CLASSIFICATION METHODS

Nearest-Neighbor classifiers

instance-based learning

eager learners

(1) constructing a classification model from data
(2) applying the model to test examples

Nearest-Neighbor classifiers

instance-based learning

eager learners

(1) constructing a classification model from data
(2) applying the model to test examples

uses specific training instances to make predictions without having to maintain an abstraction (or model) derived from data

Lazy learners

Nearest-Neighbor classifiers

instance-based learning

eager learners

(1) constructing a classification model from data
(2) applying the model to test examples

uses specific training instances to make predictions without having to maintain an abstraction (or model) derived from data

Lazy learners

Nearest-Neighbor classifiers

instance-based learning

eager learners

(1) constructing a classification model from data
(2) applying the model to test examples

uses specific training instances to make predictions without having to maintain an abstraction (or model) derived from data

Lazy learners

Set of Stored Cases

Atr1	………	AtrN	Class
			A
			B
			B
			C
			A
			C
			B

Unseen Case

Atr1	………	AtrN

Nearest-Neighbor classifiers

Rote-learner

- Memorizes entire training data and performs classification only if attributes of record match one of the training examples exactly
- problem

Nearest-Neighbor classifiers

Rote-learner

- Memorizes entire training data and performs classification only if attributes of record match one of the training examples exactly
- problem

Nearest neighbor

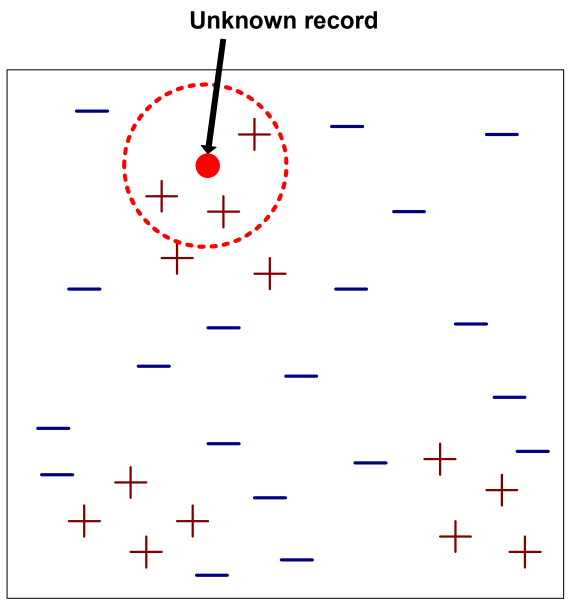
- Uses k “closest” points (nearest neighbors) for performing classification

Nearest Neighbor Classifiers

- Basic idea:
  - If it walks like a duck, quacks like a duck, then it's probably a duck

Nearest-Neighbor Classifiers

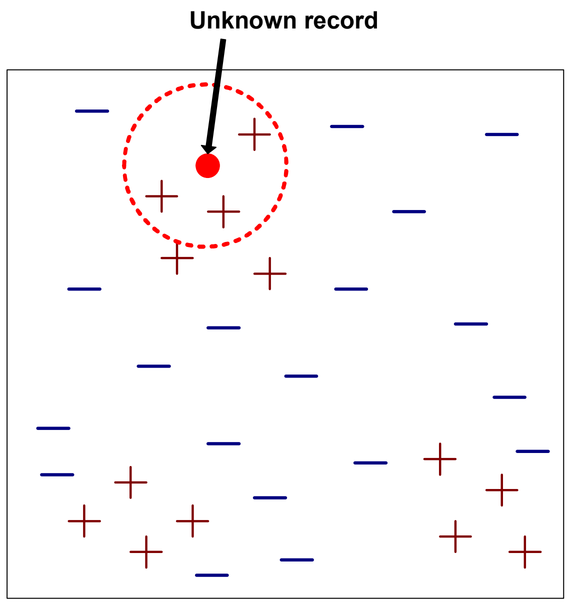
- Requires three things
  - The set of stored records
  - Distance Metric to compute distance between records
  - The value of *k*, the number of nearest neighbors to retrieve



Nearest-Neighbor Classifiers

- Requires three things
  - The set of stored records
  - Distance Metric to compute distance between records
  - The value of *k*, the number of nearest neighbors to retrieve

- To classify an unknown record:
  - Compute distance to other training records
  - Identify *k* nearest neighbors
  - Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)

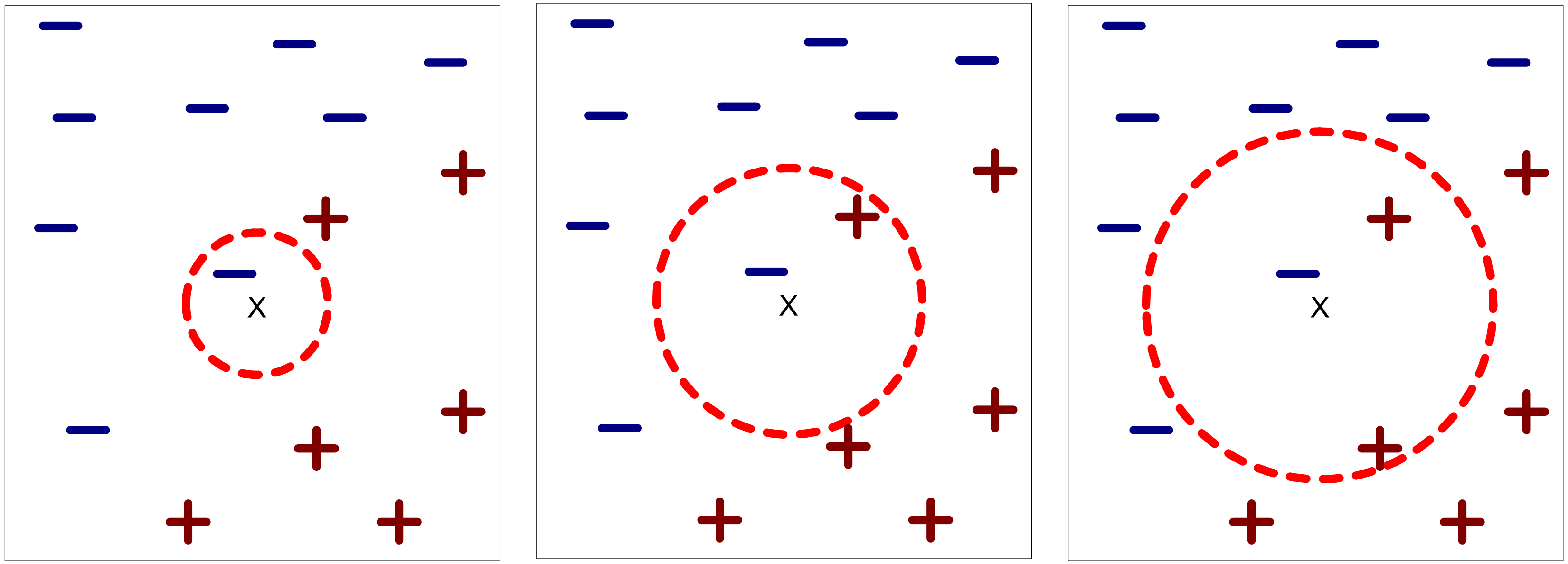


Algorithm 5.2 The k-nearest neighbor classification algorithm.

```
Let k be the number of nearest neighbors and D be the set of training examples.
for each test example z = (x', y') do
    Compute d(x', x), the distance between z and every example, (x, y) ∈ D.
    Select D_z ⊆ D, the set of k closest training examples to z.
    y' = argmax_v Σ_{(x_i, y_i) ∈ D_z} I(v = y_i)
end for
```

$$\text{Majority Voting: } y' = \underset{v}{\operatorname{argmax}} \sum_{(\mathbf{x}_i, y_i) \in D_z} I(v = y_i)$$

Definition of Nearest Neighbor



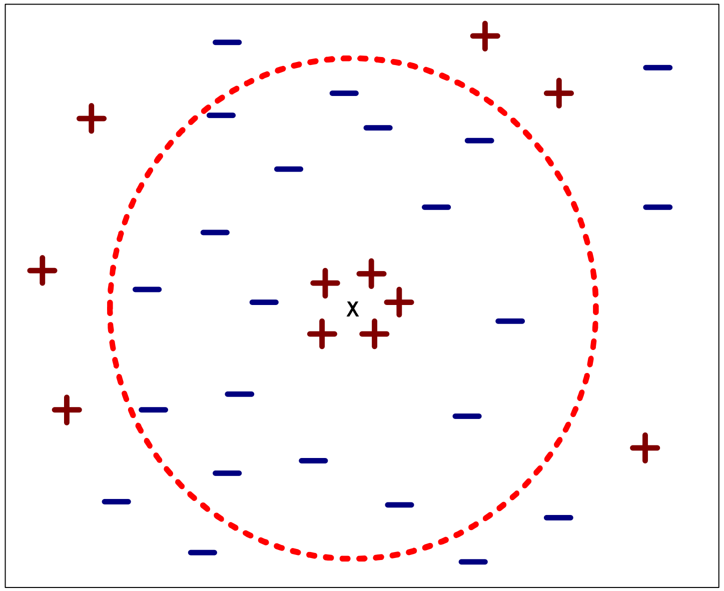
(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

K-nearest neighbors of a record x are data points that have the k smallest distance to x

Nearest-Neighbor Classifiers

Choosing the value of k:

- If k is too small, sensitive to noise points
- If k is too large, neighborhood may include points from other classes



Nearest-Neighbor Classifiers

Distance-Weighted Voting: $y' = \underset{v}{\operatorname{argmax}} \sum_{(\mathbf{x}_i, y_i) \in D_z} w_i \times I(v = y_i)$

$$\boxed{w_i = 1/d(\mathbf{x}', \mathbf{x}_i)^2}$$

Nearest-Neighbor Classifiers

- ✓ Lazy learners
- ✓ instance-based learning, which uses specific training instances to make predictions without having to maintain an abstraction (or model) derived from data.
- ✓ classifying a test example can be quite expensive
- ✓ eager learners often spend the bulk of their computing resources for model building. Once a model has been built, classifying a test example is extremely fast.
- ✓ Nearest-neighbor classifiers can produce arbitrarily shaped decision boundaries
- ✓ appropriate proximity measure, data preprocessing steps

Bayesian Classifiers

Bayes Classifier

- A probabilistic framework for solving classification problems
- Conditional Probability:

$$P(C \mid A) = \frac{P(A,C)}{P(A)}$$

$$P(A \mid C) = \frac{P(A,C)}{P(C)}$$

- Bayes theorem:

$$P(C \mid A) = \frac{P(A \mid C)P(C)}{P(A)}$$

Example of Bayes Theorem

- Given:
  - A doctor knows that meningitis causes stiff neck 50% of the time
  - Prior probability of any patient having meningitis is 1/50,000
  - Prior probability of any patient having stiff neck is 1/20

- If a patient has stiff neck, what's the probability he/she has meningitis?

Example of Bayes Theorem

- Given:
  - A doctor knows that meningitis causes stiff neck 50% of the time
  - Prior probability of any patient having meningitis is 1/50,000
  - Prior probability of any patient having stiff neck is 1/20

- If a patient has stiff neck, what's the probability he/she has meningitis?

$$P(M \mid S) = \frac{P(S \mid M)P(M)}{P(S)} = \frac{0.5 \times 1/50000}{1/20} = 0.0002$$

Bayesian Classifiers

- Consider each attribute and class label as random variables

- Given a record with attributes $(x_1, x_2, \ldots, x_n)$
  - Goal is to predict class Y

Bayesian Classifiers

- Consider each attribute and class label as random variables
- Given a record with attributes $(x_1, x_2, \ldots, x_n)$
  - Goal is to predict class Y
  - Specifically, we want to find the value of Y that maximizes $P(Y | x_1, x_2, \ldots, x_n)$
- Can we estimate $P(Y | x_1, x_2, \ldots, x_n)$ directly from data?

Bayesian Classifiers

- Approach:
  - compute the posterior probability $P(Y \mid x_1, x_2, \ldots, x_n)$ for all values of Y using the Bayes theorem

$$\boxed{P(Y \mid x_1 x_2 \ldots x_n) = \frac{P(x_1 x_2 \ldots x_n \mid Y) P(Y)}{P(x_1 x_2 \ldots x_n)}}$$

Bayesian Classifiers

- Approach:
  - compute the posterior probability $P(Y \mid x_1, x_2, \ldots, x_n)$ for all values of Y using the Bayes theorem

$$\boxed{P(Y \mid x_1 x_2 \ldots x_n) = \frac{P(x_1 x_2 \ldots x_n \mid Y) P(Y)}{P(x_1 x_2 \ldots x_n)}}$$

  - Choose value of Y that maximizes

    $P(Y \mid x_1, x_2, \ldots, x_n)$

  - Equivalent to choosing value of Y that maximizes

    $P(x_1, x_2, \ldots, x_n \mid Y)\, P(Y)$

- How to estimate $P(x_1, x_2, \ldots, x_n \mid Y)$?

Naïve Bayes Classifier

Naïve Bayes Classifier

- Assume independence among attributes when class is given:
  - $P(x_1, x_2, \ldots, x_n | Y) = P(x_1 | Y_j) P(x_2 | Y_j) \ldots P(x_n | Y_j)$
  - Can estimate $P(x_i | Y_j)$ for all x_i and Y_j.
  - New point is classified to Y_j if $P(Y_j) \prod P(x_i | Y_j)$ is maximal.

How to Estimate Probabilities from Data?

Tid	**Refund**	**Marital Status**	**Taxable Income**	**Evade**
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**

How to Estimate Probabilities from Data?

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

- Class: $P(Y_k) = N_k/N$
  - e.g., $P(\text{No}) = 7/10$, $P(\text{Yes}) = 3/10$

How to Estimate Probabilities from Data?

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

- Class: $P(Y_k) = N_k/N$
  - e.g., P(No) = 7/10,
    P(Yes) = 3/10
- For discrete attributes:

$$P(x_i \mid Y_k) = |x_{ik}| / N_k$$

  - where $|x_{ik}|$ is number of instances having attribute x_i and belongs to class Y_k
  - Examples:

    P(Status=Married|No) = 4/7
    P(Refund=Yes|Yes)=0

Tid	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**

P(Home Owner=Yes|No) = 3/7
P(Home Owner=No|No) = 4/7
P(Home Owner=Yes|Yes) = 0
P(Home Owner=No|Yes) = 1
P(Marital Status=Single|No) = 2/7
P(Marital Status=Divorced|No) = 1/7
P(Marital Status=Married|No) = 4/7
P(Marital Status=Single|Yes) = 2/3
P(Marital Status=Divorced|Yes) = 1/3
P(Marital Status=Married|Yes) = 0

How to Estimate Probabilities from Data?

- For continuous attributes:
  - Discretize the range into bins

How to Estimate Probabilities from Data?

- For continuous attributes:
  - Discretize the range into bins
  - Probability density estimation:
    - Assume attribute follows a normal distribution
    - Use data to estimate parameters of distribution (e.g., mean and standard deviation)
    - Once probability distribution is known, can use it to estimate the conditional probability $P(x_i|Y)$

How to Estimate Probabilities from Data?

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

- Normal distribution:

$$P(x_i \mid Y_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} e^{-\frac{(x_i-\mu_{ij})^2}{2\sigma_{ij}^2}}$$

  - One for each (x_i, Y_i) pair

How to Estimate Probabilities from Data?

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

- Normal distribution:

$$P(x_i \mid Y_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} e^{-\frac{(x_i - \mu_{ij})^2}{2\sigma_{ij}^2}}$$

  - One for each (x_i, Y_i) pair

- For (Income, Class=No):
  - If Class=No
    - sample mean = 110
    - sample variance = 2975

How to Estimate Probabilities from Data?

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

- Normal distribution:

$$P(x_i \mid Y_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} e^{-\frac{(x_i-\mu_{ij})^2}{2\sigma_{ij}^2}}$$

  - One for each (x_i, Y_i) pair

- For (Income, Class=No):
  - If Class=No
    - sample mean = 110
    - sample variance = 2975

$$P(Income = 120 \mid No) = \frac{1}{\sqrt{2\pi}(54.54)} e^{-\frac{(120-110)^2}{2(2975)}} = 0.0072$$

example

Tid	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**

P(Home Owner=Yes|No) = 3/7
P(Home Owner=No|No) = 4/7
P(Home Owner=Yes|Yes) = 0
P(Home Owner=No|Yes) = 1
P(Marital Status=Single|No) = 2/7
P(Marital Status=Divorced|No) = 1/7
P(Marital Status=Married|No) = 4/7
P(Marital Status=Single|Yes) = 2/3
P(Marital Status=Divorced|Yes) = 1/3
P(Marital Status=Married|Yes) = 0

For Annual Income:
If class=No: sample mean=110
sample variance=2975
If class=Yes: sample mean=90
sample variance=25

Test data: **X** = (Home Owner=No, Marital Status = Married, Income = $120K)

$$P(\texttt{No}|\mathbf{X}) \text{ and } P(\texttt{Yes}|\mathbf{X})$$

New point is classified to Y_i if $P(Y_i) \prod P(x_i | Y_i)$ is maximal.

P(Home Owner=Yes|No) = 3/7
P(Home Owner=No|No) = 4/7
P(Home Owner=Yes|Yes) = 0
P(Home Owner=No|Yes) = 1
P(Marital Status=Single|No) = 2/7
P(Marital Status=Divorced|No) = 1/7
P(Marital Status=Married|No) = 4/7
P(Marital Status=Single|Yes) = 2/3
P(Marital Status=Divorced|Yes) = 1/3
P(Marital Status=Married|Yes) = 0

For Annual Income:
If class=No: sample mean=110
sample variance=2975
If class=Yes: sample mean=90
sample variance=25

$P(\texttt{Yes}) = 0.3$ and $P(\texttt{No}) = 0.7$

$P(Y_j) \prod P(x_i \mid Y_j)$

X = (Home Owner=No, Marital Status = Married, Income = $120K)

P(Home Owner=Yes|No) = 3/7
P(Home Owner=No|No) = 4/7
P(Home Owner=Yes|Yes) = 0
P(Home Owner=No|Yes) = 1
P(Marital Status=Single|No) = 2/7
P(Marital Status=Divorced|No) = 1/7
P(Marital Status=Married|No) = 4/7
P(Marital Status=Single|Yes) = 2/3
P(Marital Status=Divorced|Yes) = 1/3
P(Marital Status=Married|Yes) = 0

For Annual Income:
If class=No: sample mean=110
sample variance=2975
If class=Yes: sample mean=90
sample variance=25

$P(\texttt{Yes}) = 0.3$ and $P(\texttt{No}) = 0.7$

$$P(Y_j) \prod P(x_i \mid Y_j)$$

X = (Home Owner=No, Marital Status = Married, Income = $120K)

$$\begin{aligned} P(\mathbf{X}|\texttt{No}) &= P(\texttt{Home Owner} = \texttt{No}|\texttt{No}) \times P(\texttt{Status} = \texttt{Married}|\texttt{No}) \\ &\quad \times P(\texttt{Annual Income} = \$120\text{K}|\texttt{No}) \\ &= 4/7 \times 4/7 \times 0.0072 = 0.0024. \end{aligned}$$

P(Home Owner=Yes|No) = 3/7
P(Home Owner=No|No) = 4/7
P(Home Owner=Yes|Yes) = 0
P(Home Owner=No|Yes) = 1
P(Marital Status=Single|No) = 2/7
P(Marital Status=Divorced|No) = 1/7
P(Marital Status=Married|No) = 4/7
P(Marital Status=Single|Yes) = 2/3
P(Marital Status=Divorced|Yes) = 1/3
P(Marital Status=Married|Yes) = 0

For Annual Income:
If class=No: sample mean=110
sample variance=2975
If class=Yes: sample mean=90
sample variance=25

$P(\texttt{Yes}) = 0.3$ and $P(\texttt{No}) = 0.7$

$$P(Y_j) \prod P(x_i | Y_j)$$

X = (Home Owner=No, Marital Status = Married, Income = $120K)

$$\begin{aligned} P(\mathbf{X}|\texttt{No}) &= P(\texttt{Home Owner} = \texttt{No}|\texttt{No}) \times P(\texttt{Status} = \texttt{Married}|\texttt{No}) \\ &\quad \times P(\texttt{Annual Income} = \$120\text{K}|\texttt{No}) \\ &= 4/7 \times 4/7 \times 0.0072 = 0.0024. \end{aligned}$$

$$\begin{aligned} P(\mathbf{X}|\texttt{Yes}) &= P(\texttt{Home Owner} = \texttt{No}|\texttt{Yes}) \times P(\texttt{Status} = \texttt{Married}|\texttt{Yes}) \\ &\quad \times P(\texttt{Annual Income} = \$120\text{K}|\texttt{Yes}) \\ &= 1 \times 0 \times 1.2 \times 10^{-9} = 0. \end{aligned}$$

$P(\text{No}|\mathbf{X}) > P(\text{Yes}|\mathbf{X})$

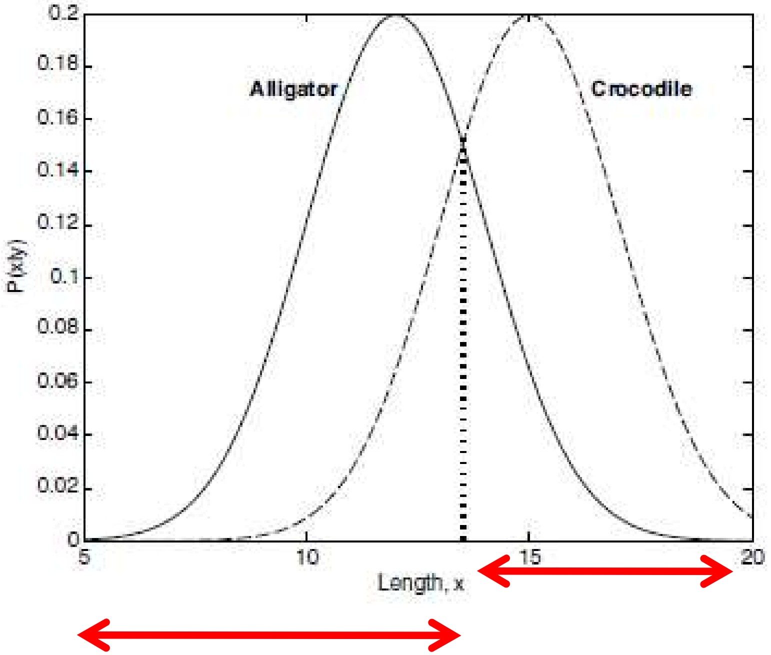
Bayes Error Rate

Example:

identifying alligators and crocodiles based on their lengths
their prior probabilities are the same

$$
\begin{aligned}
P(X|\texttt{Crocodile}) &= \frac{1}{\sqrt{2\pi}\cdot 2}\exp\left[-\frac{1}{2}\left(\frac{X-15}{2}\right)^2\right] \\
P(X|\texttt{Alligator}) &= \frac{1}{\sqrt{2\pi}\cdot 2}\exp\left[-\frac{1}{2}\left(\frac{X-12}{2}\right)^2\right]
\end{aligned}
$$

Bayes Error Rate



Directed graphical models (Bayesian Belief network)

Introduction

- multiple correlated variables, such as words in a document, pixels in an image, or genes in a microarray
- compactly *represent* the **joint distribution** $p(\mathbf{x})$ (probabilistic modeling)
- *learn* the parameters of this distribution with a reasonable amount of data (learning)
- *infer* one set of variables given another (inference)

Introduction

- **Chain rule**

$$p(x_{1:V}) = p(x_1)p(x_2|x_1)p(x_3|x_2, x_1)p(x_4|x_1, x_2, x_3)\ldots p(x_V|x_{1:V-1})$$

- becomes more and more complicated to represent the conditional distributions

Introduction

- **Chain rule**

$$p(x_{1:V}) = p(x_1)p(x_2|x_1)p(x_3|x_2,x_1)p(x_4|x_1,x_2,x_3)\ldots p(x_V|x_{1:V-1})$$

- becomes more and more complicated to represent the conditional distributions
- suppose all the variables have *K* states
- $p(x2|x1)$, $p(x3|x1, x2)$

Conditional independence(CI)

- The key to efficiently representing large joint distributions is to make some assumptions about conditional independence

$$X \perp Y | Z \iff p(X, Y | Z) = p(X | Z) p(Y | Z)$$

Conditional independence(CI)

- The key to efficiently representing large joint distributions is to make some assumptions about conditional independence

$$X \perp Y \mid Z \iff p(X, Y \mid Z) = p(X \mid Z)p(Y \mid Z)$$

$$x_{t+1} \perp \mathbf{x}_{1:t-1} \mid x_t$$

Conditional independence(CI)

- The key to efficiently representing large joint distributions is to make some assumptions about conditional independence

$$X \perp Y | Z \iff p(X, Y | Z) = p(X | Z) p(Y | Z)$$

$$x_{t+1} \perp \mathbf{x}_{1:t-1} | x_t$$

Markov Assumption

$$p(x_{1:V}) = p(x_1) p(x_2 | x_1) p(x_3 | x_2, x_1) p(x_4 | x_1, x_2, x_3) \ldots p(x_V | x_{1:V-1})$$

$$p(\mathbf{x}_{1:V}) = p(x_1) \prod_{t=1}^{V} p(x_t | x_{t-1})$$

Graphical Models

- first-order Markov assumption is useful for defining distributions on 1d sequences
- A **graphical model** (**GM**) is a way to represent a joint distribution by making CI assumptions
- nodes in the graph represent random variables
- lack of edges represents CI assumptions

Graphical Models

- first-order Markov assumption is useful for defining distributions on 1d sequences
- A **graphical model** (**GM**) is a way to represent a joint distribution by making CI assumptions
- nodes in the graph represent random variables
- lack of edges represents CI assumptions

$$\textbf{graph } G = (\mathcal{V}, \mathcal{E})$$

adjacency matrix

$$\mathcal{V} = \{1, \ldots, V\}$$

$$\mathcal{E} = \{(s,t) : s,t \in \mathcal{V}\} \qquad G(s,t) = 1 \text{ to denote } (s,t) \in \mathcal{E}$$

Graph terminology

- **Parent** For a directed graph, the **parents** of a node is the set of all nodes that feed into it: $\mathrm{pa}(s) \triangleq \{t : G(t, s) = 1\}$.

- **Child** For a directed graph, the **children** of a node is the set of all nodes that feed out of it: $\mathrm{ch}(s) \triangleq \{t : G(s, t) = 1\}$.

- **Root** For a directed graph, a **root** is a node with no parents.

- **Path or trail** A **path** or **trail** $s \rightsquigarrow t$ is a series of directed edges leading from s to t.

Graph terminology

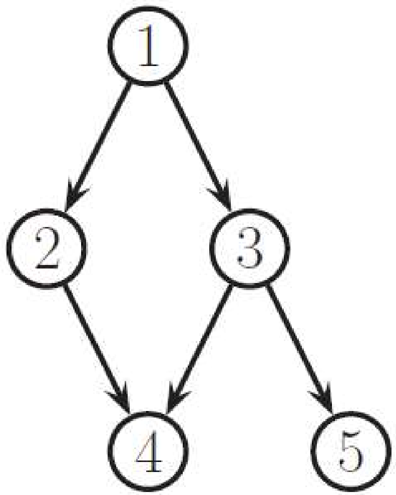
- **Ancestors** For a directed graph, the **ancestors** are the parents, grand-parents, etc of a node. That is, the ancestors of t is the set of nodes that connect to t via a trail: $\text{anc}(t) \triangleq \{s : s \rightsquigarrow t\}$.

- **Descendants** For a directed graph, the **descendants** are the children, grand-children, etc of a node. That is, the descendants of s is the set of nodes that can be reached via trails from s: $\text{desc}(s) \triangleq \{t : s \rightsquigarrow t\}$.

- **Cycle or loop** For any graph, we define a **cycle** or **loop** to be a series of nodes such that we can get back to where we started by following edges

Graph terminology

- **DAG** A **directed acyclic graph** or **DAG** is a directed graph with no directed cycles.



- **Topological ordering** For a DAG, a **topological ordering** or **total ordering** is a numbering of the nodes such that parents have lower numbers than their children.

Directed graphical models

- A **directed graphical model** or **DGM** is a GM whose graph is a DAG
- Bayesian Network, Causal Network
- topological ordering
- **ordered Markov property**

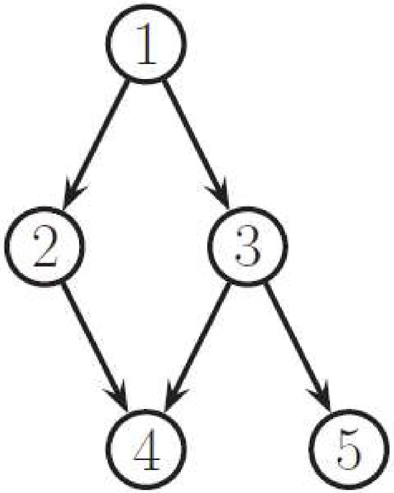
Directed graphical models

- A **directed graphical model** or **DGM** is a GM whose graph is a DAG
- Bayesian Network, Causal Network
- topological ordering
- **ordered Markov property**

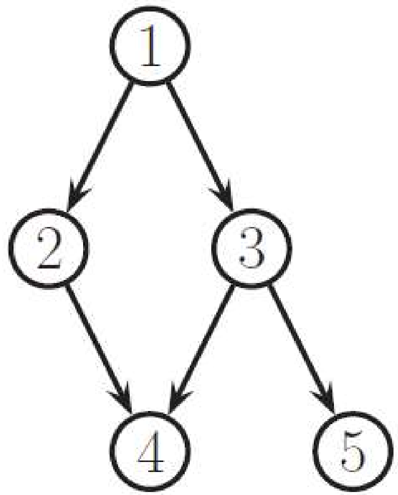
Property 1 (Conditional Independence). *A node in a Bayesian network is conditionally independent of its non-descendants, if its parents are known.*

natural generalization of the first-order Markov property

Directed graphical models

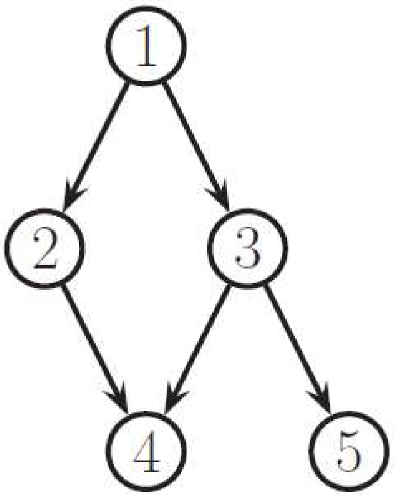


Directed graphical models



$$p(\mathbf{x}_{1:5}) \quad = \quad p(x_1)p(x_2|x_1)p(x_3|x_1,\cancel{x_2})p(x_4|\cancel{x_1},x_2,x_3)p(x_5|\cancel{x_1},\cancel{x_2},x_3,\cancel{x_4})$$

Directed graphical models



$$
\begin{aligned}
p(\mathbf{x}_{1:5}) &= p(x_1)p(x_2|x_1)p(x_3|x_1,\cancel{x_2})p(x_4|\cancel{x_1},x_2,x_3)p(x_5|\cancel{x_1},\cancel{x_2},x_3,\cancel{x_4}) \\
&= p(x_1)p(x_2|x_1)p(x_3|x_1)p(x_4|x_2,x_3)p(x_5|x_3)
\end{aligned}
$$

$$
\boxed{p(\mathbf{x}_{1:V}|G) = \prod_{t=1}^{V} p(x_t|\mathbf{x}_{\mathrm{pa}(t)})}
$$

What are the random variables in Bayesian learning that we want to jointly model them?

What are the random variables in Bayesian learning that we want to jointly model them?

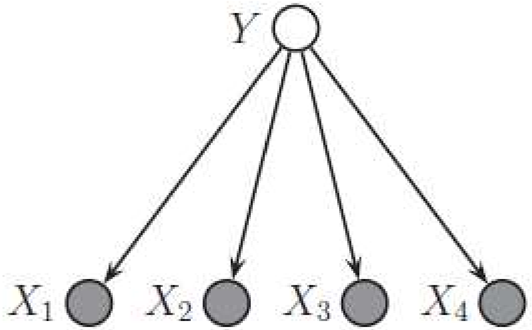
$$\boxed{P(Y \mid x_1 x_2 \ldots x_n) = \frac{P(x_1 x_2 \ldots x_n \mid Y)\,P(Y)}{P(x_1 x_2 \ldots x_n)}}$$

Example:Naive Bayes classifiers

features are conditionally independent given the class label

Example:Naive Bayes classifiers

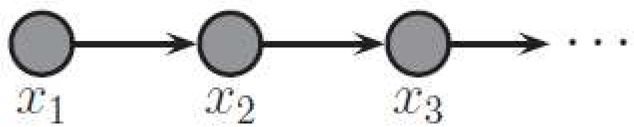
features are conditionally independent given the class label



$$p(y, \mathbf{x}) = p(y) \prod_{j=1}^{D} p(x_j | y)$$

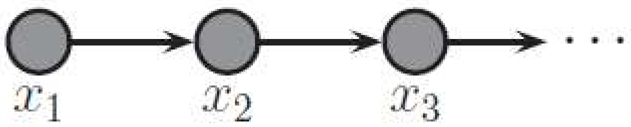
Example: Markov Chain

first-order Markov chain

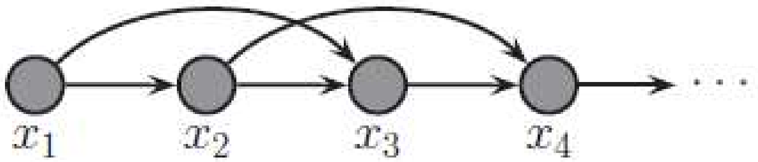


Example: Markov Chain

first-order Markov chain

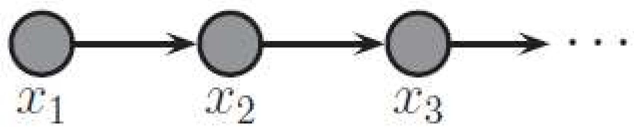


second order Markov chain

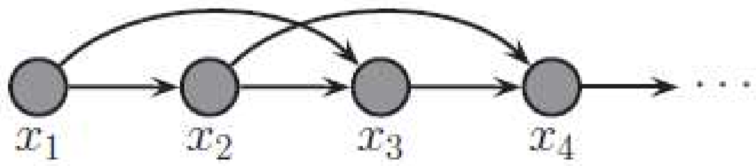


Example: Markov Chain

first-order Markov chain

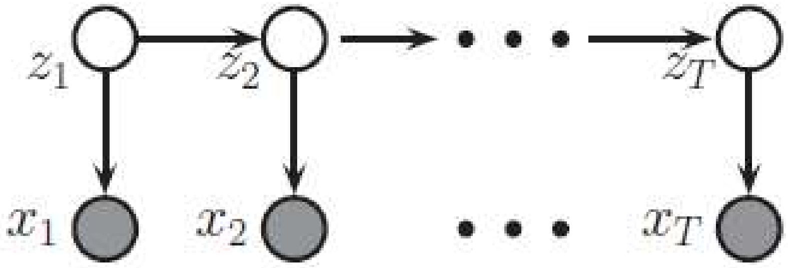


second order Markov chain



$$p(\mathbf{x}_{1:T}) = p(x_1, x_2)p(x_3|x_1, x_2)p(x_4|x_2, x_3) \ldots = p(x_1, x_2) \prod_{t=3}^{T} p(x_t|x_{t-1}, x_{t-2})$$

Example: HMM



- ❖ hidden variables often represent quantities of interest
- ❖ estimate the hidden state given the data

Inference

- ✓ graphical models : compact way to define joint probability distributions
- ✓ The main use for such a joint distribution is to perform **probabilistic inference**

task of estimating unknown quantities from known quantities

set of correlated random variables with joint distribution

$$p(\mathbf{x}_{1:V}|\theta)$$

visible variables and hidden variables

Inference

$$p(\mathbf{x}_h|\mathbf{x}_v,\theta) = \frac{p(\mathbf{x}_h,\mathbf{x}_v|\theta)}{p(\mathbf{x}_v|\theta)} = \frac{p(\mathbf{x}_h,\mathbf{x}_v|\theta)}{\sum_{\mathbf{x}'_h} p(\mathbf{x}'_h,\mathbf{x}_v|\theta)}$$

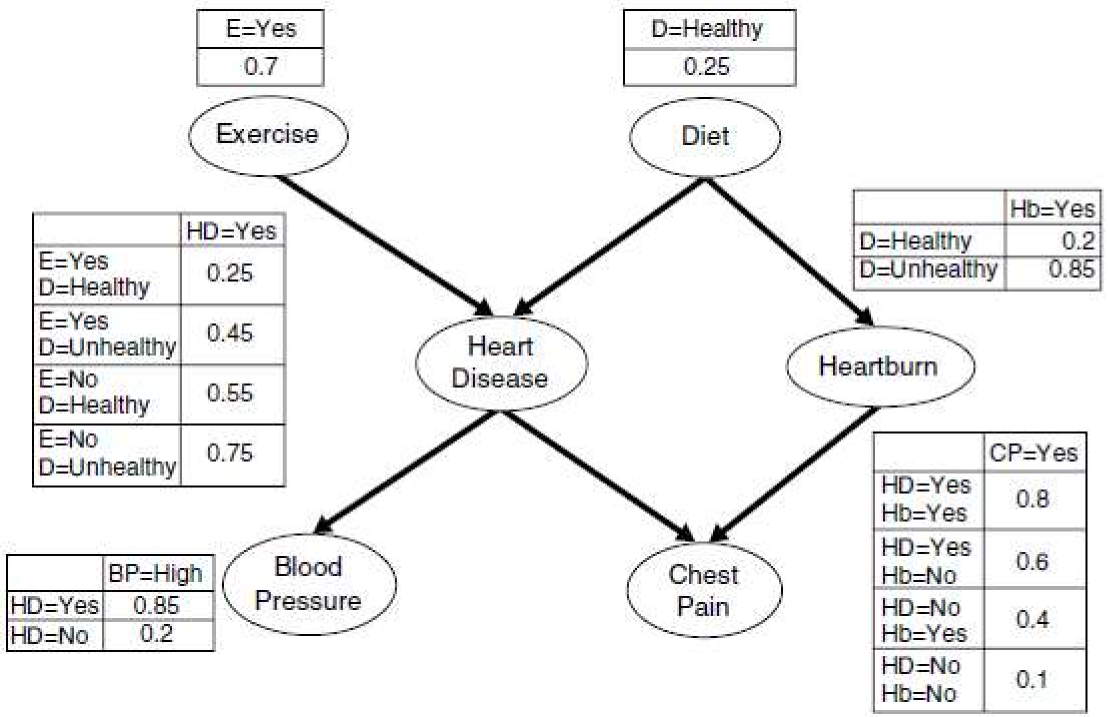
In Classification : class label :Y

Model Paramters

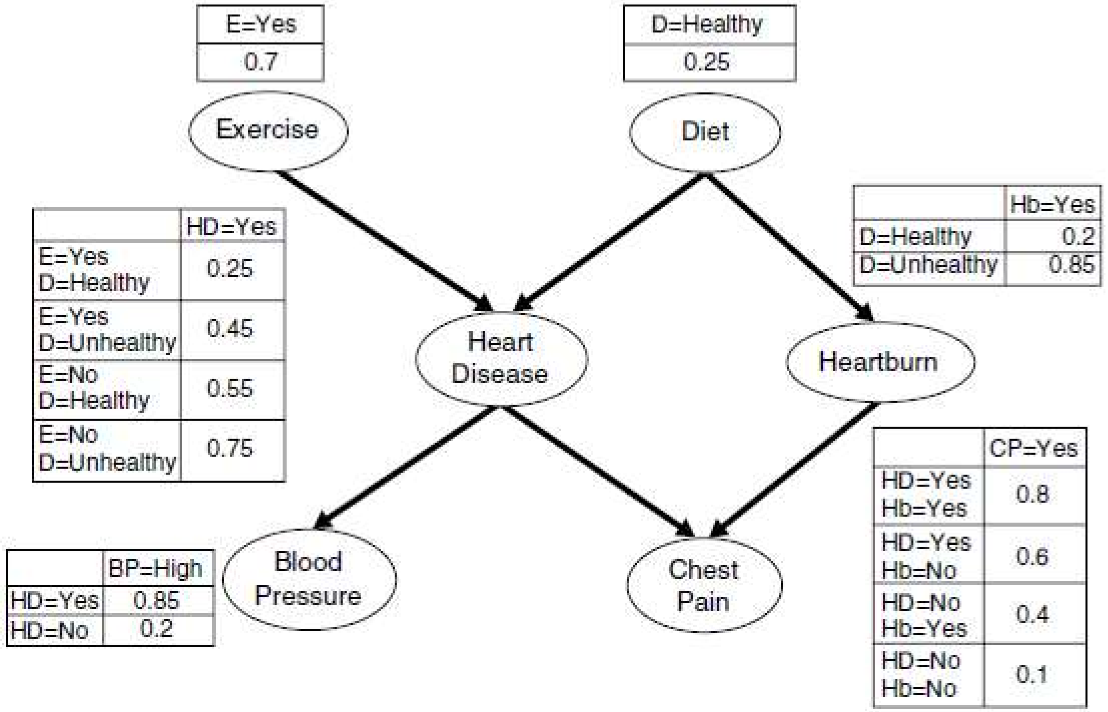
Example of Inferencing Using

Case 1: No Prior Information

Without any prior information, we can determine whether the person is likely to have heart disease



Example of Inferencing Using



E=Yes
0.7

D=Healthy
0.25

	HD=Yes
E=Yes D=Healthy	0.25
E=Yes D=Unhealthy	0.45
E=No D=Healthy	0.55
E=No D=Unhealthy	0.75

	Hb=Yes
D=Healthy	0.2
D=Unhealthy	0.85

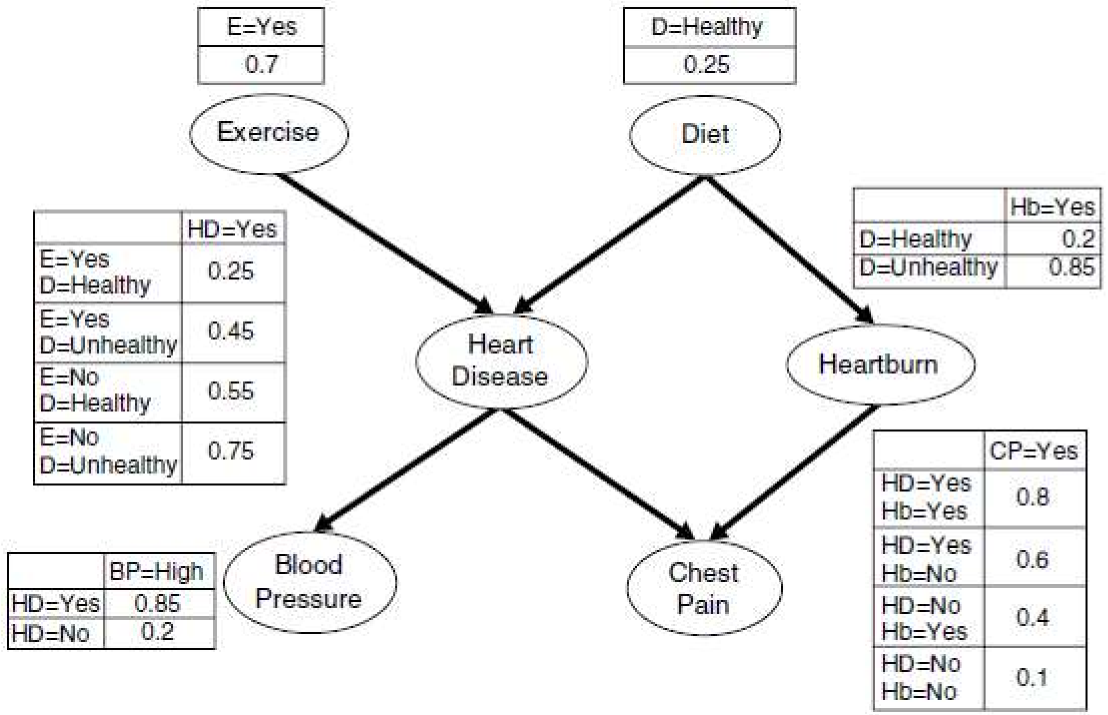
	BP=High
HD=Yes	0.85
HD=No	0.2

	CP=Yes
HD=Yes Hb=Yes	0.8
HD=Yes Hb=No	0.6
HD=No Hb=Yes	0.4
HD=No Hb=No	0.1

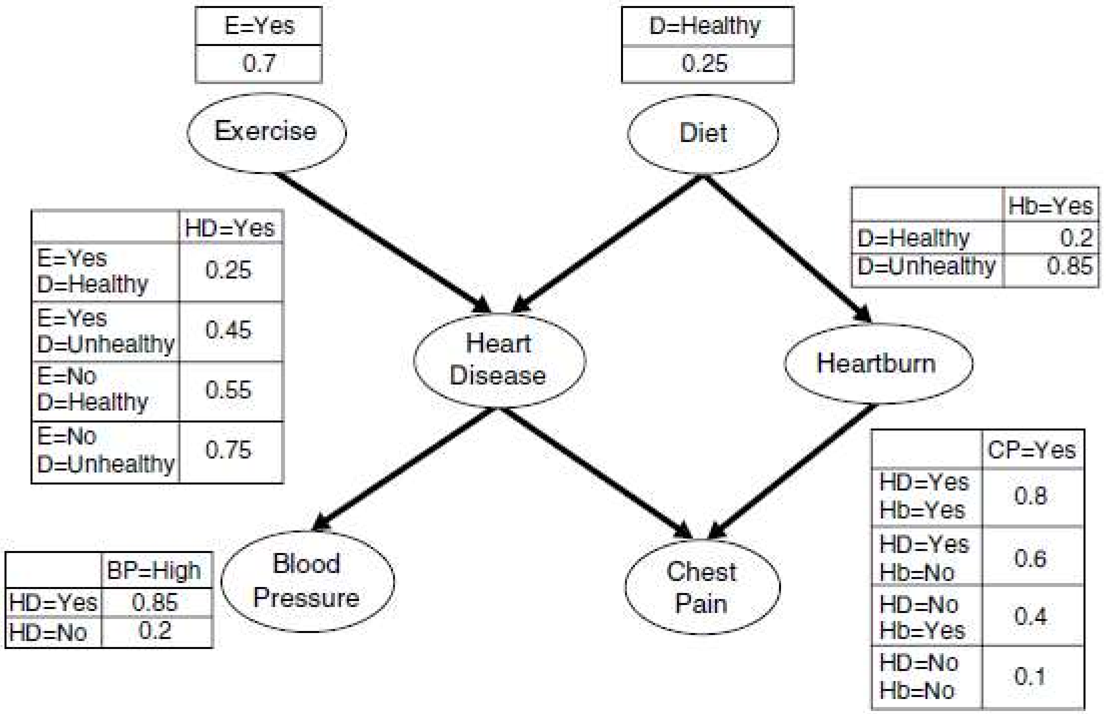
$$\begin{aligned} P(\mathtt{HD}=\mathtt{Yes}) &= \sum_{\alpha}\sum_{\beta} P(\mathtt{HD}=\mathtt{Yes}|E=\alpha,D=\beta)P(E=\alpha,D=\beta) \\ &= \sum_{\alpha}\sum_{\beta} P(\mathtt{HD}=\mathtt{Yes}|E=\alpha,D=\beta)P(E=\alpha)P(D=\beta) \\ &= 0.25\times 0.7\times 0.25+0.45\times 0.7\times 0.75+0.55\times 0.3\times 0.25 \\ &\quad +0.75\times 0.3\times 0.75 \\ &= 0.49. \end{aligned}$$

Example

Case 2: High Blood Pressure
If the person has high blood pressure, we can make a diagnosis about heart disease



Example



E=Yes
0.7

D=Healthy
0.25

	HD=Yes
E=Yes D=Healthy	0.25
E=Yes D=Unhealthy	0.45
E=No D=Healthy	0.55
E=No D=Unhealthy	0.75

	Hb=Yes
D=Healthy	0.2
D=Unhealthy	0.85

	BP=High
HD=Yes	0.85
HD=No	0.2

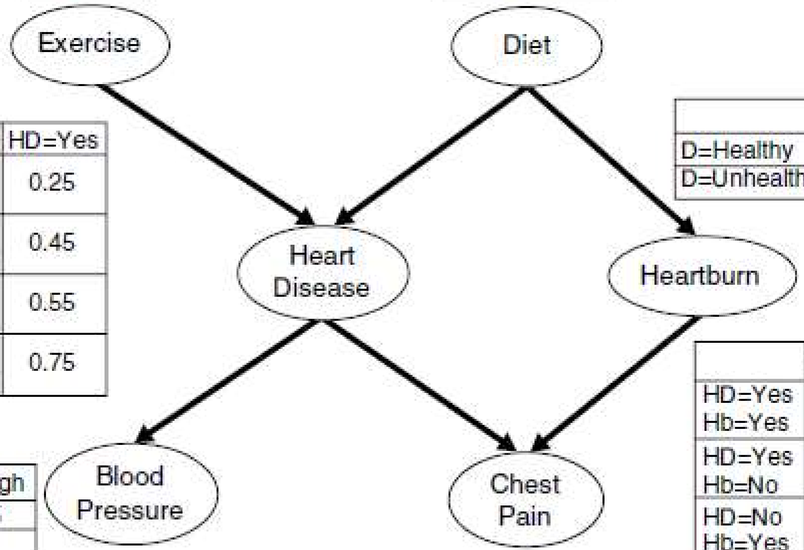
	CP=Yes
HD=Yes Hb=Yes	0.8
HD=Yes Hb=No	0.6
HD=No Hb=Yes	0.4
HD=No Hb=No	0.1

$$P(\mathtt{HD} = \mathtt{Yes} | \mathtt{BP} = \mathtt{High})$$

$$= \frac{P(\mathtt{BP} = \mathtt{High} | \mathtt{HD} = \mathtt{Yes}) P(\mathtt{HD} = \mathtt{Yes})}{P(\mathtt{BP} = \mathtt{High})}$$

$$= \frac{0.85 \times 0.49}{0.5185} = 0.8033.$$

Example



E=Yes
0.7

D=Healthy
0.25

	HD=Yes
E=Yes D=Healthy	0.25
E=Yes D=Unhealthy	0.45
E=No D=Healthy	0.55
E=No D=Unhealthy	0.75

	Hb=Yes
D=Healthy	0.2
D=Unhealthy	0.85

	CP=Yes
HD=Yes Hb=Yes	0.8
HD=Yes Hb=No	0.6
HD=No Hb=Yes	0.4
HD=No Hb=No	0.1

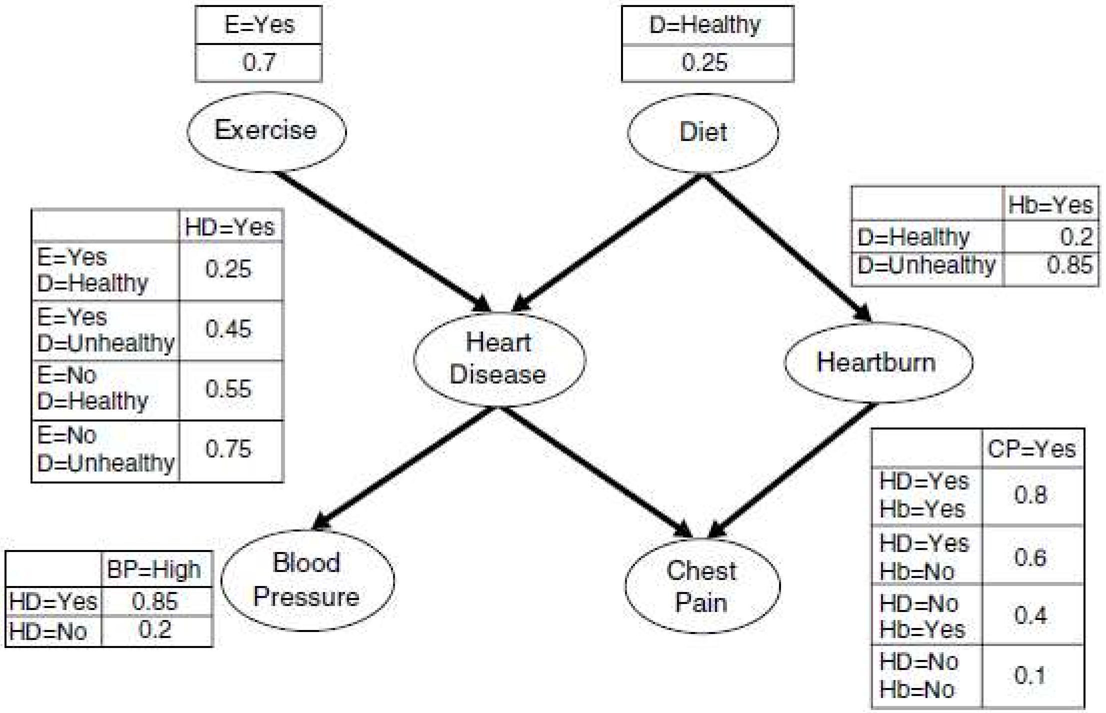
	BP=High
HD=Yes	0.85
HD=No	0.2

$$
\begin{aligned}
P(\mathtt{BP}=\mathtt{High}) &= \sum_{\gamma} P(\mathtt{BP}=\mathtt{High}|\mathtt{HD}=\gamma)P(\mathtt{HD}=\gamma) \\
&= 0.85 \times 0.49 + 0.2 \times 0.51 = 0.5185.
\end{aligned}
$$

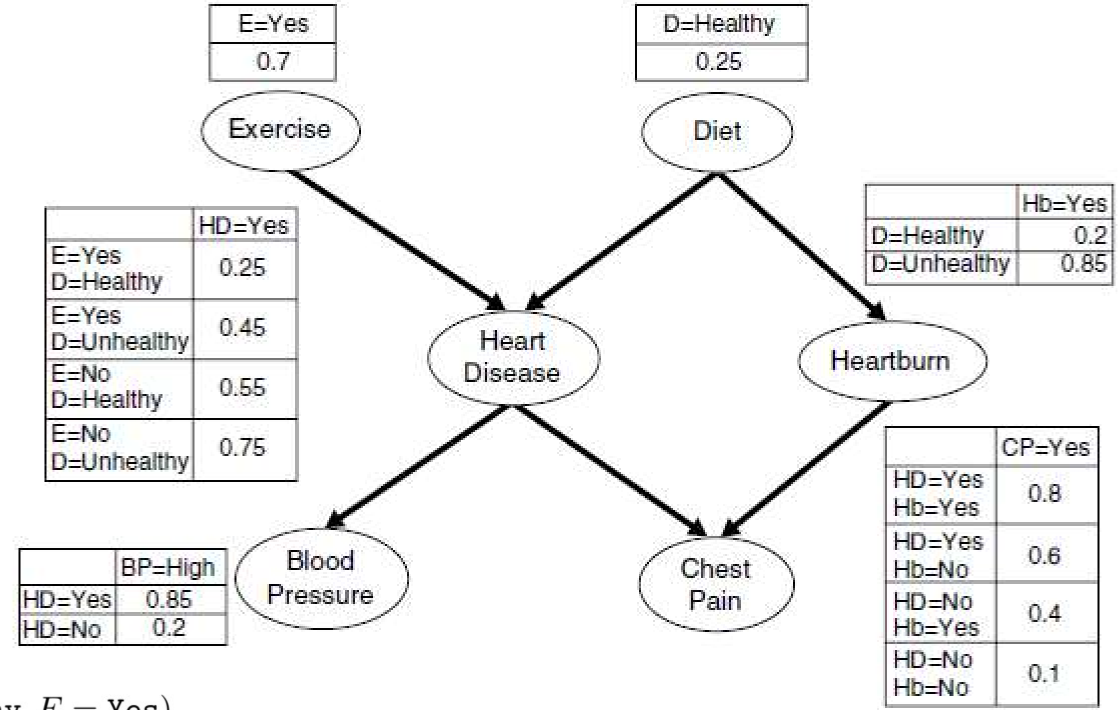
$$
\begin{aligned}
&P(\mathtt{HD}=\mathtt{Yes}|\mathtt{BP}=\mathtt{High}) \\
&= \frac{P(\mathtt{BP}=\mathtt{High}|\mathtt{HD}=\mathtt{Yes})P(\mathtt{HD}=\mathtt{Yes})}{P(\mathtt{BP}=\mathtt{High})} \\
&= \frac{0.85 \times 0.49}{0.5185} = 0.8033.
\end{aligned}
$$

Example

Case 3: High Blood Pressure, Healthy Diet, and Regular Exercise
Suppose we are told that the person exercises regularly and eats a healthy diet. How does the new information affect our diagnosis?



Example



E=Yes
0.7

D=Healthy
0.25

	HD=Yes
E=Yes D=Healthy	0.25
E=Yes D=Unhealthy	0.45
E=No D=Healthy	0.55
E=No D=Unhealthy	0.75

	Hb=Yes
D=Healthy	0.2
D=Unhealthy	0.85

	BP=High
HD=Yes	0.85
HD=No	0.2

	CP=Yes
HD=Yes Hb=Yes	0.8
HD=Yes Hb=No	0.6
HD=No Hb=Yes	0.4
HD=No Hb=No	0.1

$$
\begin{aligned}
& P(\texttt{HD} = \texttt{Yes}|\texttt{BP} = \texttt{High}, D = \texttt{Healthy}, E = \texttt{Yes}) \\
= & \left[\frac{P(\texttt{BP} = \texttt{High}|\texttt{HD} = \texttt{Yes}, D = \texttt{Healthy}, E = \texttt{Yes})}{P(\texttt{BP} = \texttt{High}|D = \texttt{Healthy}, E = \texttt{Yes})}\right] \\
& \qquad \times P(\texttt{HD} = \texttt{Yes}|D = \texttt{Healthy}, E = \texttt{Yes}) \\
= & \frac{P(\texttt{BP} = \texttt{High}|\texttt{HD} = \texttt{Yes})P(\texttt{HD} = \texttt{Yes}|D = \texttt{Healthy}, E = \texttt{Yes})}{\sum_\gamma P(\texttt{BP} = \texttt{High}|\texttt{HD} = \gamma)P(\texttt{HD} = \gamma|D = \texttt{Healthy}, E = \texttt{Yes})} \\
= & \frac{0.85 \times 0.25}{0.85 \times 0.25 + 0.2 \times 0.75} \\
= & \; 0.5862,
\end{aligned}
$$

Support Vector Machine (SVM)

Hyperplanes

A *hyperplane* is a set of the form

$$\{x \mid a^T x = b\},$$

$a \in \mathbf{R}^n$, $a \neq 0$, and $b \in \mathbf{R}$.

a is the normal vector

Hyperplanes

A *hyperplane* is a set of the form

$$\{x \mid a^T x = b\},$$

$a \in \mathbf{R}^n$, $a \neq 0$, and $b \in \mathbf{R}$.

a is the normal vector

Geometrical interpretation

x_0 is any point in the hyperplane $\quad a^T x_0 = b \quad \{x \mid a^T(x - x_0) = 0\},$

Hyperplanes

A *hyperplane* is a set of the form

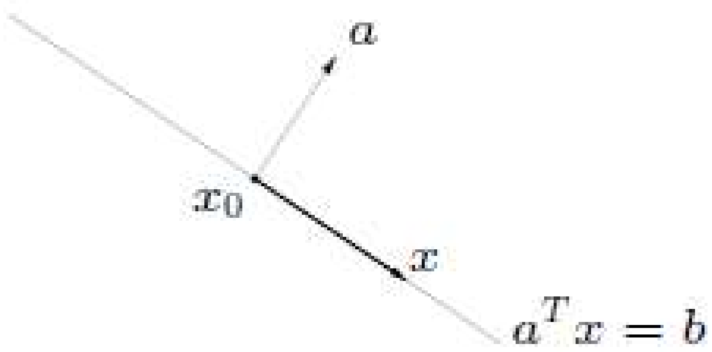
$$\{x \mid a^T x = b\},$$

$a \in \mathbf{R}^n$, $a \neq 0$, and $b \in \mathbf{R}$.

a is the normal vector

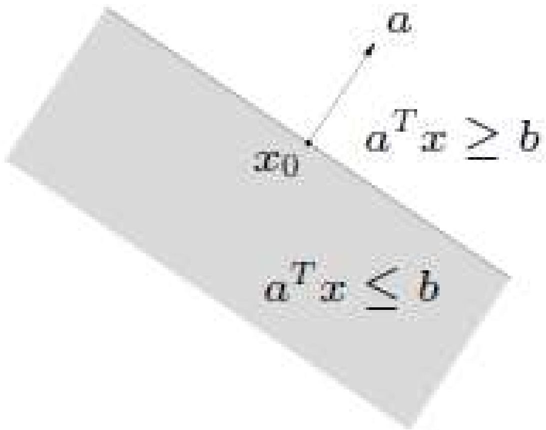
Geometrical interpretation

x_0 is any point in the hyperplane $\quad a^T x_0 = b \quad \{x \mid a^T (x - x_0) = 0\},$

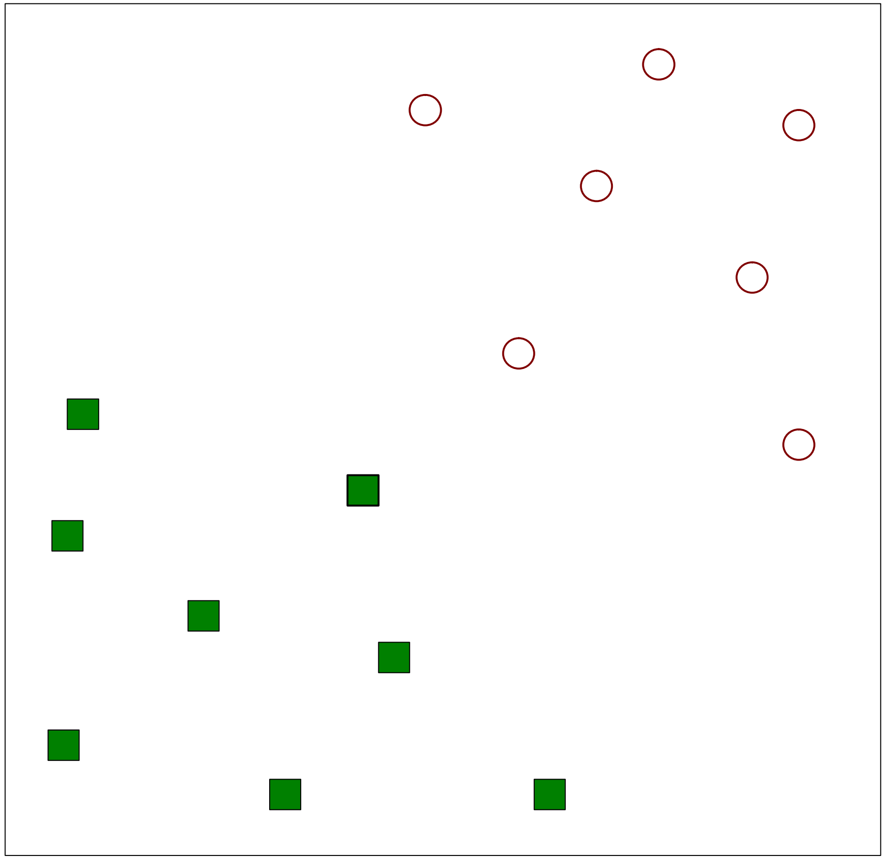


halfspaces

halfspace: set of the form $\{x \mid a^T x \leq b\}$ $(a \neq 0)$

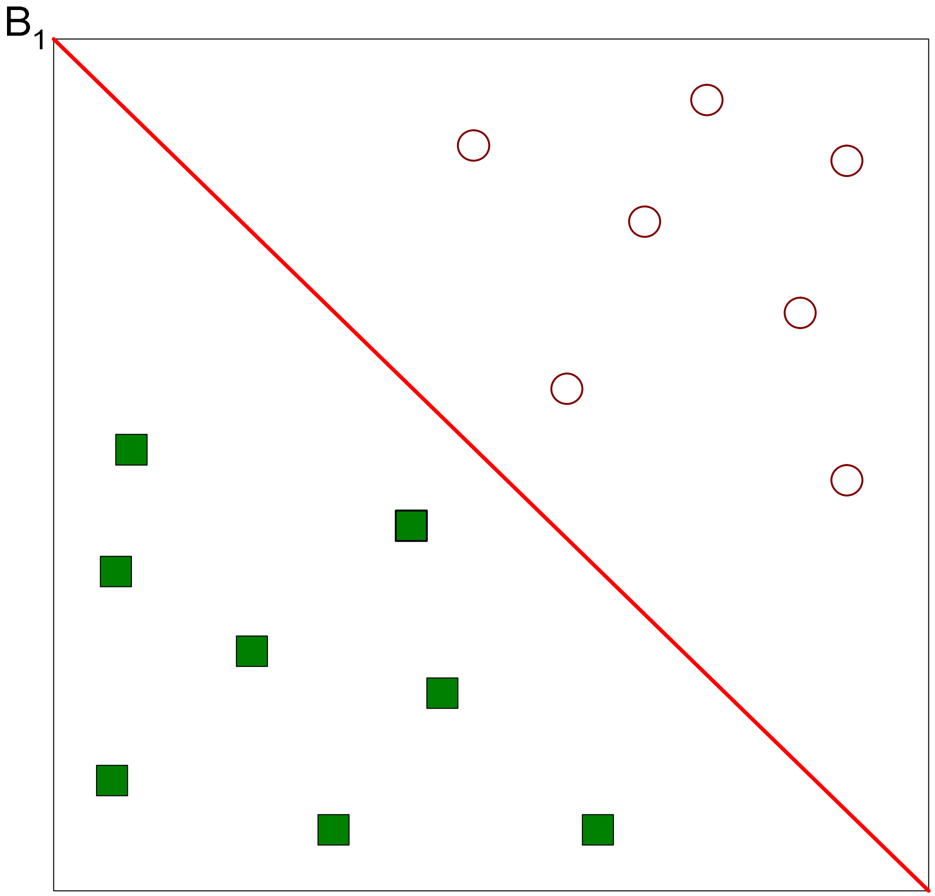


Maximum Margin Hyperplanes



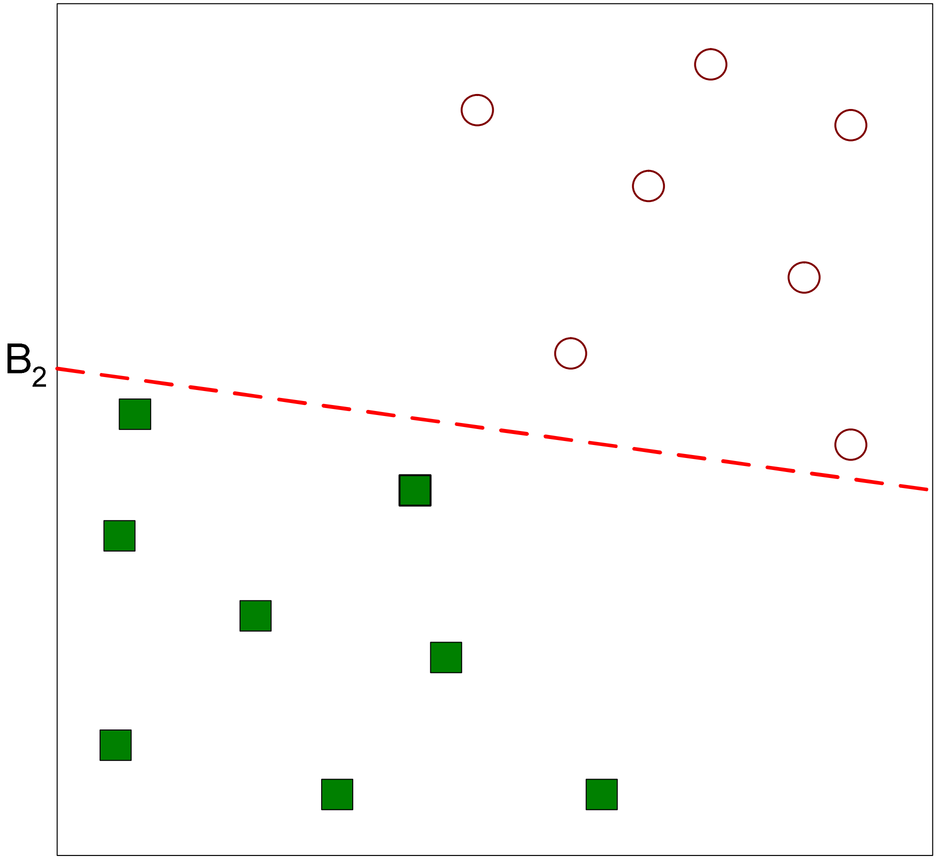
- Find a linear hyperplane (decision boundary) that will separate the data

Maximum Margin Hyperplanes



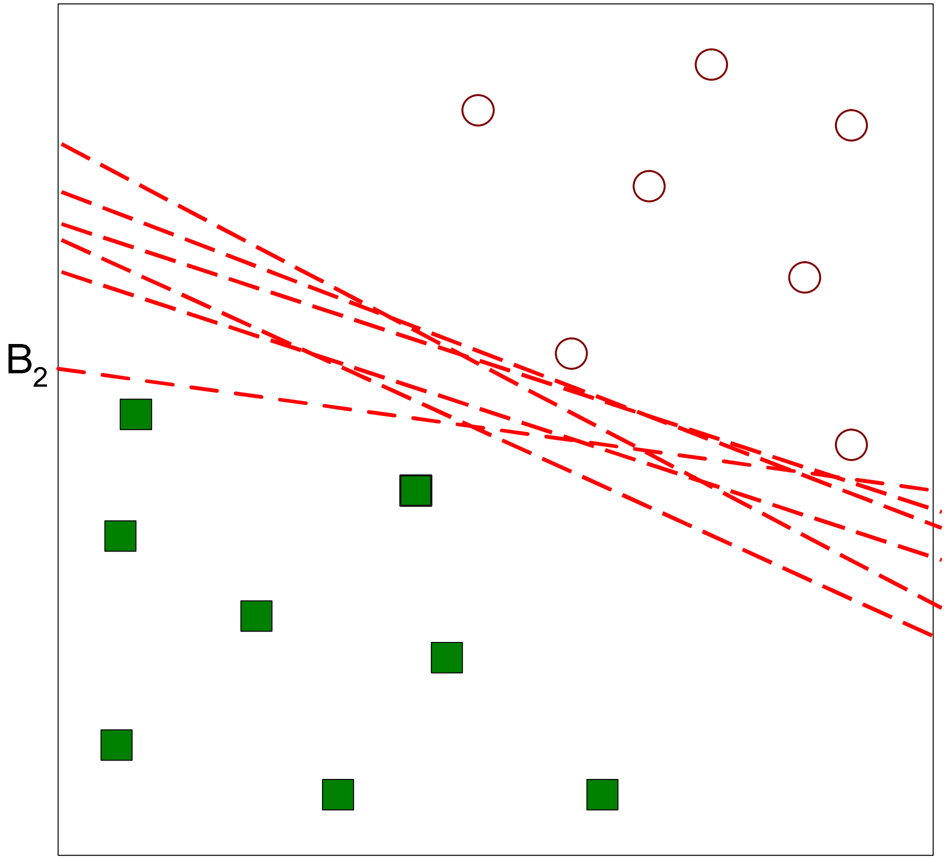
- One Possible Solution

Maximum Margin Hyperplanes



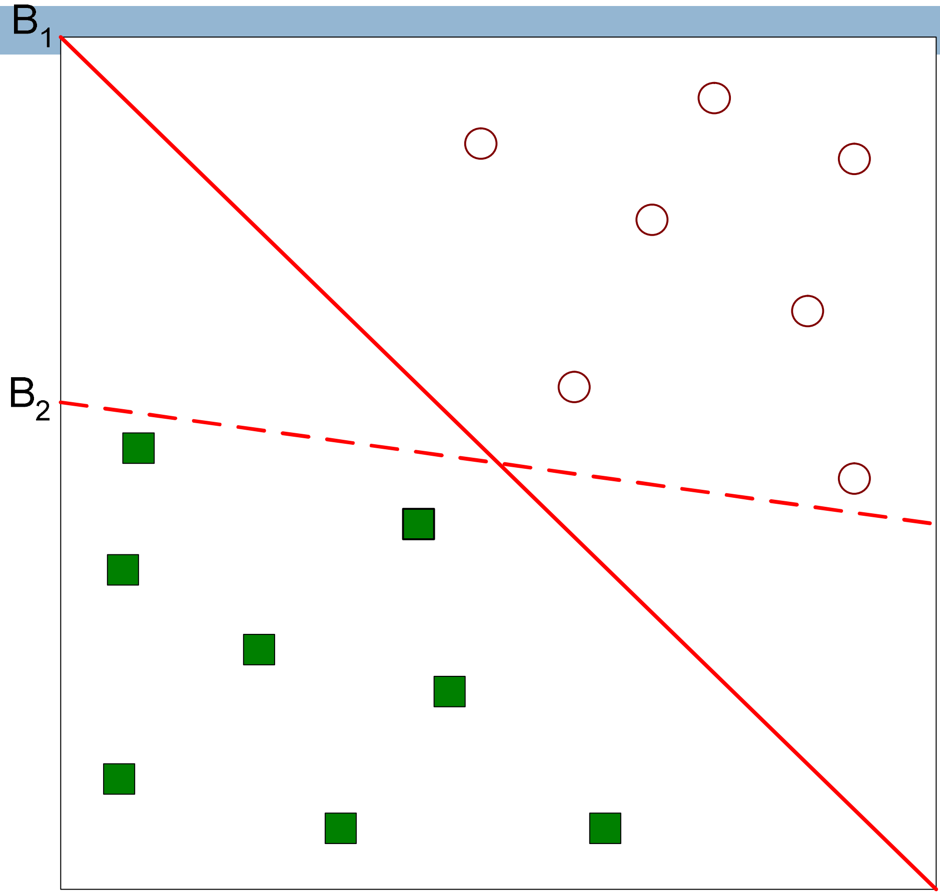
- Another possible solution

Maximum Margin Hyperplanes



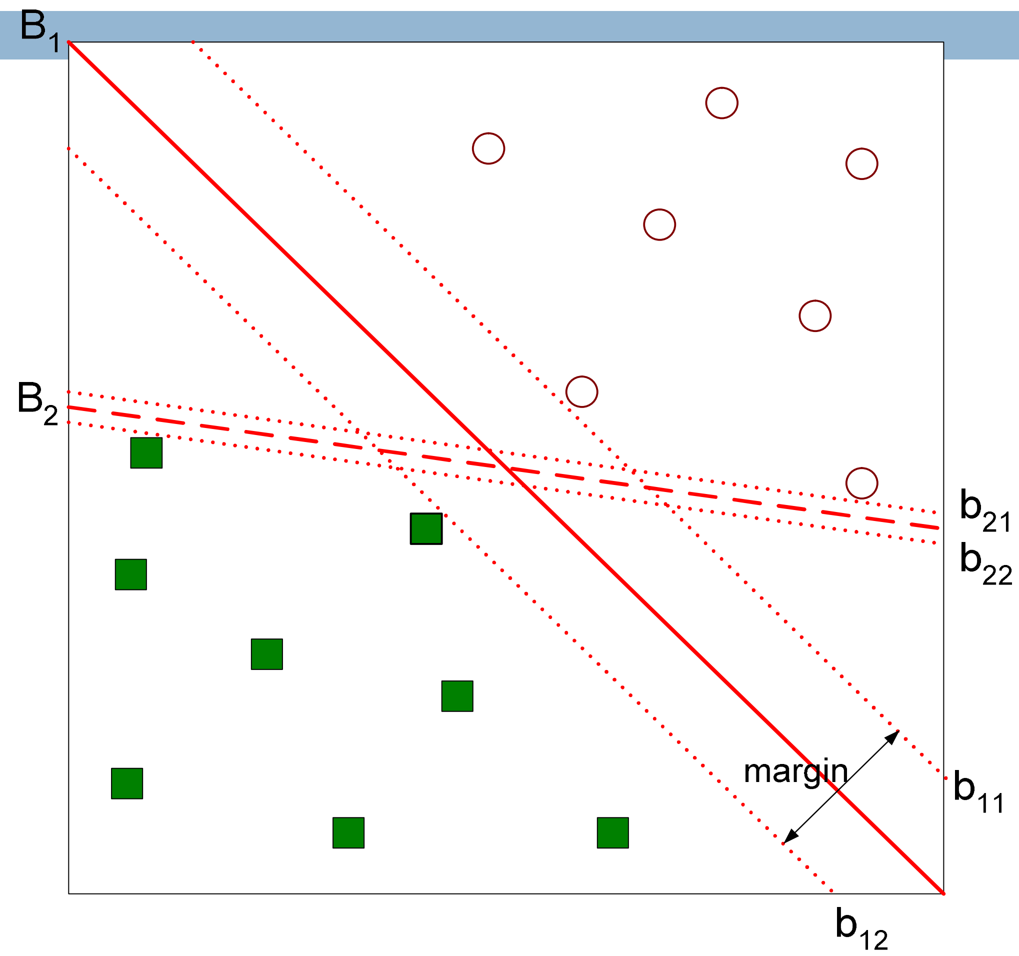
- Other possible solutions

Maximum Margin Hyperplanes



- Which one is better? B1 or B2?
- How do you define better?

Maximum Margin Hyperplanes



- Find hyperplane maximizes the margin => B1 is better than B2
- Generalization error

Linear SVM: Separable Case

A linear SVM is a classifier that searches for a hyperplane with the largest margin

$$(\mathbf{x_i}, y_i) \ (i = 1, 2, \ldots, N)$$

$$(x_{i1}, x_{i2}, \ldots, x_{id})^T \qquad y_i \in \{-1, 1\}$$

Linear SVM: Separable Case

A linear SVM is a classifier that searches for a hyperplane with the largest margin

$$(\mathbf{x_i}, y_i) \ (i = 1, 2, \ldots, N)$$

$$(x_{i1}, x_{i2}, \ldots, x_{id})^T \qquad y_i \in \{-1, 1\}$$

decision boundary of a linear classifier $\quad \mathbf{w} \cdot \mathbf{x} + b = 0,$

w and *b* are parameters of the model

$$\mathbf{w} \cdot \mathbf{x}_s + b = k, \qquad k > 0.$$

$$\mathbf{w} \cdot \mathbf{x}_c + b = k', \qquad k' < 0.$$

Linear SVM: Separable Case

A linear SVM is a classifier that searches for a hyperplane with the largest margin

$$(\mathbf{x_i}, y_i) \ (i = 1, 2, \ldots, N)$$

$$(x_{i1}, x_{i2}, \ldots, x_{id})^T \qquad y_i \in \{-1, 1\}$$

decision boundary of a linear classifier

$$\mathbf{w} \cdot \mathbf{x} + b = 0,$$

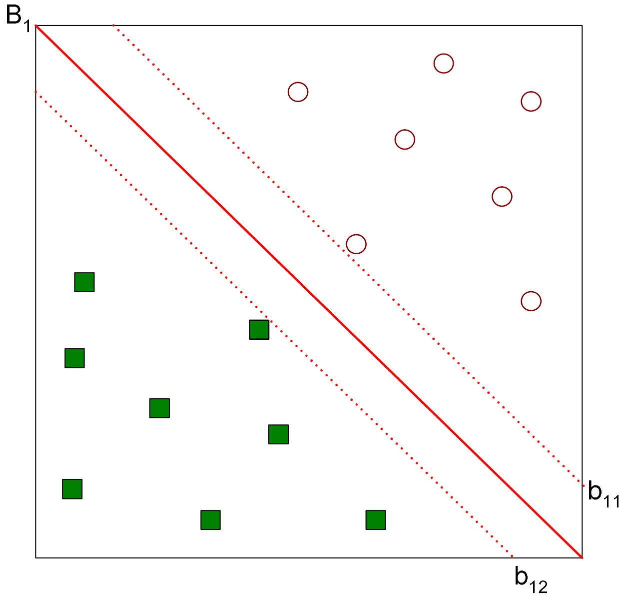
w and *b* are parameters of the model

$$\mathbf{w} \cdot \mathbf{x}_s + b = k, \qquad k > 0.$$

$$\mathbf{w} \cdot \mathbf{x}_c + b = k', \qquad k' < 0.$$

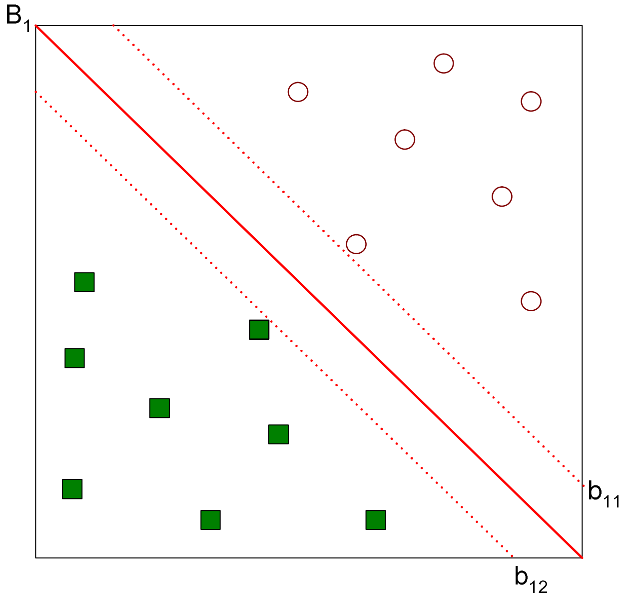
$$y = \begin{cases} 1, & \text{if } \mathbf{w} \cdot \mathbf{z} + b > 0; \\ -1, & \text{if } \mathbf{w} \cdot \mathbf{z} + b < 0. \end{cases}$$

Linear SVM: Separable Case



$$b_{i1}: \ \mathbf{w} \cdot \mathbf{x} + b = 1,$$
$$b_{i2}: \ \mathbf{w} \cdot \mathbf{x} + b = -1.$$

Linear SVM: Separable Case



$$b_{i1} : \ \mathbf{w} \cdot \mathbf{x} + b = 1,$$
$$b_{i2} : \ \mathbf{w} \cdot \mathbf{x} + b = -1.$$

margin of the decision boundary is given by the distance between these two hyperplanes

$$d = \frac{2}{||\mathbf{w}||}$$

SVM

$$\mathbf{w} \cdot \mathbf{x_i} + b \geq \ 1 \text{ if } y_i = 1,$$
$$\mathbf{w} \cdot \mathbf{x_i} + b \leq -1 \text{ if } y_i = -1$$

$$y_i(\mathbf{w} \cdot \mathbf{x_i} + b) \geq 1, \quad i = 1, 2, \ldots, N.$$

SVM

$$\mathbf{w} \cdot \mathbf{x_i} + b \geq 1 \text{ if } y_i = 1,$$
$$\mathbf{w} \cdot \mathbf{x_i} + b \leq -1 \text{ if } y_i = -1$$

$$y_i(\mathbf{w} \cdot \mathbf{x_i} + b) \geq 1, \quad i = 1, 2, \ldots, N.$$

Definition 5.1 (Linear SVM: Separable Case). The learning task in SVM can be formalized as the following constrained optimization problem:

$$\min_{\mathbf{w}} \frac{\|\mathbf{w}\|^2}{2}$$
$$\text{subject to} \quad y_i(\mathbf{w} \cdot \mathbf{x_i} + b) \geq 1, \quad i = 1, 2, \ldots, N.$$

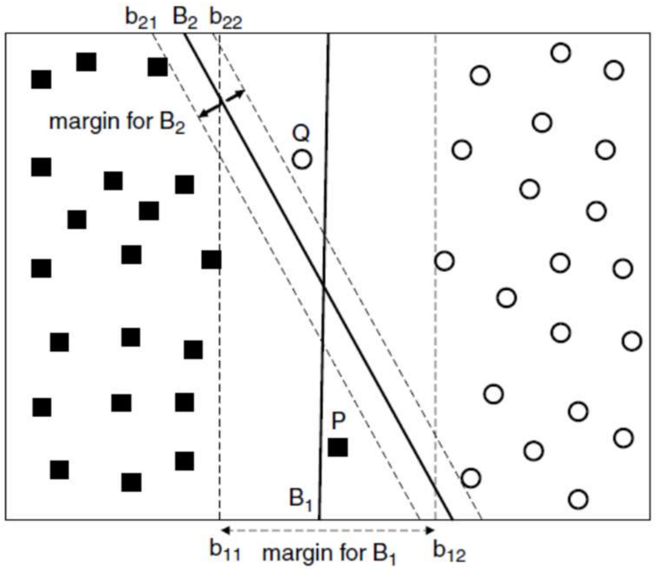
This is a constrained optimization problem

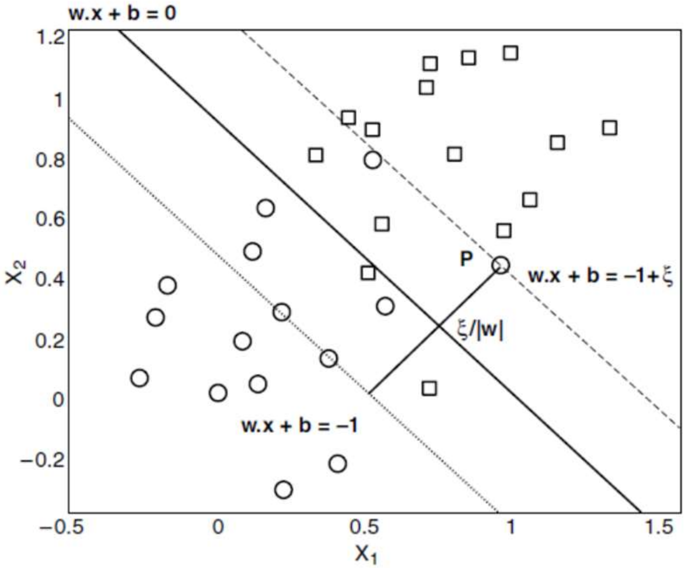
Numerical approaches to solve it (e.g., quadratic programming)

$$L_D = \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j \mathbf{x_i} \cdot \mathbf{x_j}.$$

$$\left(\sum_{i=1}^{N} \lambda_i y_i \mathbf{x_i} \cdot \mathbf{x} \right) + b = 0.$$

Linear SVM: Nonseparable Case

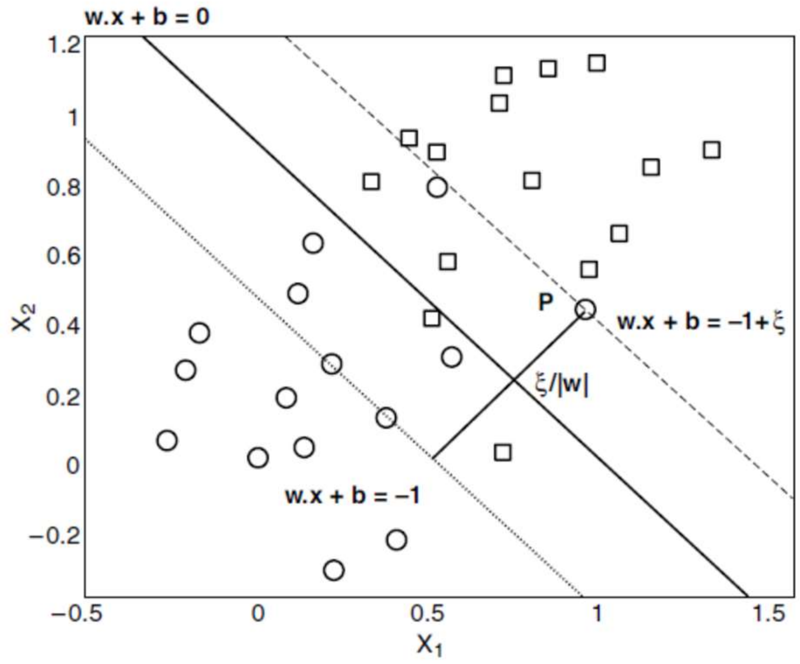




while B_2 classifies them correctly, this does not mean that B_2 is a better decision boundary than B_1

Linear SVM: Nonseparable Case

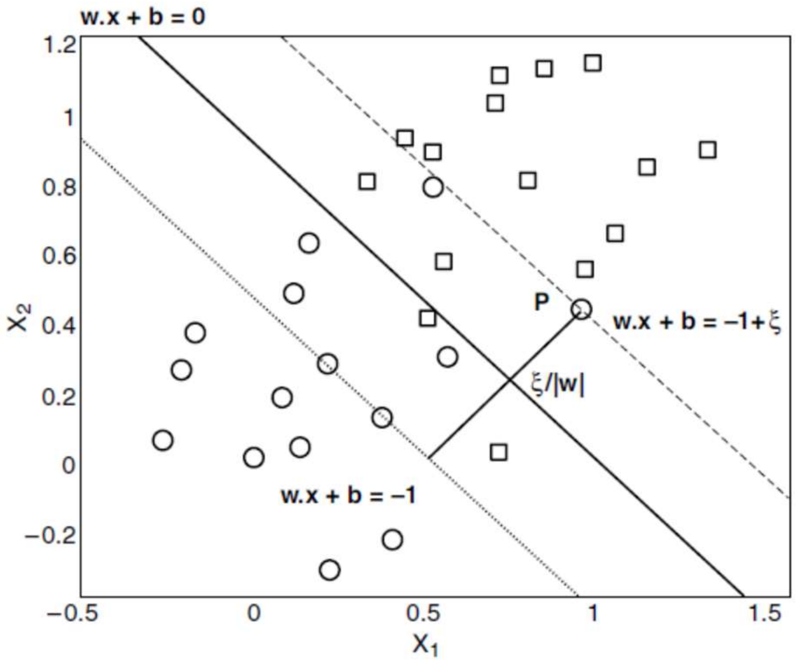
- ✓ learn a decision boundary that is tolerable to small training errors
- ✓ construct a linear decision boundary even in situations where the classes are not linearly separable
- ✓ inequality constraints must therefore be relaxed to accommodate the nonlinearly separable data



Linear SVM: Nonseparable Case

- ✓ learn a decision boundary that is tolerable to small training errors
- ✓ construct a linear decision boundary even in situations where the classes are not linearly separable
- ✓ inequality constraints must therefore be relaxed to accommodate the nonlinearly separable data

$$\mathbf{w} \cdot \mathbf{x_i} + b \geq 1 - \xi_i \quad \text{if } y_i = 1,$$
$$\mathbf{w} \cdot \mathbf{x_i} + b \leq -1 + \xi_i \quad \text{if } y_i = -1,$$



ξ provides an estimate of the error of the decision boundary

Linear SVM: Nonseparable Case

$$\min_{\mathbf{w}} \frac{\|\mathbf{w}\|^2}{2}$$

$$\mathbf{w} \cdot \mathbf{x_i} + b \geq 1 - \xi_i \quad \text{if } y_i = 1,$$

$$\mathbf{w} \cdot \mathbf{x_i} + b \leq -1 + \xi_i \quad \text{if } y_i = -1,$$

problem

Linear SVM: Nonseparable Case

$$\min_{\mathbf{w}} \frac{\|\mathbf{w}\|^2}{2}$$

$$\mathbf{w} \cdot \mathbf{x_i} + b \geq 1 - \xi_i \quad \text{if } y_i = 1,$$
$$\mathbf{w} \cdot \mathbf{x_i} + b \leq -1 + \xi_i \quad \text{if } y_i = -1,$$

problem

$$f(\mathbf{w}) = \frac{\|\mathbf{w}\|^2}{2} + C\left(\sum_{i=1}^{N} \xi_i\right)^k,$$

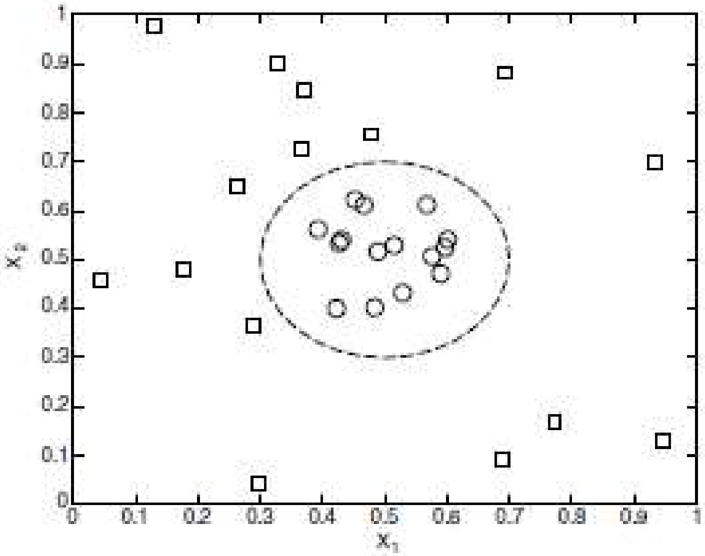
C and *k* are user-specified parameters

Nonlinear SVM

✓applying SVM to data sets that have nonlinear decision boundaries
✓transform the data **x into a new space Φ(x) so that a linear decision** boundary can be used to separate the instances in the transformed space

Nonlinear SVM

✓applying SVM to data sets that have nonlinear decision boundaries
✓transform the data **x into a new space Φ(x) so that a linear decision** boundary can be used to separate the instances in the transformed space



(a) Decision boundary in the original two-dimensional space.

$$y(x_1, x_2) = \begin{cases} 1 & \text{if } \sqrt{(x_1 - 0.5)^2 + (x_2 - 0.5)^2} > 0.2, \\ -1 & \text{otherwise.} \end{cases}$$

$$x_1^2 - x_1 + x_2^2 - x_2 = -0.46$$

$$\Phi : (x_1, x_2) \longrightarrow (x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, 1).$$

$$\mathbf{w} = (w_0, w_1, \ldots, w_4)$$

$$w_4 x_1^2 + w_3 x_2^2 + w_2\sqrt{2}x_1 + w_1\sqrt{2}x_2 + w_0 = 0.$$

Nonlinear SVM

Definition 5.2 (Nonlinear SVM). The learning task for a nonlinear SVM can be formalized as the following optimization problem:

$$\min_{\mathbf{w}} \frac{\|\mathbf{w}\|^2}{2}$$

$$\text{subject to} \quad y_i(\mathbf{w} \cdot \Phi(\mathbf{x}_i) + b) \geq 1, \quad i = 1, 2, \ldots, N.$$

Nonlinear SVM

Definition 5.2 (Nonlinear SVM). The learning task for a nonlinear SVM can be formalized as the following optimization problem:

$$\min_{\mathbf{w}} \frac{\|\mathbf{w}\|^2}{2}$$

subject to $\quad y_i(\mathbf{w} \cdot \Phi(\mathbf{x}_i) + b) \geq 1, \quad i = 1, 2, \ldots, N.$

$$L_D = \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j \mathbf{x_i} \cdot \mathbf{x_j}.$$

$$L_D = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$$

Nonlinear SVM

Definition 5.2 (Nonlinear SVM). The learning task for a nonlinear SVM can be formalized as the following optimization problem:

$$\min_{\mathbf{w}} \frac{\|\mathbf{w}\|^2}{2}$$

$$\text{subject to} \quad y_i(\mathbf{w} \cdot \Phi(\mathbf{x}_i) + b) \geq 1, \quad i = 1, 2, \ldots, N.$$

$$\begin{aligned}
\Phi(\mathbf{u}) \cdot \Phi(\mathbf{v}) &= (u_1^2, u_2^2, \sqrt{2}u_1, \sqrt{2}u_2, 1) \cdot (v_1^2, v_2^2, \sqrt{2}v_1, \sqrt{2}v_2, 1) \\
&= u_1^2 v_1^2 + u_2^2 v_2^2 + 2u_1 v_1 + 2u_2 v_2 + 1 \\
&= (\mathbf{u} \cdot \mathbf{v} + 1)^2.
\end{aligned}$$

$$K(\mathbf{u}, \mathbf{v}) = \Phi(\mathbf{u}) \cdot \Phi(\mathbf{v}) = (\mathbf{u} \cdot \mathbf{v} + 1)^2$$

Nonlinear SVM

The main requirement for the kernel function used in nonlinear SVM is that there must exist a corresponding transformation such that the kernel function computed for a pair of vectors is equivalent to the dot product between the vectors in the transformed space.

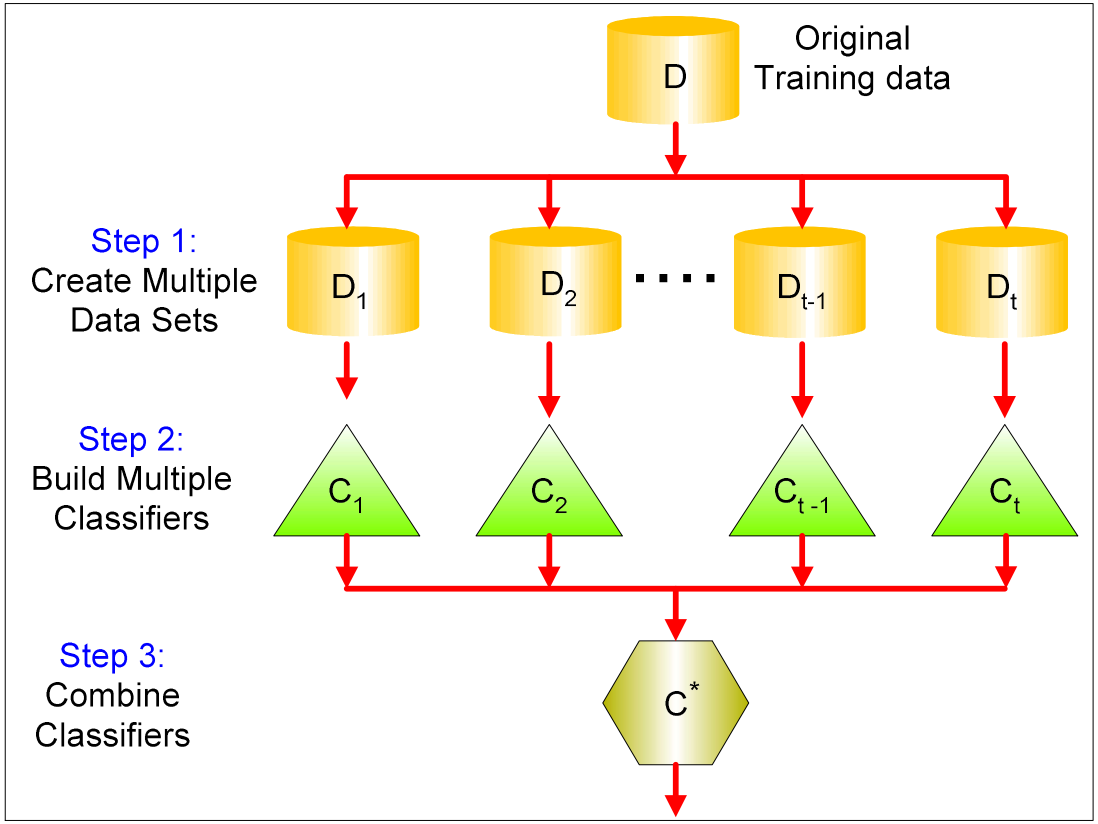
$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^p$$
$$K(\mathbf{x}, \mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|^2/(2\sigma^2)}$$
$$K(\mathbf{x}, \mathbf{y}) = \tanh(k\mathbf{x} \cdot \mathbf{y} - \delta)$$

105

Ensemble Methods

Ensemble Methods

- ✓ improving classification accuracy by aggregating the predictions of multiple classifiers
- ✓ Construct a set of base classifiers from the training data
- ✓ Predict class label of previously unseen records by aggregating predictions made by multiple classifiers

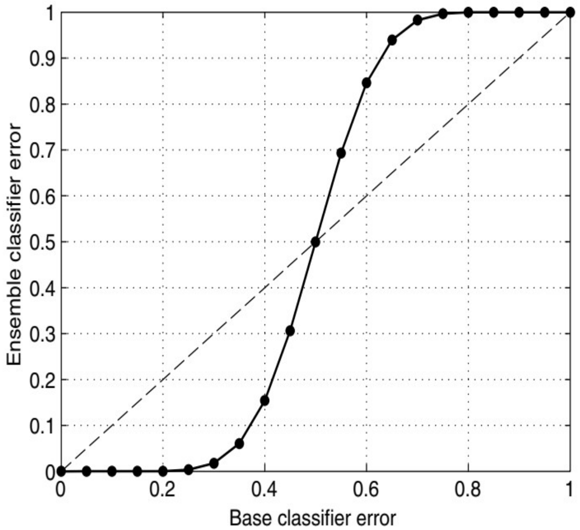


Ensemble Methods

- Suppose there are 25 base classifiers
  - Each classifier has error rate, ε = 0.35
  - Assume classifiers are independent
  - Probability that the ensemble classifier makes a wrong prediction:

$$\sum_{i=13}^{25} \binom{25}{i} \varepsilon^{i} (1-\varepsilon)^{25-i} = 0.06$$

base classifiers should be independent of each other
base classifiers should do better than a classifier that performs random guessing.



How to generate an ensemble of classifiers?

- Bagging: bootstrap aggregating
- Boosting

Bagging

- Sampling with replacement
- Build classifier on each bootstrap sample

Original Data	1	2	3	4	5	6	7	8	9	10
Bagging (Round 1)	7	8	10	8	2	5	10	10	5	9
Bagging (Round 2)	1	4	9	1	2	3	2	7	3	2
Bagging (Round 3)	1	8	5	10	5	5	9	6	3	7

Algorithm 5.6 Bagging algorithm.

1: Let k be the number of bootstrap samples.
2: **for** $i = 1$ to k **do**
3: Create a bootstrap sample of size N, D_i.
4: Train a base classifier C_i on the bootstrap sample D_i.
5: **end for**
6: $C^*(x) = \underset{y}{\operatorname{argmax}} \sum_i \delta\big(C_i(x) = y\big)$.
$\{\delta(\cdot) = 1$ if its argument is true and 0 otherwise$\}$.

Boosting

- An iterative procedure to adaptively change distribution of training data by focusing more on previously misclassified records
  - Initially, all N records are assigned equal weights
  - Unlike bagging, weights may change at the end of boosting round

1. how the weights of the training examples are updated at the end of each boosting round
2. how the predictions made by each classifier are combined.

Boosting: AdaBoost

Let $\{(\mathbf{x}_j, y_j) \mid j = 1, 2, \ldots, N\}$ denote a set of N training examples.
Unlike bagging, importance of a base classifier Ci depends on its error rate

$$\epsilon_i = \frac{1}{N}\left[\sum_{j=1}^{N} w_j \, I\Big(C_i(\mathbf{x}_j) \neq y_j\Big)\right], \qquad \alpha_i = \frac{1}{2}\ln\left(\frac{1-\epsilon_i}{\epsilon_i}\right)$$

weight assigned to example ($\mathbf{x}i$, yi) during the *jth* boosting round

$$w_i^{(j+1)} = \frac{w_i^{(j)}}{Z_j} \times \begin{cases} \exp^{-\alpha_j} & \text{if } C_j(\mathbf{x_i}) = y_i \\ \exp^{\alpha_j} & \text{if } C_j(\mathbf{x_i}) \neq y_i \end{cases}$$

Normalization factor

intermediate rounds produce an error rate higher than 50%, the weights wi = 1/N

AdaBoost

Algorithm 5.7 AdaBoost algorithm.

1: $\mathbf{w} = \{w_j = 1/N \mid j = 1, 2, \ldots, N\}$. {Initialize the weights for all N examples.}
2: Let k be the number of boosting rounds.
3: **for** $i = 1$ to k **do**
4: Create training set D_i by sampling (with replacement) from D according to $\mathbf{w}$.
5: Train a base classifier C_i on D_i.
6: Apply C_i to all examples in the original training set, D.
7: $\epsilon_i = \frac{1}{N}\left[\sum_j w_j\ \delta\big(C_i(x_j) \neq y_j\big)\right]$ {Calculate the weighted error.}
8: **if** $\epsilon_i > 0.5$ **then**
9: $\mathbf{w} = \{w_j = 1/N \mid j = 1, 2, \ldots, N\}$. {Reset the weights for all N examples.}
10: Go back to Step 4.
11: **end if**
12: $\alpha_i = \frac{1}{2} \ln \frac{1-\epsilon_i}{\epsilon_i}$.
13: Update the weight of each example according to Equation 5.69.
14: **end for**
15: $C^*(\mathbf{x}) = \underset{y}{\operatorname{argmax}} \sum_{j=1}^{T} \alpha_j \delta(C_j(\mathbf{x}) = y)$.

Example

x	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
y	1	1	1	-1	-1	-1	-1	1	1	1

Boosting Round 1:

x	0.1	0.4	0.5	0.6	0.6	0.7	0.7	0.7	0.8	1
y	1	-1	-1	-1	-1	-1	-1	-1	1	1

Boosting Round 2:

x	0.1	0.1	0.2	0.2	0.2	0.2	0.3	0.3	0.3	0.3
y	1	1	1	1	1	1	1	1	1	1

Boosting Round 3:

x	0.2	0.2	0.4	0.4	0.4	0.4	0.5	0.6	0.6	0.7
y	1	1	-1	-1	-1	-1	-1	-1	-1	-1

Round	x=0.1	x=0.2	x=0.3	x=0.4	x=0.5	x=0.6	x=0.7	x=0.8	x=0.9	x=1.0
1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
2	0.311	0.311	0.311	0.01	0.01	0.01	0.01	0.01	0.01	0.01
3	0.029	0.029	0.029	0.228	0.228	0.228	0.228	0.009	0.009	0.009

Random Forest

- ✓ ensemble methods specifically designed for decision tree classifiers
- ✓ multiple decision trees where each tree is generated based on the values of an independent set of random vectors

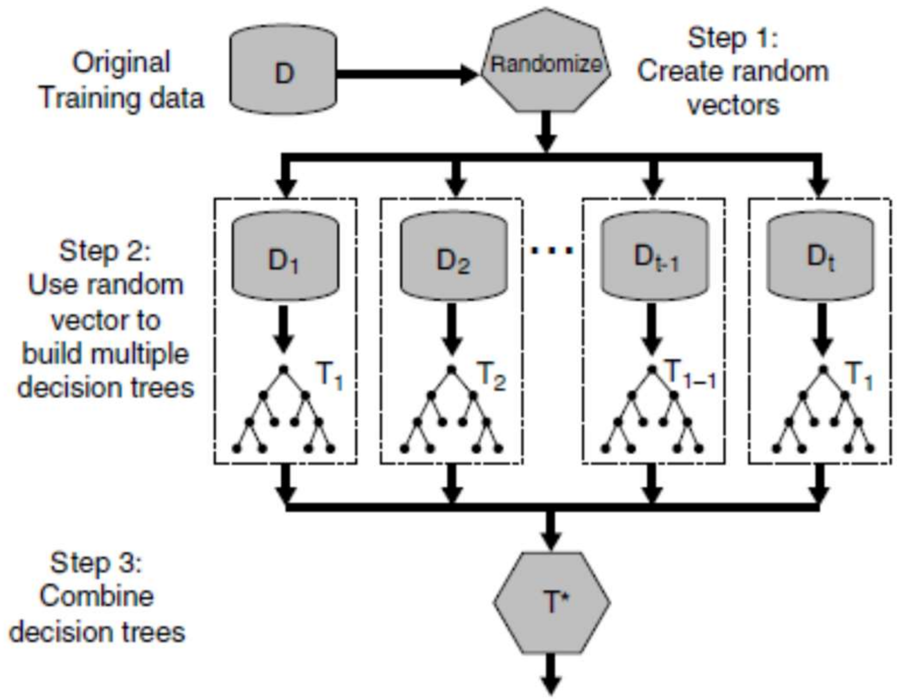


Figure 5.40. Random forests.

Random Forest

- ✓ Bagging using decision trees is a special case of random forests
- ✓ randomly select *F* input features to split at each node of the decision tree
- ✓ majority voting scheme
- ✓ To increase randomness, bagging can also be used to generate bootstrap samples for Forest-RI

Metrics for class imbalance problem

Imbalance

- ✓ Data sets with imbalanced class distributions
- ✓ in credit card fraud detection, fraudulent transactions are outnumbered by legitimate transactions
- ✓ accuracy measure, used extensively for classifiers, may not be well suited for evaluating models derived from imbalanced data sets

example : 1% of the credit card transactions fraudulent,
a model that predicts every transaction as legitimate
accuracy 99%
it fails to detect any of the fraudulent activities.

binary classification, the rare class is often denoted as the positive class against negative class

		Predicted Class	
		+	−
Actual	+	f_{++} (TP)	f_{+-} (FN)
Class	−	f_{-+} (FP)	f_{--} (TN)

confusion matrix

Imbalance

Precision : fraction of records that actually turns out to be positive in the group the classifier has declared as a positive class

$$\boxed{\text{Precision}, p = \frac{TP}{TP + FP}}$$

Recall measures the fraction of positive examples correctly predicted by the classifier

$$\boxed{\text{Recall}, r = \frac{TP}{TP + FN}}$$

maximizes both precision and recall

Imbalance

Precision and recall can be summarized into another metric known as the F1 measure

$$\boxed{F_1 = \frac{2}{\frac{1}{r} + \frac{1}{p}}.}$$

tends to be closer to the smaller of the two numbers
a high value of *F*1-measure ensures that both precision and recall are reasonably high

$$\boxed{\text{Weighted accuracy} = \frac{w_1 TP + w_4 TN}{w_1 TP + w_2 FP + w_3 FN + w_4 TN}.}$$