Machine Learning
Machine Learning in a Nutshell

Data \[\rightarrow\] Machine Learner \[\rightarrow\] Model \[\rightarrow\] Performance Measure
Machine Learning in a Nutshell

Data with attributes

<table>
<thead>
<tr>
<th>ID</th>
<th>A1</th>
<th>Reflex</th>
<th>RefLow</th>
<th>RefHigh</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.6</td>
<td>Normal</td>
<td>3.4</td>
<td>7</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>5.5</td>
<td>Normal</td>
<td>2.4</td>
<td>5.7</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>5.3</td>
<td>Normal</td>
<td>2.4</td>
<td>5.7</td>
<td>Yes</td>
</tr>
<tr>
<td>4</td>
<td>5.3</td>
<td>Elevated</td>
<td>2.4</td>
<td>5.7</td>
<td>No</td>
</tr>
<tr>
<td>5</td>
<td>6.3</td>
<td>Normal</td>
<td>3.4</td>
<td>7</td>
<td>No</td>
</tr>
<tr>
<td>6</td>
<td>3.3</td>
<td>Normal</td>
<td>2.4</td>
<td>5.7</td>
<td>Yes</td>
</tr>
<tr>
<td>7</td>
<td>5.1</td>
<td>Decreased</td>
<td>2.4</td>
<td>5.7</td>
<td>Yes</td>
</tr>
<tr>
<td>8</td>
<td>4.2</td>
<td>Normal</td>
<td>2.4</td>
<td>5.7</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Instance $x_i \in \mathcal{X}$

with label $y_i \in \mathcal{Y}$
Machine Learning in a Nutshell

Data

Machine Learner

Model

Performance Measure

Data with attributes

<table>
<thead>
<tr>
<th>ID</th>
<th>A1</th>
<th>Reflex</th>
<th>RefLow</th>
<th>RefHigh</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.6</td>
<td>Normal</td>
<td>3.4</td>
<td>7</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>5.5</td>
<td>Normal</td>
<td>2.4</td>
<td>5.7</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>5.3</td>
<td>Normal</td>
<td>2.4</td>
<td>5.7</td>
<td>Yes</td>
</tr>
<tr>
<td>4</td>
<td>5.3</td>
<td>Elevated</td>
<td>2.4</td>
<td>5.7</td>
<td>No</td>
</tr>
<tr>
<td>5</td>
<td>6.3</td>
<td>Normal</td>
<td>3.4</td>
<td>7</td>
<td>No</td>
</tr>
<tr>
<td>6</td>
<td>3.3</td>
<td>Normal</td>
<td>2.4</td>
<td>5.7</td>
<td>Yes</td>
</tr>
<tr>
<td>7</td>
<td>5.1</td>
<td>Decreased</td>
<td>2.4</td>
<td>5.7</td>
<td>Yes</td>
</tr>
<tr>
<td>8</td>
<td>4.2</td>
<td>Normal</td>
<td>2.4</td>
<td>5.7</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Instance \( x_i \in \mathcal{X} \) with label \( y_i \in \mathcal{Y} \)

Model \( f : \mathcal{X} \mapsto \mathcal{Y} \)

Logistic regression
Support vector machines

Mixture Models
Hierarchical Bayesian Networks
Machine Learning in a Nutshell

Data with attributes

<table>
<thead>
<tr>
<th>ID</th>
<th>A1</th>
<th>Reflex</th>
<th>RefLow</th>
<th>RefHigh</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.6</td>
<td>Normal</td>
<td>3.4</td>
<td>7</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>5.5</td>
<td>Normal</td>
<td>2.4</td>
<td>5.7</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>5.3</td>
<td>Normal</td>
<td>2.4</td>
<td>5.7</td>
<td>Yes</td>
</tr>
<tr>
<td>4</td>
<td>5.3</td>
<td>Elevated</td>
<td>2.4</td>
<td>5.7</td>
<td>No</td>
</tr>
<tr>
<td>5</td>
<td>6.3</td>
<td>Normal</td>
<td>3.4</td>
<td>7</td>
<td>No</td>
</tr>
<tr>
<td>6</td>
<td>3.3</td>
<td>Normal</td>
<td>2.4</td>
<td>5.7</td>
<td>Yes</td>
</tr>
<tr>
<td>7</td>
<td>5.1</td>
<td>Decreased</td>
<td>2.4</td>
<td>5.7</td>
<td>Yes</td>
</tr>
<tr>
<td>8</td>
<td>4.2</td>
<td>Normal</td>
<td>2.4</td>
<td>5.7</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Instance \( x_i \in \mathcal{X} \) with label \( y_i \in \mathcal{Y} \)

Model \( f : \mathcal{X} \rightarrow \mathcal{Y} \)

Logistic regression
Support vector machines
Hierarchical Bayesian Networks
Mixture Models

Evaluation
Measure predicted labels vs actual labels on test data

Learning Curve

Performance

# Training Examples
A training set

<table>
<thead>
<tr>
<th>Example</th>
<th>Attr</th>
<th>Bear</th>
<th>Fii</th>
<th>Hin</th>
<th>Pat</th>
<th>Price</th>
<th>Rain</th>
<th>Res</th>
<th>Type</th>
<th>Est</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>X₁</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Some</td>
<td>$$$</td>
<td>No</td>
<td>Yes</td>
<td>French</td>
<td>0–10</td>
<td>Yes</td>
</tr>
<tr>
<td>X₂</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Full</td>
<td>$$</td>
<td>No</td>
<td>No</td>
<td>Thai</td>
<td>30–60</td>
<td>No</td>
</tr>
<tr>
<td>X₃</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Some</td>
<td>$$</td>
<td>No</td>
<td>No</td>
<td>Burger</td>
<td>0–10</td>
<td>Yes</td>
</tr>
<tr>
<td>X₄</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Full</td>
<td>$$</td>
<td>No</td>
<td>No</td>
<td>Thai</td>
<td>10–30</td>
<td>Yes</td>
</tr>
<tr>
<td>X₅</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Full</td>
<td>$$$</td>
<td>No</td>
<td>Yes</td>
<td>French</td>
<td>&gt;60</td>
<td>No</td>
</tr>
<tr>
<td>X₆</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Some</td>
<td>$$</td>
<td>Yes</td>
<td>Yes</td>
<td>Italian</td>
<td>0–10</td>
<td>Yes</td>
</tr>
<tr>
<td>X₇</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>None</td>
<td>$$</td>
<td>Yes</td>
<td>No</td>
<td>Burger</td>
<td>0–10</td>
<td>No</td>
</tr>
<tr>
<td>X₈</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Some</td>
<td>$$</td>
<td>Yes</td>
<td>Yes</td>
<td>Thai</td>
<td>0–10</td>
<td>Yes</td>
</tr>
<tr>
<td>X₉</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Full</td>
<td>$$</td>
<td>Yes</td>
<td>No</td>
<td>Burger</td>
<td>&gt;60</td>
<td>No</td>
</tr>
<tr>
<td>X₁₀</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Full</td>
<td>$$$</td>
<td>No</td>
<td>Yes</td>
<td>Italian</td>
<td>10–30</td>
<td>No</td>
</tr>
<tr>
<td>X₁₁</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>None</td>
<td>$$</td>
<td>No</td>
<td>No</td>
<td>Thai</td>
<td>0–10</td>
<td>No</td>
</tr>
<tr>
<td>X₁₂</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Full</td>
<td>$$</td>
<td>No</td>
<td>No</td>
<td>Burger</td>
<td>30–60</td>
<td>Yes</td>
</tr>
</tbody>
</table>
ID3-induced decision tree
Model spaces

Decision tree

Nearest neighbor

Version space
Decision tree-induced partition – example

Color

Size

Shape

Size

big  small

big  small

big  small

round  square

red  blue  green

+  -

+  -

+  -
$k$-Nearest Neighbor
Instance-Based Learning

Some material adapted from slides by Andrew Moore, CMU.

1-Nearest Neighbor

- One of the simplest of all machine learning classifiers
- Simple idea: label a new point the same as the closest known point

Label it red.
1-Nearest Neighbor

- A type of instance-based learning
  - Also known as “memory-based” learning
- Forms a Voronoi tessellation of the instance space
**Distance Metrics**

- Different metrics can change the decision surface

\[
\text{Dist}(\mathbf{a}, \mathbf{b}) = (a_1 - b_1)^2 + (a_2 - b_2)^2
\]

\[
\text{Dist}(\mathbf{a}, \mathbf{b}) = (a_1 - b_1)^2 + (3a_2 - 3b_2)^2
\]

- Standard Euclidean distance metric:
  - Two-dimensional: \( \text{Dist}(\mathbf{a}, \mathbf{b}) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2} \)
  - Multivariate: \( \text{Dist}(\mathbf{a}, \mathbf{b}) = \sqrt{\sum (a_i - b_i)^2} \)

Adapted from “Instance-Based Learning” lecture slides by Andrew Moore, CMU.
Four Aspects of an Instance-Based Learner:

1. A distance metric
2. How many nearby neighbors to look at?
3. A weighting function (optional)
4. How to fit with the local points?

Adapted from “Instance-Based Learning” lecture slides by Andrew Moore, CMU.
1-NN’s Four Aspects as an Instance-Based Learner:

1. A distance metric
   - *Euclidian*

2. How many nearby neighbors to look at?
   - *One*

3. A weighting function (optional)
   - *Unused*

4. How to fit with the local points?
   - *Just predict the same output as the nearest neighbor.*

Adapted from “Instance-Based Learning” lecture slides by Andrew Moore, CMU.
Mystery of renowned zen garden revealed | CNN Article
Thursday, September 26, 2002 Posted: 10:11 AM EDT (1411 GMT)

LONDON (Reuters) -- For centuries visitors to the renowned Ryoanji Temple garden in Kyoto, Japan have been entranced and mystified by the simple arrangement of rocks.

The five sparse clusters on a rectangle of raked gravel are said to be pleasing to the eyes of the hundreds of thousands of tourists who visit the garden each year.

Scientists in Japan said on Wednesday they now believe they have discovered its mysterious appeal.

"We have uncovered the implicit structure of the Ryoanji garden's visual ground and have shown that it includes an abstract, minimalist depiction of natural scenery," said Gert Van Tonder of Kyoto University.

The researchers discovered that the empty space of the garden evokes a hidden image of a branching tree that is sensed by the unconscious mind.

"We believe that the unconscious perception of this pattern contributes to the enigmatic appeal of the garden," Van Tonder added.

He and his colleagues believe that whoever created the garden during the Muromachi era between 1333-1573 knew exactly what they were doing and placed the rocks around the tree image.

By using a concept called medial-axis transformation, the scientists showed that the hidden branched tree converges on the main area from which the garden is viewed.

The trunk leads to the prime viewing site in the ancient temple that once overlooked the garden. It is thought that abstract art may have a similar impact.

"There is a growing realisation that scientific analysis can reveal unexpected structural features hidden in controversial abstract paintings," Van Tonder said

Adapted from “Instance-Based Learning” lecture slides by Andrew Moore, CMU.
**k – Nearest Neighbor**

- Generalizes 1-NN to smooth away noise in the labels
- A new point is now assigned the most frequent label of its $k$ nearest neighbors

Label it red, when $k = 3$

Label it blue, when $k = 7$
**k-Nearest Neighbor (k = 9)**

A magnificent job of noise smoothing. Three cheers for 9-nearest-neighbor. But the lack of gradients and the jerkiness isn’t good.

Appalling behavior! Loses all the detail that 1-nearest neighbor would give. The tails are horrible!

Fits much less of the noise, captures trends. But still, frankly, pathetic compared with linear regression.

Adapted from “Instance-Based Learning” lecture slides by Andrew Moore, CMU.
The Naïve Bayes Classifier

Some material adapted from slides by Tom Mitchell, CMU.
The Naïve Bayes Classifier

- Recall Bayes rule:

\[
P(Y_i \mid X_j) = \frac{P(Y_i)P(X_j \mid Y_i)}{P(X_j)}
\]

- Which is short for:

\[
P(Y = y_i \mid X = x_j) = \frac{P(Y = y_i)P(X = x_j \mid Y = y_i)}{P(X = x_j)}
\]

- We can re-write this as:

\[
P(Y = y_i \mid X = x_j) = \frac{\sum_k P(X = x_j \mid Y = y_k)P(Y = y_k)}{P(X = x_j)}
\]
Deriving Naïve Bayes

- Idea: use the training data to directly estimate:
  \[ P(X \mid Y) \quad \text{and} \quad P(Y) \]

- Then, we can use these values to estimate
  \[ P(Y \mid X_{\text{new}}) \] using Bayes rule.

- Recall that representing the full joint probability
  \[ P(X_1, X_2, \ldots, X_n \mid Y) \] is not practical.
Deriving Naïve Bayes

However, if we make the assumption that the attributes are independent, estimation is easy!

\[ P(X_1, \ldots, X_n \mid Y) = \prod_i P(X_i \mid Y) \]

In other words, we assume all attributes are conditionally independent given Y.

Often this assumption is violated in practice, but more on that later…
Deriving Naïve Bayes

Let $X = \langle X_1, \ldots, X_n \rangle$ and label $Y$ be discrete.

Then, we can estimate $P(X_i | Y_i)$ and $P(Y_i)$ directly from the training data by counting!

<table>
<thead>
<tr>
<th>Sky</th>
<th>Temp</th>
<th>Humid</th>
<th>Wind</th>
<th>Water</th>
<th>Forecast</th>
<th>Play?</th>
</tr>
</thead>
<tbody>
<tr>
<td>sunny</td>
<td>warm</td>
<td>normal</td>
<td>strong</td>
<td>warm</td>
<td>same</td>
<td>yes</td>
</tr>
<tr>
<td>sunny</td>
<td>warm</td>
<td>high</td>
<td>strong</td>
<td>warm</td>
<td>same</td>
<td>yes</td>
</tr>
<tr>
<td>rainy</td>
<td>cold</td>
<td>high</td>
<td>strong</td>
<td>warm</td>
<td>change</td>
<td>no</td>
</tr>
<tr>
<td>sunny</td>
<td>warm</td>
<td>high</td>
<td>strong</td>
<td>cool</td>
<td>change</td>
<td>yes</td>
</tr>
</tbody>
</table>
The Naïve Bayes Classifier

- Now we have:

\[
P(Y = y_j \mid X_1, \ldots, X_n) = \frac{P(Y = y_j) \prod_i P(X_i \mid Y = y_j)}{\sum_k P(Y = y_k) \prod_i P(X_i \mid Y = y_k)}
\]

which is just a one-level Bayesian Network

- To classify a new point \( X_{\text{new}} \):

\[
Y_{\text{new}} \leftarrow \arg \max_{y_k} P(Y = y_k) \prod_i P(X_i \mid Y = y_k)
\]
The Naïve Bayes Algorithm

- For each value $y_k$
  - Estimate $P(Y = y_k)$ from the data.
  - For each value $x_{ij}$ of each attribute $X_i$
    - Estimate $P(X_i = x_{ij} \mid Y = y_k)$

- Classify a new point via:
  $$Y_{new} \leftarrow \arg \max_{y_k} P(Y = y_k) \prod_i P(X_i \mid Y = y_k)$$

- In practice, the independence assumption doesn’t often hold true, but Naïve Bayes performs very well despite it.
Naïve Bayes Applications

- **Text classification**
  - Which e-mails are spam?
  - Which e-mails are meeting notices?
  - Which author wrote a document?

- **Classifying mental states**

Learning $P(\text{BrainActivity} \mid \text{WordCategory})$

Pairwise Classification
Accuracy: 85%

People Words  Animal Words
Polynomial Curve Fitting

Slides adapted from Pattern Recognition and Machine Learning by Christopher Bishop
Polynomial Curve Fitting

\[ y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j \]
Sum-of-Squares Error Function

\[ E(w) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, w) - t_n\}^2 \]
$0^{th}$ Order Polynomial

$M = 0$
1st Order Polynomial

\[ M = 1 \]
3\textsuperscript{rd} Order Polynomial

$M = 3$
9\textsuperscript{th} Order Polynomial
Over-fitting

![Graph showing training and test error over different M values. The training error decreases with increasing M, while the test error increases and then decreases急剧地。]
## Polynomial Coefficients

<table>
<thead>
<tr>
<th></th>
<th>$M = 0$</th>
<th>$M = 1$</th>
<th>$M = 3$</th>
<th>$M = 9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_0^*$</td>
<td>0.19</td>
<td>0.82</td>
<td>0.31</td>
<td>0.35</td>
</tr>
<tr>
<td>$w_1^*$</td>
<td></td>
<td>-1.27</td>
<td>7.99</td>
<td>232.37</td>
</tr>
<tr>
<td>$w_2^*$</td>
<td></td>
<td></td>
<td>-25.43</td>
<td>-5321.83</td>
</tr>
<tr>
<td>$w_3^*$</td>
<td></td>
<td></td>
<td>17.37</td>
<td>48568.31</td>
</tr>
<tr>
<td>$w_4^*$</td>
<td></td>
<td></td>
<td></td>
<td>-231639.30</td>
</tr>
<tr>
<td>$w_5^*$</td>
<td></td>
<td></td>
<td></td>
<td>640042.26</td>
</tr>
<tr>
<td>$w_6^*$</td>
<td></td>
<td></td>
<td></td>
<td>-1061800.52</td>
</tr>
<tr>
<td>$w_7^*$</td>
<td></td>
<td></td>
<td></td>
<td>1042400.18</td>
</tr>
<tr>
<td>$w_8^*$</td>
<td></td>
<td></td>
<td></td>
<td>-557682.99</td>
</tr>
<tr>
<td>$w_9^*$</td>
<td></td>
<td></td>
<td></td>
<td>125201.43</td>
</tr>
</tbody>
</table>
Data Set Size: $N = 15$

9th Order Polynomial
Data Set Size: $N = 100$

9th Order Polynomial
Regularization

Penalize large coefficient values

\[ \tilde{E}(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\lambda}{2} (\|\mathbf{w}\|_2)^2 \]

L₂ Norm

\[ \|\mathbf{w}\|_2 = \sqrt{\sum_{i} w_i^2} \] Measures the “complexity” of \( \mathbf{w} \)
Penalize large coefficient values

$$\tilde{E}(w) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, w) - t_n \}^2 + \frac{\lambda}{2} (\|w\|_2)^2$$

$$\tilde{E}(w) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, w) - t_n \}^2 + \frac{\lambda}{2} \sum_i w_i^2$$

L₂ Norm

$$\|w\|_2 = \sqrt{\sum_i w_i^2}$$

Measures the "complexity" of w

\(\lambda\) regularization parameter

higher \(\lambda\) \(\rightarrow\) more regularization
Regularization: $\lambda = 1.5\times10^{-8}$
Regularization: $\lambda = 1$
Regularization: $E_{RMS}$ vs. $\ln \lambda$
Polynomial Coefficients

<table>
<thead>
<tr>
<th>$w_k^*$</th>
<th>$\ln \lambda = -\infty$</th>
<th>$\ln \lambda = -18$</th>
<th>$\ln \lambda = 0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_0^*$</td>
<td>0.35</td>
<td>0.35</td>
<td>0.13</td>
</tr>
<tr>
<td>$w_1^*$</td>
<td>232.37</td>
<td>4.74</td>
<td>-0.05</td>
</tr>
<tr>
<td>$w_2^*$</td>
<td>-5321.83</td>
<td>-0.77</td>
<td>-0.06</td>
</tr>
<tr>
<td>$w_3^*$</td>
<td>48568.31</td>
<td>-31.97</td>
<td>-0.05</td>
</tr>
<tr>
<td>$w_4^*$</td>
<td>-231639.30</td>
<td>-3.89</td>
<td>-0.03</td>
</tr>
<tr>
<td>$w_5^*$</td>
<td>640042.26</td>
<td>55.28</td>
<td>-0.02</td>
</tr>
<tr>
<td>$w_6^*$</td>
<td>-1061800.52</td>
<td>41.32</td>
<td>-0.01</td>
</tr>
<tr>
<td>$w_7^*$</td>
<td>1042400.18</td>
<td>-45.95</td>
<td>-0.00</td>
</tr>
<tr>
<td>$w_8^*$</td>
<td>-557682.99</td>
<td>-91.53</td>
<td>0.00</td>
</tr>
<tr>
<td>$w_9^*$</td>
<td>125201.43</td>
<td>72.68</td>
<td>0.01</td>
</tr>
</tbody>
</table>
Learning via Gradient Descent

\[ \tilde{E}(w) = \frac{1}{2} \sum_{n=1}^{N} \left( y(x_n, w) - t_n \right)^2 + \frac{\lambda}{2} \sum_{j} w_j^2 \]

\[ \nabla_j \tilde{E}(w) = \sum_{n=1}^{N} x_n^j \left( y(x_n, w) - t_n \right) + \lambda w_j \]

Choose \( w \) randomly, where \( w_j \sim N(0, \sigma^2) \)

Repeat until \( w \) converges (i.e., \( ||w - w_{\text{old}}|| < \varepsilon \))

\[ w_{\text{old}} = w \]

For \( j = 0 \ldots M \):

\[ w_j = w_j - \alpha \nabla_j \tilde{E}(w) \]
The Gaussian Distribution

\[ \mathcal{N}(x|\mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp \left\{ -\frac{1}{2\sigma^2}(x - \mu)^2 \right\} \]

\[ \mathcal{N}(x|\mu, \sigma^2) > 0 \]

\[ \int_{-\infty}^{\infty} \mathcal{N}(x|\mu, \sigma^2) \, dx = 1 \]
The Multivariate Gaussian

\[
\mathcal{N}(x|\mu, \Sigma) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} \exp \left\{ -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right\}
\]
Gaussian Parameter Estimation

\[ p(x) \]

Likelihood function

\[ \mathcal{N}(x_n | \mu, \sigma^2) \]

\[
p(x|\mu, \sigma^2) = \prod_{n=1}^{N} \mathcal{N}(x_n | \mu, \sigma^2)
\]
Maximum (Log) Likelihood

\[
\ln p(x|\mu, \sigma^2) = -\frac{1}{2\sigma^2} \sum_{n=1}^{N} (x_n - \mu)^2 - \frac{N}{2} \ln \sigma^2 - \frac{N}{2} \ln(2\pi)
\]

\[
\mu_{ML} = \frac{1}{N} \sum_{n=1}^{N} x_n \\
\sigma_{ML}^2 = \frac{1}{N} \sum_{n=1}^{N} (x_n - \mu_{ML})^2
\]
Curve Fitting Re-visited

\[ y(x_0, \mathbf{w}) \]

\[ p(t|x_0, \mathbf{w}, \beta) = \mathcal{N}(t|y(x_0, \mathbf{w}), \beta^{-1}) \]
Maximum Likelihood

\[ p(t|x, w, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n | y(x_n, w), \beta^{-1}) \]

\[ \ln p(t|x, w, \beta) = -\frac{\beta}{2} \sum_{n=1}^{N} \{y(x_n, w) - t_n\}^2 + \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi) \]

\[ \beta E(w) \]

Determine \( w_{ML} \) by minimizing sum-of-squares error, \( E(w) \).

\[ \frac{1}{\beta_{ML}} = \frac{1}{N} \sum_{n=1}^{N} \{y(x_n, w_{ML}) - t_n\}^2 \]
Predictive Distribution

\[ p(t|x, w_{ML}, \beta_{ML}) = \mathcal{N}(t|y(x, w_{ML}), \beta_{ML}^{-1}) \]
MAP: A Step towards Bayes

\[ p(w|\alpha) = \mathcal{N}(w|0, \alpha^{-1}I) = \left( \frac{\alpha}{2\pi} \right)^{(M+1)/2} \exp \left\{ -\frac{\alpha}{2} w^T w \right\} \]

\[ p(w|x, t, \alpha, \beta) \propto p(t|x, w, \beta)p(w|\alpha) \]

\[ \beta \tilde{E}(w) = \frac{\beta}{2} \sum_{n=1}^{N} \{ y(x_n, w) - t_n \}^2 + \frac{\alpha}{2} w^T w \]

Determine \( w_{MAP} \) by minimizing regularized sum-of-squares error, \( \tilde{E}(w) \).
Bayesian Curve Fitting

\[ p(t|x, x, t) = \int p(t|x, w)p(w|x, t) \, dw = \mathcal{N}(t|m(x), s^2(x)) \]

\[ m(x) = \beta \phi(x)^T S \sum_{n=1}^{N} \phi(x_n) t_n \]
\[ s^2(x) = \beta^{-1} + \phi(x)^T S \phi(x) \]

\[ S^{-1} = \alpha I + \beta \sum_{n=1}^{N} \phi(x_n) \phi(x_n)^T \]
\[ \phi(x_n) = (x_n^0, \ldots, x_n^M)^T \]
Bayesian Predictive Distribution

\[ p(t|x, x, t) = \mathcal{N}(t|m(x), s^2(x)) \]
Model Selection

Cross-Validation

run 1
run 2
run 3
run 4
Curse of Dimensionality
Curse of Dimensionality

Polynomial curve fitting, $M = 3$

$$y(x, w) = w_0 + \sum_{i=1}^{D} w_i x_i + \sum_{i=1}^{D} \sum_{j=1}^{D} w_{ij} x_i x_j + \sum_{i=1}^{D} \sum_{j=1}^{D} \sum_{k=1}^{D} w_{ijk} x_i x_j x_k$$

Gaussian Densities in higher dimensions