.

Cross-validated misclassification rate:

$$\textit{Err}_{cv}(y, D, k) = \frac{1}{k} \sum_{i=1}^{k} \textit{Err}(y_i, D_i)$$

Remarks:

- Rationale: For large k the set $D \setminus D_i$ is of similar size as D. Hence *Err*$^*(y_i)$ is close to *Err*$^*(y)$.
- For the construction of tree classifiers, tenfold cross-validation has been reported to give good results. [Breiman]

Measuring Performance

Cross Validation: Leave One Out

The special case of cross validation with $k = n$:

- Determine the cross-validated misclassification rate for $D_i = D \setminus \{(\mathbf{x}_i, c(\mathbf{x}_i))\}$, $k \in \{1, \ldots, n\}$.

Measuring Performance

Cross Validation: Leave One Out

The special case of cross validation with $k = n$:

- Determine the cross-validated misclassification rate for $D_i = D \setminus \{(\mathbf{x}_i, c(\mathbf{x}_i))\}$, $k \in \{1, \ldots, n\}$.

Problems:

- High computational effort if D is large.
- Singleton test sets ($|D_i| = 1$) are not stratified but contain only one class.
- Pessimistic error estimation becomes possible.

  Consider the learning algorithm “Majority decision on the basis of D”, which leads for $|C| = 2$ and stratified samples to a misclassification rate of 1.

Measuring Performance

Bootstrapping

The idea of a multiple exploitation of D:

- For $i = 1, \ldots, k$ do:

  1. Form training set D_i by drawing n examples from D *with replacement*.
  2. $y_i : X \to C$ is a classifier learned on the basis of D_i
  3. $\mathit{Err}(y_i, D \setminus D_i) = \dfrac{|\{(\mathbf{x}, c(\mathbf{x})) \in D \setminus D_i : y_i(\mathbf{x}) \neq c(\mathbf{x})\}|}{|D \setminus D_i|}$

- $y : X \to C$ is a classifier learned on the basis of D.

Bootstrapped misclassification rate:

$$\mathit{Err}_{bt}(y, D) = \frac{1}{k} \sum_{i=1}^{k} \mathit{Err}(y_i, D \setminus D_i)$$

Remarks:

- The probability that an example is not considered is $(1 - 1/n)^n$. As a consequence, the probability that an example is considered at least once is $1 - (1 - 1/n)^n$.

- If n is large then $1 - (1 - 1/n)^n \approx 1 - 1/e \approx 0.632$. I.e., each training set contains about 63.2% of the examples in D.

- The classifiers $y_1, \ldots, y_k$ can be used in a combined fashion as *ensemble* where the class is determined by means of a majority decision:

$$y(\mathbf{x}) = \underset{j \in C}{\text{argmax}} \, |\{i \in \{1, \ldots, k\} : y_i(\mathbf{x}) = j\}|$$

- For the construction of tree classifiers, bootstrapping has been reported to improve the misclassification rate about 20% - 47% compared to a standard approach.

Measuring Performance

Misclassification Cost

Use of a cost measure for the misclassification of a feature vector $\mathbf{x}$ in class c' instead of in class c:

$$\textit{cost}(c' \mid c) \begin{cases} \geq 0 & \text{if } c' \neq c \\ = 0 & \text{otherwise} \end{cases}$$

Estimation of $\textit{Err}^{*}_{\textit{cost}}(y)$ given a finite subset $X' \subseteq X$:

$$\textit{Err}_{\textit{cost}}(y, X') = \sum_{\mathbf{x} \in X'} \textit{cost}(y(\mathbf{x}) \mid c(\mathbf{x}))$$

Requirement:

- $c(\mathbf{x})$ must be known. I.e., $X' \subseteq \{\mathbf{x} \in X : (\mathbf{x}, c(\mathbf{x})) \in D\}$

The misclassification rate, *Err*, is a special case of $\textit{Err}_{\textit{cost}}$ with cost 1.