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

1. Learning Algorithms
2. Lazy Learning
3. Decision trees
4. Generative Models
5. Linear Models
6. Summary
Outline

1 Learning Algorithms

2 Lazy Learning

3 Decision trees

4 Generative Models

5 Linear Models

6 Summary
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

1. Learning Algorithms
2. Lazy Learning
3. Decision trees
4. Generative Models
5. Linear Models
6. Summary
Lazy learning

- Based on empirical distribution and direct application of the Bayes rule to local estimates of $P(y | x)$.

  ▶ Based on group-bys and simple counting.

  ▶ Needs a lot of data to get reasonable estimates!!!
Lazy learning

- Based on empirical distribution and direct application of the Bayes rule to local estimates of $P(y \mid x)$.
- The simplest approach estimates conditional probabilities $P(y \mid x)$ for any $x$ 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? | \( P(y = 1|\text{gold} = 1 \land \text{price} = 1) \) | \( P(y = 0|\text{gold} = 1 \land \text{price} = 1) \) |
|------------|-------|--------------------------------------------------|--------------------------------------------------|
| 1          | 1     | 0.75                                             | 0.25                                             |
| 1          | 1     | \( P(y = 1|\text{gold} = 0 \land \text{price} = 0) \) | \( P(y = 0|\text{gold} = 0 \land \text{price} = 0) \) | 0.33 | 0.66 |
| 0          | 0     | \( P(y = 1|\text{gold} = 0 \land \text{price} = 1) \) | \( P(y = 0|\text{gold} = 0 \land \text{price} = 1) \) | 0.5 | 0.5 |
| 0          | 0     | \( P(y = 1|\text{gold} = 1 \land \text{price} = 0) \) | \( P(y = 0|\text{gold} = 1 \land \text{price} = 0) \) | ? | ? |
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.
• 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 tuning $k$ 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 tuning $k$ 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 tuning $k$ 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 tuning $k$ 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.
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

\[ X_2 \]

\[ R_1 \quad R_3 \]

\[ t_1 \]

\[ X_1 \]

\[ X_1 \leq t_1 \]

\[ R_1 \quad R_3 \]
Decision trees
Decision trees

X_2

R_2

R_1

t_2

t_1

X_1

R_3

R_4

t_3

X_1 \leq t_1

X_2 \leq t_2

X_1 \leq t_3

R_1

R_2

R_3

R_4
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.
• 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

1 Learning Algorithms
2 Lazy Learning
3 Decision trees
4 Generative Models
5 Linear Models
6 Summary
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:

\[ P(y = k | \mathbf{x}) = \frac{P(y = k, \mathbf{x})}{P(\mathbf{x})} = \frac{P(\mathbf{x} | y = k)P(y = k)}{P(\mathbf{x})} \]

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

• For classification, we have:

\[ P(y = k | x) = \frac{P(y = k, x)}{P(x)} = \frac{P(x | y = k)P(y = k)}{P(x)} \]

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

\[ P(x) = \sum_{k'} P(x | y = k')P(y = k') \]

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 $\mathbf{x} = (x_1, x_2, \ldots, x_m)$ are independent:

$$P(\mathbf{x} \mid y) = \prod_{j=1}^{m} P(x_j \mid y)$$

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

$$N(E(x_j \mid y = k), \text{Var}(x_j \mid y = k))$$

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

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

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

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

\[
P(\mathbf{x}|y) = \prod_{j=1}^{m} P(x_j|y).
\]

• The model takes the following form:

\[
P(y = k|\mathbf{x}) =
\]
Naive Bayes

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

\[
P(\mathbf{x} | y) = \prod_{j=1}^{m} P(x_j | y).
\]

- The model takes the following form:

\[
P(y = k | \mathbf{x}) = \frac{P(y = k) \prod_{j=1}^{m} P(x_j | y = k)}{\sum_{k'} P(y = k') \prod_{j=1}^{m} P(x_j | y = k')}
\]
Naive Bayes

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

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

• The model takes the following form:

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

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

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

• If a component $x_j$ of $\mathbf{x}$ is discrete, then an appropriate histogram estimate can be used.
Naive Bayes

Example

<table>
<thead>
<tr>
<th>gold</th>
<th>price</th>
<th>spam?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

\[ P(y = 1) = \]

\[ P(y = 0) = 0.5 \]

\[ P(\text{gold} = 1 \mid Y = 1) = 0.5 \]

\[ P(\text{gold} = 0 \mid Y = 1) = 0.5 \]

\[ P(\text{gold} = 1 \mid Y = 0) = 0.17 \]

\[ P(\text{gold} = 0 \mid Y = 0) = 0.83 \]

\[ P(\text{price} = 1 \mid Y = 1) = 0.66 \]

\[ P(\text{price} = 0 \mid Y = 1) = 0.33 \]

\[ P(\text{price} = 1 \mid Y = 0) = 0.33 \]

\[ P(\text{price} = 0 \mid Y = 0) = 0.66 \]
Naive Bayes

Example

<table>
<thead>
<tr>
<th>gold price</th>
<th>spam?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1</td>
<td>1</td>
</tr>
<tr>
<td>1 1</td>
<td>1</td>
</tr>
<tr>
<td>1 1</td>
<td>1</td>
</tr>
<tr>
<td>1 1</td>
<td>0</td>
</tr>
<tr>
<td>0 0</td>
<td>0</td>
</tr>
<tr>
<td>0 0</td>
<td>0</td>
</tr>
<tr>
<td>0 0</td>
<td>0</td>
</tr>
<tr>
<td>0 0</td>
<td>0</td>
</tr>
<tr>
<td>0 0</td>
<td>0</td>
</tr>
<tr>
<td>0 0</td>
<td>1</td>
</tr>
<tr>
<td>0 0</td>
<td>1</td>
</tr>
<tr>
<td>0 1</td>
<td>0</td>
</tr>
<tr>
<td>0 1</td>
<td>1</td>
</tr>
</tbody>
</table>

\[ P(y = 1) = 0.5 \]
\[ P(y = 0) = \]

\[ P(y = 1) = 0.5 \times 0.33 \times 0.5 = 0.0825 \]
\[ P(y = 0) = 1 - 0.0825 = 0.9175 \]
**Naive Bayes**

**Example**

<table>
<thead>
<tr>
<th>gold</th>
<th>price</th>
<th>spam?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

\[
P(y = 1) = 0.5 \]

\[
P(y = 0) = 0.5 \]

\[
P(\text{gold} = 1|Y = 1) = \]

\[
P(\text{gold} = 0|Y = 1) = \]

\[
P(\text{price} = 1|Y = 1) = \]

\[
P(\text{price} = 0|Y = 1) = \]

\[
P(\text{price} = 1|Y = 0) = \]

\[
P(\text{price} = 0|Y = 0) = \]

We can, for example, compute:

\[
P(y = 1|\text{gold} = 1 \land \text{price} = 0) = 0.5 \times 0.33 \times 0.5 = 0.0825 \]

\[
P(y = 0|\text{gold} = 1 \land \text{price} = 0) = 1 - 0.0825 = 0.405 \]
## Naive Bayes

### Example

<table>
<thead>
<tr>
<th>gold</th>
<th>price</th>
<th>spam?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

$P(y = 1) = 0.5$

$P(y = 0) = 0.5$

$P(gold = 1|Y = 1) = 0.5$

$P(gold = 0|Y = 1) =$

$P(price = 1|Y = 1) = 0.66$

$P(price = 0|Y = 1) = 0.33$
## Naive Bayes

### Example

<table>
<thead>
<tr>
<th>gold price</th>
<th>spam?</th>
<th>( P(y = 1) = 0.5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1</td>
<td>1</td>
<td>( P(y = 0) = 0.5 )</td>
</tr>
<tr>
<td>1 1</td>
<td>1</td>
<td>( P(\text{gold} = 1</td>
</tr>
<tr>
<td>1 1</td>
<td>1</td>
<td>( P(\text{gold} = 0</td>
</tr>
<tr>
<td>1 1</td>
<td>0</td>
<td>( P(\text{gold} = 1</td>
</tr>
<tr>
<td>0 0</td>
<td>0</td>
<td>( P(\text{gold} = 0</td>
</tr>
<tr>
<td>0 0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0 0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0 0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0 0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0 0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0 1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0 1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
Naive Bayes

Example

<table>
<thead>
<tr>
<th>gold</th>
<th>price</th>
<th>spam?</th>
<th>P(y = 1) = 0.5</th>
<th>P(y = 0) = 0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>P(gold = 1</td>
<td>Y = 1) = 0.5</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>P(gold = 1</td>
<td>Y = 0) = 0.17</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Example

<table>
<thead>
<tr>
<th>gold</th>
<th>price</th>
<th>spam?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

\[
P(y = 1) = 0.5 \]

\[
P(y = 0) = 0.5 \]

\[
P(\text{gold} = 1|Y = 1) = 0.5 \]

\[
P(\text{gold} = 0|Y = 1) = 0.5 \]

\[
P(\text{gold} = 1|Y = 0) = 0.17 \]

\[
P(\text{gold} = 0|Y = 0) = 0.83 \]

\[
P(\text{price} = 1|Y = 1) = \]

\[
P(\text{price} = 0|Y = 1) = \]

\[
P(\text{price} = 1|Y = 0) = \]

\[
P(\text{price} = 0|Y = 0) = \]
Naive Bayes

Example

<table>
<thead>
<tr>
<th>gold</th>
<th>price</th>
<th>spam?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

\[
P(y = 1) = 0.5 \\
P(y = 0) = 0.5 \\
P(\text{gold} = 1|Y = 1) = 0.5 \\
P(\text{gold} = 0|Y = 1) = 0.5 \\
P(\text{gold} = 1|Y = 0) = 0.17 \\
P(\text{gold} = 0|Y = 0) = 0.83 \\
P(\text{price} = 1|Y = 1) = 0.66 \\
P(\text{price} = 0|Y = 1) =
\]
Naive Bayes

Example

<table>
<thead>
<tr>
<th>gold price</th>
<th>spam?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1</td>
<td>1</td>
</tr>
<tr>
<td>1 1</td>
<td>1</td>
</tr>
<tr>
<td>1 1</td>
<td>1</td>
</tr>
<tr>
<td>1 1</td>
<td>0</td>
</tr>
<tr>
<td>0 0</td>
<td>0</td>
</tr>
<tr>
<td>0 0</td>
<td>0</td>
</tr>
<tr>
<td>0 0</td>
<td>0</td>
</tr>
<tr>
<td>0 0</td>
<td>0</td>
</tr>
<tr>
<td>0 0</td>
<td>1</td>
</tr>
<tr>
<td>0 0</td>
<td>1</td>
</tr>
<tr>
<td>0 1</td>
<td>0</td>
</tr>
<tr>
<td>0 1</td>
<td>1</td>
</tr>
</tbody>
</table>

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

Example

<table>
<thead>
<tr>
<th>gold</th>
<th>price</th>
<th>spam?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

\[P(y = 1) = 0.5\]
\[P(y = 0) = 0.5\]
\[P(\text{gold} = 1 | Y = 1) = 0.5\]
\[P(\text{gold} = 0 | Y = 1) = 0.5\]
\[P(\text{gold} = 1 | Y = 0) = 0.17\]
\[P(\text{gold} = 0 | Y = 0) = 0.83\]
\[P(\text{price} = 1 | Y = 1) = 0.66\]
\[P(\text{price} = 0 | Y = 1) = 0.33\]
\[P(\text{price} = 1 | Y = 0) = 0.33\]
\[P(\text{price} = 0 | Y = 0) = \]
**Naive Bayes**

### Example

<table>
<thead>
<tr>
<th>gold</th>
<th>price</th>
<th>spam?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

We can, for example, compute:

\[
P(y = 1 | \text{gold} = 1 \land \text{price} = 0) = \]

\[
P(y = 1) = 0.5
\]

\[
P(y = 0) = 0.5
\]

\[
P(\text{gold} = 1 | Y = 1) = 0.5
\]

\[
P(\text{gold} = 0 | Y = 1) = 0.5
\]

\[
P(\text{gold} = 1 | Y = 0) = 0.17
\]

\[
P(\text{gold} = 0 | Y = 0) = 0.83
\]

\[
P(\text{price} = 1 | Y = 1) = 0.66
\]

\[
P(\text{price} = 0 | Y = 1) = 0.33
\]

\[
P(\text{price} = 1 | Y = 0) = 0.33
\]

\[
P(\text{price} = 0 | Y = 0) = 0.66
\]
Naive Bayes

Example

<table>
<thead>
<tr>
<th>gold</th>
<th>price</th>
<th>spam?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

We can, for example, compute:

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

\[
P(y = 0 | \text{gold} = 1 \land \text{price} = 0) =
\]
### Naive Bayes

#### Example

<table>
<thead>
<tr>
<th>gold</th>
<th>price</th>
<th>spam?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

We can, for example, compute:

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

\[
P(y = 0|\text{gold} = 1 \land \text{price} = 0) = 1 - 0.595 = 0.405
\]
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.
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.
Linear models

Consider a linear model of the form:

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

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

It is often convenient to use vector notation:

\[ f(x) = w \cdot x \]

where \( x = (1, x_1, x_2, \ldots, x_n) \) has an additional 1 in the first position.
• Consider a linear model of the form:

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

where \( w = (w_0, w_1, \ldots, w_m) \) are the parameters of the model and \( x = (x_1, x_2, \ldots, x_n) \) is a feature vector describing an example.
• Consider a linear model of the form:

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

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

• It is often convenient to use vector notation:

\[ f(x) = \mathbf{w} \cdot \mathbf{x} \]

where \( \mathbf{x} = (1, x_1, x_2, \ldots, x_n) \) has an additional 1 in the first position.
• Linear model fits perfectly.
• 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 $\mathbf{w}$ of a linear model using training data

$$\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$$

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

• We fit parameters $\mathbf{w}$ of a linear model using training data

$$\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$$

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

• We use loss function $\ell(y, f(x))$ to guide the learning process.
Fitting linear models

• We fit parameters $\mathbf{w}$ of a linear model using training data

$$\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$$

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

• We use loss function $\ell(y, f(\mathbf{x}))$ to guide the learning process.

• Since direct optimization of $\ell(y, f(\mathbf{x}))$ can be hard (e.g., 0/1 loss is neither convex nor differentiable), we use the so-called surrogate loss functions $\ell_s$:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \sum_{i=1}^{n} \ell_s(y_i, \mathbf{w} \mathbf{x}_i)$$
Linear regression

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

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

• Let $f(x)$ be a linear function of the input variables:

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

• We minimize the squared error loss:

$$\ell_{\text{sq}}(y, f(x)) = (y - f(x))^2.$$
Linear regression

• Let \( f(\mathbf{x}) \) be a linear function of the input variables:

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

• We minimize the squared error loss:

\[
    \ell_{sq}(y, f(\mathbf{x})) = (y - f(\mathbf{x}))^2.
\]

• Minimizing squared error loss is equivalent to estimating:

\[
    \mathbb{E}(y|\mathbf{x}) = w_0 + \sum_{j=1}^{n} w_j x_j = \mathbf{w} \cdot \mathbf{x},
\]

the **conditional mean value.**
Linear regression

- The task of a learning algorithm is to estimate

\[ \mathbf{w} = (w_0, w_1, \ldots, w_m) \]

by solving the following optimization problem:

\[
\hat{\mathbf{w}} = \arg \min_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} \ell_{sq}(y_i, w_0 + \sum_{j=1}^{m} w_j x_{ij})
\]

\[
= \arg \min_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} (y_i - w_0 - \sum_{j=1}^{m} w_j x_{ij})^2
\]

\[
= \arg \min_{\mathbf{w}} \sum_{i=1}^{n} (y_i - \mathbf{w} \cdot \mathbf{x}_i)^2.
\]
Linear regression

• The task of a learning algorithm is to estimate

\[ \mathbf{w} = (w_0, w_1, \ldots, w_m) \]

by solving the following optimization problem:

\[
\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \sum_{i=1}^{n} \ell_{\text{sq}}(y_i, w_0 + \sum_{j=1}^{m} w_j x_{ij}) \\
= \arg\min_{\mathbf{w}} \sum_{i=1}^{n} (y_i - \mathbf{w} \cdot \mathbf{x}_i)^2.
\]

• Let us solve this problem in a simple one-dimension case \((m = 1)\) \ldots
Linear regression

• Define:

\[ \tilde{L}(w_0, w_1) = \sum_{i=1}^{n} (y_i - w_0 - w_1 x_i)^2. \]
Linear regression

• Define:

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

• We take derivative of \( \hat{L} \) with respect to \( w_0 \) and equate it to zero:

\[ \frac{\partial \hat{L}}{\partial w_0} = 0 \iff -2 \sum_{i=1}^{n} (y_i - w_0 - w_1 x_i) = 0 \]

\[ n w_0 = \frac{1}{n} \sum_{i=1}^{n} y_i - w_1 \frac{1}{n} \sum_{i=1}^{n} x_i \]

\[ w_0 = \bar{y} - w_1 \bar{x}, \]
Linear regression

• Define:

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

• We take derivative of \( \hat{L} \) with respect to \( w_0 \) and equate it to zero:

\[
\frac{\partial \hat{L}}{\partial w_0} = 0 \iff -2 \sum_{i=1}^{n} (y_i - w_0 - w_1 x_i) = 0
\]

\[
nw_0 = \sum_{i=1}^{n} y_i - w_1 \sum_{i=1}^{n} x_i
\]
Linear regression

• Define:

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

• We take derivative of \( \hat{L} \) with respect to \( w_0 \) and equate it to zero:

\[ \frac{\partial \hat{L}}{\partial w_0} = 0 \iff -2 \sum_{i=1}^{n} (y_i - w_0 - w_1 x_i) = 0 \]

\[ nw_0 = \sum_{i=1}^{n} y_i - w_1 \sum_{i=1}^{n} x_i \]

\[ w_0 = \bar{y} - w_1 \bar{x}, \]

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 $\hat{L}$ with respect to $w_1$ and equate it to zero:

$$\frac{\partial \hat{L}}{\partial w_1} = 0 \iff -2 \sum_{i=1}^{n} (y_i - w_0 - w_1 x_i)x_i = 0$$
• In the next step we take derivative of $\hat{L}$ with respect to $w_1$ and equate it to zero:

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

$$\sum_{i=1}^{n} y_i x_i - w_0 \sum_{i=1}^{n} x_i - w_1 \sum_{i=1}^{n} x_i^2 = 0$$
• In the next step we take derivative of $\hat{L}$ with respect to $w_1$ and equate it to zero:

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

$$\sum_{i=1}^{n} y_i x_i - w_0 \sum_{i=1}^{n} x_i - w_1 \sum_{i=1}^{n} x_i^2 = 0$$

$$w_0 = \bar{y} - w_1 \bar{x}$$

$$(w_0 = \bar{y} - w_1 \bar{x}) \sum_{i=1}^{n} (y_i - \bar{y})(x_i - \bar{x}) - w_1 \sum_{i=1}^{n} (x_i - \bar{x})^2 = 0$$
Linear regression

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

\[
\frac{\partial \hat{L}}{\partial w_1} = 0 \iff -2 \sum_{i=1}^{n} (y_i - w_0 - w_1 x_i) x_i = 0
\]

\[
\sum_{i=1}^{n} y_i x_i - w_0 \sum_{i=1}^{n} x_i - w_1 \sum_{i=1}^{n} x_i^2 = 0
\]

\[
(w_0 = \bar{y} - w_1 \bar{x}) \quad \sum_{i=1}^{n} (y_i - \bar{y})(x_i - \bar{x}) - w_1 \sum_{i=1}^{n} (x_i - \bar{x})^2 = 0
\]

so that we get:

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

- We solve the equation for $w_1$:

\[
\sum_{i=1}^{n} y_i x_i - (\bar{y} - w_1 \bar{x}) \sum_{i=1}^{n} x_i - w_1 \sum_{i=1}^{n} x_i^2 = 0
\]

\[
\sum_{i=1}^{n} y_i x_i - \bar{y} \sum_{i=1}^{n} x_i - w_1 \left( \sum_{i=1}^{n} x_i^2 - \bar{x} \sum_{i=1}^{n} x_i \right) = 0
\]
Linear regression

- We solve the equation for $w_1$:

$$
\sum_{i=1}^{n} y_i x_i - (\bar{y} - w_1 \bar{x}) \sum_{i=1}^{n} x_i - w_1 \sum_{i=1}^{n} x_i^2 = 0
$$

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

- We have:

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

• We solve the equation for $w_1$:

$$
\sum_{i=1}^{n} y_i x_i - (\bar{y} - w_1 \bar{x}) \sum_{i=1}^{n} x_i - w_1 \sum_{i=1}^{n} x_i^2 = 0
$$

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

• We have:

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

$$
= \sum_{i=1}^{n} x_i^2 - 2\bar{x} \sum_{i=1}^{n} x_i + n \bar{x}^2 = \sum_{i=1}^{n} \left( x_i^2 - 2\bar{x} x_i + \bar{x}^2 \right)
$$
Linear regression

• We solve the equation for $w_1$:

$$
\sum_{i=1}^{n} y_i x_i - (\bar{y} - w_1 \bar{x}) \sum_{i=1}^{n} x_i - w_1 \sum_{i=1}^{n} x_i^2 = 0
$$

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

• We have:

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

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

$$
= \sum_{i=1}^{n} (x_i - \bar{x})^2
$$
Linear regression

• Similarly, we have:

\[ \sum_{i=1}^{n} y_i x_i - \bar{y} \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i x_i - \bar{y} \sum_{i=1}^{n} x_i + \bar{y} \sum_{i=1}^{n} x_i - \bar{y} \sum_{i=1}^{n} x_i \]
Linear regression

• Similarly, we have:

\[
\sum_{i=1}^{n} y_i x_i - \bar{y} \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i x_i - \bar{y} \sum_{i=1}^{n} x_i + \bar{y} \sum_{i=1}^{n} x_i - \bar{y} \sum_{i=1}^{n} x_i
\]

\[
= \sum_{i=1}^{n} y_i x_i - \bar{y} \sum_{i=1}^{n} x_i - \bar{x} \sum_{i=1}^{n} y_i + n \bar{y} \bar{x}
\]
Linear regression

• Similarly, we have:

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

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

$$= \sum_{i=1}^{n} (y_i x_i - \bar{y} x_i - \bar{x} y_i + \bar{x} \bar{y})$$
Linear regression

- Similarly, we have:

\[
\sum_{i=1}^{n} y_i x_i - \bar{y} \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i x_i - \bar{y} \sum_{i=1}^{n} x_i + \bar{y} \sum_{i=1}^{n} x_i - \bar{y} \sum_{i=1}^{n} x_i
\]

\[
= \sum_{i=1}^{n} y_i x_i - \bar{y} \sum_{i=1}^{n} x_i - \bar{x} \sum_{i=1}^{n} y_i + n \bar{y} \bar{x}
\]

\[
= \sum_{i=1}^{n} (y_i x_i - \bar{y} x_i - \bar{x} y_i + \bar{x} \bar{y})
\]

\[
= \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})
\]

- Finally, we get:

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

• The solution for the one-dimensional problem is:

\[
\hat{w}_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2},
\]

\[
\hat{w}_0 = \bar{y} - \hat{w}_1 \bar{x}.
\]

• The final model is given by:

\[
f(x) = \hat{w}_0 + \hat{w}_1 x
\]
Linear regression – general case

- The criterion to be minimized:

$$\hat{L}(w) = \sum_{i=1}^{n} (y_i - w \cdot x_i)^2.$$
Linear regression – general case

• The criterion to be minimized:

\[ \hat{L}(\mathbf{w}) = \sum_{i=1}^{n} (y_i - \mathbf{w} \cdot \mathbf{x}_i)^2. \]

• Differentiating with respect to \( \mathbf{w} \) and setting the gradient to 0:

\[
\frac{\partial \hat{L}}{\partial \mathbf{w}} = 0 \iff 2 \sum_{i=1}^{n} (y_i - \mathbf{w} \cdot \mathbf{x}_i) \mathbf{x}_i = 0
\]
Linear regression – general case

• The criterion to be minimized:

\[
\hat{L}(\mathbf{w}) = \sum_{i=1}^{n} (y_i - \mathbf{w} \cdot \mathbf{x}_i)^2.
\]

• Differentiating with respect to \( \mathbf{w} \) and setting the gradient to 0:

\[
\frac{\partial \hat{L}}{\partial \mathbf{w}} = 0 \quad \iff \quad 2 \sum_{i=1}^{n} (y_i - \mathbf{w} \cdot \mathbf{x}_i) \mathbf{x}_i = 0
\]

\[
\sum_{i=1}^{n} y_i \mathbf{x}_i - \left( \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^\top \right) \mathbf{w} = 0
\]
Linear regression – general case

- The criterion to be minimized:
  \[ \hat{L}(\mathbf{w}) = \sum_{i=1}^{n} (y_i - \mathbf{w} \cdot \mathbf{x}_i)^2. \]

- Differentiating with respect to \( \mathbf{w} \) and setting the gradient to 0:
  \[ \frac{\partial \hat{L}}{\partial \mathbf{w}} = 0 \iff 2 \sum_{i=1}^{n} (y_i - \mathbf{w} \cdot \mathbf{x}_i) \mathbf{x}_i = 0 \]
  \[ \sum_{i=1}^{n} y_i \mathbf{x}_i - \left( \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^\top \right) \mathbf{w} = 0 \]

- Assuming \( \sum_{i} \mathbf{x}_i \mathbf{x}_i^\top \) is nonsingular, the solution is:
  \[ \hat{\mathbf{w}} = \left( \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^\top \right)^{-1} \left( \sum_{i=1}^{n} y_i \mathbf{x}_i \right). \]
Linear regression – Example

- Two scatter plots showing the relationship between program size and coding time.
- The left plot is a scatter plot with points scattered across the graph, indicating no clear linear relationship.
- The right plot is a scatter plot with a fitted line, indicating a linear relationship between program size and coding time.
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.
• Learning algorithms
  ▶ Lazy learning,
  ▶ Decision trees,
  ▶ Generative models,
  ▶ Linear models.


  http://www.cs.ucl.ac.uk/staff/d.barber/brml/