Lecture 5: MDL Principle

IEEE Golden Jubilee Award for Technological Innovation (for the invention of arithmetic coding) 1998; IEEE Richard W. Hamming Medal (for fundamental contribution to information theory, statistical inference, control theory, and the theory of complexity) 1993; Kolmogorov Medal 2006; IBM Corporate Award (for the MDL/PMDL principles and stochastic complexity) 1991; IBM Outstanding Innovation Award (for work in statistical inference, information theory, and the theory of complexity) 1988; ...
1 Occam’s Razor

- House
- Visual Recognition
- Astronomy
- Razor
1 Occam’s Razor
   - House
   - Visual Recognition
   - Astronomy
   - Razor

2 MDL Principle
   - Idea
   - Rules & Exceptions
   - Probabilistic Models
   - Old-Style MDL
House
Brandon has

1. cough,
2. severe abdominal pain,
3. nausea,
4. low blood pressure,
5. fever.

No single disease causes all of these. Each symptom can be caused by some (possibly different) disease...

Dr. House explains the symptoms with two simple causes:
1. common cold, causing the cough and fever,
2. pharmacy error: cough medicine replaced by gout medicine.
Brandon has

1. cough,
2. severe abdominal pain,
3. nausea,
4. low blood pressure,
5. fever.

No single disease causes all of these.
Brandon has

1. cough,
2. severe abdominal pain,
3. nausea,
4. low blood pressure,
5. fever.

No single disease causes all of these.

Each symptom can be caused by some (possibly different) disease...
Brandon has

1. cough,
2. severe abdominal pain,
3. nausea,
4. low blood pressure,
5. fever.

No single disease causes all of these.

Each symptom can be caused by some (possibly different) disease...
Brandon has

1. cough,
2. severe abdominal pain,
3. nausea,
4. low blood pressure,
5. fever.

No single disease causes all of these.

Each symptom can be caused by some (possibly different) disease...
Brandon has

1. cough,
2. severe abdominal pain,
3. nausea,
4. low blood pressure,
5. fever.

No single disease causes all of these.

Each symptom can be caused by *some* (possibly different) disease...
Brandon has

1. cough,
2. severe abdominal pain,
3. nausea,
4. low blood pressure,
5. fever.

No single disease causes all of these.

Each symptom can be caused by some (possibly different) disease...
Brandon has

1. cough,
2. severe abdominal pain,
3. nausea,
4. low blood pressure,
5. fever.

No single disease causes all of these.

Each symptom can be caused by *some* (possibly different) disease...
Brandon has

1. cough,
2. severe abdominal pain,
3. nausea,
4. low blood pressure,
5. fever.

No single disease causes all of these.

Each symptom can be caused by some (possibly different) disease...

Dr. House explains the symptoms with two simple causes:
Brandon has
1. cough,
2. severe abdominal pain,
3. nausea,
4. low blood pressure,
5. fever.

No single disease causes all of these.

Each symptom can be caused by some (possibly different) disease...

Dr. House explains the symptoms with two simple causes:
1. common cold, causing the cough and fever,
Brandon has:

1. cough,
2. severe abdominal pain,
3. nausea,
4. low blood pressure,
5. fever.

No single disease causes all of these.

Each symptom can be caused by some (possibly different) disease...

Dr. House explains the symptoms with two simple causes:

1. common cold, causing the cough and fever,
2. pharmacy error: cough medicine replaced by gout medicine.
Visual Recognition

Three Concepts: Information
Visual Recognition
Visual Recognition
Visual Recognition

Three Concepts: Information
Visual Recognition
Visual Recognition
Visual Recognition
Visual Recognition
Schema huius praeminze divisionis Sphaearum.
Astronomy

Outline
- Occam’s Razor
- MDL Principle
- Astronomy

Copyright © 2005 Pearson Prentice Hall, Inc.
William of Ockham (c. 1288–1348)
Occam’s Razor

Entities should not be multiplied beyond necessity.
Occam’s Razor

Entities should not be multiplied beyond necessity.

Isaac Newton: “We are to admit no more causes of natural things than such as are both true and sufficient to explain their appearances.”
Occam’s Razor

Entities should not be multiplied beyond necessity.

Isaac Newton: “We are to admit no more causes of natural things than such as are both true and sufficient to explain their appearances.”

Diagnostic parsimony: Find the fewest possible causes that explain the symptoms.
Occam’s Razor

Entities should not be multiplied beyond necessity.

Isaac Newton: “We are to admit no more causes of natural things than such as are both true and sufficient to explain their appearances.”

Diagnostic parsimony: Find the fewest possible causes that explain the symptoms.

(Hickam’s dictum: “Patients can have as many diseases as they damn well please.”)
Visual Recognition
Visual Recognition
Visual Recognition
Visual Recognition
<table>
<thead>
<tr>
<th>Outline</th>
<th>Idea</th>
<th>Rules &amp; Exceptions</th>
<th>Probabilistic Models</th>
<th>Old-Style MDL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Occam’s Razor</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>House</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Visual Recognition</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Astronomy</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Razor</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MDL Principle</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Idea</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rules &amp; Exceptions</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Probabilistic Models</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Old-Style MDL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Minimum Description Length (MDL) Principle (2-part)

Choose the hypothesis which minimizes the sum of
1. the codelength of the hypothesis, and
2. the codelength of the data with the help of the hypothesis.
Minimum Description Length (MDL) Principle (2-part)

Choose the hypothesis which minimizes the sum of
1. the codelength of the hypothesis, and
2. the codelength of the data with the help of the hypothesis.

How to encode data with the help of a hypothesis?
Encoding Data: Rules & Exceptions

Idea 1: Hypothesis = rule; encode exceptions.
**Encoding Data: Rules & Exceptions**

**Idea 1:** Hypothesis = rule; encode exceptions.

Black box of size $25 \times 25 = 625$, white dots at $(x_1, y_1), (x_2, y_2), (x_3, y_3)$. 
Encoding Data: Rules & Exceptions

**Idea 1:** Hypothesis = rule; encode exceptions.

Black box of size $25 \times 25 = 625$, white dots at $(x_1, y_1), (x_2, y_2), (x_3, y_3)$.

For image of size $n = 625$, there are $2^n$ different images, and

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

different groups of $k$ exceptions.
Encoding Data: Rules & Exceptions

**Idea 1:** Hypothesis = rule; encode exceptions.

Black box of size $25 \times 25 = 625$, white dots at $(x_1, y_1), (x_2, y_2), (x_3, y_3)$.

For image of size $n = 625$, there are $2^n$ different images, and

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

different groups of $k$ exceptions.

$k = 1: \binom{n}{1} = 625 \ll 2^{625}$.

Codelength $\log_2(n + 1) + \log_2 \binom{n}{k} \approx 19$ vs. 625
Encoding Data: Rules & Exceptions

**Idea 1:** Hypothesis = rule; encode exceptions.

Black box of size $25 \times 25 = 625$, white dots at $(x_1, y_1), (x_2, y_2), (x_3, y_3)$. 

For image of size $n = 625$, there are $2^n$ different images, and

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

different groups of $k$ exceptions.

$k = 2 : \binom{n}{2} = 195\,000 \ll 2^{625}$.

Codelength $\log_2(n + 1) + \log_2 \left( \binom{n}{k} \right) \approx 27$ vs. 625
Encoding Data: Rules & Exceptions

**Idea 1:** Hypothesis = rule; encode exceptions.

Black box of size $25 \times 25 = 625$, white dots at $(x_1, y_1), (x_2, y_2), (x_3, y_3)$.

For image of size $n = 625$, there are $2^n$ different images, and

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

different groups of $k$ exceptions.

$k = 3 : \binom{625}{3} = 40\,495\,000 \ll 2^{625}$.

Codelength $\log_2(n + 1) + \log_2 \binom{n}{k} \approx 35$ vs. 625
Idea 1: Hypothesis = rule; encode exceptions.

Black box of size $25 \times 25 = 625$, white dots at $(x_1, y_1), (x_2, y_2), (x_3, y_3)$.

For image of size $n = 625$, there are $2^n$ different images, and

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

different groups of $k$ exceptions.

$k = 10: \binom{625}{10} = 233135400000000000000000 \ll 2^{625}$.

Codelength $\log_2(n+1) + \log_2 \binom{n}{k} \approx 80$ vs. 625
Encoding Data: Rules & Exceptions

**Idea 1:** Hypothesis = rule; encode exceptions.

Black box of size $25 \times 25 = 625$, white dots at $(x_1, y_1), (x_2, y_2), (x_3, y_3)$.

For image of size $n = 625$, there are $2^n$ different images, and

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

different groups of $k$ exceptions.

$k = 100 : \left( \begin{array}{c} n \\ 100 \end{array} \right) \approx 9.5 \times 10^{117} \ll 2^{625}$.

Codelength $\log_2(n + 1) + \log_2 \left( \begin{array}{c} n \\ k \end{array} \right) \approx 401$ vs. $625$
**Idea 1:** Hypothesis = rule; encode exceptions.

Black box of size $25 \times 25 = 625$, white dots at $(x_1, y_1), (x_2, y_2), (x_3, y_3)$.

For image of size $n = 625$, there are $2^n$ different images, and

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

different groups of $k$ exceptions.

$k = 300 : \binom{625}{300} \approx 2.7 \times 10^{186} < 2^{625}$.

Codelength $\log_2(n + 1) + \log_2 \left( \frac{n}{k} \right) \approx 629$ vs. 625
Idea 2: Hypothesis = probability distribution.

<table>
<thead>
<tr>
<th>Noisy</th>
<th>PSNR=19.8</th>
<th>MDL (A-B)</th>
<th>PSNR=32.9</th>
</tr>
</thead>
</table>

Teemu Roos  Three Concepts: Information
Encoding Data: Probabilistic Models

Idea 2: Hypothesis = probability distribution.

<table>
<thead>
<tr>
<th>Noisy</th>
<th>PSNR=19.8</th>
<th>MDL (A-B)</th>
<th>PSNR=32.9</th>
</tr>
</thead>
</table>

How to encode a distribution?
Two-Part Codes

Let $\mathcal{M} = \{ p_\theta : \theta \in \Theta \}$ be a parametric probabilistic model class, i.e., a set of distributions $p_\theta$ indexed by parameter $\theta$. 
Two-Part Codes

Let \( \mathcal{M} = \{ p_\theta : \theta \in \Theta \} \) be a parametric probabilistic model class, i.e., a set of distributions \( p_\theta \) indexed by parameter \( \theta \).

If the parameter space \( \Theta \) is discrete, we can construct a (prefix) code \( C_1 : \Theta \to \{0,1\}^* \) which maps each parameter value to a codeword.
Two-Part Codes

Let $\mathcal{M} = \{p_\theta : \theta \in \Theta\}$ be a parametric probabilistic model class, i.e., a set of distributions $p_\theta$ indexed by parameter $\theta$.

If the parameter space $\Theta$ is discrete, we can construct a (prefix) code $C_1 : \Theta \rightarrow \{0,1\}^*$ which maps each parameter value to a codeword.

For each distribution $p_\theta$ there is a prefix code $C_\theta : \mathcal{D} \rightarrow \{0,1\}^*$ where $D \in \mathcal{D}$ is a data-set to be encoded, such that the codeword lengths satisfy

$$\ell_\theta(D) \approx \log_2 \frac{1}{p_\theta(D)}.$$
Two-Part Codes

Let $\mathcal{M} = \{ p_\theta : \theta \in \Theta \}$ be a parametric probabilistic model class, i.e., a set of distributions $p_\theta$ indexed by parameter $\theta$.

If the parameter space $\Theta$ is discrete, we can construct a (prefix) code $C_1 : \Theta \to \{0, 1\}^*$ which maps each parameter value to a codeword.

For each distribution $p_\theta$ there is a prefix code $C_\theta : D \to \{0, 1\}^*$ where $D \in \mathcal{D}$ is a data-set to be encoded, such that the codeword lengths satisfy

$$\ell_\theta(D) \approx \log_2 \frac{1}{p_\theta(D)}.$$  

Using parameter value $\theta$, the total codelength becomes ($\approx$)

$$\ell_1(\theta) + \log_2 \frac{1}{p_\theta(D)}.$$
Two-Part Codes

The parameter value minimizing the codelength is given by the maximum likelihood parameter $\hat{\theta}$:

$$\min_{\theta \in \Theta} \log_2 \frac{1}{p_\theta(D)} = \log_2 \frac{1}{\max_{\theta \in \Theta} p_\theta(D)} = \log_2 \frac{1}{p_{\hat{\theta}}(D)}.$$
Two-Part Codes

The parameter value minimizing the codelength is given by the maximum likelihood parameter $\hat{\theta}$:

$$\min_{\theta \in \Theta} \log_2 \frac{1}{p_{\theta}(D)} = \log_2 \frac{1}{\max_{\theta \in \Theta} p_{\theta}(D)} = \log_2 \frac{1}{p_{\hat{\theta}}(D)}.$$

It could of course be that $\ell_1(\hat{\theta})$ is so large that some other parameter value gives a shorter total codelength.
Multi-Part Codes

If there are more than one model classes, $M_1, M_2, \ldots$ it is possible to construct **multi-part codes** where the parts are:

1. Encoding of the model class index: $C_0(i), i \in \mathbb{N}$. 
2. Encoding of the parameter (vector): $C_i(\theta), \theta \in \Theta_i$.
3. Encoding of the data: $C_{\theta}(D), D \in D$.

For instance, the models could be polynomials with different degrees, the parameters are the coefficients $\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \ldots + \theta_k x^k$. 

The more complex the model class (the higher the degree), the better it fits the data but the longer the second part $C_i(\theta)$ becomes.
Multi-Part Codes

If there are more than one model classes, \( \mathcal{M}_1, \mathcal{M}_2, \ldots \) it is possible to construct **multi-part codes** where the parts are

1. Encoding of the model class index: \( C_0(i), \ i \in \mathbb{N} \).
2. Encoding of the parameter (vector): \( C_i(\theta), \ \theta \in \Theta_i \).

For instance, the models could be polynomials with different degrees, the parameters are the coefficients \( \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \ldots + \theta_k x^k \). The more complex the model class (the higher the degree), the better it fits the data but the longer the second part \( C_i(\theta) \) becomes.
Multi-Part Codes

If there are more than one model classes, $M_1, M_2, \ldots$ it is possible to construct multi-part codes where the parts are:

1. Encoding of the model class index: $C_0(i), i \in \mathbb{N}$.
2. Encoding of the parameter (vector): $C_i(\theta), \theta \in \Theta_i$.
3. Encoding of the data: $C_\theta(D), D \in \mathcal{D}$.
Multi-Part Codes

If there are more than one model classes, $\mathcal{M}_1, \mathcal{M}_2, \ldots$ it is possible to construct **multi-part codes** where the parts are

1. Encoding of the model class index: $C_0(i), i \in \mathbb{N}$.
2. Encoding of the parameter (vector): $C_i(\theta), \theta \in \Theta_i$.
3. Encoding of the data: $C_\theta(D), D \in \mathcal{D}$.

For instance, the models could be polynomials with different degrees, the parameters are the **coefficients**

$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \ldots + \theta_k x^k.$$
Multi-Part Codes

If there are more than one model classes, $\mathcal{M}_1, \mathcal{M}_2, \ldots$ it is possible to construct **multi-part codes** where the parts are:

1. Encoding of the model class index: $C_0(i), i \in \mathbb{N}$.
2. Encoding of the parameter (vector): $C_i(\theta), \theta \in \Theta_i$.
3. Encoding of the data: $C_\theta(D), D \in \mathcal{D}$.

For instance, the models could be polynomials with different degrees, the parameters are the **coefficients**

$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \ldots + \theta_k x^k .$$

The more complex the model class (the higher the degree), the better it fits the data but the longer the second part $C_i(\theta)$ becomes.
Polynomials

Figure 1: A simple (1.1), complex (1.2) and a trade-off (3rd degree) polynomial.
Continuous Parameters

What if the parameters are continuous (like polynomial coefficients)? How to encode continuous values?
Continuous Parameters

What if the parameters are continuous (like polynomial coefficients)? How to encode continuous values?

**Solution**: Quantization. Choose a discrete subset of points, \( \theta^{(1)}, \theta^{(2)}, \ldots \), and use only them.
Continuous Parameters

What if the parameters are continuous (like polynomial coefficients)? How to encode continuous values?

**Solution:** Quantization. Choose a discrete subset of points, $\theta^{(1)}, \theta^{(2)}, \ldots,$ and use only them.
Continuous Parameters

What if the parameters are continuous (like polynomial coefficients)? How to encode continuous values?

**Solution**: Quantization. Choose a discrete subset of points, $\theta^{(1)}, \theta^{(2)}, \ldots$, and use only them.
Continuous Parameters

What if the parameters are continuous (like polynomial coefficients)? How to encode continuous values?

**Solution:** Quantization. Choose a discrete subset of points, $\theta^{(1)}, \theta^{(2)}, \ldots$, and use only them.
Continuous Parameters

What if the parameters are continuous (like polynomial coefficients)? How to encode continuous values?

**Solution:** Quantization. Choose a discrete subset of points, $\theta^{(1)}, \theta^{(2)}, \ldots$, and use only them.
Continuous Parameters

What if the parameters are continuous (like polynomial coefficients)? How to encode continuous values?

**Solution:** Quantization. Choose a discrete subset of points, \( \theta^{(1)}, \theta^{(2)}, \ldots \), and use only them.
Continuous Parameters

What if the parameters are continuous (like polynomial coefficients)? How to encode continuous values?

**Solution:** Quantization. Choose a discrete subset of points, \( \theta^{(1)}, \theta^{(2)}, \ldots \), and use only them.

If the points are sufficiently *dense* (in a codelength sense) then the codelength for data is still almost as short as \( \min_{\theta \in \Theta} \ell_\theta(D) \).
Continuous Parameters

What if the parameters are continuous (like polynomial coefficients)? How to encode continuous values?

**Solution:** Quantization. Choose a discrete subset of points, \( \theta^{(1)}, \theta^{(2)}, \ldots \), and use only them.

If the points are sufficiently dense (in a codelength sense) then the codelength for data is still almost as short as \( \min_{\theta \in \Theta} \ell_{\theta}(D) \).

Information Geometry!
About Quantization

How many points should there be in the subset $\theta(1), \theta(2), \ldots$?
About Quantization

How many points should there be in the subset $\theta^{(1)}, \theta^{(2)}, \ldots$?

**Intuition:** Estimation accuracy of order $\frac{1}{\sqrt{n}}$. 
About Quantization

How many points should there be in the subset $\theta^{(1)}, \theta^{(2)}, \ldots$?

**Intuition:** Estimation accuracy of order $\frac{1}{\sqrt{n}}$.

**Theorem**

Optimal quantization accuracy is of order $\frac{1}{\sqrt{n}}$.

$\Rightarrow$ number of points $\approx \sqrt{n^k} = n^{k/2}$, where $k = \text{dim}(\Theta)$. 
About Quantization

How many points should there be in the subset $\theta^{(1)}, \theta^{(2)}, \ldots$?

**Intuition:** Estimation accuracy of order $\frac{1}{\sqrt{n}}$.

**Theorem**

Optimal quantization accuracy is of order $\frac{1}{\sqrt{n}}$.

$\Rightarrow$ number of points $\approx \sqrt{n^k} = n^{k/2}$, where $k = \dim(\Theta)$. 
About Quantization

How many points should there be in the subset $\theta(1), \theta(2), \ldots$?

**Intuition:** Estimation accuracy of order $\frac{1}{\sqrt{n}}$.

**Theorem**

Optimal quantization accuracy is of order $\frac{1}{\sqrt{n}}$.

⇒ number of points $\approx \sqrt{n^k} = n^{k/2}$, where $k = \text{dim}(\Theta)$.

The codelength for the quantized parameters becomes

$$\ell(\theta^q) \approx \log_2 n^{k/2} = \frac{k}{2} \log_2 n.$$
Old-Style MDL

With the precision $\frac{1}{\sqrt{n}}$ the codelength for data is almost optimal:

$$\min_{\theta q \in \{\theta^{(1)}, \theta^{(2)}, \ldots\}} \ell_{\theta q}(D) \approx \min_{\theta \in \Theta} \ell_{\theta}(D) = \log_2 \frac{1}{p_\hat{\theta}(D)}.$$
With the precision $\frac{1}{\sqrt{n}}$ the codelength for data is almost optimal:

$$\min_{\theta \in \{\theta^{(1)}, \theta^{(2)}, \ldots\}} \ell_{\theta^q}(D) \approx \min_{\theta \in \Theta} \ell_{\theta}(D) = \log_2 \frac{1}{p^*_\theta(D)}.$$  

This gives the total codelength formula:

"Steam MDL"

$$\ell_{\theta^q}(D) + \ell(\theta^q) \approx \log_2 \frac{1}{p^*_\theta(D)} + \frac{k}{2} \log_2 n.$$
The $\frac{k}{2} \log_2 n$ formula is only a rough approximation, and works well only for very large samples.
The $\frac{k}{2} \log_2 n$ formula is only a rough approximation, and works well only for very large samples.

**Next week:**

- More advanced codes: mixtures, normalized maximum likelihood, etc.
- Foundations of MDL.