Classification and Regression II

Krzysztof Dembczyński

Intelligent Decision Support Systems Laboratory (IDSS)
Poznań University of Technology, Poland

Software Development Technologies
Master studies, second semester
Academic year 2018/19 (winter course)

Review of the previous lectures

- Mining of massive datasets.
- Classification and regression
  - What is machine learning?
  - Supervised learning: statistical decision/learning theory, loss functions, risk.
  - Learning paradigms and principles.

Outline

Outline

Learning algorithms

- Lazy learning (histogram-based classifiers, nearest neighbors),
- Decision trees,
- Generative models,
- Linear models (linear regression, logistic regression, SVM),
- Kernel methods,
- Ensemble methods,
- Deep learning.

Outline

Lazy learning

- Based on empirical distribution and direct application of the Bayes rule to local estimates of $P(y \mid \boldsymbol{x})$.

Lazy learning

- Based on empirical distribution and direct application of the Bayes rule to local estimates of $P(y \mid \boldsymbol{x})$.
- The simplest approach estimates conditional probabilities $P(y \mid \boldsymbol{x})$ for any $\boldsymbol{x}$ from training data:
  - Based on *group-bys* and simple counting.
  - Needs a lot of data to get reasonable estimates!!!
  - Data should be discrete/nominal or we need to discretize numerical data before.

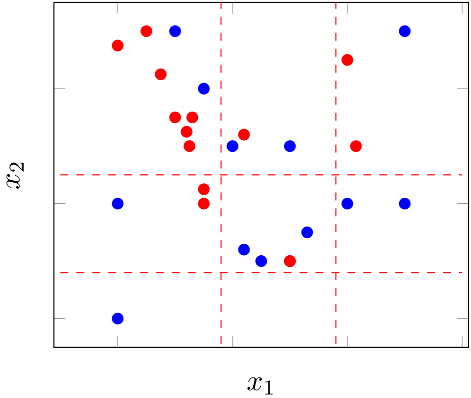
Example

gold	price	spam?
1	1	1
1	1	1
1	1	1
1	1	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	1
0	0	1
0	1	0
0	1	1

$$\begin{aligned}
P(y=1|\text{gold}=1 \wedge \text{price}=1) &= 0.75 \\
P(y=0|\text{gold}=1 \wedge \text{price}=1) &= 0.25 \\
P(y=1|\text{gold}=0 \wedge \text{price}=0) &= 0.33 \\
P(y=0|\text{gold}=0 \wedge \text{price}=0) &= 0.66 \\
P(y=1|\text{gold}=0 \wedge \text{price}=1) &= 0.5 \\
P(y=0|\text{gold}=0 \wedge \text{price}=1) &= 0.5 \\
P(y=1|\text{gold}=1 \wedge \text{price}=0) &= ? \\
P(y=0|\text{gold}=1 \wedge \text{price}=0) &= ?
\end{aligned}$$

Histogram-based methods

- Build a multidimensional grid and estimate the conditional probability in each element of the grid,
- Plug the estimates to the Bayes classifier for a given $\ell(y, \hat{y})$ to obtain prediction.



Histogram-based methods

- The predictive performance depends on the grid resolution, dimensionality of data and the size of training data.

Histogram-based methods

- The predictive performance depends on the grid resolution, dimensionality of data and the size of training data.
- With some tricks can be efficiently implemented.

Histogram-based methods

- The predictive performance depends on the grid resolution, dimensionality of data and the size of training data.
- With some tricks can be efficiently implemented.
- Piecewise-constant prediction for a given region.

Histogram-based methods

- The predictive performance depends on the grid resolution, dimensionality of data and the size of training data.
- With some tricks can be efficiently implemented.
- Piecewise-constant prediction for a given region.
- Computation of the estimates in the region: well-know statistical problem, properties of estimates, maximum likelihood estimates, regularization.

Histogram-based methods

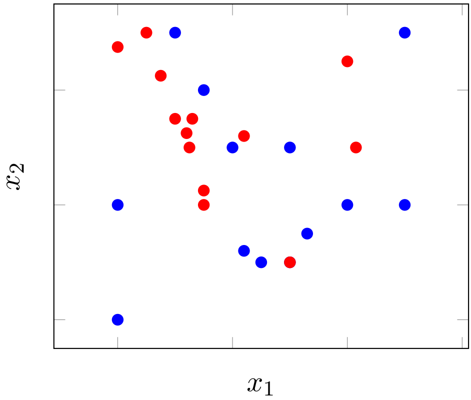
- The predictive performance depends on the grid resolution, dimensionality of data and the size of training data.
- With some tricks can be efficiently implemented.
- Piecewise-constant prediction for a given region.
- Computation of the estimates in the region: well-know statistical problem, properties of estimates, maximum likelihood estimates, regularization.
- The grid can be given as a domain knowledge, simple discretization, or random splits.

Histogram-based methods

- The predictive performance depends on the grid resolution, dimensionality of data and the size of training data.
- With some tricks can be efficiently implemented.
- Piecewise-constant prediction for a given region.
- Computation of the estimates in the region: well-know statistical problem, properties of estimates, maximum likelihood estimates, regularization.
- The grid can be given as a domain knowledge, simple discretization, or random splits.
- One can use more intelligent methods to obtain a grid, for example, supervised discretization or supervised recursive splitting like in decision trees.

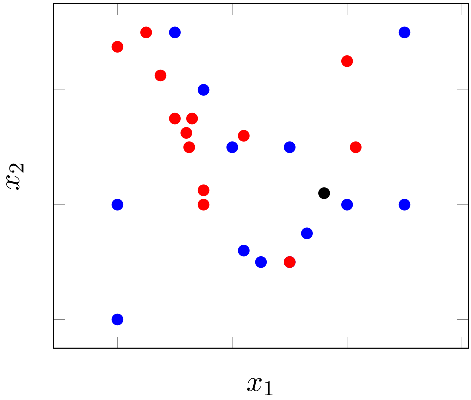
Nearest neighbor methods

- Find k-nearest neighbors of the test example.
- Estimate the Bayes classifier based on the neighborhood.



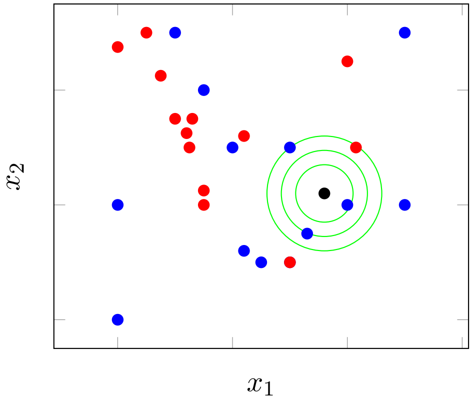
Nearest neighbor methods

- Find k-nearest neighbors of the test example.
- Estimate the Bayes classifier based on the neighborhood.



Nearest neighbor methods

- Find k-nearest neighbors of the test example.
- Estimate the Bayes classifier based on the neighborhood.



Nearest neighbor methods

- Prediction for a test example is computed based on nearest training examples – there is no learning.

Nearest neighbor methods

- Prediction for a test example is computed based on nearest training examples – there is no learning.
- The same principles for computing the prediction as in histogram-based and tree classifiers.

Nearest neighbor methods

- Prediction for a test example is computed based on nearest training examples – there is no learning.
- The same principles for computing the prediction as in histogram-based and tree classifiers.
- Training set can be used for tunning k and finding a metric.

Nearest neighbor methods

- Prediction for a test example is computed based on nearest training examples – there is no learning.
- The same principles for computing the prediction as in histogram-based and tree classifiers.
- Training set can be used for tunning k and finding a metric.
- Specialized data structures for efficient search of nearest neighbors.

Nearest neighbor methods

- Prediction for a test example is computed based on nearest training examples – there is no learning.
- The same principles for computing the prediction as in histogram-based and tree classifiers.
- Training set can be used for tunning k and finding a metric.
- Specialized data structures for efficient search of nearest neighbors.
- Reduction of training data: prototypes, feature selection, dimensionality reduction by PCA or similar methods.

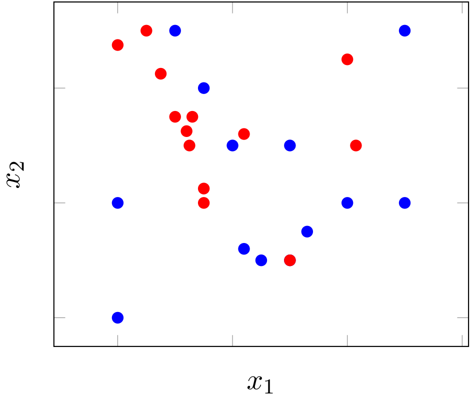
Nearest neighbor methods

- Prediction for a test example is computed based on nearest training examples – there is no learning.
- The same principles for computing the prediction as in histogram-based and tree classifiers.
- Training set can be used for tunning k and finding a metric.
- Specialized data structures for efficient search of nearest neighbors.
- Reduction of training data: prototypes, feature selection, dimensionality reduction by PCA or similar methods.
- Approximate nearest neighbors.

Outline

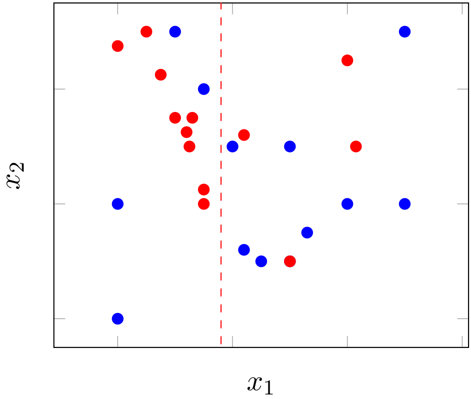
Decision trees

- Recursively make a partition of the feature space (in a smart way).
- Compute (Bayes) decision in each region.



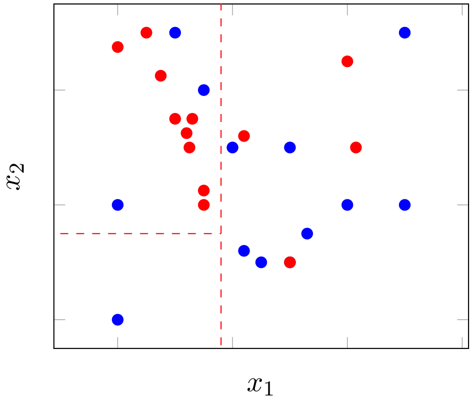
Decision trees

- Recursively make a partition of the feature space (in a smart way).
- Compute (Bayes) decision in each region.



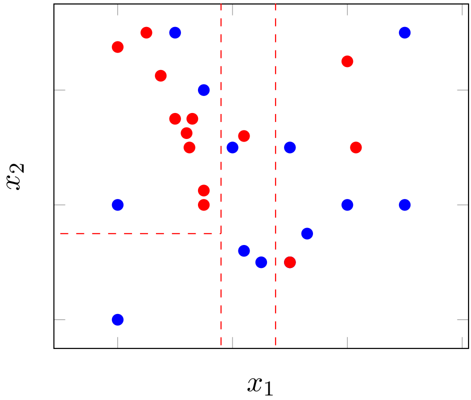
Decision trees

- Recursively make a partition of the feature space (in a smart way).
- Compute (Bayes) decision in each region.



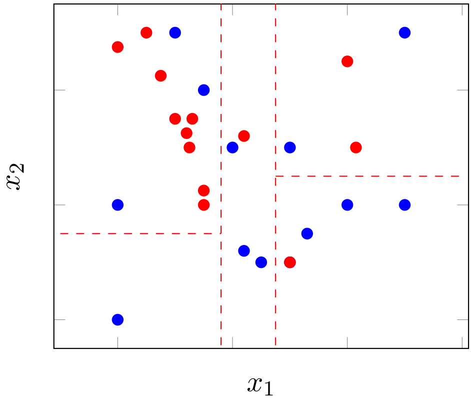
Decision trees

- Recursively make a partition of the feature space (in a smart way).
- Compute (Bayes) decision in each region.

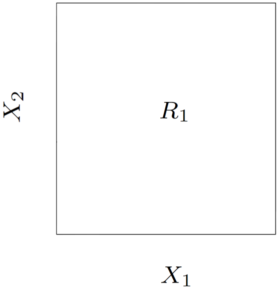


Decision trees

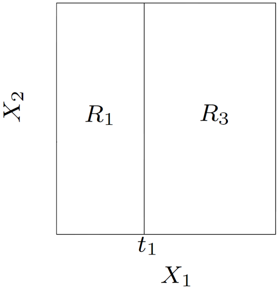
- Recursively make a partition of the feature space (in a smart way).
- Compute (Bayes) decision in each region.

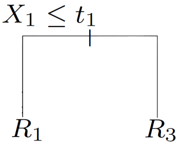


Decision trees

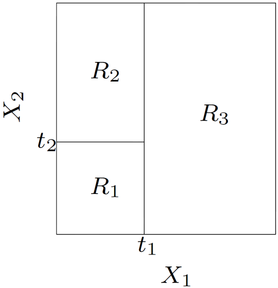


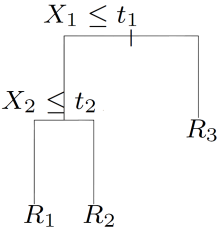
Decision trees



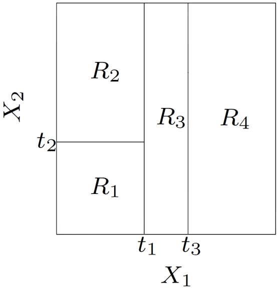


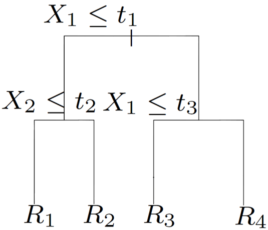
Decision trees



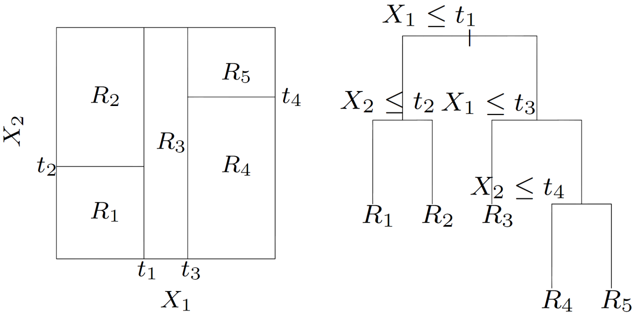


Decision trees





Decision trees



Decision trees

- The learning method seeks an optimal tree shape (e.g. feature space partition) by minimizing the empirical risk (usually expressed in terms of a surrogate loss).

Decision trees

- The learning method seeks an optimal tree shape (e.g. feature space partition) by minimizing the empirical risk (usually expressed in terms of a surrogate loss).
- Greedy methods used for constructing a tree.

Decision trees

- The learning method seeks an optimal tree shape (e.g. feature space partition) by minimizing the empirical risk (usually expressed in terms of a surrogate loss).
- Greedy methods used for constructing a tree.
- The resulting model can be easily interpreted.

Decision trees

- The learning method seeks an optimal tree shape (e.g. feature space partition) by minimizing the empirical risk (usually expressed in terms of a surrogate loss).
- Greedy methods used for constructing a tree.
- The resulting model can be easily interpreted.
- The most influential splits are close to the root (like in the 20-question game).

Decision trees

- The learning method seeks an optimal tree shape (e.g. feature space partition) by minimizing the empirical risk (usually expressed in terms of a surrogate loss).
- Greedy methods used for constructing a tree.
- The resulting model can be easily interpreted.
- The most influential splits are close to the root (like in the 20-question game).
- Learning and prediction is very efficient.

Decision trees

- The learning method seeks an optimal tree shape (e.g. feature space partition) by minimizing the empirical risk (usually expressed in terms of a surrogate loss).
- Greedy methods used for constructing a tree.
- The resulting model can be easily interpreted.
- The most influential splits are close to the root (like in the 20-question game).
- Learning and prediction is very efficient.
- Estimation of the decision in each leaf – the same problem like in histogram-based methods.

Outline

Generative models

- A **generative model** is a model for randomly generating observable-data values, typically given some hidden parameters.
- It specifies a joint probability distribution over observations and outcomes.
- Generative models rely on the Bayes theorem.

Generative models

- For classification, we have:

$$P(y=k|\boldsymbol{x}) = \frac{P(y=k,\boldsymbol{x})}{P(\boldsymbol{x})} = \frac{P(\boldsymbol{x}|y=k)P(y=k)}{P(\boldsymbol{x})}$$

  where $P(\boldsymbol{x}|y=k)$ is the density function $f_k(\boldsymbol{x})$ (for example, multivariate Gaussian distribution), and $P(\boldsymbol{x})$ is given by:

Generative models

- For classification, we have:

$$P(y = k|\boldsymbol{x}) = \frac{P(y = k, \boldsymbol{x})}{P(\boldsymbol{x})} = \frac{P(\boldsymbol{x}|y = k)P(y = k)}{P(\boldsymbol{x})}$$

  where $P(\boldsymbol{x}|y = k)$ is the density function $f_k(\boldsymbol{x})$ (for example, multivariate Gaussian distribution), and $P(\boldsymbol{x})$ is given by:

$$P(\boldsymbol{x}) = \sum_{k'} P(\boldsymbol{x}|y = k')P(y = k')$$

  from the law of total probability.

Generative models

- The main algorithms:
  - Linear and quadratic discriminant analysis that use Gaussian densities,
  - General nonparametric density estimates for each class density,
  - Naive Bayes model that assumes that each of the class densities are products of marginal densities, i.e., the features are conditionally independent in each class.

Naive Bayes

- The naive Bayes model assumes that given a class $y = k$, the features $\boldsymbol{x} = (x_1, x_2, \ldots, x_m)$ are independent:

$$P(\boldsymbol{x}|y) =$$

Naive Bayes

- The naive Bayes model assumes that given a class $y = k$, the features $\boldsymbol{x} = (x_1, x_2, \ldots, x_m)$ are independent:

$$P(\boldsymbol{x}|y) = \prod_{j=1}^{m} P(x_j|y).$$

Naive Bayes

- The naive Bayes model assumes that given a class $y = k$, the features $\boldsymbol{x} = (x_1, x_2, \ldots, x_m)$ are independent:

$$P(\boldsymbol{x}|y) = \prod_{j=1}^{m} P(x_j|y).$$

- The model takes the following form:

$$P(y = k|\boldsymbol{x}) =$$

Naive Bayes

- The naive Bayes model assumes that given a class $y = k$, the features $\boldsymbol{x} = (x_1, x_2, \ldots, x_m)$ are independent:

$$P(\boldsymbol{x}|y) = \prod_{j=1}^{m} P(x_j|y).$$

- The model takes the following form:

$$P(y = k|\boldsymbol{x}) = \frac{P(y = k) \prod_{j=1}^{m} P(x_j|y = k)}{\sum_{k'} P(y = k') \prod_{j=1}^{m} P(x_j|y = k')}$$

Naive Bayes

- The naive Bayes model assumes that given a class $y = k$, the features $\boldsymbol{x} = (x_1, x_2, \ldots, x_m)$ are independent:

$$P(\boldsymbol{x}|y) = \prod_{j=1}^{m} P(x_j|y).$$

- The model takes the following form:

$$P(y = k|\boldsymbol{x}) = \frac{P(y = k) \prod_{j=1}^{m} P(x_j|y = k)}{\sum_{k'} P(y = k') \prod_{j=1}^{m} P(x_j|y = k')}$$

- The individual class-conditional marginal densities f_{jk} can each be estimated separately using univariate Gaussian distributions:

$$N(\mathbb{E}(x_j|y = k), \mathrm{Var}(x_j|y = k))$$

- If a component x_j of $\boldsymbol{x}$ is discrete, then an appropriate histogram estimate can be used.

Naive Bayes

Example

gold	price	spam?
1	1	1
1	1	1
1	1	1
1	1	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	1
0	0	1
0	1	0
0	1	1

$P(y = 1) =$

Naive Bayes

Example

gold	price	spam?
1	1	1
1	1	1
1	1	1
1	1	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	1
0	0	1
0	1	0
0	1	1

$P(y = 1) = 0.5$

$P(y = 0) =$

Naive Bayes

Example

gold	price	spam?
1	1	1
1	1	1
1	1	1
1	1	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	1
0	0	1
0	1	0
0	1	1

$$P(y = 1) = 0.5$$
$$P(y = 0) = 0.5$$
$$P(\text{gold} = 1 | Y = 1) =$$

Naive Bayes

Example

gold	price	spam?
1	1	1
1	1	1
1	1	1
1	1	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	1
0	0	1
0	1	0
0	1	1

$$P(y=1) = 0.5$$
$$P(y=0) = 0.5$$
$$P(\text{gold}=1|Y=1) = 0.5$$
$$P(\text{gold}=0|Y=1) =$$

Naive Bayes

Example

gold	price	spam?
1	1	1
1	1	1
1	1	1
1	1	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	1
0	0	1
0	1	0
0	1	1

$$P(y = 1) = 0.5$$
$$P(y = 0) = 0.5$$
$$P(\text{gold} = 1|Y = 1) = 0.5$$
$$P(\text{gold} = 0|Y = 1) = 0.5$$
$$P(\text{gold} = 1|Y = 0) =$$

Naive Bayes

Example

gold	price	spam?
1	1	1
1	1	1
1	1	1
1	1	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	1
0	0	1
0	1	0
0	1	1

$$P(y = 1) = 0.5$$
$$P(y = 0) = 0.5$$
$$P(\text{gold} = 1|Y = 1) = 0.5$$
$$P(\text{gold} = 0|Y = 1) = 0.5$$
$$P(\text{gold} = 1|Y = 0) = 0.17$$
$$P(\text{gold} = 0|Y = 0) =$$

Naive Bayes

Example

gold	price	spam?
1	1	1
1	1	1
1	1	1
1	1	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	1
0	0	1
0	1	0
0	1	1

$$P(y = 1) = 0.5$$
$$P(y = 0) = 0.5$$
$$P(\text{gold} = 1|Y = 1) = 0.5$$
$$P(\text{gold} = 0|Y = 1) = 0.5$$
$$P(\text{gold} = 1|Y = 0) = 0.17$$
$$P(\text{gold} = 0|Y = 0) = 0.83$$
$$P(\text{price} = 1|Y = 1) =$$

Naive Bayes

Example

gold	price	spam?
1	1	1
1	1	1
1	1	1
1	1	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	1
0	0	1
0	1	0
0	1	1

$$P(y = 1) = 0.5$$
$$P(y = 0) = 0.5$$
$$P(\text{gold} = 1|Y = 1) = 0.5$$
$$P(\text{gold} = 0|Y = 1) = 0.5$$
$$P(\text{gold} = 1|Y = 0) = 0.17$$
$$P(\text{gold} = 0|Y = 0) = 0.83$$
$$P(\text{price} = 1|Y = 1) = 0.66$$
$$P(\text{price} = 0|Y = 1) =$$

Naive Bayes

Example

gold	price	spam?
1	1	1
1	1	1
1	1	1
1	1	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	1
0	0	1
0	1	0
0	1	1

$P(y = 1) = 0.5$
$P(y = 0) = 0.5$
$P(\text{gold} = 1|Y = 1) = 0.5$
$P(\text{gold} = 0|Y = 1) = 0.5$
$P(\text{gold} = 1|Y = 0) = 0.17$
$P(\text{gold} = 0|Y = 0) = 0.83$
$P(\text{price} = 1|Y = 1) = 0.66$
$P(\text{price} = 0|Y = 1) = 0.33$
$P(\text{price} = 1|Y = 0) =$

Naive Bayes

Example

gold	price	spam?
1	1	1
1	1	1
1	1	1
1	1	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	1
0	0	1
0	1	0
0	1	1

$$
\begin{aligned}
&P(y=1) = 0.5 \\
&P(y=0) = 0.5 \\
&P(\text{gold}=1|Y=1) = 0.5 \\
&P(\text{gold}=0|Y=1) = 0.5 \\
&P(\text{gold}=1|Y=0) = 0.17 \\
&P(\text{gold}=0|Y=0) = 0.83 \\
&P(\text{price}=1|Y=1) = 0.66 \\
&P(\text{price}=0|Y=1) = 0.33 \\
&P(\text{price}=1|Y=0) = 0.33 \\
&P(\text{price}=0|Y=0) =
\end{aligned}
$$

Naive Bayes

Example

gold	price	spam?
1	1	1
1	1	1
1	1	1
1	1	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	1
0	0	1
0	1	0
0	1	1

$$
\begin{aligned}
&P(y=1) = 0.5\\
&P(y=0) = 0.5\\
&P(\text{gold}=1|Y=1) = 0.5\\
&P(\text{gold}=0|Y=1) = 0.5\\
&P(\text{gold}=1|Y=0) = 0.17\\
&P(\text{gold}=0|Y=0) = 0.83\\
&P(\text{price}=1|Y=1) = 0.66\\
&P(\text{price}=0|Y=1) = 0.33\\
&P(\text{price}=1|Y=0) = 0.33\\
&P(\text{price}=0|Y=0) = 0.66
\end{aligned}
$$

We can, for example, compute:

$$P(y=1|\text{gold}=1 \wedge \text{price}=0) =$$

Example

gold	price	spam?
1	1	1
1	1	1
1	1	1
1	1	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	1
0	0	1
0	1	0
0	1	1

$$P(y=1) = 0.5$$
$$P(y=0) = 0.5$$
$$P(\text{gold}=1|Y=1) = 0.5$$
$$P(\text{gold}=0|Y=1) = 0.5$$
$$P(\text{gold}=1|Y=0) = 0.17$$
$$P(\text{gold}=0|Y=0) = 0.83$$
$$P(\text{price}=1|Y=1) = 0.66$$
$$P(\text{price}=0|Y=1) = 0.33$$
$$P(\text{price}=1|Y=0) = 0.33$$
$$P(\text{price}=0|Y=0) = 0.66$$

We can, for example, compute:

$$P(y=1|\text{gold}=1 \wedge \text{price}=0) = \frac{0.5 \times 0.33 \times 0.5}{0.1386} = \frac{0.0825}{0.1386} = 0.595$$
$$P(y=0|\text{gold}=1 \wedge \text{price}=0) =$$

Naive Bayes

Example

gold	price	spam?
1	1	1
1	1	1
1	1	1
1	1	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	1
0	0	1
0	1	0
0	1	1

$$P(y=1) = 0.5$$
$$P(y=0) = 0.5$$
$$P(\text{gold}=1|Y=1) = 0.5$$
$$P(\text{gold}=0|Y=1) = 0.5$$
$$P(\text{gold}=1|Y=0) = 0.17$$
$$P(\text{gold}=0|Y=0) = 0.83$$
$$P(\text{price}=1|Y=1) = 0.66$$
$$P(\text{price}=0|Y=1) = 0.33$$
$$P(\text{price}=1|Y=0) = 0.33$$
$$P(\text{price}=0|Y=0) = 0.66$$

We can, for example, compute:

$$P(y=1|\text{gold}=1 \wedge \text{price}=0) = \frac{0.5 \times 0.33 \times 0.5}{0.1386} = \frac{0.0825}{0.1386} = 0.595$$
$$P(y=0|\text{gold}=1 \wedge \text{price}=0) = 1 - 0.595 = 0.405$$

Naive Bayes

Naive Bayes

- If the independence assumption is not valid, then the model can provide very bad predictions.

Naive Bayes

- If the independence assumption is not valid, then the model can provide very bad predictions.
- In many applications, however, this assumption seems to be at least partially satisfied, for example, in text classification.

Naive Bayes

- If the independence assumption is not valid, then the model can provide very bad predictions.
- In many applications, however, this assumption seems to be at least partially satisfied, for example, in text classification.
- Training is very efficient: one pass over training data to collect all necessary statistics.

Naive Bayes

- If the independence assumption is not valid, then the model can provide very bad predictions.
- In many applications, however, this assumption seems to be at least partially satisfied, for example, in text classification.
- Training is very efficient: one pass over training data to collect all necessary statistics.
- Prediction is linear in a number of features.

Naive Bayes

- If the independence assumption is not valid, then the model can provide very bad predictions.
- In many applications, however, this assumption seems to be at least partially satisfied, for example, in text classification.
- Training is very efficient: one pass over training data to collect all necessary statistics.
- Prediction is linear in a number of features.
- Some tricks to improve quality of computed statistics: Laplace correction and similar.

Naive Bayes

- If the independence assumption is not valid, then the model can provide very bad predictions.
- In many applications, however, this assumption seems to be at least partially satisfied, for example, in text classification.
- Training is very efficient: one pass over training data to collect all necessary statistics.
- Prediction is linear in a number of features.
- Some tricks to improve quality of computed statistics: Laplace correction and similar.

Question

Is Naive Bayes a linear classifier? **Prove** under which conditions it is true.

Outline

Linear models

Linear models

- Consider a linear model of the form:

$$f(\boldsymbol{x}) = w_0 + \sum_{j=1}^{m} w_j x_j.$$

where $\boldsymbol{w} = (w_0, w_1, \ldots, w_m)$ are the parameters of the model and $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ is a feature vector describing an example.

Linear models

- Consider a linear model of the form:

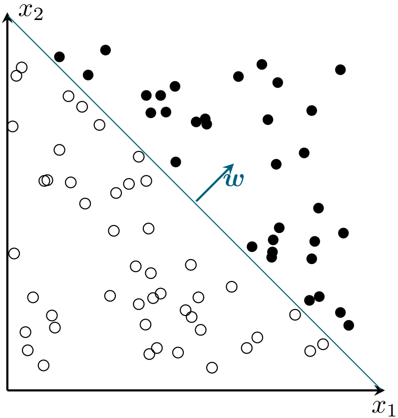
$$f(\boldsymbol{x}) = w_0 + \sum_{j=1}^{m} w_j x_j.$$

  where $\boldsymbol{w} = (w_0, w_1, \ldots, w_m)$ are the parameters of the model and $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ is a feature vector describing an example.
- It is often convenient to use vector notation:

$$f(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x}$$

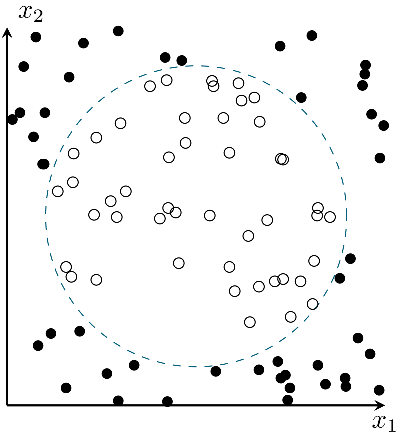
  where $\boldsymbol{x} = (1, x_1, x_2, \ldots, x_n)$ has an additional 1 in the first position.

Linear models



- Linear model fits perfectly.

Linear models



- What if the data is not even close to linear?

Linear models

- Linear models constitute a very general class of models:
  - Basic transformations and expansion of original features,
  - Kernel trick (SVM),
  - Linear combination of weak classifiers (AdaBoost),
  - Deep learning: hierarchical structure of generalized linear models.

Fitting linear models

- We fit parameters $\boldsymbol{w}$ of a linear model using training data

$$\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)\}$$

where $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{im})$ is a feature vector of the i-th training example.

Fitting linear models

- We fit parameters $\boldsymbol{w}$ of a linear model using training data

$$\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)\}$$

  where $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{im})$ is a feature vector of the i-th training example.
- We use loss function $\ell(y, f(\boldsymbol{x}))$ to guide the learning process.

Fitting linear models

- We fit parameters $\boldsymbol{w}$ of a linear model using training data

$$\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)\}$$

  where $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{im})$ is a feature vector of the i-th training example.
- We use loss function $\ell(y, f(\boldsymbol{x}))$ to guide the learning process.
- Since direct optimization of $\ell(y, f(\boldsymbol{x}))$ can be hard (e.g., 0/1 loss is neither convex nor differentiable), we use the so-called surrogate loss functions ℓ_{s}:

$$\widehat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \sum_{i=1}^{n} \ell_{\mathrm{s}}(y_i, \boldsymbol{w}\boldsymbol{x}_i)$$

Linear regression

- Let $f(\boldsymbol{x})$ be a linear function of the input variables:

$$f(\boldsymbol{x}) = w_0 + \sum_{j=1}^{m} w_j x_j \ = \ \boldsymbol{w} \cdot \boldsymbol{x}.$$

Linear regression

- Let $f(\boldsymbol{x})$ be a linear function of the input variables:

$$f(\boldsymbol{x}) = w_0 + \sum_{j=1}^{m} w_j x_j \; = \; \boldsymbol{w} \cdot \boldsymbol{x}.$$

- We minimize the squared error loss:

$$\ell_{\mathrm{sq}}(y, f(\boldsymbol{x})) = (y - f(\boldsymbol{x}))^2 \, .$$

Linear regression

- Let $f(\boldsymbol{x})$ be a linear function of the input variables:

$$f(\boldsymbol{x}) = w_0 + \sum_{j=1}^{m} w_j x_j \; = \; \boldsymbol{w} \cdot \boldsymbol{x}.$$

- We minimize the squared error loss:

$$\ell_{\mathrm{sq}}(y, f(\boldsymbol{x})) = (y - f(\boldsymbol{x}))^2 \,.$$

- Minimizing squared error loss is equivalent to estimating:

$$\mathbb{E}(y|\boldsymbol{x}) = w_0 + \sum_{j=1}^{n} w_j x_j \; = \; \boldsymbol{w} \cdot \boldsymbol{x},$$

the **conditional mean value**.

Linear regression

- The task of a learning algorithm is to estimate

$$\boldsymbol{w} = (w_0, w_1, \ldots, w_m)$$

by solving the following optimization problem:

$$\begin{aligned}
\widehat{\boldsymbol{w}} &= \arg\min_{\boldsymbol{w}} \sum_{i=1}^{n} \ell_{\mathrm{sq}}(y_i, w_0 + \sum_{j=1}^{m} w_j x_{ij}) \\
&= \arg\min_{\boldsymbol{w}} \sum_{i=1}^{n} (y_i - w_0 - \sum_{j=1}^{m} w_j x_{ij})^2 \\
&= \arg\min_{\boldsymbol{w}} \sum_{i=1}^{n} (y_i - \boldsymbol{w} \cdot \boldsymbol{x}_i)^2.
\end{aligned}$$

Linear regression

- The task of a learning algorithm is to estimate

$$\boldsymbol{w} = (w_0, w_1, \ldots, w_m)$$

  by solving the following optimization problem:

$$\begin{aligned} \widehat{\boldsymbol{w}} &= \arg\min_{\boldsymbol{w}} \sum_{i=1}^{n} \ell_{\mathrm{sq}}(y_i, w_0 + \sum_{j=1}^{m} w_j x_{ij}) \\ &= \arg\min_{\boldsymbol{w}} \sum_{i=1}^{n} (y_i - w_0 - \sum_{j=1}^{m} w_j x_{ij})^2 \\ &= \arg\min_{\boldsymbol{w}} \sum_{i=1}^{n} (y_i - \boldsymbol{w} \cdot \boldsymbol{x}_i)^2. \end{aligned}$$

- Let us solve this problem in a simple one-dimension case $(m = 1)$ ...

Linear regression

- Define:

$$\widehat{L}(w_0, w_1) = \sum_{i=1}^{n} (y_i - w_0 - w_1 x_i)^2.$$

- Define:

$$\widehat{L}(w_0, w_1) = \sum_{i=1}^{n} (y_i - w_0 - w_1 x_i)^2.$$

- We take derivative of $\widehat{L}$ with respect to w_0 and equate it to zero:

$$\frac{\partial \widehat{L}}{\partial w_0} = 0 \iff -2 \sum_{i=1}^{n} (y_i - w_0 - w_1 x_i) \quad = \quad 0$$

- Define:

$$\widehat{L}(w_0, w_1) = \sum_{i=1}^{n} (y_i - w_0 - w_1 x_i)^2.$$

- We take derivative of $\widehat{L}$ with respect to w_0 and equate it to zero:

$$\begin{aligned} \frac{\partial \widehat{L}}{\partial w_0} = 0 \iff -2\sum_{i=1}^{n}(y_i - w_0 - w_1 x_i) &= 0 \\ n w_0 &= \sum_{i=1}^{n} y_i - w_1 \sum_{i=1}^{n} x_i \end{aligned}$$

- Define:

$$\widehat{L}(w_0, w_1) = \sum_{i=1}^{n}(y_i - w_0 - w_1 x_i)^2.$$

- We take derivative of $\widehat{L}$ with respect to w_0 and equate it to zero:

$$\begin{aligned}
\frac{\partial \widehat{L}}{\partial w_0} = 0 \iff -2\sum_{i=1}^{n}(y_i - w_0 - w_1 x_i) &= 0 \\
n w_0 &= \sum_{i=1}^{n} y_i - w_1 \sum_{i=1}^{n} x_i \\
w_0 &= \bar{y} - w_1 \bar{x}\,,
\end{aligned}$$

where:

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \quad \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Linear regression

- In the next step we take derivative of $\widehat{L}$ with respect to w_1 and equate it to zero:

$$\frac{\partial \widehat{L}}{\partial w_1} = 0 \iff -2\sum_{i=1}^{n}(y_i - w_0 - w_1 x_i)x_i \quad = \quad 0$$

- In the next step we take derivative of $\widehat{L}$ with respect to w_1 and equate it to zero:

$$
\begin{aligned}
\frac{\partial \widehat{L}}{\partial w_1} = 0 \iff -2\sum_{i=1}^{n}(y_i - w_0 - w_1 x_i)x_i &= 0 \\
\sum_{i=1}^{n} y_i x_i - w_0 \sum_{i=1}^{n} x_i - w_1 \sum_{i=1}^{n} x_i^2 &= 0
\end{aligned}
$$

Linear regression

- In the next step we take derivative of $\widehat{L}$ with respect to w_1 and equate it to zero:

$$
\begin{aligned}
\frac{\partial \widehat{L}}{\partial w_1} = 0 \iff -2\sum_{i=1}^{n}(y_i - w_0 - w_1 x_i)x_i &= 0 \\
\sum_{i=1}^{n} y_i x_i - w_0 \sum_{i=1}^{n} x_i - w_1 \sum_{i=1}^{n} x_i^2 &= 0 \\
(w_0 = \bar{y} - w_1 \bar{x}) \qquad \sum_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x}) - w_1 \sum_{i=1}^{n}(x_i - \bar{x})^2 &= 0
\end{aligned}
$$

- In the next step we take derivative of $\widehat{L}$ with respect to w_1 and equate it to zero:

$$\begin{aligned}
\frac{\partial \widehat{L}}{\partial w_1} = 0 \iff -2\sum_{i=1}^{n}(y_i - w_0 - w_1 x_i)x_i &= 0 \\
\sum_{i=1}^{n} y_i x_i - w_0 \sum_{i=1}^{n} x_i - w_1 \sum_{i=1}^{n} x_i^2 &= 0 \\
(w_0 = \bar{y} - w_1\bar{x}) \qquad \sum_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x}) - w_1 \sum_{i=1}^{n}(x_i - \bar{x})^2 &= 0
\end{aligned}$$

so that we get:

$$w_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

- We solve the equation for w_1:

$$\begin{aligned}
\sum_{i=1}^{n} y_i x_i - (\bar{y} - w_1 \bar{x}) \sum_{i=1}^{n} x_i - w_1 \sum_{i=1}^{n} x_i^2 &= 0 \\
\sum_{i=1}^{n} y_i x_i - \bar{y} \sum_{i=1}^{n} x_i - w_1 \left(\sum_{i=1}^{n} x_i^2 - \bar{x} \sum_{i=1}^{n} x_i \right) &= 0
\end{aligned}$$

- We solve the equation for w_1:

$$\begin{aligned}\sum_{i=1}^{n} y_i x_i - (\bar{y} - w_1 \bar{x}) \sum_{i=1}^{n} x_i - w_1 \sum_{i=1}^{n} x_i^2 &= 0 \\ \sum_{i=1}^{n} y_i x_i - \bar{y} \sum_{i=1}^{n} x_i - w_1 \left(\sum_{i=1}^{n} x_i^2 - \bar{x} \sum_{i=1}^{n} x_i \right) &= 0\end{aligned}$$

- We have:

$$\sum_{i=1}^{n} x_i^2 - \bar{x} \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} x_i^2 - \bar{x} \sum_{i=1}^{n} x_i + \bar{x} \sum_{i=1}^{n} x_i - \bar{x} \sum_{i=1}^{n} x_i$$

- We solve the equation for w_1:

$$\begin{aligned}
\sum_{i=1}^{n} y_i x_i - (\bar{y} - w_1 \bar{x}) \sum_{i=1}^{n} x_i - w_1 \sum_{i=1}^{n} x_i^2 &= 0 \\
\sum_{i=1}^{n} y_i x_i - \bar{y} \sum_{i=1}^{n} x_i - w_1 \left(\sum_{i=1}^{n} x_i^2 - \bar{x} \sum_{i=1}^{n} x_i \right) &= 0
\end{aligned}$$

- We have:

$$\begin{aligned}
\sum_{i=1}^{n} x_i^2 - \bar{x} \sum_{i=1}^{n} x_i &= \sum_{i=1}^{n} x_i^2 - \bar{x} \sum_{i=1}^{n} x_i + \bar{x} \sum_{i=1}^{n} x_i - \bar{x} \sum_{i=1}^{n} x_i \\
&= \sum_{i=1}^{n} x_i^2 - 2\bar{x} \sum_{i=1}^{n} x_i + n\bar{x}^2 = \sum_{i=1}^{n} \left(x_i^2 - 2\bar{x} x_i + \bar{x}^2 \right)
\end{aligned}$$

Linear regression

- We solve the equation for w_1:

$$\begin{aligned} \sum_{i=1}^{n} y_i x_i - (\bar{y} - w_1 \bar{x}) \sum_{i=1}^{n} x_i - w_1 \sum_{i=1}^{n} x_i^2 &= 0 \\ \sum_{i=1}^{n} y_i x_i - \bar{y} \sum_{i=1}^{n} x_i - w_1 \left(\sum_{i=1}^{n} x_i^2 - \bar{x} \sum_{i=1}^{n} x_i \right) &= 0 \end{aligned}$$

- We have:

$$\begin{aligned} \sum_{i=1}^{n} x_i^2 - \bar{x} \sum_{i=1}^{n} x_i &= \sum_{i=1}^{n} x_i^2 - \bar{x} \sum_{i=1}^{n} x_i + \bar{x} \sum_{i=1}^{n} x_i - \bar{x} \sum_{i=1}^{n} x_i \\ &= \sum_{i=1}^{n} x_i^2 - 2\bar{x} \sum_{i=1}^{n} x_i + n\bar{x}^2 = \sum_{i=1}^{n} \left(x_i^2 - 2\bar{x} x_i + \bar{x}^2 \right) \\ &= \sum_{i=1}^{n} (x_i - \bar{x})^2 \end{aligned}$$

- Similarly, we have:

$$\sum_{i=1}^{n} y_i x_i - \bar{y} \sum_{i=1}^{n} x_i \;=\; \sum_{i=1}^{n} y_i x_i - \bar{y} \sum_{i=1}^{n} x_i + \bar{y} \sum_{i=1}^{n} x_i - \bar{y} \sum_{i=1}^{n} x_i$$

- Similarly, we have:

$$\begin{aligned}\sum_{i=1}^{n} y_i x_i - \bar{y}\sum_{i=1}^{n} x_i &= \sum_{i=1}^{n} y_i x_i - \bar{y}\sum_{i=1}^{n} x_i + \bar{y}\sum_{i=1}^{n} x_i - \bar{y}\sum_{i=1}^{n} x_i \\ &= \sum_{i=1}^{n} y_i x_i - \bar{y}\sum_{i=1}^{n} x_i - \bar{x}\sum_{i=1}^{n} y_i + n\bar{y}\bar{x}\end{aligned}$$

- Similarly, we have:

$$\begin{aligned}\sum_{i=1}^{n} y_i x_i - \bar{y}\sum_{i=1}^{n} x_i &= \sum_{i=1}^{n} y_i x_i - \bar{y}\sum_{i=1}^{n} x_i + \bar{y}\sum_{i=1}^{n} x_i - \bar{y}\sum_{i=1}^{n} x_i \\ &= \sum_{i=1}^{n} y_i x_i - \bar{y}\sum_{i=1}^{n} x_i - \bar{x}\sum_{i=1}^{n} y_i + n\bar{y}\bar{x} \\ &= \sum_{i=1}^{n} (y_i x_i - \bar{y}x_i - \bar{x}y_i + \bar{x}\bar{y})\end{aligned}$$

- Similarly, we have:

$$\begin{aligned}
\sum_{i=1}^{n} y_i x_i - \bar{y}\sum_{i=1}^{n} x_i &= \sum_{i=1}^{n} y_i x_i - \bar{y}\sum_{i=1}^{n} x_i + \bar{y}\sum_{i=1}^{n} x_i - \bar{y}\sum_{i=1}^{n} x_i \\
&= \sum_{i=1}^{n} y_i x_i - \bar{y}\sum_{i=1}^{n} x_i - \bar{x}\sum_{i=1}^{n} y_i + n\bar{y}\bar{x} \\
&= \sum_{i=1}^{n} \left(y_i x_i - \bar{y}x_i - \bar{x}y_i + \bar{x}\bar{y}\right) \\
&= \sum_{i=1}^{n} \left(x_i - \bar{x}\right)\left(y_i - \bar{y}\right)
\end{aligned}$$

- Finally, we get:

$$w_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

- The solution for the one-dimensional problem is:

$$\begin{aligned}\widehat{w}_1 &= \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2},\\ \widehat{w}_0 &= \bar{y}_1 - \widehat{w}_1\bar{x}.\end{aligned}$$

- The final model is given by:

$$f(\boldsymbol{x}) = \widehat{w}_0 + \widehat{w}_1 x$$

Linear regression – general case

- The criterion to be minimized:

$$\widehat{L}(\boldsymbol{w}) = \sum_{i=1}^{n} (y_i - \boldsymbol{w} \cdot \boldsymbol{x}_i)^2.$$

Linear regression – general case

- The criterion to be minimized:

$$\widehat{L}(\boldsymbol{w}) = \sum_{i=1}^{n} (y_i - \boldsymbol{w} \cdot \boldsymbol{x}_i)^2.$$

- Differentiating with respect to $\boldsymbol{w}$ and setting the gradient to 0:

$$\frac{\partial \widehat{L}}{\partial \boldsymbol{w}} = \boldsymbol{0} \iff 2 \sum_{i=1}^{n} (y_i - \boldsymbol{w} \cdot \boldsymbol{x}_i) \boldsymbol{x}_i \quad = \quad \boldsymbol{0}$$

Linear regression – general case

- The criterion to be minimized:

$$\widehat{L}(\boldsymbol{w}) = \sum_{i=1}^{n} (y_i - \boldsymbol{w} \cdot \boldsymbol{x}_i)^2.$$

- Differentiating with respect to $\boldsymbol{w}$ and setting the gradient to 0:

$$\begin{aligned} \frac{\partial \widehat{L}}{\partial \boldsymbol{w}} = \mathbf{0} \iff 2 \sum_{i=1}^{n} (y_i - \boldsymbol{w} \cdot \boldsymbol{x}_i) \boldsymbol{x}_i &= \mathbf{0} \\ \sum_{i=1}^{n} y_i \boldsymbol{x}_i - \left(\sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^\top \right) \boldsymbol{w} &= \mathbf{0} \end{aligned}$$

Linear regression – general case

- The criterion to be minimized:

$$\widehat{L}(\boldsymbol{w}) = \sum_{i=1}^{n} (y_i - \boldsymbol{w} \cdot \boldsymbol{x}_i)^2.$$

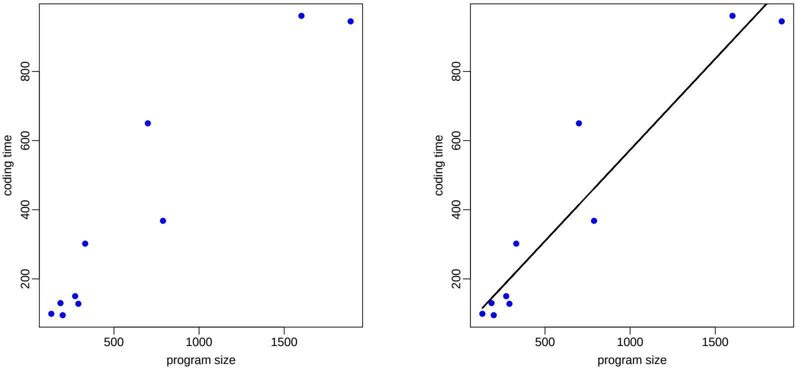
- Differentiating with respect to $\boldsymbol{w}$ and setting the gradient to 0:

$$\begin{aligned} \frac{\partial \widehat{L}}{\partial \boldsymbol{w}} = \mathbf{0} \iff 2 \sum_{i=1}^{n} (y_i - \boldsymbol{w} \cdot \boldsymbol{x}_i) \boldsymbol{x}_i &= \mathbf{0} \\ \sum_{i=1}^{n} y_i \boldsymbol{x}_i - \left(\sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^\top \right) \boldsymbol{w} &= \mathbf{0} \end{aligned}$$

- Assuming $\sum_i \boldsymbol{x}_i \boldsymbol{x}_i^\top$ is nonsingular, the solution is:

$$\widehat{\boldsymbol{w}} = \left(\sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^\top \right)^{-1} \left(\sum_{i=1}^{n} y_i \boldsymbol{x}_i \right).$$

Linear regression – Example



Linear regression

- Very efficient method for a small or moderate number of features.

Linear regression

- Very efficient method for a small or moderate number of features.
- For large number of features different learning algorithms should be used.

Linear regression

- Very efficient method for a small or moderate number of features.
- For large number of features different learning algorithms should be used.
- Statistical properties of linear regression are very well-studied – a very mature statistical procedure.

Linear regression

- Very efficient method for a small or moderate number of features.
- For large number of features different learning algorithms should be used.
- Statistical properties of linear regression are very well-studied – a very mature statistical procedure.
- Can also be used for binary classification – quite popular in large scale problems.

Outline

Summary

- Learning algorithms
  - Lazy learning,
  - Decision trees,
  - Generative models,
  - Linear models.

Bibliography

- T. Hastie, R. Tibshirani, and J. Friedman. *Elements of Statistical Learning: Second Edition*.
  Springer, 2009
  `http://www-stat.stanford.edu/~tibs/ElemStatLearn/`
- Christopher M. Bishop. *Pattern Recognition and Machine Learning*.
  Springer-Verlag, 2006
- David Barber. *Bayesian Reasoning and Machine Learning*.
  Cambridge University Press, 2012
  `http://www.cs.ucl.ac.uk/staff/d.barber/brml/`
- Yaser S. Abu-Mostafa, Malik Magdon-Ismail, and Hsuan-Tien Lin. *Learning From Data*.
  AMLBook, 2012