Information-Theoretic Modeling
Lecture 9: The MDL Principle

Jyrki Kivinen
Department of Computer Science, University of Helsinki

Autumn 2012
Lecture 9: 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 Outstanding Innovation Award (for work in statistical inference, information theory, and the theory of complexity) 1988; IEEE Claude E. Shannon Award 2009; ...
Occam’s Razor

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

2 MDL Principle
   - Rules & Exceptions
   - Probabilistic Models
   - Old-Style MDL
   - Modern MDL
House
Brandon has
1. cough,
2. severe abdominal pain,
3. nausea,
4. low blood pressure,
5. fever.
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.

1. pneumonia,
2. appendicitis,
3. food poisoning,
4. hemorrhage,
5. meningitis.

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
Visual Recognition
Visual Recognition
Visual Recognition
Visual Recognition
Visual Recognition
Visual Recognition
Visual Recognition

- Pac-Man
- Ghost
- Ghost

Jyrki Kivinen

Information-Theoretic Modeling
Schema huius praemissae divisionis Sphærarum.
Astronomy
Astronomy

Copyright © 2005 Pearson Prentice Hall, Inc.
William of Ockham (c. 1288–1348)
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.”)
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
Occam’s Razor
- House
- Visual Recognition
- Astronomy
- Razor

MDL Principle
- Rules & Exceptions
- Probabilistic Models
- Old-Style MDL
- Modern MDL
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.
**Idea 1:** Hypothesis = rule; encode exceptions.

Black box of size $25 \times 25 = 625$, white dots at $(x_1, y_1), \ldots, (x_k, y_k)$. 
**Encoding Data: Rules & Exceptions**

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

Black box of size $25 \times 25 = 625$, white dots at $(x_1, y_1), \ldots, (x_k, y_k)$.

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 $(x_1, y_1), \ldots, (x_k, y_k)$.

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.

$k = 1 : \binom{n}{1} = 625 \ll 2^{625} \approx 1.4 \cdot 10^{188}$.

Codelength $\log_2(n + 1) + \log_2 \binom{n}{k} \approx 19$ vs. $\log_2 2^{625} = 625$
Encoding Data: Rules & Exceptions

Idea 1: Hypothesis = rule; encode exceptions.

Black box of size $25 \times 25 = 625$, white dots at $(x_1, y_1), \ldots, (x_k, y_k)$.

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.

$k = 2 : \binom{625}{2} = 195000 \ll 2^{625} \approx 1.4 \cdot 10^{188}$.

Codelength $\log_2(n+1) + \log_2\left(\begin{pmatrix}n \\ k \end{pmatrix}\right) \approx 27$ vs. $\log_2 2^{625} = 625$
Encoding Data: Rules & Exceptions

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

Black box of size $25 \times 25 = 625$, white dots at $(x_1, y_1), \ldots, (x_k, y_k)$.

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.

$k = 3 : \binom{n}{3} = 40\,495\,000 \ll 2^{625} \approx 1.4 \cdot 10^{188}$.

Codelength $\log_2(n + 1) + \log_2 \binom{n}{k} \approx 35$ vs. $\log_2 2^{625} = 625$
Encoding Data: Rules & Exceptions

Idea 1: Hypothesis = rule; encode exceptions.

Black box of size $25 \times 25 = 625$, white dots at $(x_1, y_1), \ldots, (x_k, y_k)$.

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.

$k = 10 : \binom{625}{10} = 2331\,354\,000\,000\,000\,000\,000 \ll 2^{625}$.

Codelength $\log_2(n + 1) + \log_2 \binom{n}{k} \approx 80$ vs. $\log_2 2^{625} = 625$. 
Encoding Data: Rules & Exceptions

Idea 1: Hypothesis = rule; encode exceptions.

Black box of size $25 \times 25 = 625$, white dots at $(x_1, y_1), \ldots, (x_k, y_k)$.

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.

$k = 100 : \binom{n}{100} \approx 9.5 \cdot 10^{117} \ll 2^{625} \approx 1.4 \cdot 10^{188}$.

Codelength $\log_2(n + 1) + \log_2 \left( \binom{n}{k} \right) \approx 401$ vs. $\log_2 2^{625} = 625$
Encoding Data: Rules & Exceptions

Idea 1: Hypothesis = rule; encode exceptions.

Black box of size $25 \times 25 = 625$, white dots at $(x_1, y_1), \ldots, (x_k, y_k)$.

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.

$k = 300 : \binom{625}{300} \approx 2.7 \cdot 10^{186} < 2^{625} \approx 1.4 \cdot 10^{188}$.

Codelength $\log_2(n + 1) + \log_2 \binom{n}{k} \approx 629$ vs. $\log_2 2^{625} = 625$
Idea 1: Hypothesis = rule; encode exceptions.

Black box of size $25 \times 25 = 625$, white dots at $(x_1, y_1), \ldots, (x_k, y_k)$.

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.

$k = 372: \binom{n}{372} \approx 5.1 \cdot 10^{181} \ll 2^{625} \approx 1.4 \cdot 10^{188}$.

Codelength $\log_2(n+1) + \log_2 \binom{n}{k} \approx 613$ vs. $\log_2 2^{625} = 625$. 
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>

*Noisy PSNR=19.8, MDL (A-B) PSNR=32.9*
Idea 2: Hypothesis = probability distribution.

Rissanen & Shannon: $\log_2 \frac{1}{p_\hat{\theta}(D)} + \frac{k}{2} \log_2 n.$
Polynomials

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

From P. Grünewald
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:

$$\min_{\theta_q \in \{\theta(1), \theta(2), \ldots\}} \ell_{\theta_q}(D) \approx \min_{\theta \in \Theta} \ell_{\theta}(D) = \log_2 \frac{1}{p_{\hat{\theta}}(D)}.$$ 

This gives the total codelength formula:

"Steam MDL"

$$\ell_{\theta_q}(D) + \ell(\theta^q) \approx \log_2 \frac{1}{p_{\hat{\theta}}(D)} + \frac{k}{2} \log_2 n.$$
Old-Style MDL

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.

**MDL in the 21st century:**

- More advanced codes: mixtures, normalized maximum likelihood, etc.
Perhaps the best known use for MDL is for *model class selection* (which is often called just model selection).
MDL Model Selection

Perhaps the best known use for MDL is for *model class selection* (which is often called just model selection).

Suppose we have a family of model classes $\mathcal{M}_1, \ldots, M_m$, each with their own parameter set $\Theta_1, \ldots, \Theta_m$.

We want to pick the one that seems best suited for our data $D$. 
MDL Model Selection

Perhaps the best known use for MDL is for *model class selection* (which is often called just model selection).

Suppose we have a family of model classes $\mathcal{M}_1, \ldots, \mathcal{M}_m$, each with their own parameter set $\Theta_1, \ldots, \Theta_m$.

We want to pick the one that seems best suited for our data $D$.

Typically $\mathcal{M}_1 \subset \cdots \subset \mathcal{M}_m$, so $\mathcal{M}_m$ always achieves the best fit to data, but may be *overfitting* (see *Introduction to machine learning*).

Therefore we use MDL and pick $\mathcal{M}_i$ that minimizes the total code length.
MDL Model Selection

Actually here we have a “three-part” code:
Actually here we have a “three-part” code:

1. Encoding of the model class: $\ell(M_i), \ i \in \{1, \ldots, m\}$. 

$\ell(M_i)$ represents the encoding of the model class $M_i$. This term is used to calculate the complexity of a particular model class in the context of the Minimum Description Length (MDL) principle. The MDL principle is a method for model selection that balances the trade-off between model complexity and goodness of fit to the data.
Actually here we have a “three-part” code:

1. Encoding of the model class: $\ell(M_i), i \in \{1, \ldots, m\}$.
2. Encoding of the parameter (vector): $\ell_1(\theta), \theta \in \Theta_i$. 
Actually here we have a “three-part” code:

1. Encoding of the model class: $\ell(M_i), i \in \{1, \ldots, m\}$.
2. Encoding of the parameter (vector): $\ell_1(\theta), \theta \in \Theta_i$.
3. Encoding of the data: $\log_2 \frac{1}{p_\theta(D)}, D \in \mathcal{D}$. 
MDL Model Selection

Actually here we have a “three-part” code:

1. Encoding of the model class: $\ell(M_i), \ i \in \{1, \ldots, m\}$.
2. Encoding of the parameter (vector): $\ell_1(\theta), \ \theta \in \Theta_i$.
3. Encoding of the data: $\log_2 \frac{1}{p_\theta(D)}, \ D \in \mathcal{D}$.

If we have a finite family of $m$ model classes, we can do Part 1 with $\ell(M_i) = \log_2 m$ for all $i$.

This can be generalized to infinite families of model classes, in which case we often to pick $p$ which is “as uniform as possible” over $\mathbb{N}$ and $\ell(M_i) = \log_2(1/p(i))$. 

Jyrki Kivinen
Information-Theoretic Modeling
If we are interested in choosing a model class (and not the parameters), we can improve parts 2 & 3 by combining them into a better universal code than two-part:
MDL Model Selection

If we are interested in choosing a model class (and not the parameters), we can improve parts 2 & 3 by combining them into a better universal code than two-part:

1. Encoding of the model class index: $\ell(M_i), i \in \mathbb{N}$. 

---

Jyrki Kivinen
Information-Theoretic Modeling
If we are interested in choosing a model class (and not the parameters), we can improve parts 2 & 3 by combining them into a better universal code than two-part:

1. Encoding of the model class index: $\ell(M_i), \ i \in \mathbb{N}$.
2. Encoding of the data: $\ell_{M_i}(D), \ D \in \mathcal{D}$, where $\ell_{M_i}$ is a universal code-length (e.g., mixture, NML) based on model class $M_i$. 
MDL Model Selection

**MDL Explanation of MDL**

The success in extracting the structure from data can be measured by the codelength.
The success in extracting the structure from data can be measured by the codelength.

In practice, we only find the structure that is “visible” to the used model class(es). For instance, the Bernoulli (coin flipping) model only sees the number of 1s.
MDL & Bayes

The MDL model selection criterion

\[
\text{minimize } \ell(\theta) + \ell_\theta(D)
\]

can be interpreted (via \( p = 2^{-\ell} \)) as

\[
\text{maximize } p(\theta) \cdot p_\theta(D)
\]
MDL & Bayes

The MDL model selection criterion

\[
\text{minimize } \ell(\theta) + \ell_\theta(D)
\]

can be interpreted (via \( p = 2^{-\ell} \)) as

\[
\text{maximize } p(\theta) \cdot p_\theta(D)
\]

In Bayesian probability, this is equivalent to **maximization of posterior probability**:

\[
p(\theta \mid D) = \frac{p(\theta) p(D \mid \theta)}{p(D)}
\]

where the term \( p(D) \) (the marginal probability of \( D \)) is constant wrt. \( \theta \) and doesn’t affect model selection.
MDL & Bayes

The MDL model selection criterion

$$\text{minimize } \ell(\theta) + \ell_\theta(D)$$

can be interpreted (via $p = 2^{-\ell}$) as

$$\text{maximize } p(\theta) \cdot p_\theta(D) .$$

In Bayesian probability, this is equivalent to maximization of posterior probability:

$$p(\theta \mid D) = \frac{p(\theta) p(D \mid \theta)}{p(D)} ,$$

where the term $p(D)$ (the marginal probability of $D$) is constant wrt. $\theta$ and doesn’t affect model selection.

$\Rightarrow$ Probabilistic Modelling
Example: Denoising

\[
\text{Complexity} = \text{Information} + \text{Noise} \\
= \text{Regularity} + \text{Randomness} \\
= \text{Algorithm} + \text{Compressed file}
\]
Example: Denoising

\[
\text{Complexity} = \text{Information} + \text{Noise} \\
= \text{Regularity} + \text{Randomness} \\
= \text{Algorithm} + \text{Compressed file}
\]

Denoising means the process of removing noise from a signal.
Example: Denoising

\[
\text{Complexity} = \text{Information} + \text{Noise} = \text{Regularity} + \text{Randomness} = \text{Algorithm} + \text{Compressed file}
\]

Denoising means the process of removing noise from a signal.

The MDL principle gives a natural method for denoising since the very idea of MDL is to separate the total complexity of a signal into information and noise.
Example: Denoising

\[
\text{Complexity} = \text{Information} + \text{Noise} \\
= \text{Regularity} + \text{Randomness} \\
= \text{Algorithm} + \text{Compressed file}
\]

**Denoising** means the process of removing noise from a signal.

The MDL principle gives a natural method for denoising since the very idea of MDL is to separate the total complexity of a signal into information and noise.

First encode a smooth signal (information), and then the difference to the observed signal (noise).
Example: Denoising

Noisy PSNR=19.8 MDL (A-B) PSNR=32.9
Example: Denoising
Example: Denoising
Example: Denoising
Example: Denoising
Example: Denoising
We’ll see some more MDL:

- MDL on continuous data
- examples of MDL in applications.

Then we move on to *Kolmogorov complexity*. 