Introduction to Information-Theoretic Modeling

Teemu Roos

Helsinki Institute for Information Technology HIIT,
Department of Computer Science, University of Helsinki

BCSMIF, Maresias, April 11–12, 2011

Jorma Rissanen (left) receiving the IEEE Information Theory Society Best Paper Award from Claude Shannon in 1986.

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; ...

1 Occam's Razor

House

House

Brandon has

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

House

Brandon has

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

No single disease causes all of these.

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...

House

Brandon has

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

1. pneumonia,

No single disease causes all of these.

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

House

Brandon has

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

1. pneumonia,
2. appendicitis,

No single disease causes all of these.

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

House

Brandon has

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

1. pneumonia,
2. appendicitis,
3. food poisoning,

No single disease causes all of these.

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

House

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,

No single disease causes all of these.

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

House

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...

House

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:

House

Brandon has

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

1. common cold,
2. appendicitis,
3. food poisoning,
4. hemorrhage,
5. common cold.

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,

House

Brandon has

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

1. common cold,
2. gout medicine,
3. gout medicine,
4. gout medicine,
5. common cold.

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.

William of Ockham (c. 1288–1348)

Occam's Razor

Occam's Razor

Entities should not be multiplied beyond necessity.

Occam's Razor

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

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

Occam's Razor

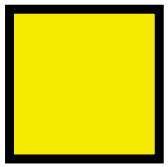
Entities should not be multiplied beyond necessity.

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

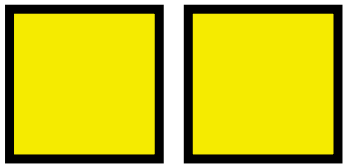
Diagnostic parsimony: Find the fewest possible causes that explain the symptoms.

(**Hickam's dictum:** "Patients can have as many diseases as they damn well please.")

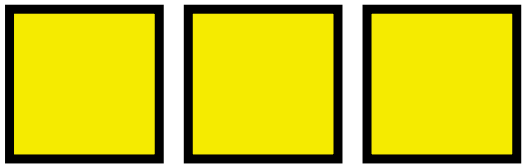
Guessing Game



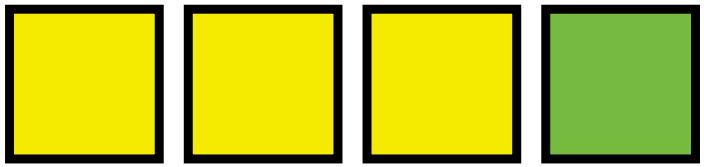
Guessing Game



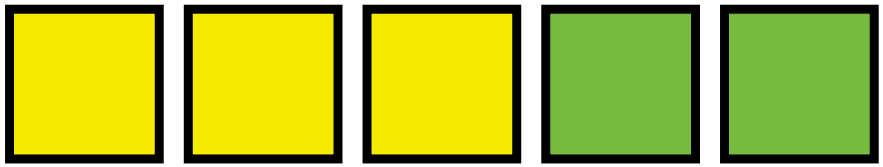
Guessing Game



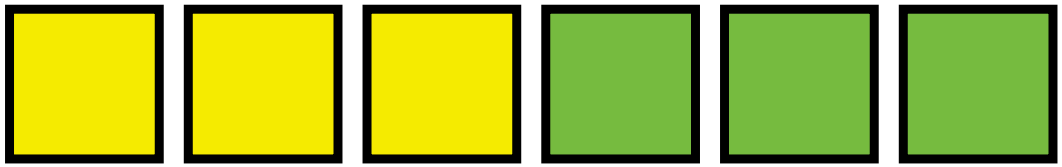
Guessing Game



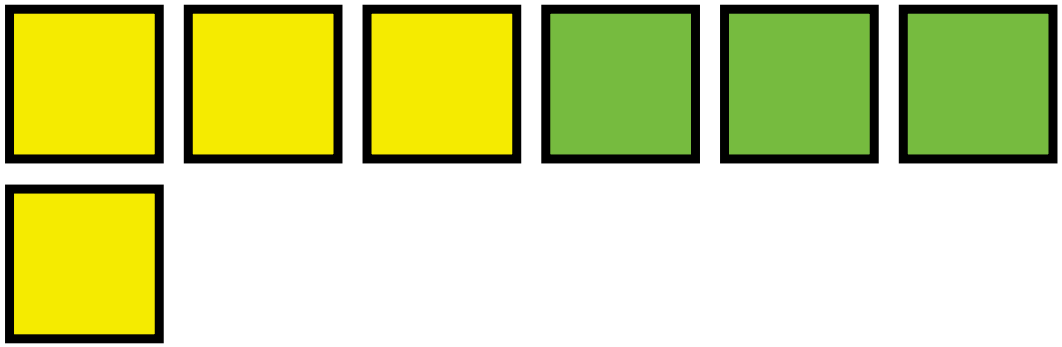
Guessing Game



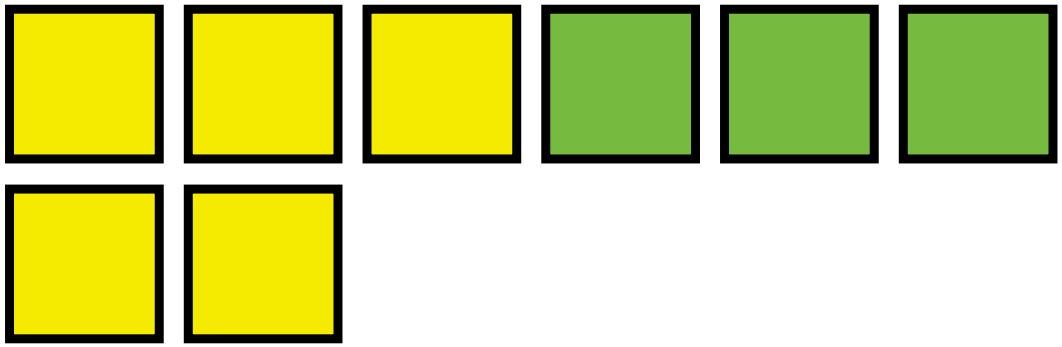
Guessing Game



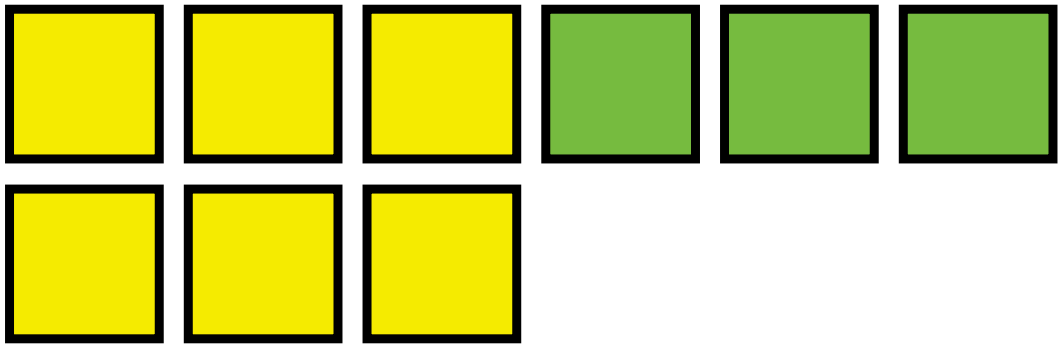
Guessing Game



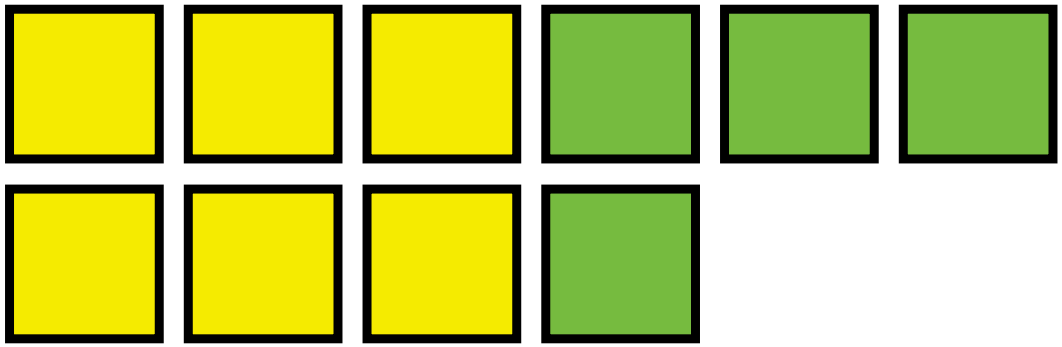
Guessing Game



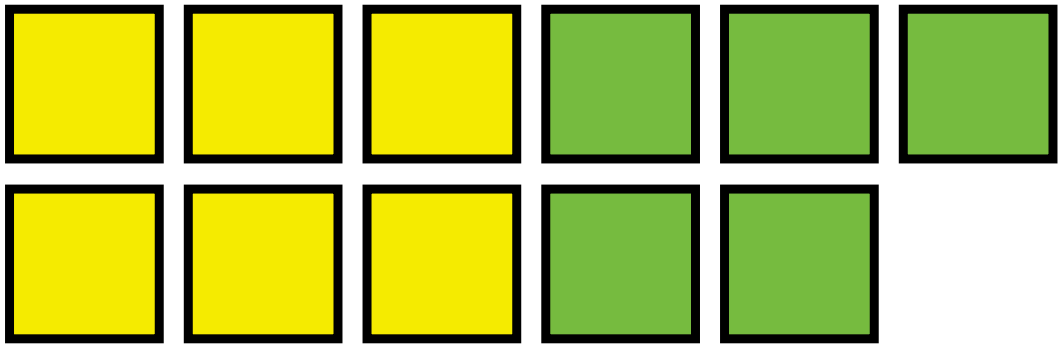
Guessing Game



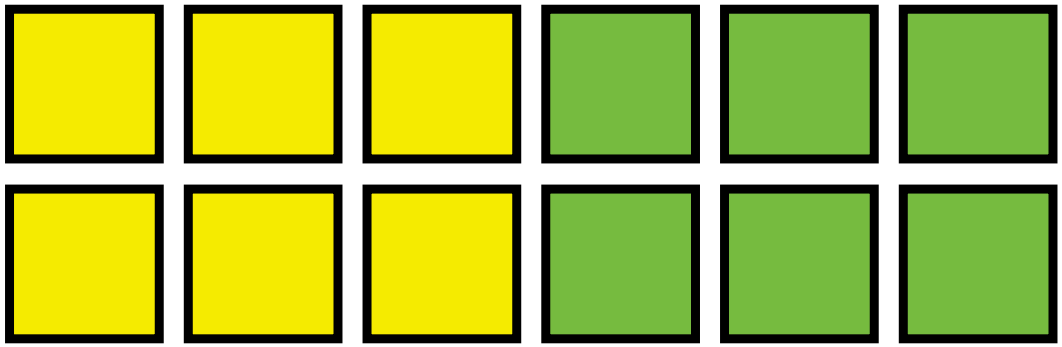
Guessing Game



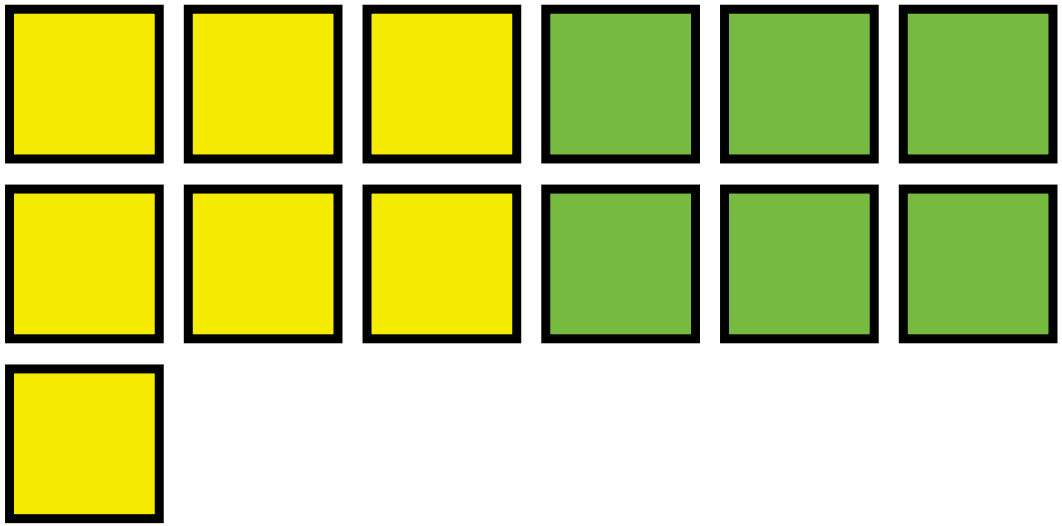
Guessing Game



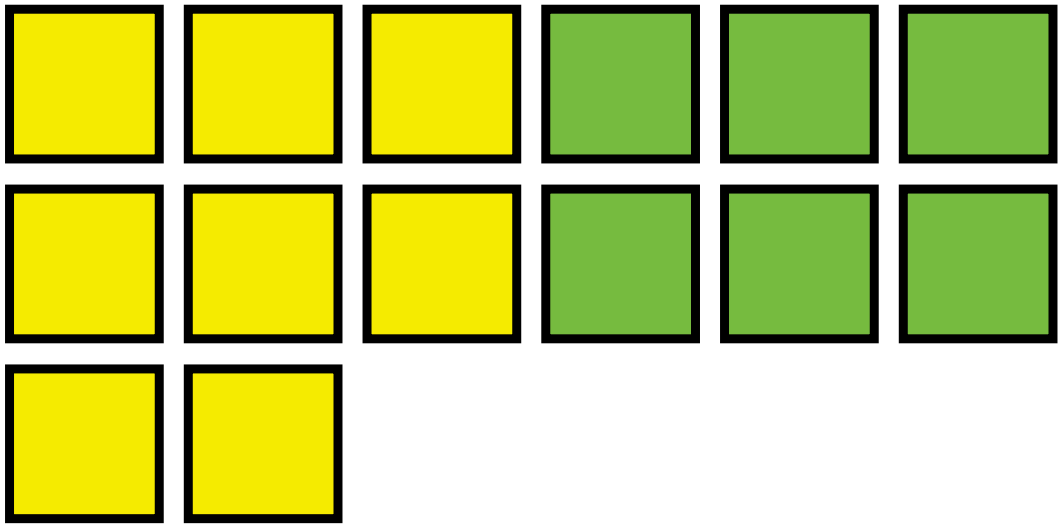
Guessing Game



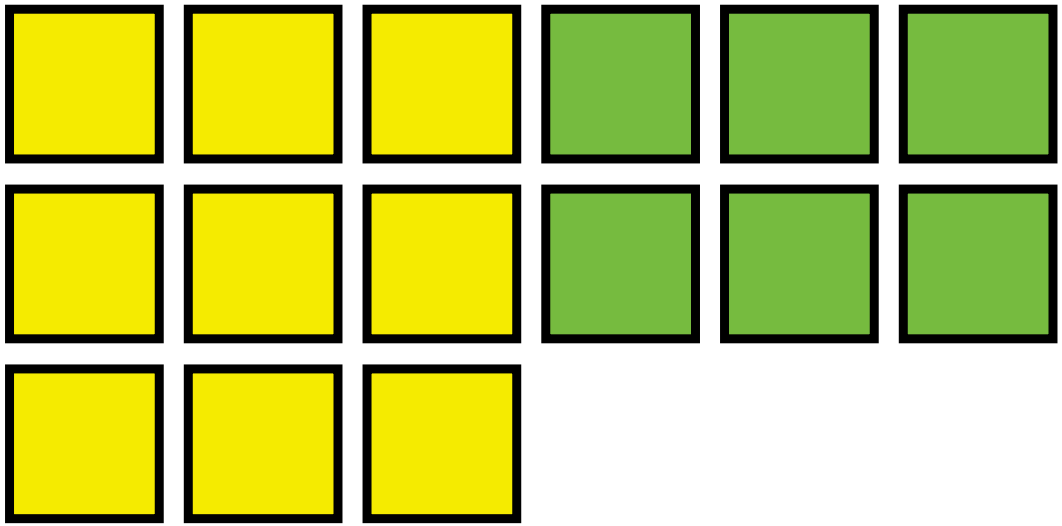
Guessing Game



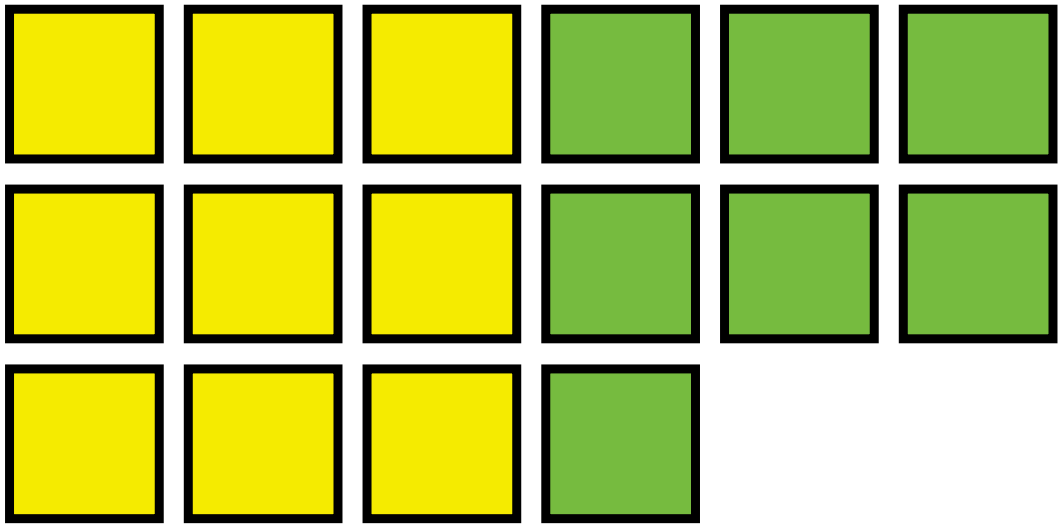
Guessing Game



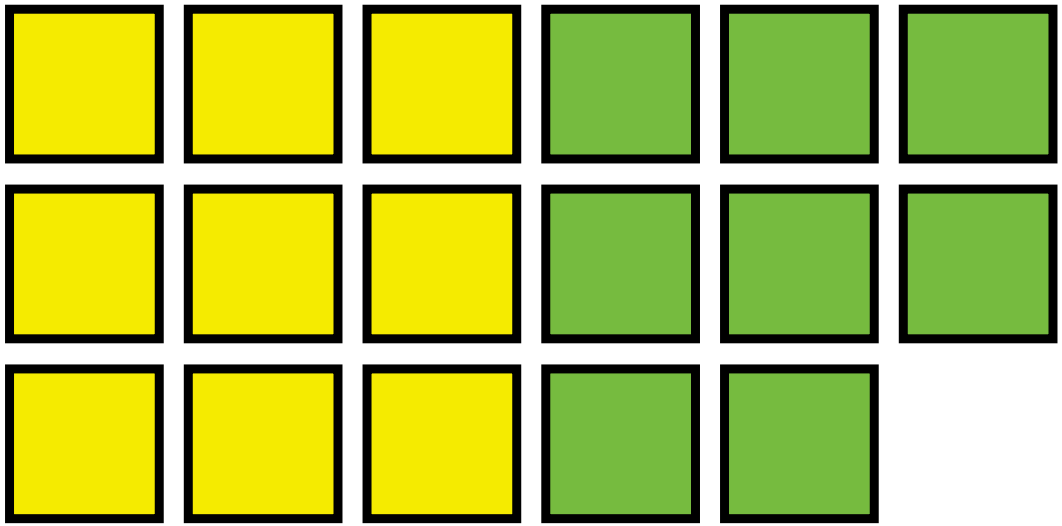
Guessing Game



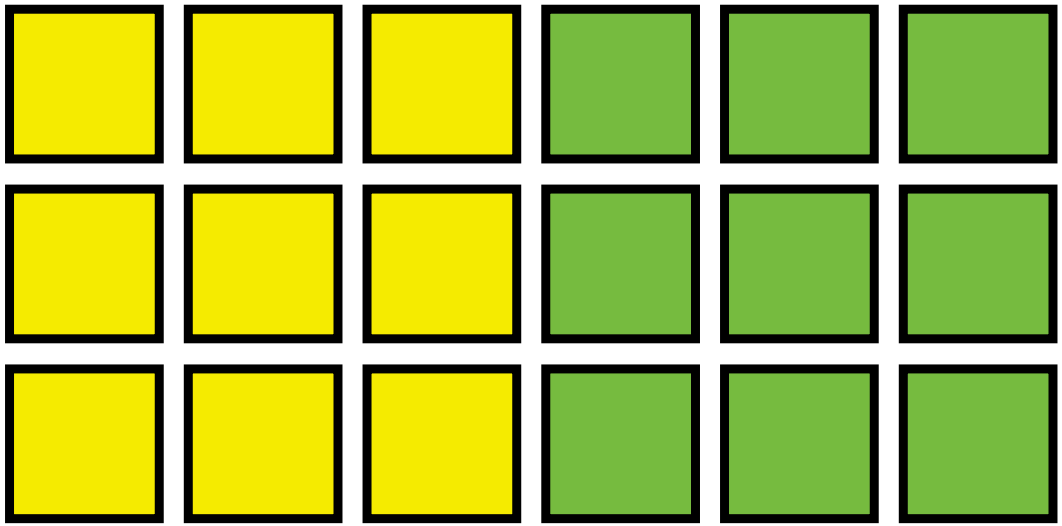
Guessing Game



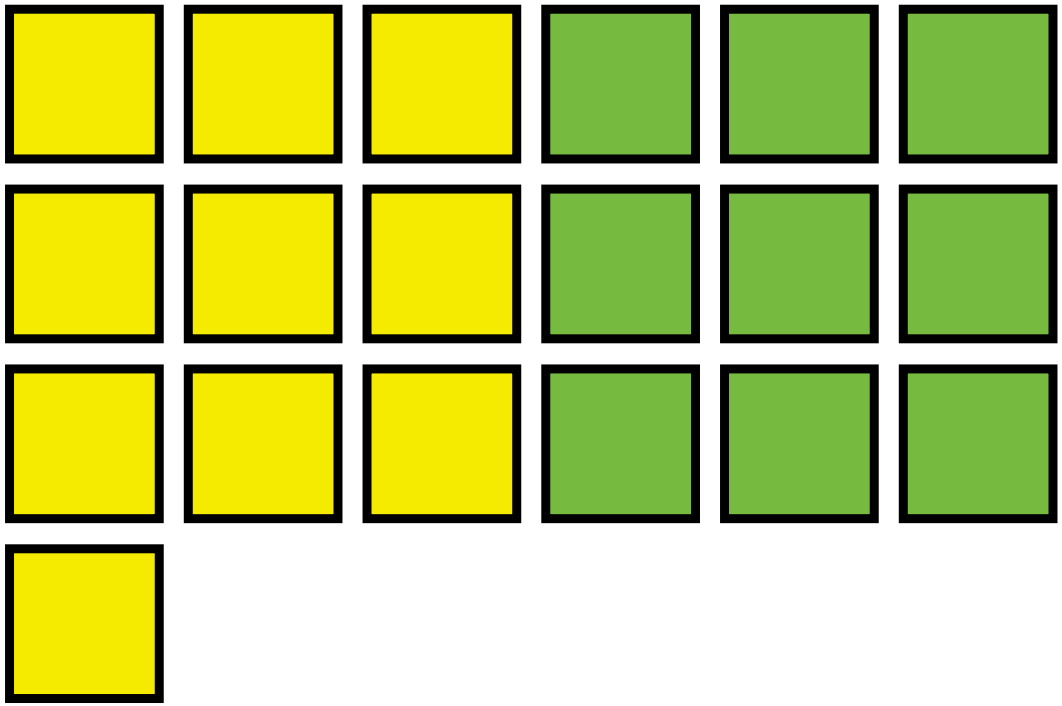
Guessing Game



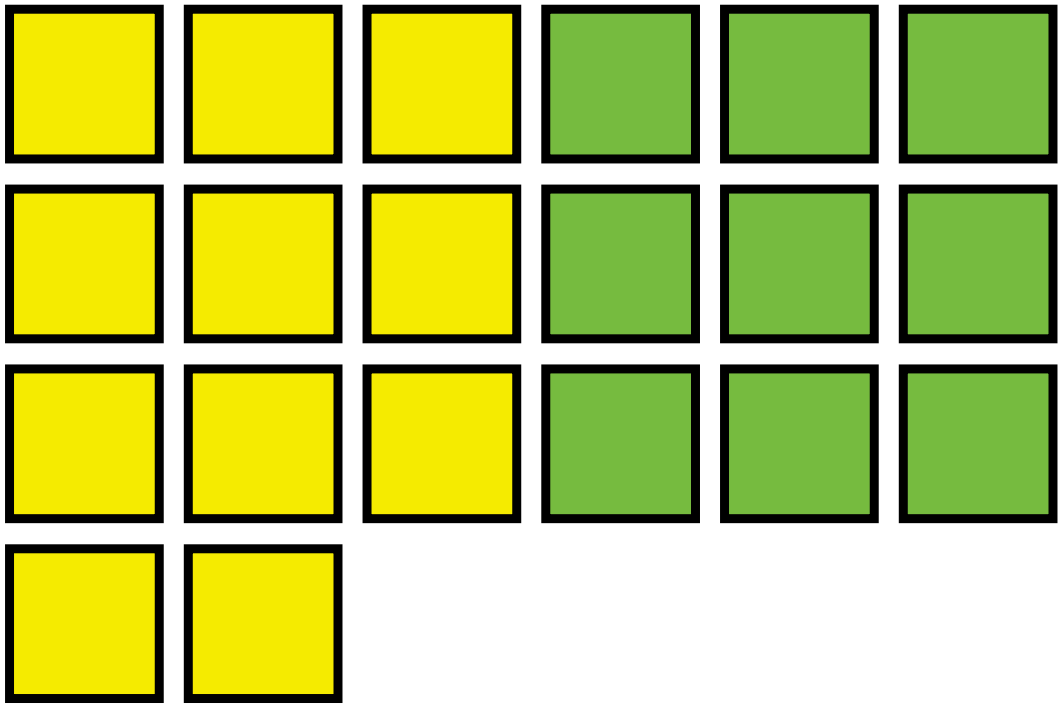
Guessing Game



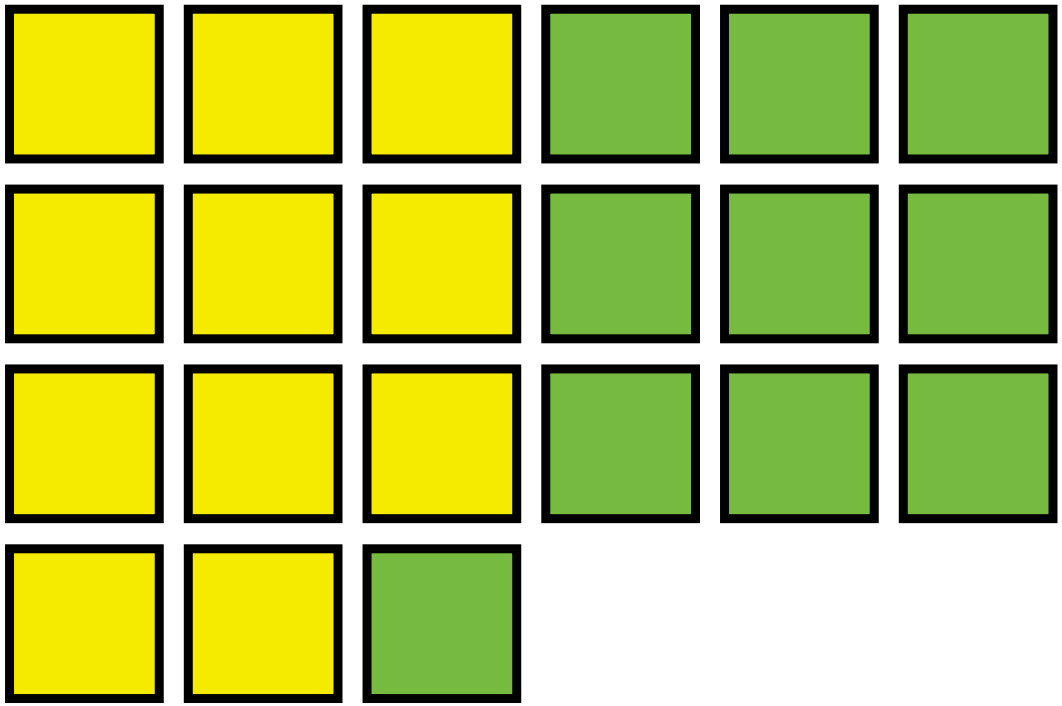
Guessing Game



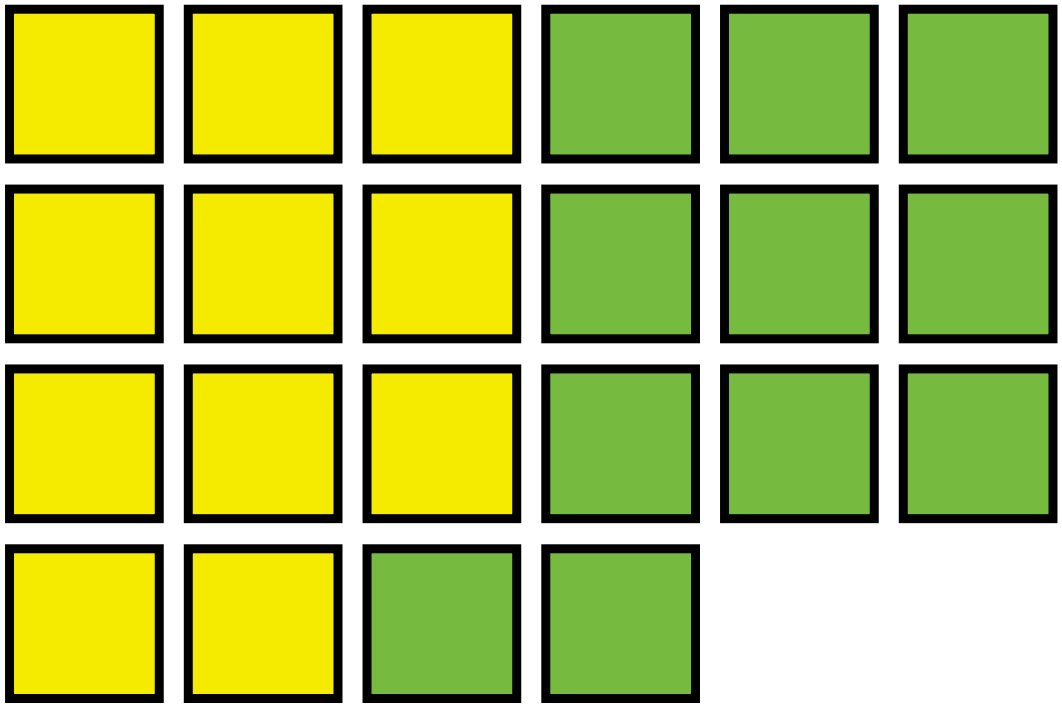
Guessing Game



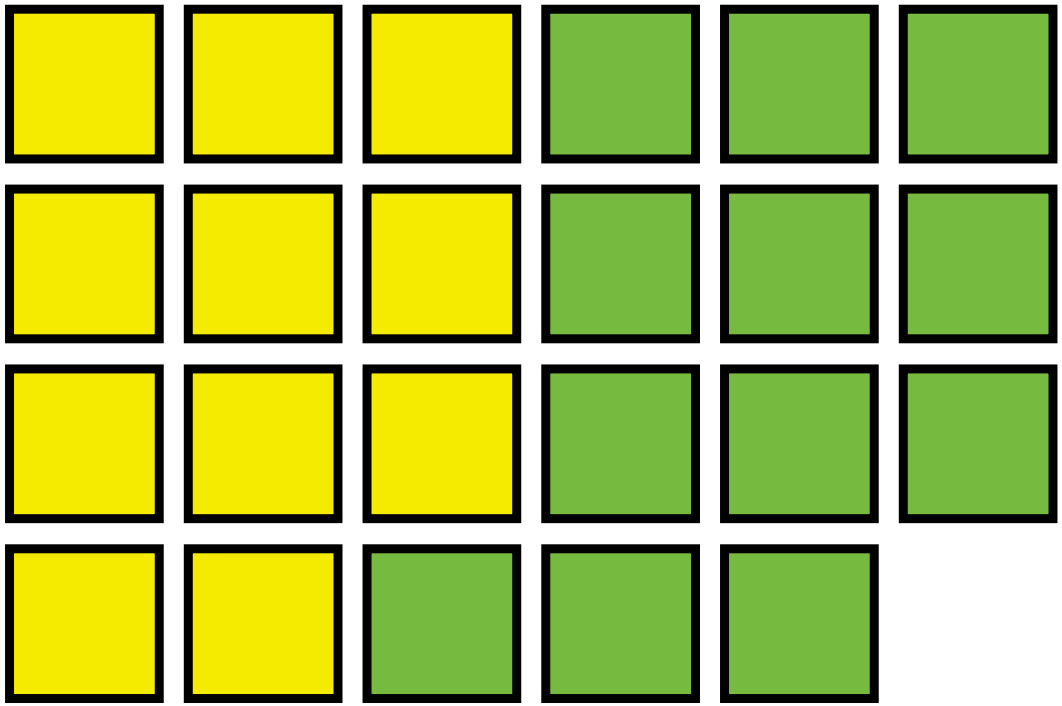
Guessing Game



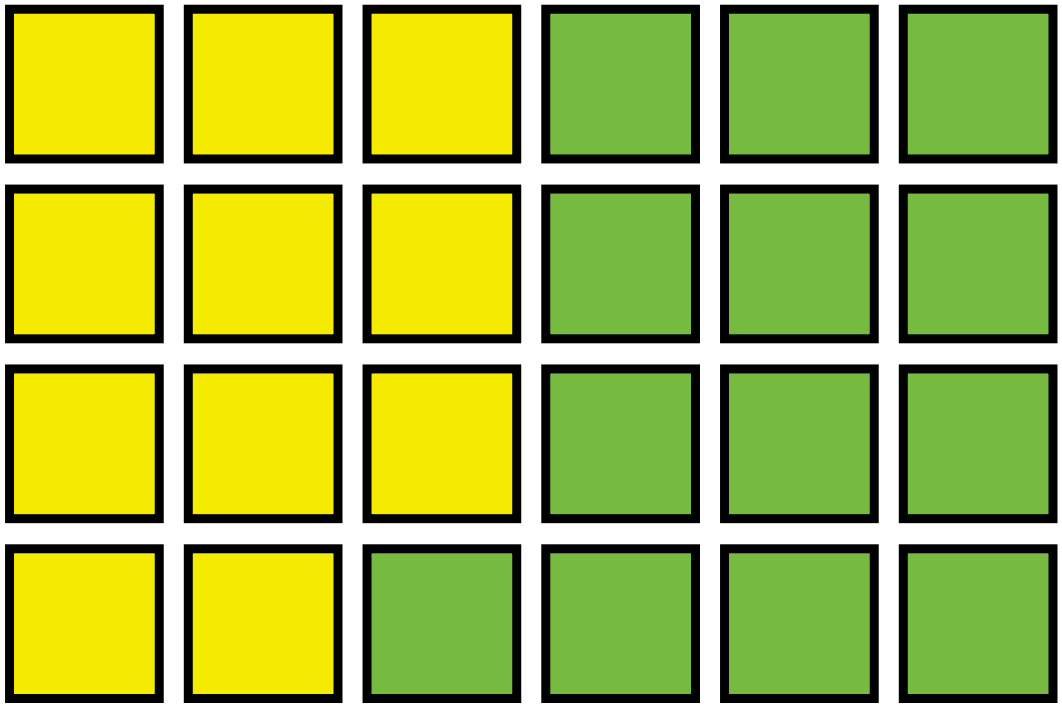
Guessing Game



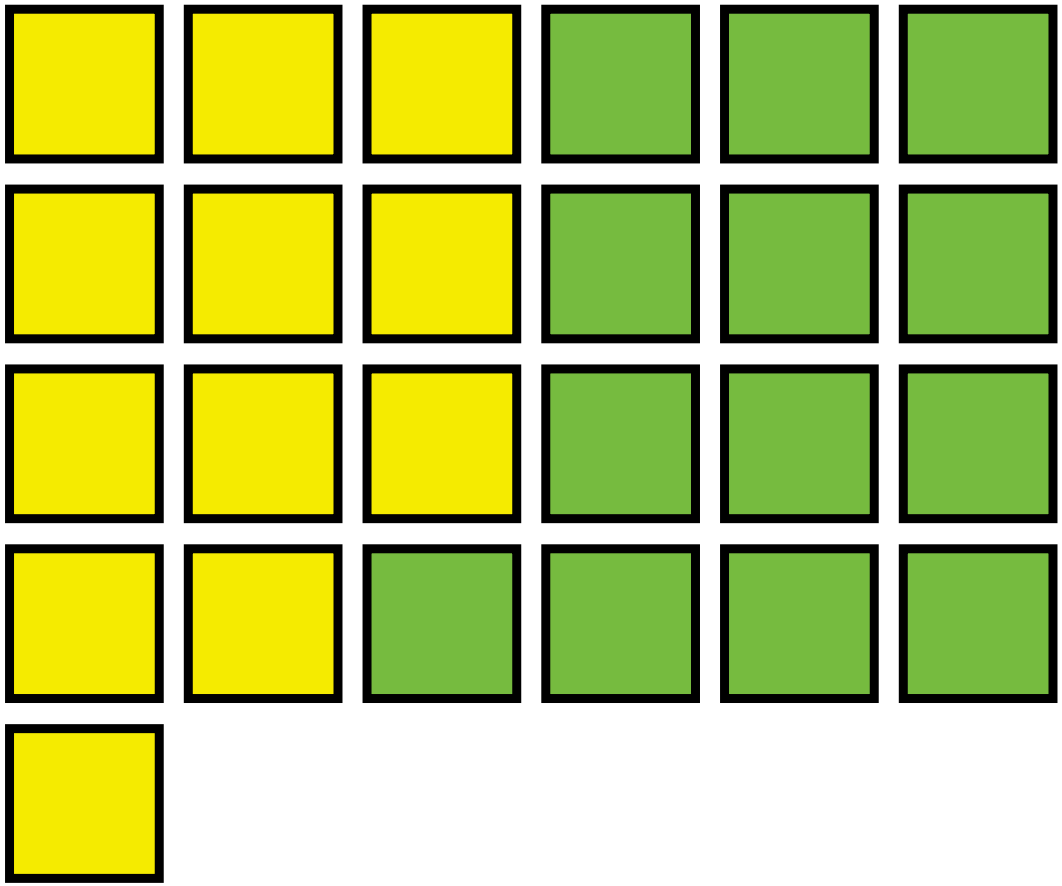
Guessing Game



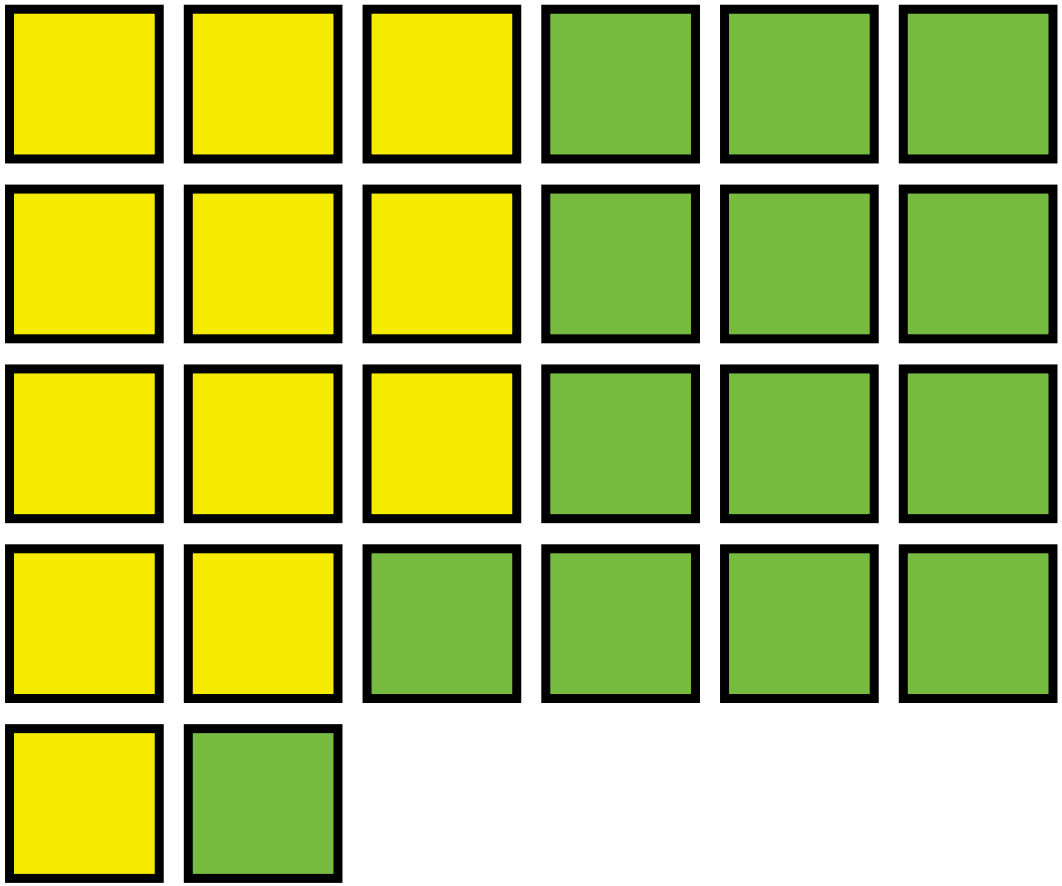
Guessing Game



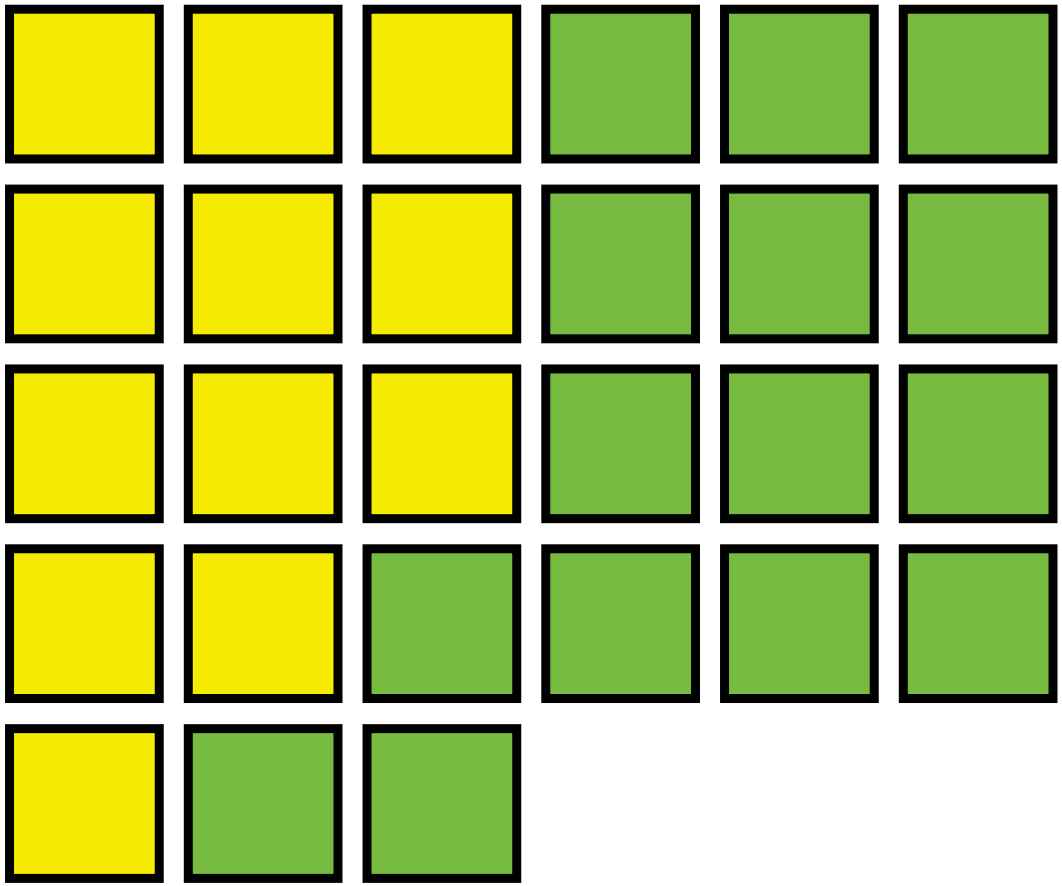
Guessing Game



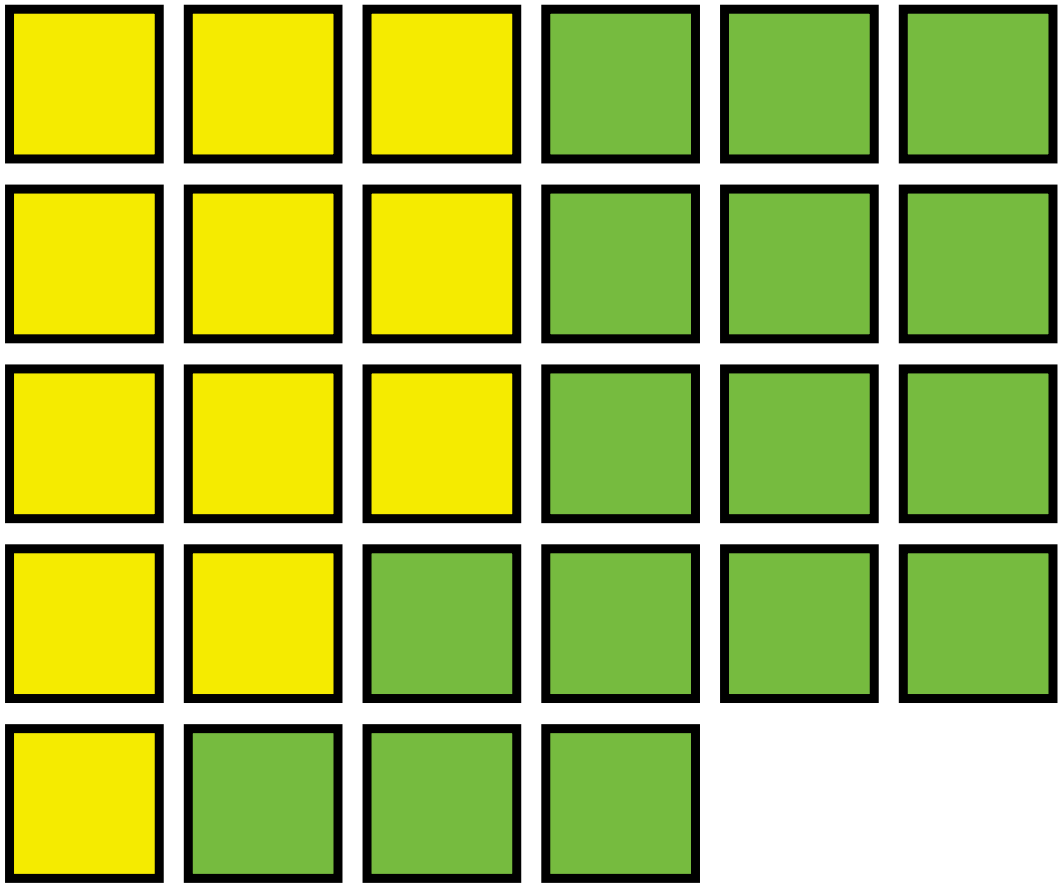
Guessing Game



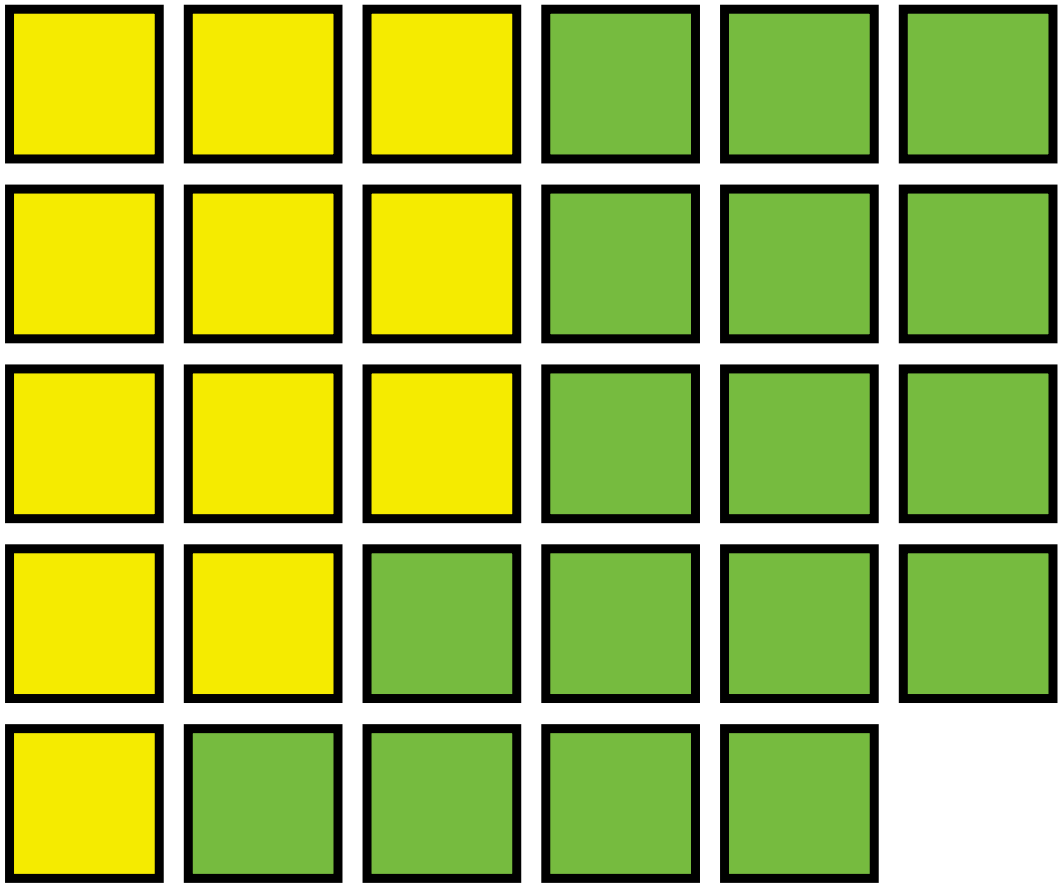
Guessing Game



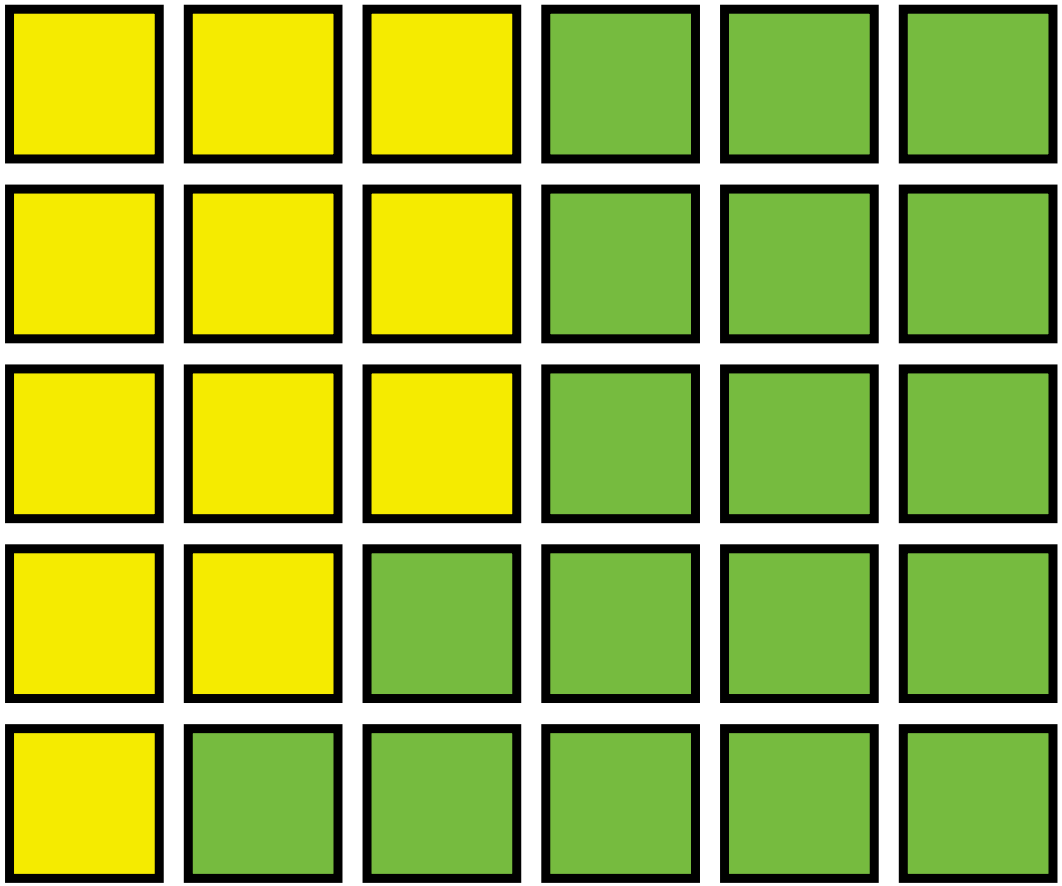
Guessing Game



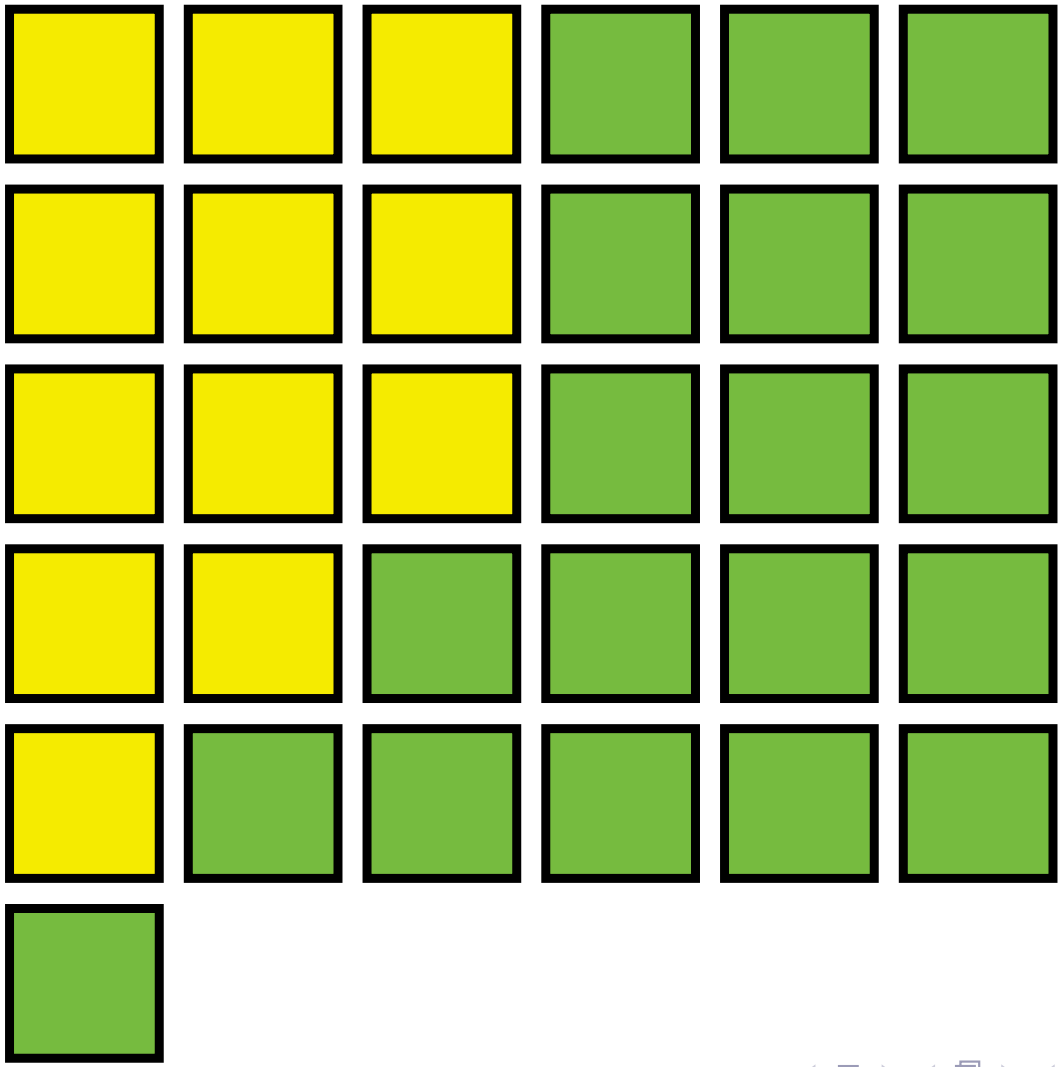
Guessing Game



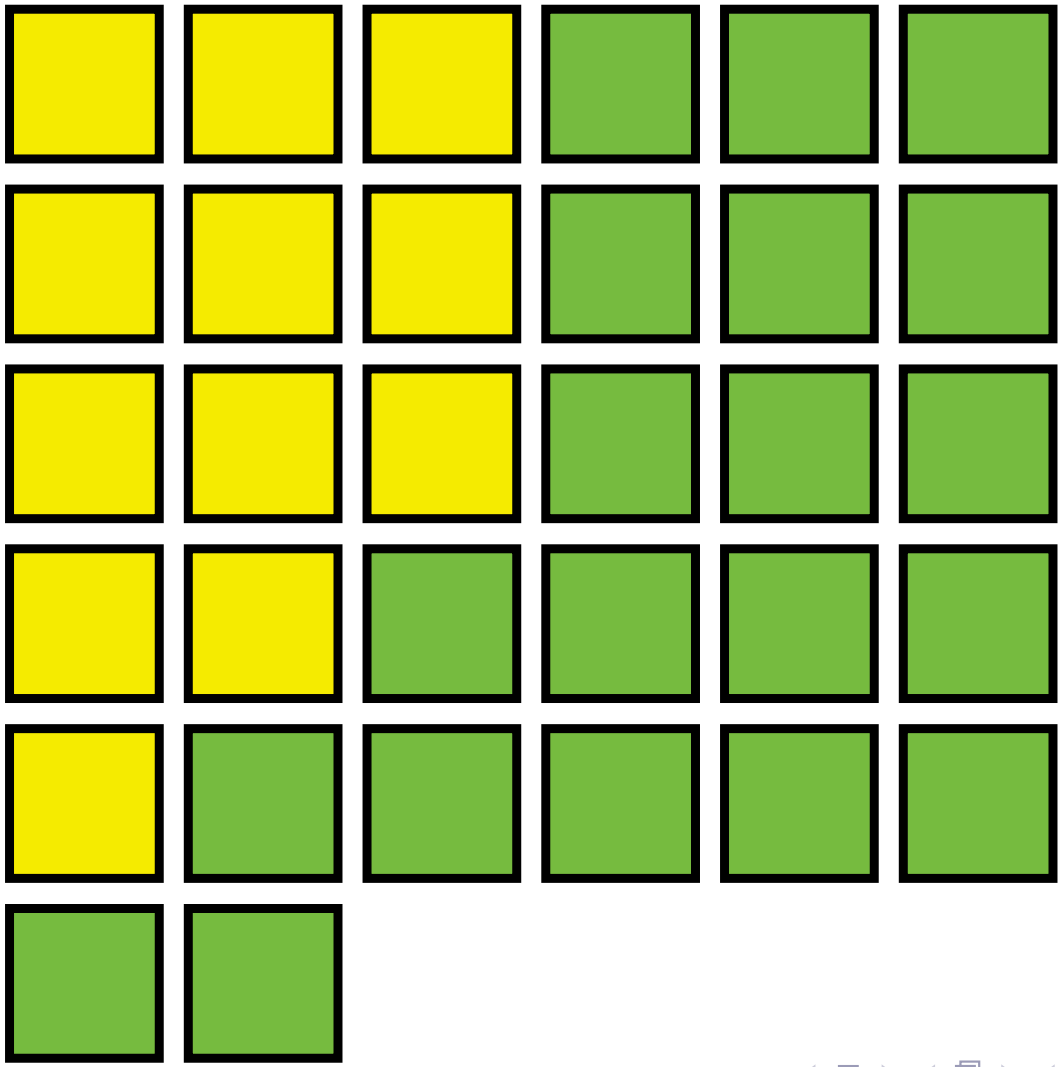
Guessing Game



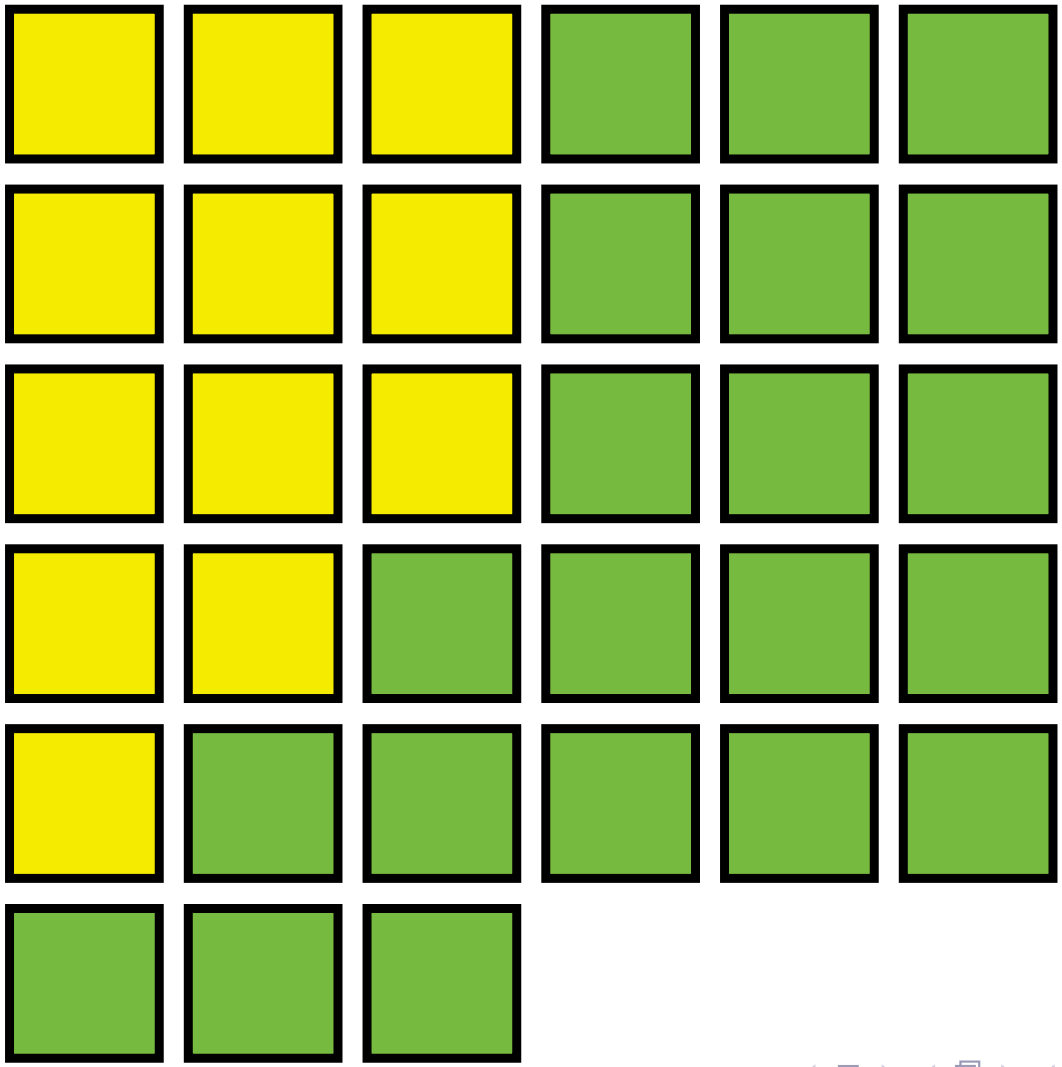
Guessing Game



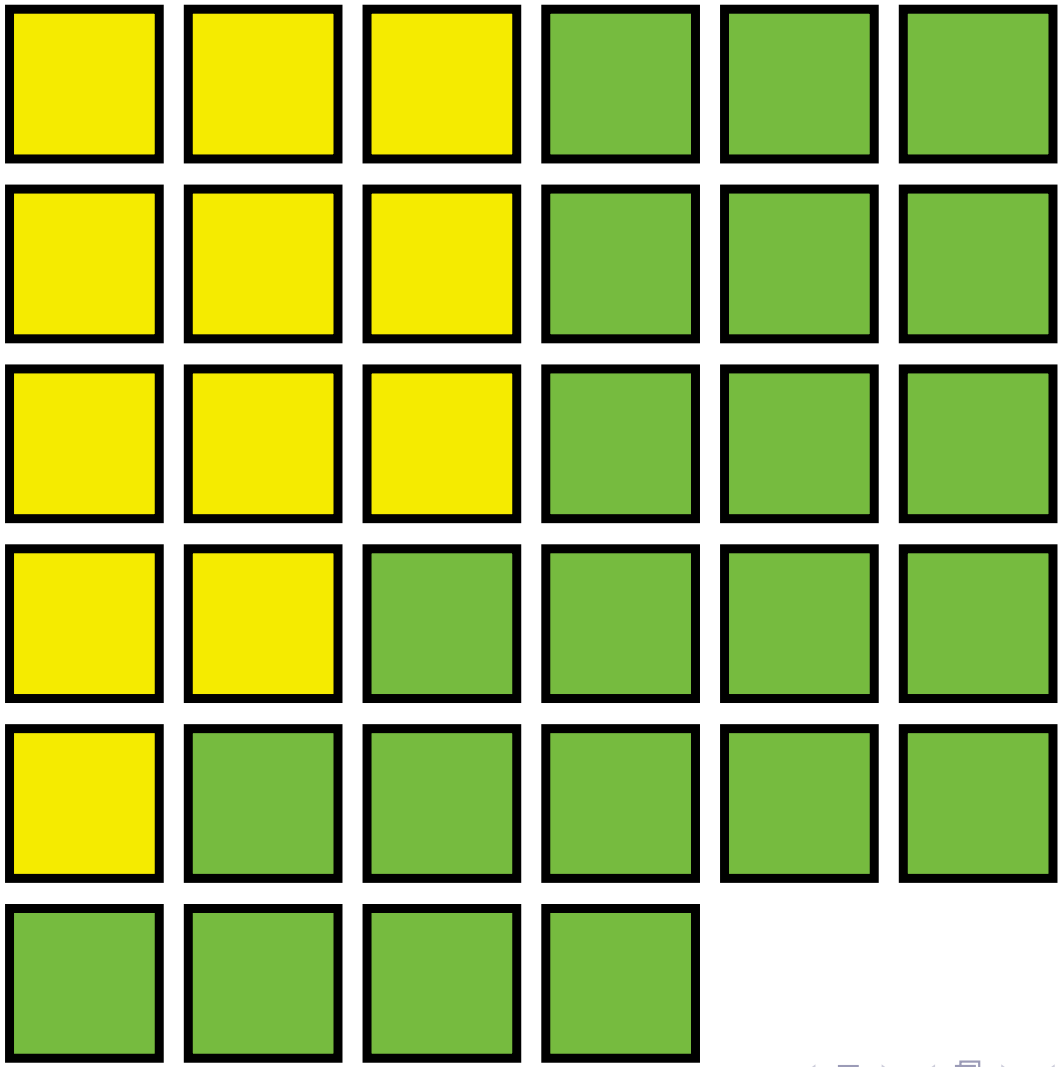
Guessing Game



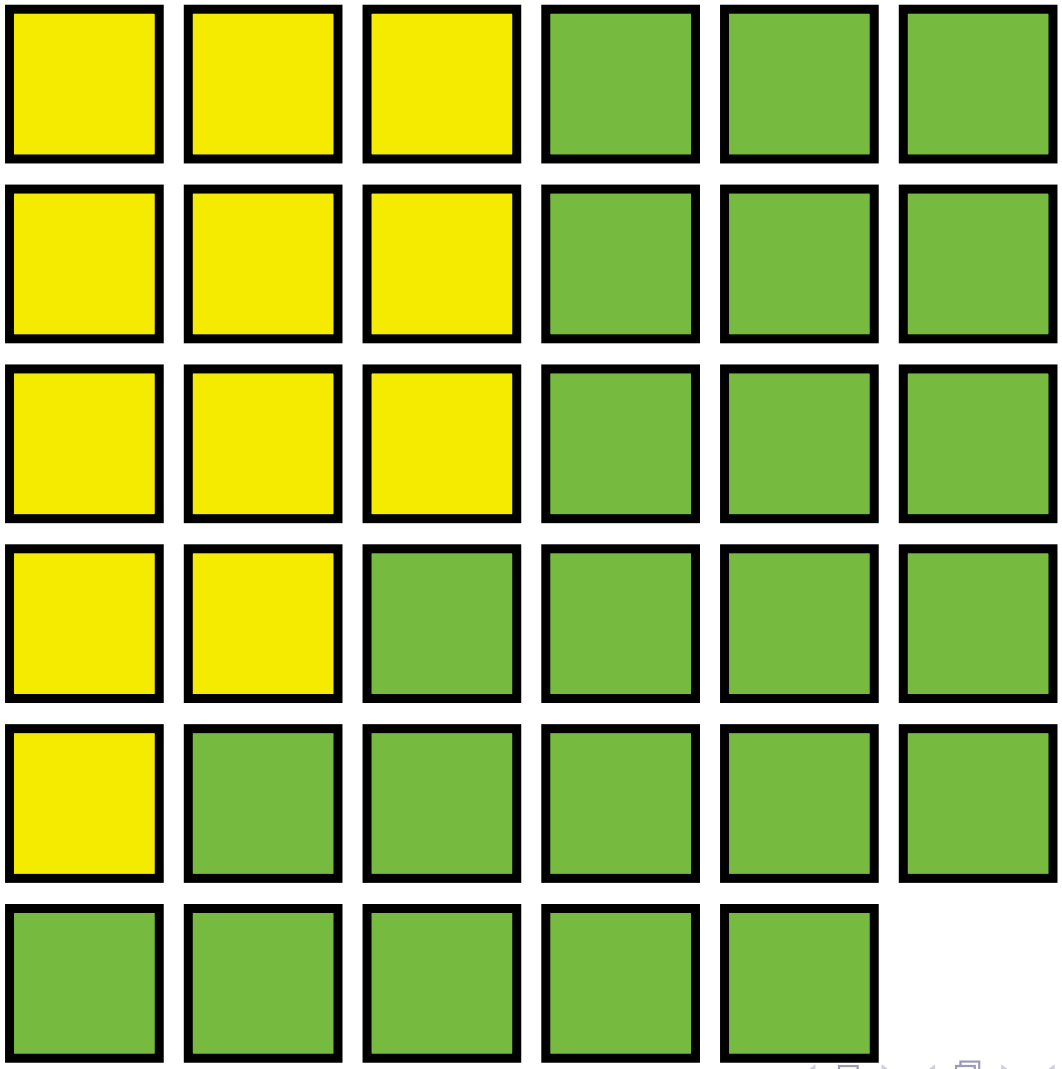
Guessing Game



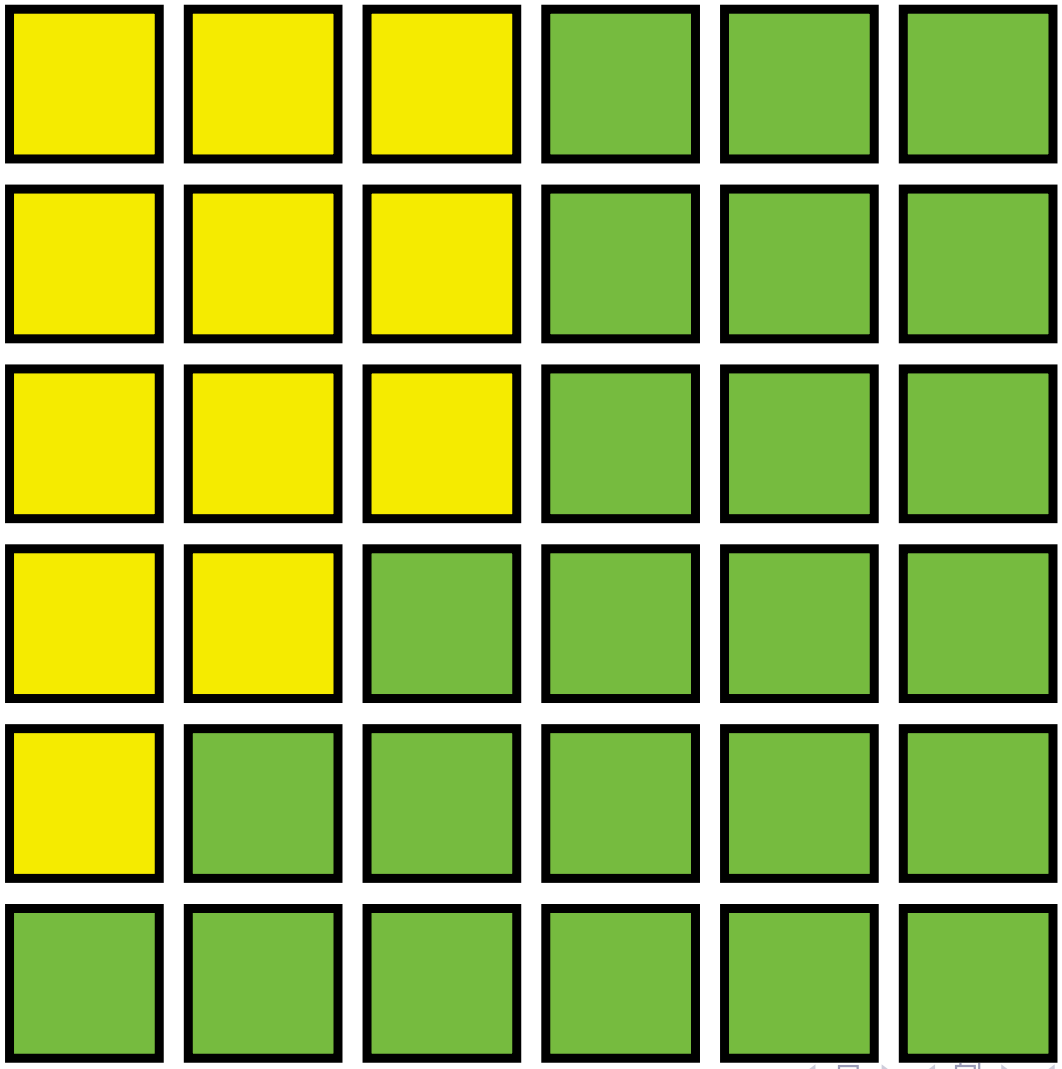
Guessing Game



Guessing Game



Guessing Game



MDL Principle

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.

MDL Principle

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.

$$\min_{h \in \mathcal{H}} (\ell(h) + \ell(D\,;\,h))$$

MDL Principle

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.

$$\min_{h \in \mathcal{H}} (\ell(h) + \ell(D\,;\,h))$$

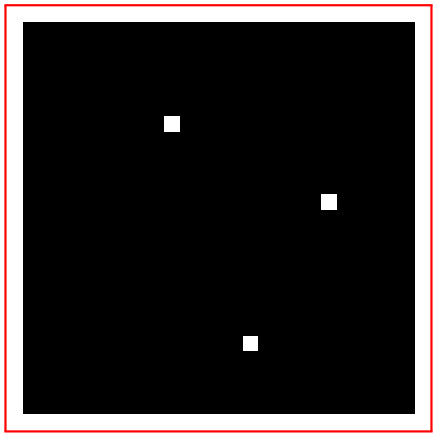
How to encode data *with the help of a hypothesis*?

Encoding Data: Rules & Exceptions

Idea 1: Hypothesis = rule; encode exceptions.

Encoding Data: Rules & Exceptions

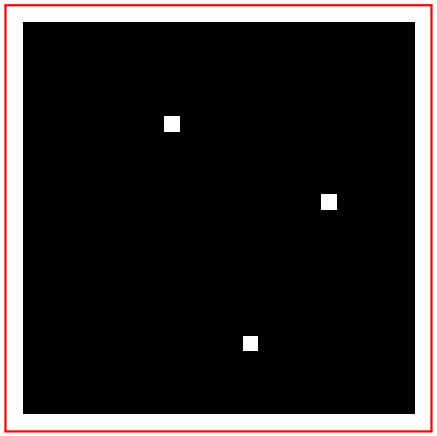
Idea 1: Hypothesis = rule; encode exceptions.



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 $(x_1, y_1), (x_2, y_2), (x_3, y_3)$.

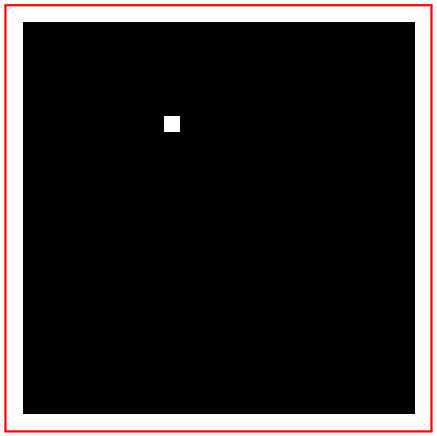
For image of size $n = 625$, there are 2^n different images, and

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

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

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

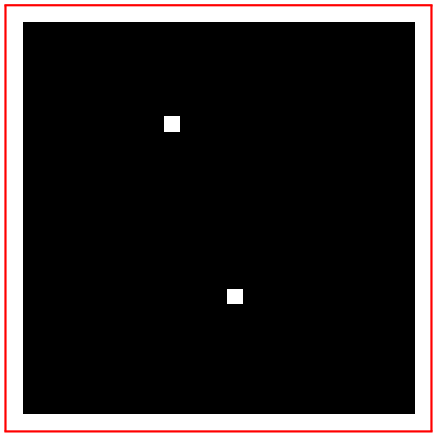
different groups of k exceptions.

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

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 $(x_1, y_1), (x_2, y_2), (x_3, y_3)$.

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

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

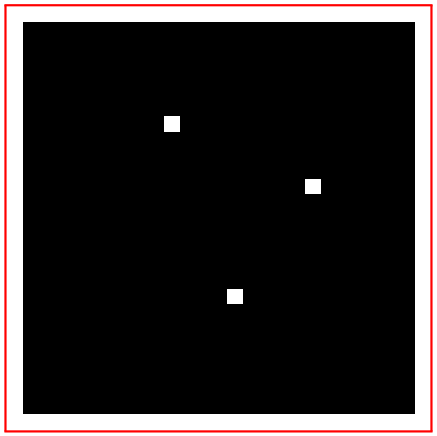
different groups of k exceptions.

$k = 2 : \binom{n}{2} = 195\,000 \ll 2^{625} \approx 1.4 \times 10^{188}.$

Codelength $\log_2(n+1) + \log_2 \binom{n}{k} \approx 27$ 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 $(x_1, y_1), (x_2, y_2), (x_3, y_3)$.

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

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

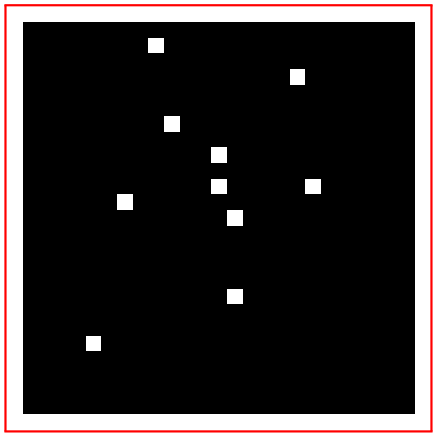
different groups of k exceptions.

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

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 $(x_1, y_1), (x_2, y_2), (x_3, y_3)$.

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

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

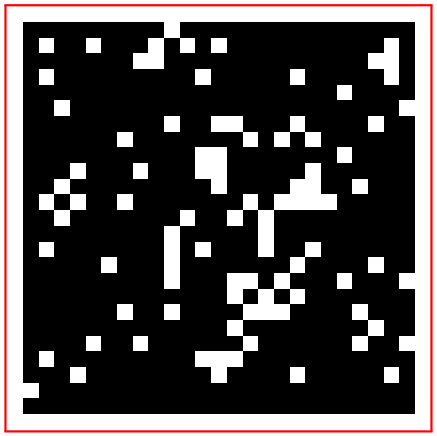
different groups of k exceptions.

$$k = 10 : \binom{n}{10} = 2\,331\,354\,000\,000\,000\,000\,000 \ll 2^{625}.$$

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 $(x_1, y_1), (x_2, y_2), (x_3, y_3)$.

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

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

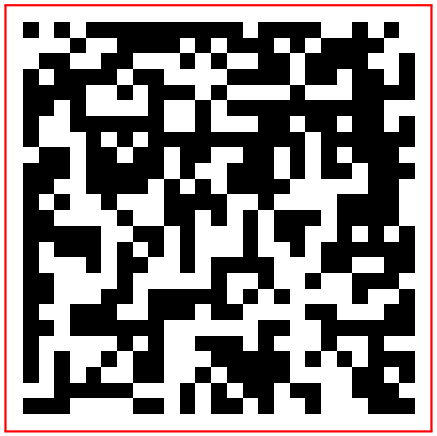
different groups of k exceptions.

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

Codelength $\log_2(n+1) + \log_2 \binom{n}{k} \approx 401$ 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 $(x_1, y_1), (x_2, y_2), (x_3, y_3)$.

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

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

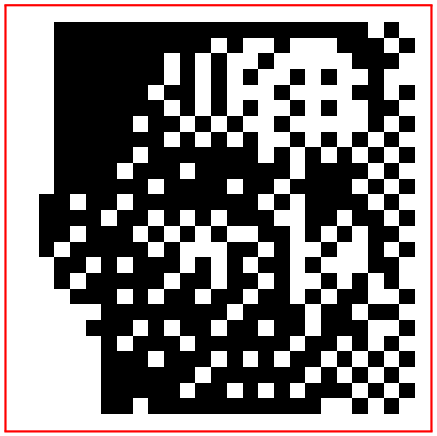
different groups of k exceptions.

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

Codelength $\log_2(n+1) + \log_2 \binom{n}{k} \approx 629$ 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 $(x_1, y_1), (x_2, y_2), (x_3, y_3)$.

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.

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

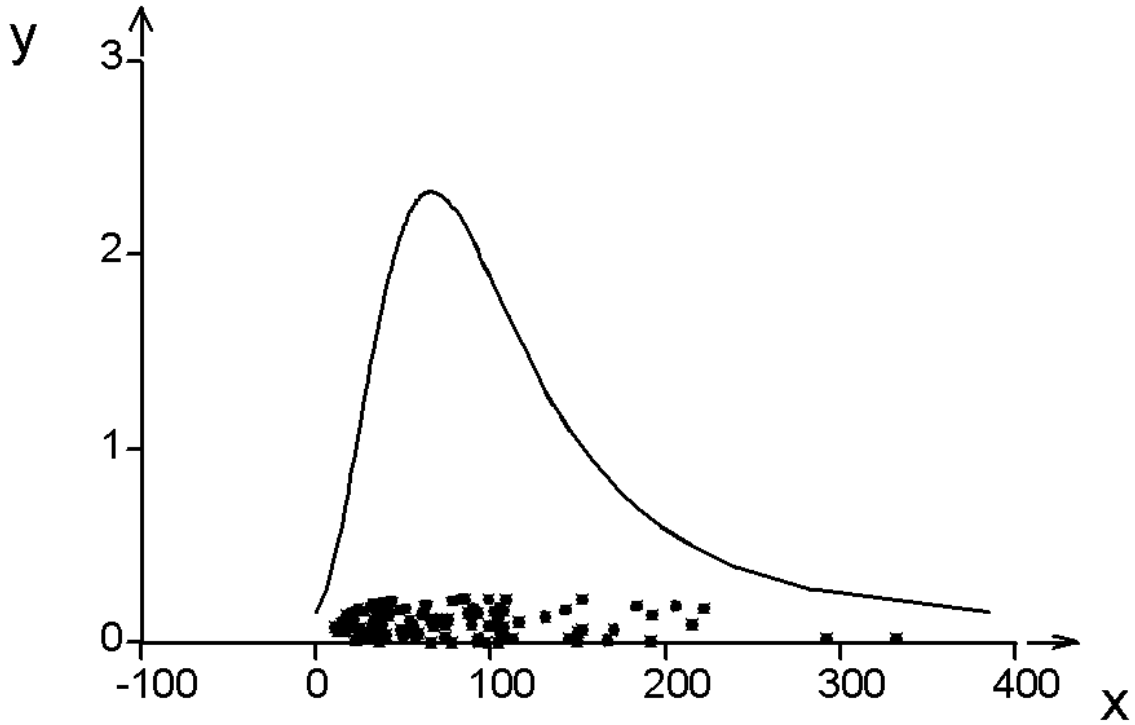
Codelength $\log_2(n+1) + \log_2 \binom{n}{k} \approx 613$ vs. $\log_2 2^{625} = 625$

Encoding Data: Probabilistic Models

Idea 2: Hypothesis = probability distribution.

Encoding Data: Probabilistic Models

Idea 2: Hypothesis = probability distribution.



Encoding Data: Probabilistic Models

Idea 2: Hypothesis = probability distribution.

Important Observation

Probability distributions are codes are probability distributions!

Encoding Data: Probabilistic Models

Idea 2: Hypothesis = probability distribution.

Important Observation

Probability distributions are codes are probability distributions!

The code-length of the data is given by

$$\ell(D\,;\,h) = \log_2 \frac{1}{p_h(D)} \quad .$$

Encoding Data: Probabilistic Models

Idea 2: Hypothesis = probability distribution.

Important Observation

Probability distributions are codes are probability distributions!

The code-length of the data is given by

$$\ell(D\,;\,h) = \log_2 \frac{1}{p_h(D)} \quad .$$

Remember to encode distribution too: $\ell(h) > 0$.

MDL & Bayes

The MDL model selection criterion

$$\text{minimize } \ell(h) + \ell(D\,;\,h)$$

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

$$\text{maximize } p(h) \times p_h(D) \ .$$

MDL & Bayes

The MDL model selection criterion

$$\text{minimize } \ell(h) + \ell(D\,;\,h)$$

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

$$\text{maximize } p(h) \times p_h(D) \ .$$

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

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

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

MDL & Bayes

ENVIRONMETRICS
Environmetrics 2001; **12**: 559–568 (DOI: 10.1002/env.482)

Model selection: Full Bayesian approach

Carlos Alberto de Bragança Pereira*,† and Julio Michael Stern‡

BIOINFO and IME-USP – University of Sao Paulo, Brazil

MDL & Bayes

ENVIRONMETRICS
Environmetrics 2001; **12**: 559–568 (DOI: 10.1002/env.482)

We can use the FBST as a model selection criterion, testing the hypothesis of some of its parameters being null, and using the following version of the 'Ockham razor: Do not include in the model a new parameter unless there is strong evidence it is not null.'

The FBST selection criterion has an intrinsic regularization mechanism, under some general circumstances discussed later.

Carlos Alberto de Bragança Pereira*,† and Julio Michael Stern‡

BIOINFO and IME-USP – University of Sao Paulo, Brazil

MDL & Bayes

ENVIRONMETRICS
Environmetrics 2001; **12**: 559–568 (DOI: 10.1002/env.482)

We can use the FBST as a model selection criterion, testing the hypothesis of some of its parameters being null, and using the following version of the 'Ockham razor: Do not include in the model a new parameter unless there is strong evidence it is not null.'

The FBST selection criterion has an intrinsic regularization mechanism, under some general circumstances discussed later.

Carlos Alberto de Bragança Pereira*,† and Julio Michael Stern‡

BIOINFO and IME-USP – University of Sao Paulo, Brazil

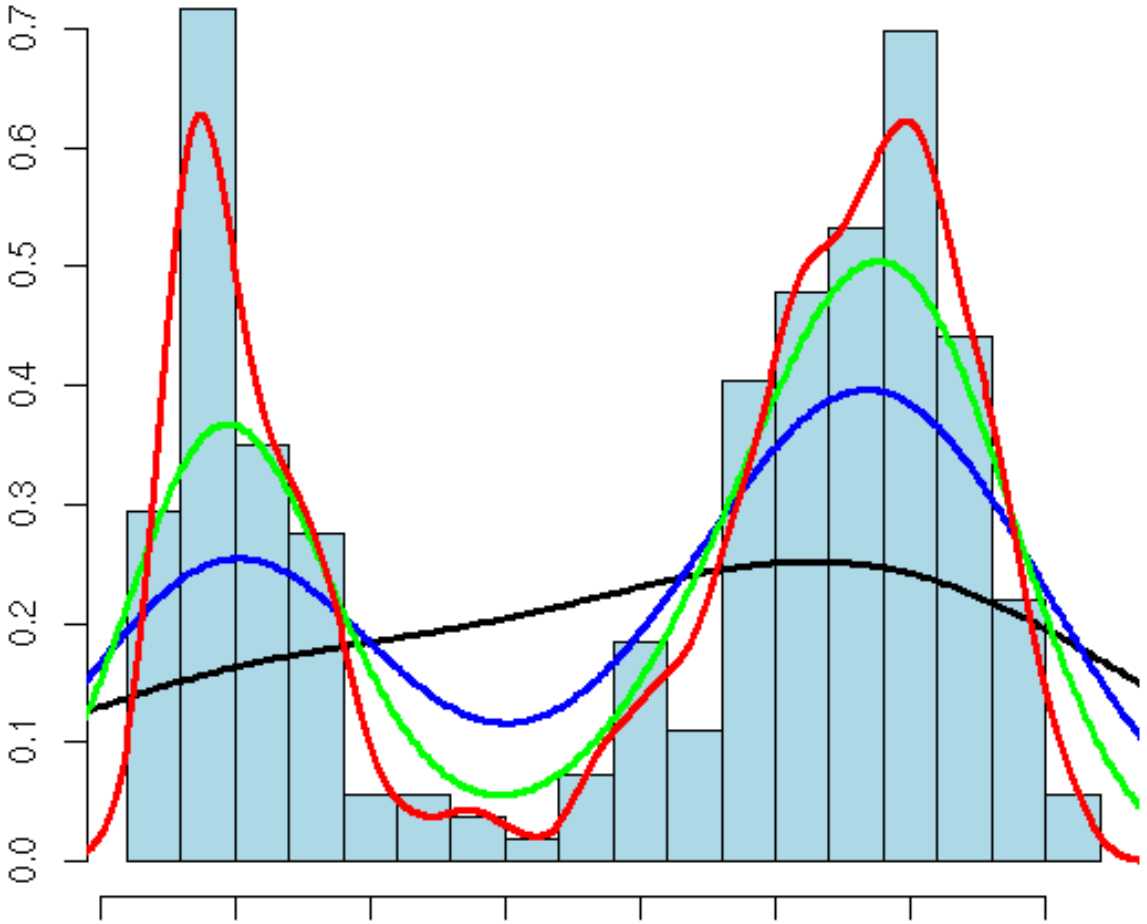
"Do not include in the model a new parameter unless there is strong evidence it is not null."

Encoding Data: Probabilistic Models

Idea 3: Hypothesis = set of probability distributions = model class.

Encoding Data: Probabilistic Models

Idea 3: Hypothesis = set of probability distributions = model class.



Encoding Data: Probabilistic Models

Idea 3: Hypothesis = set of probability distributions = model class.

Universal Coding

A **universal code** achieves almost as short a code-length as the code based on the **best distribution** in the model class.

Encoding Data: Probabilistic Models

Idea 3: Hypothesis = set of probability distributions
= model class.

Universal Coding

A **universal code** achieves almost as short a code-length as the code based on the **best distribution** in the model class.

Different types of universal codes:

Encoding Data: Probabilistic Models

Idea 3: Hypothesis = set of probability distributions = model class.

Universal Coding

A **universal code** achieves almost as short a code-length as the code based on the **best distribution** in the model class.

Different types of universal codes:

1. two-part code,

Encoding Data: Probabilistic Models

Idea 3: Hypothesis = set of probability distributions = model class.

Universal Coding

A **universal code** achieves almost as short a code-length as the code based on the **best distribution** in the model class.

Different types of universal codes:

1. two-part code,
2. mixture code,

Encoding Data: Probabilistic Models

Idea 3: Hypothesis = set of probability distributions = model class.

Universal Coding

A **universal code** achieves almost as short a code-length as the code based on the **best distribution** in the model class.

Different types of universal codes:

1. two-part code,
2. mixture code,
3. normalized maximum likelihood (NML) code.

Universal models

The typical situation might be as follows:

Universal models

The typical situation might be as follows:

1. We know (think) that the source symbols are generated by a Bernoulli model with parameter $p \in [0, 1]$.

Universal models

The typical situation might be as follows:

1. We know (think) that the source symbols are generated by a Bernoulli model with parameter $p \in [0, 1]$.
2. However, we do not know p in advance.

Universal models

The typical situation might be as follows:

1. We know (think) that the source symbols are generated by a Bernoulli model with parameter $p \in [0, 1]$.
2. However, we do not know p in advance.
3. We'd like to encode data at rate $H(p)$.

Two-Part Codes

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

Two-Part Codes

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

For any distribution p_θ, the Shannon code-lengths satisfy

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

Two-Part Codes

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

For any distribution p_θ, the Shannon code-lengths satisfy

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

Using parameter value θ, the total code-length becomes

$$\ell_1(\theta) + \log_2 \frac{1}{p_\theta(D)} \ .$$

Continuous Parameters

What if the parameters are continuous (like polynomial coefficients)?

Continuous Parameters

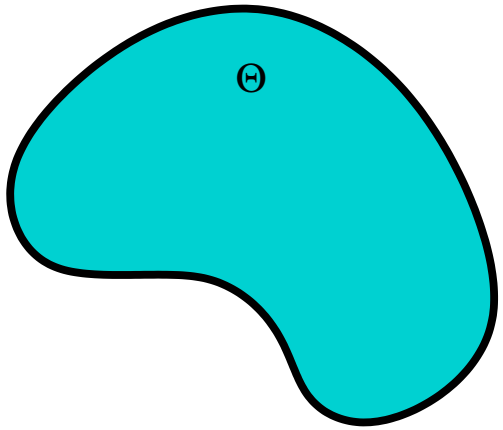
What if the parameters are continuous (like polynomial coefficients)?

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)?

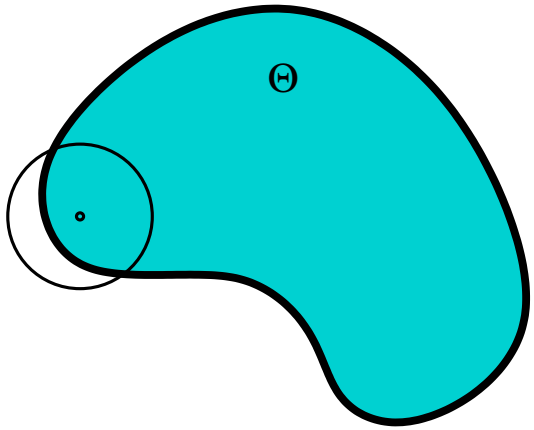
Solution: Quantization. Choose a discrete subset of points, $\theta^{(1)}, \theta^{(2)}, \ldots$, and use only them.



Continuous Parameters

What if the parameters are continuous (like polynomial coefficients)?

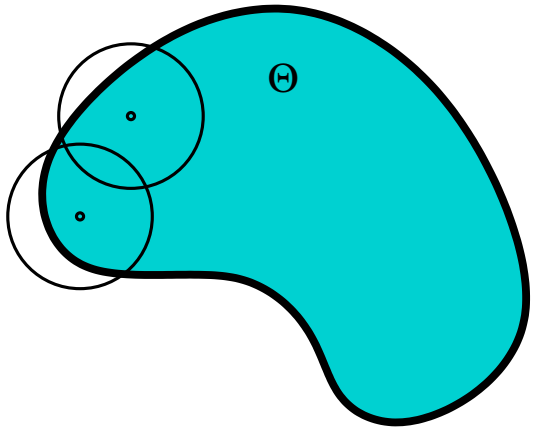
Solution: Quantization. Choose a discrete subset of points, $\theta^{(1)}, \theta^{(2)}, \ldots$, and use only them.



Continuous Parameters

What if the parameters are continuous (like polynomial coefficients)?

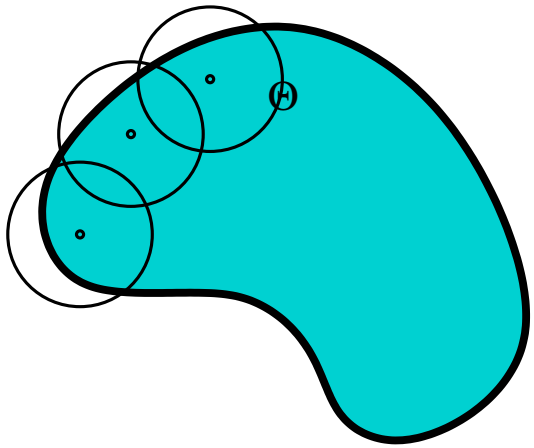
Solution: Quantization. Choose a discrete subset of points, $\theta^{(1)}, \theta^{(2)}, \ldots$, and use only them.



Continuous Parameters

What if the parameters are continuous (like polynomial coefficients)?

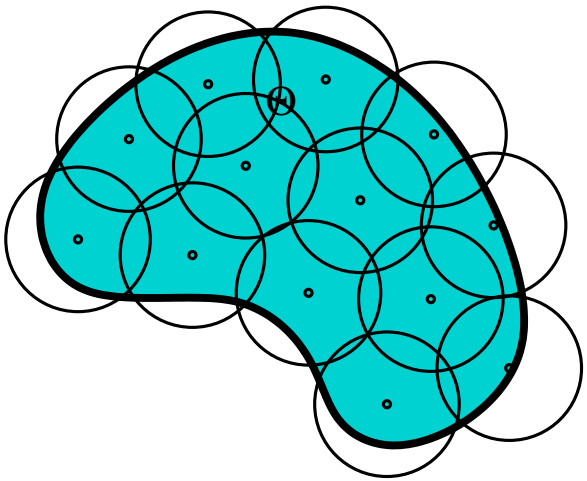
Solution: Quantization. Choose a discrete subset of points, $\theta^{(1)}, \theta^{(2)}, \ldots$, and use only them.



Continuous Parameters

What if the parameters are continuous (like polynomial coefficients)?

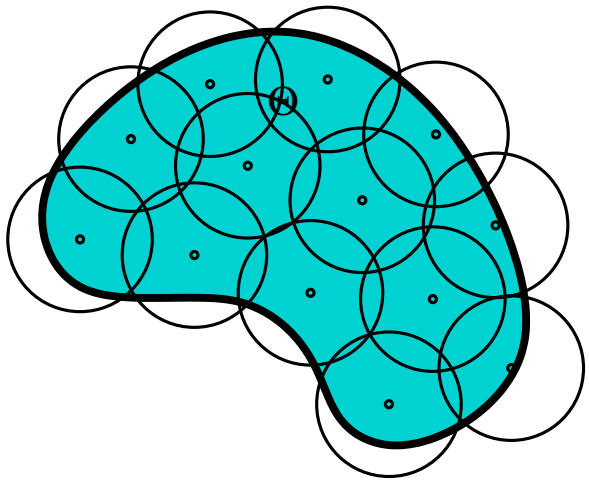
Solution: Quantization. Choose a discrete subset of points, $\theta^{(1)}, \theta^{(2)}, \ldots$, and use only them.



Continuous Parameters

What if the parameters are continuous (like polynomial coefficients)?

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

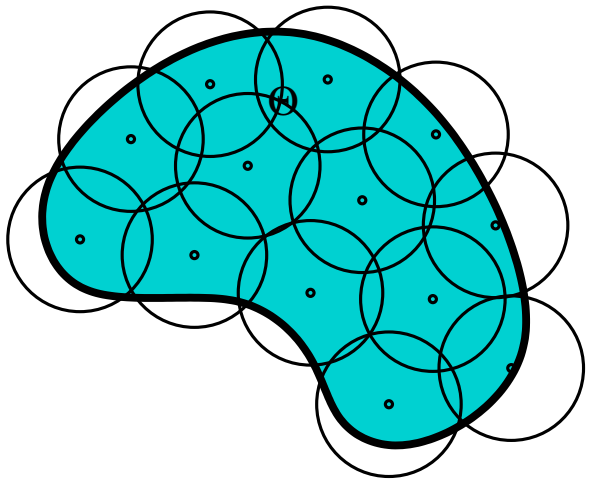


If the points are sufficiently *dense* (in a code-length sense) then the code-length for data is still almost as short as $\min_{\theta \in \Theta} \ell_\theta(D)$.

Continuous Parameters

What if the parameters are continuous (like polynomial coefficients)?

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



Information Geometry!

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

Mixture Universal Model

There are universal codes that are strictly better than the two-part code.

Mixture Universal Model

There are universal codes that are strictly better than the two-part code.

For instance, given a code for the parameters, let w be a distribution over the parameter space Θ (quantized if necessary) defined as

$$w(\theta) = 2^{-\ell(\theta)} \quad .$$

Mixture Universal Model

There are universal codes that are strictly better than the two-part code.

For instance, given a code for the parameters, let w be a distribution over the parameter space Θ (quantized if necessary) defined as

$$w(\theta) = 2^{-\ell(\theta)} \quad .$$

Let p^w be a **mixture distribution** over the data-sets $D \in \mathcal{D}$, defined as

$$p^w(D) = \sum_{\theta \in \Theta} p_\theta(D)\, w(\theta) \quad ,$$

i.e., an "average" distribution, where each p_θ is weighted by $w(\theta)$.

Normalized Maximum Likelihood

Consider again the maximum likelihood model

$$p_{\hat{\theta}}(D) = \max_{\theta \in \Theta} p_\theta(D) \ .$$

It is the best probability assignment achievable under model $\mathcal{M}$.

Normalized Maximum Likelihood

Consider again the maximum likelihood model

$$p_{\hat{\theta}}(D) = \max_{\theta \in \Theta} p_\theta(D) \ .$$

It is the best probability assignment achievable under model $\mathcal{M}$.

Unfortunately, it is not possible to use the ML model for coding because is not a probability distribution, i.e.,

$$C = \sum_{D \in \mathcal{D}} p_{\hat{\theta}}(D) > 1 \ ,$$

unless $\hat{\theta}$ is constant wrt. D.

Normalized Maximum Likelihood

Normalized Maximum Likelihood

The **normalized maximum likelihood (NML) model** is obtained by normalizing the ML model:

$$p_{\mathrm{nml}}(D) = \frac{p_{\hat{\theta}}(D)}{C} \quad , \quad \text{where } C = \sum_{D \in \mathcal{D}} p_{\hat{\theta}}(D) \ .$$

Normalized Maximum Likelihood

Normalized Maximum Likelihood

The **normalized maximum likelihood (NML) model** is obtained by normalizing the ML model:

$$p_{\mathrm{nml}}(D) = \frac{p_{\hat{\theta}}(D)}{C} \quad , \quad \text{where } C = \sum_{D \in \mathcal{D}} p_{\hat{\theta}}(D) \ .$$

NML code-length:

$$\ell(D) = \log_2 \frac{1}{p_{\mathrm{nml}}(D)} = \log_2 \frac{1}{p_{\hat{\theta}}(D)} + \log_2 C \ .$$

Normalized Maximum Likelihood

Normalized Maximum Likelihood

The **normalized maximum likelihood (NML) model** is obtained by normalizing the ML model:

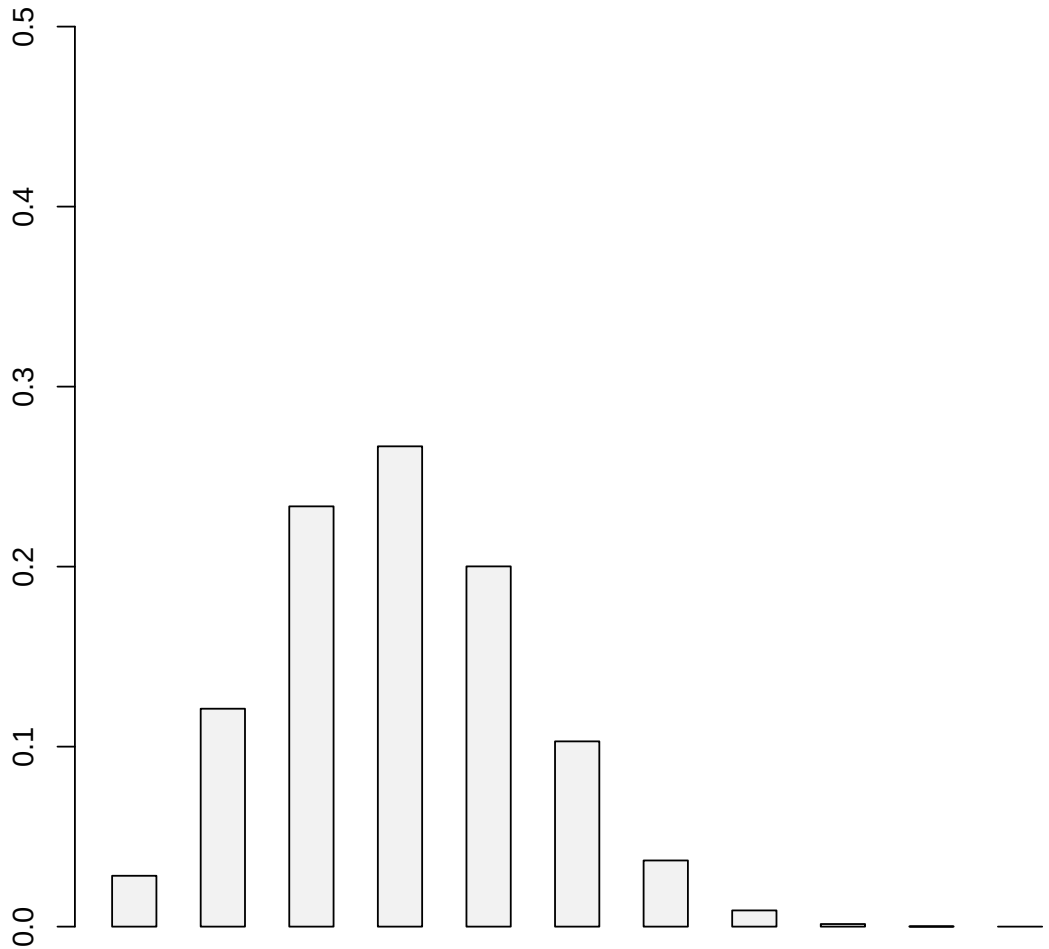
$$p_{\mathrm{nml}}(D) = \frac{p_{\hat{\theta}}(D)}{C} \quad , \quad \text{where } C = \sum_{D \in \mathcal{D}} p_{\hat{\theta}}(D) \ .$$

NML code-length:

$$\ell(D) = \log_2 \frac{1}{p_{\mathrm{nml}}(D)} = \log_2 \frac{1}{p_{\hat{\theta}}(D)} + \log_2 C \ .$$

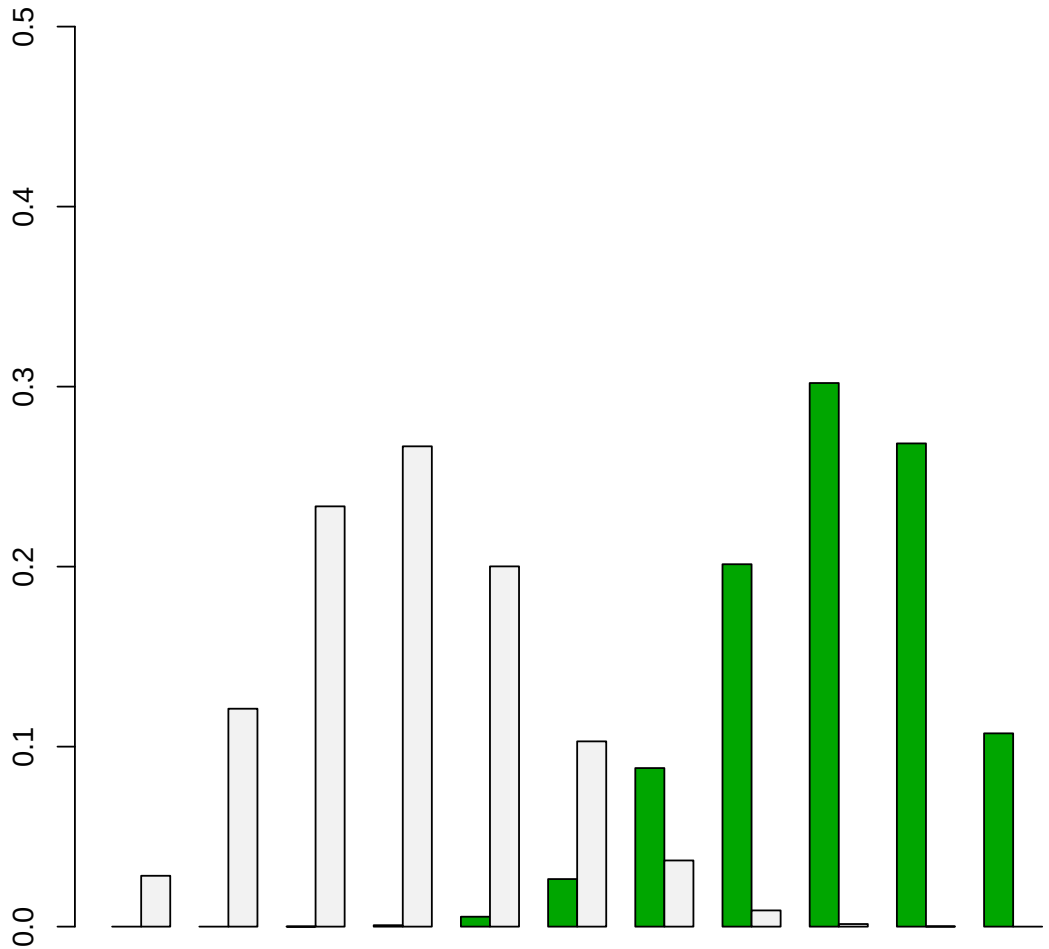
The more flexible (complex) the model class, the greater the normalizing constant.

Normalized Maximum Likelihood



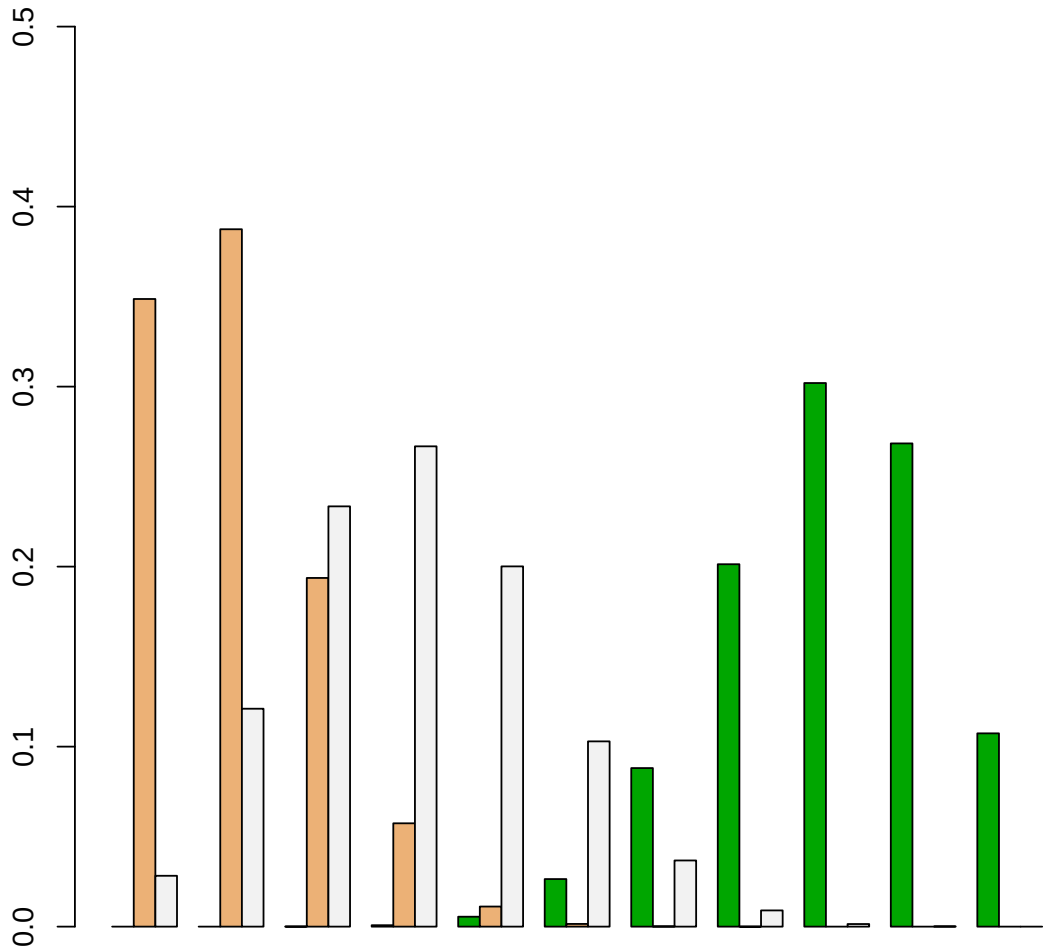
$C = 1.00$

Normalized Maximum Likelihood



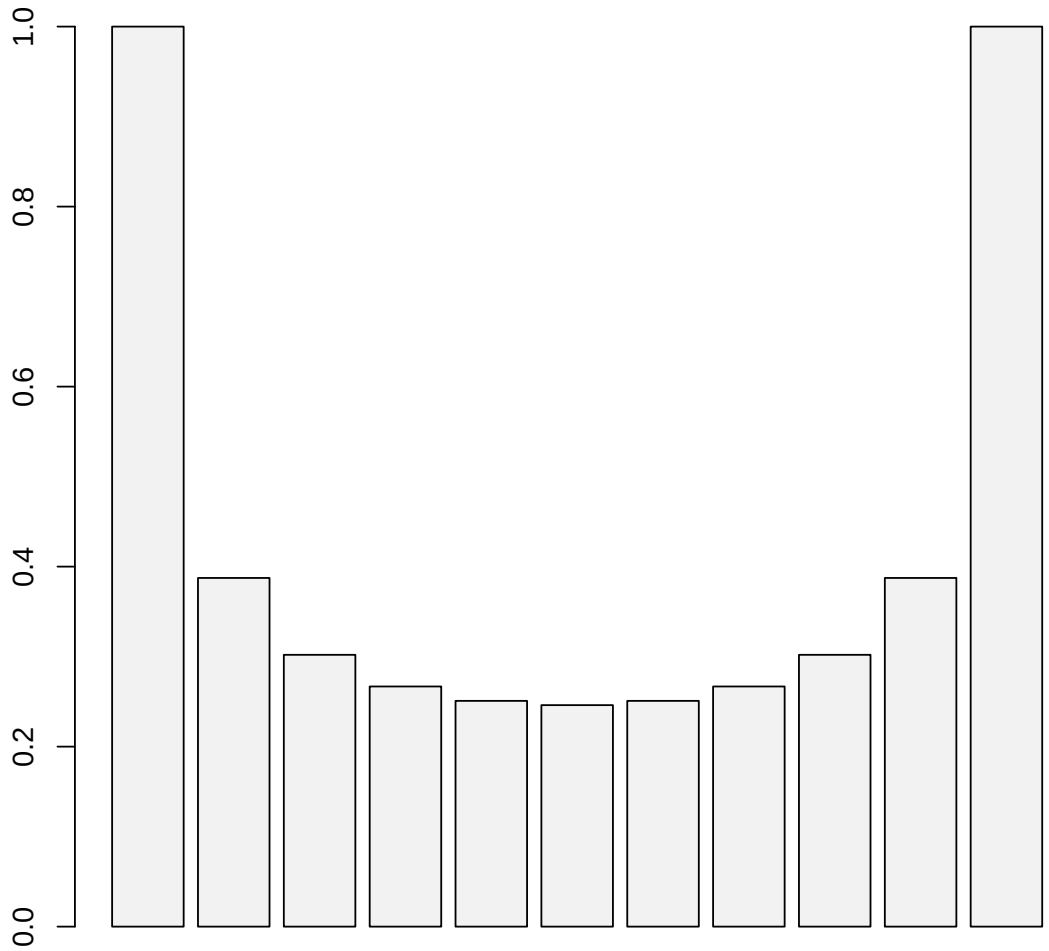
$C = 1.92$

Normalized Maximum Likelihood



$$C = 2.51$$

Normalized Maximum Likelihood



$C = 4.66$

Universal Models

We have seen three kinds of universal codes:

1. two-part,
2. mixture,
3. NML.

Universal Models

We have seen three kinds of universal codes:

1. two-part,
2. mixture,
3. NML.

There are also universal codes that are not based on any (explicit) model class: Lempel-Ziv (`WinZip, gzip`)!

Uses of Universal Codes

So what do we do with them?

Uses of Universal Codes

So what do we do with them?

We can use universal codes for (at least) three purposes:

Uses of Universal Codes

So what do we do with them?

We can use universal codes for (at least) three purposes:

1. compression,

Uses of Universal Codes

So what do we do with them?

We can use universal codes for (at least) three purposes:

1. compression,
2. prediction,

Uses of Universal Codes

So what do we do with them?

We can use universal codes for (at least) three purposes:

1. compression,
2. prediction,
3. model selection.

Universal Prediction

By the connection $p(D) = 2^{-\ell(D)}$, the following are equivalent:

- **good compression:** $\ell(D)$ is small,

Universal Prediction

By the connection $p(D) = 2^{-\ell(D)}$, the following are equivalent:

- **good compression:** $\ell(D)$ is small,
- **good predictions:** $p(D_i \mid D_1, \ldots, D_{i-1})$ is high for most $i \in \{1, \ldots, n\}$.

Universal Prediction

By the connection $p(D) = 2^{-\ell(D)}$, the following are equivalent:

- **good compression:** $\ell(D)$ is small,
- **good predictions:** $p(D_i \mid D_1, \ldots, D_{i-1})$ is high for most $i \in \{1, \ldots, n\}$.

For instance, the mixture code gives a natural predictor which is equivalent to **Bayesian prediction**.

Universal Prediction

By the connection $p(D) = 2^{-\ell(D)}$, the following are equivalent:

- **good compression:** $\ell(D)$ is small,
- **good predictions:** $p(D_i \mid D_1, \ldots, D_{i-1})$ is high for most $i \in \{1, \ldots, n\}$.

For instance, the mixture code gives a natural predictor which is equivalent to **Bayesian prediction**.

The NML model gives predictions that are good relative to the best model in the model class, **no matter what happens**.

Model (Class) Selection

Since a model class that enables good compression of the data must be based on exploiting the **regular features in the data**, the code-length can be used as a **yard-stick** for comparing model classes.

Time for a break?

MDL Principle

MDL Principle

"Old-style":

- Choose the model $p_\theta \in \mathcal{M}$ that yields the shortest *two-part code-length*

$$\min_{\theta,\mathcal{M}} \ \ell(\mathcal{M}) + \ell_1(\theta) + \log_2 \frac{1}{p_\theta(D)}.$$

Modern:

- Choose the model class $\mathcal{M}$ that yields the shortest *universal code-length*

$$\min_{\mathcal{M}} \ \ell(\mathcal{M}) + \ell_{\mathcal{M}}(D).$$

MDL Model Selection

Intuitive Explanation of MDL

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

MDL Model Selection

Intuitive Explanation of MDL

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

We can only extract 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 Model Selection

Intuitive Explanation of MDL

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

We can only extract the structure that is "visible" to the used model class(es). For instance, the Bernoulli (coin flipping) model only sees the number of 1s.

When the model can express the structural properties pertaining to the data *but not more*, the total code-length is minimal.

MDL Model Selection

Intuitive Explanation of MDL

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

We can only extract the structure that is "visible" to the used model class(es). For instance, the Bernoulli (coin flipping) model only sees the number of 1s.

When the model can express the structural properties pertaining to the data *but not more*, the total code-length is minimal.

Important: Too complex models lead to a long total code-length (Occam's Razor!).

Multinomial Models

The multinomial model — the generalization of Bernoulli — is very simple:

$$\Pr(X = j) = \theta_j, \quad \text{for } j \in \{1, \ldots, m\}.$$

Multinomial Models

The multinomial model — the generalization of Bernoulli — is very simple:

$$\Pr(X = j) = \theta_j, \quad \text{for } j \in \{1, \ldots, m\}.$$

Maximum likelihood:

$$\hat{\theta}_j = \frac{\#\{x_i = j\}}{n}.$$

Multinomial Models

The multinomial model — the generalization of Bernoulli — is very simple:

$$\Pr(X = j) = \theta_j, \quad \text{for } j \in \{1, \ldots, m\}.$$

Maximum likelihood:

$$\hat{\theta}_j = \frac{\#\{x_i = j\}}{n}.$$

Two-part, mixture, and NML models readily defined.

Multinomial Models

The multinomial model — the generalization of Bernoulli — is very simple:

$$\Pr(X = j) = \theta_j, \quad \text{for } j \in \{1, \ldots, m\}.$$

Maximum likelihood:

$$\hat{\theta}_j = \frac{\#\{x_i = j\}}{n}.$$

Two-part, mixture, and NML models readily defined.
$\Rightarrow$ Exercises 7–9.

Fast NML for Multinomials

The naïve way to compute the normalizing constant in the NML model

$$\frac{p_{\hat{\theta}}(x^n)}{C_n^m}, \qquad C_n^m = \sum_{y^n \in \mathcal{X}^n} p_{\hat{\theta}}(y^n),$$

takes exponential time ($\Omega(m^n)$).

Fast NML for Multinomials

The naïve way to compute the normalizing constant in the NML model

$$\frac{p_{\hat{\theta}}(x^n)}{C_n^m}, \qquad C_n^m = \sum_{y^n \in \mathcal{X}^n} p_{\hat{\theta}}(y^n),$$

takes exponential time ($\Omega(m^n)$).

However, there is a trick (Myllymäki & Kontkanen, 2007) to do it in linear time.

Fast NML for Multinomials

The naïve way to compute the normalizing constant in the NML model

$$\frac{p_{\hat{\theta}}(x^n)}{C_n^m}, \qquad C_n^m = \sum_{y^n \in \mathcal{X}^n} p_{\hat{\theta}}(y^n),$$

takes exponential time ($\Omega(m^n)$).

However, there is a trick (Myllymäki & Kontkanen, 2007) to do it in linear time.

Kontkanen & Myllymäki, "A linear-time algorithm for computing the multinomial stochastic complexity", *Information Processing Letters* **103** (2007), 6, pp. 227–233

Histogram Density Estimation

A histogram density is defined by

- The break-points between the *bins*,

Histogram Density Estimation

A histogram density is defined by

- The break-points between the *bins*,
- The heights of the bins.

Histogram Density Estimation

A histogram density is defined by

- The break-points between the *bins*,
- The heights of the bins.

Choosing the number *and the positions* of break-points can be done by MDL.

Histogram Density Estimation

A histogram density is defined by

- The break-points between the *bins*,
- The heights of the bins.

Choosing the number *and the positions* of break-points can be done by MDL.

The code-length is equivalent (up to additive constants) to the code-length in a multinomial model.

Histogram Density Estimation

A histogram density is defined by

- The break-points between the *bins*,
- The heights of the bins.

Choosing the number *and the positions* of break-points can be done by MDL.

The code-length is equivalent (up to additive constants) to the code-length in a multinomial model.
$\Rightarrow$ Linear-time algorithm can be used.

Histogram Density Estimation

MDL Histogram Density Estimation

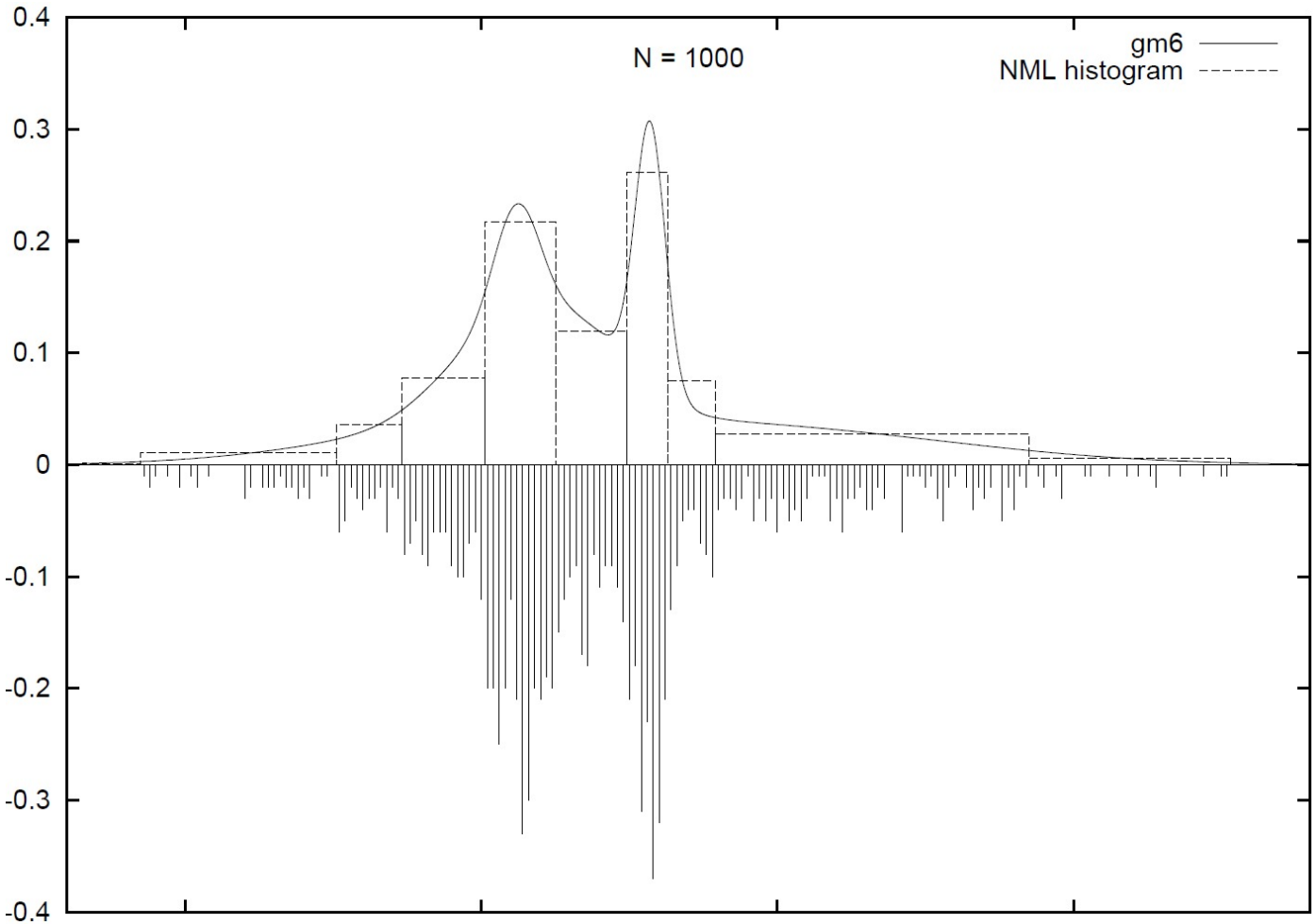
Petri Kontkanen, Petri Myllymäki
Complex Systems Computation Group (CoSCo)
Helsinki Institute for Information Technology (HIIT)
University of Helsinki and Helsinki University of Technology
P.O.Box 68 (Department of Computer Science)
FIN-00014 University of Helsinki, Finland
{Firstname}.{Lastname}@hiit.fi

Abstract

We regard histogram density estimation as a model selection problem. Our approach is based on the information-theoretic minimum description length (MDL) principle, which can be applied for tasks such as data clustering, density estimation, image denois-

only on finding the optimal bin count. These *regular* histograms are, however, often problematic. It has been argued (Rissanen, Speed, & Yu, 1992) that regular histograms are only good for describing roughly uniform data. If the data distribution is strongly non-uniform, the bin count must necessarily be high if one wants to capture the details of the high density portion of the data. This in turn means that an unnecessary large amount of bins is wasted in the low density re-

Histogram Density Estimation



Clustering

Consider the problem of clustering vectors of (independent) multinomial variables.

Clustering

Consider the problem of clustering vectors of (independent) multinomial variables.

This can be seen as a way to encode (compress) the data:

Clustering

Consider the problem of clustering vectors of (independent) multinomial variables.

This can be seen as a way to encode (compress) the data:

1. first encode the cluster index of each observation vector,

Clustering

Consider the problem of clustering vectors of (independent) multinomial variables.

This can be seen as a way to encode (compress) the data:

1. first encode the cluster index of each observation vector,
2. then encode the observations using separate (multinomial) models.

Clustering

Consider the problem of clustering vectors of (independent) multinomial variables.

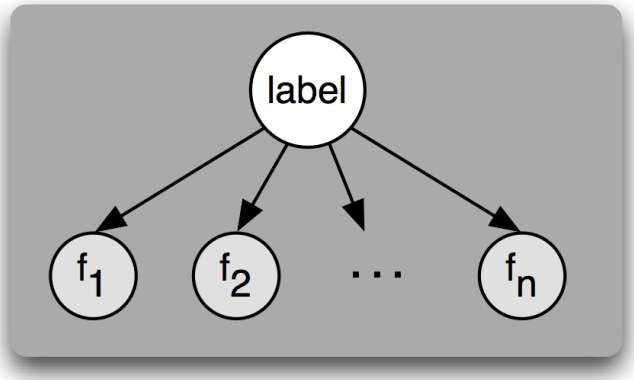
This can be seen as a way to encode (compress) the data:

1. first encode the cluster index of each observation vector,
2. then encode the observations using separate (multinomial) models.

Again, the problem is reduced to the multinomial case, and the fast NML algorithm can be applied.

Clustering

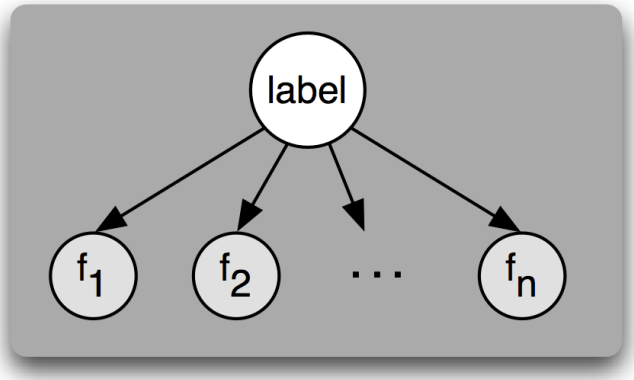
The clustering model can be interpreted as the **naïve Bayes** structure:



label = cluster index $\qquad$ $f_1, \ldots, f_n$ are *features*

Clustering

The clustering model can be interpreted as the **naïve Bayes** structure:

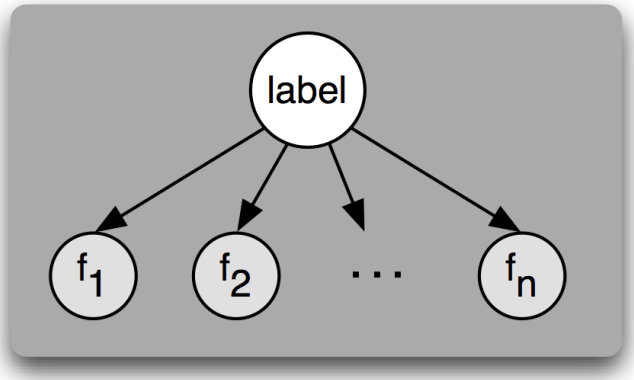


label = cluster index $\qquad$ $f_1, \ldots, f_n$ are *features*

The structure is very restrictive. Generalization achieved by **Bayesian networks**.

Clustering

The clustering model can be interpreted as the **naïve Bayes** structure:

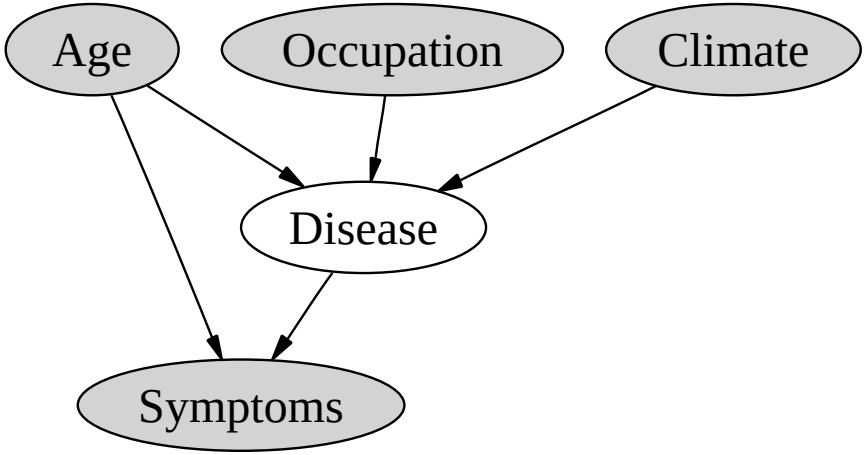


label = cluster index $\qquad$ $f_1, \ldots, f_n$ are *features*

The structure is very restrictive. Generalization achieved by **Bayesian networks**.

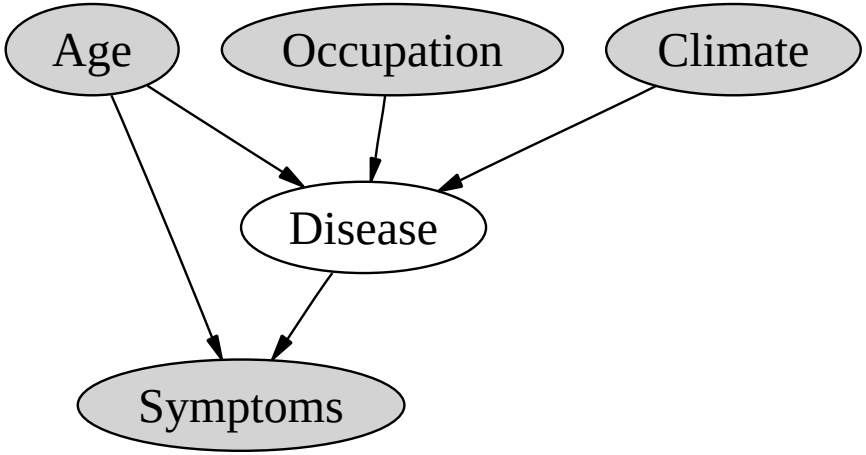
MDL criterion for learning Bayesian network structures again based on *fast NML for multinomials*.

Bayesian Networks



Directed asyclic graph (DAG) describing conditional independencies (causality?).

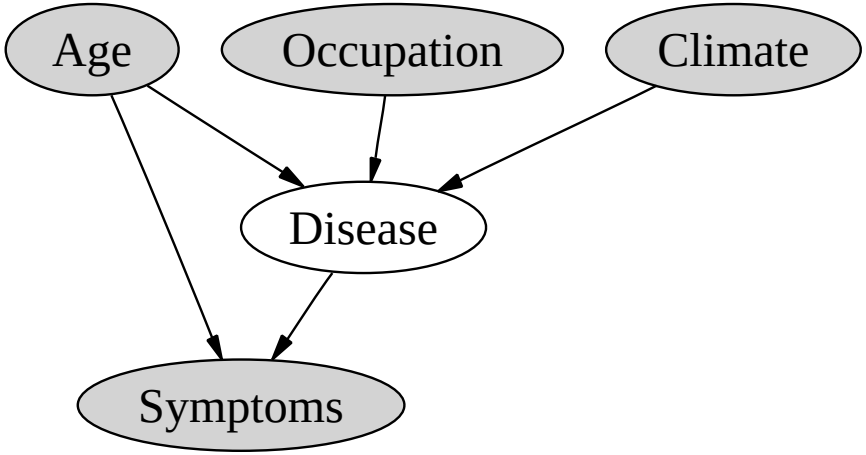
Bayesian Networks



Directed asyclic graph (DAG) describing conditional independencies (causality?).

Denote parents by $\mathrm{Pa}_i \subset \{X_1, \ldots, X_m\} \backslash X_i$.

Bayesian Networks



Directed asyclic graph (DAG) describing conditional independencies (causality?).

Denote parents by $\mathrm{Pa}_i \subset \{X_1, \ldots, X_m\} \backslash X_i$.

Joint probability

$$p(x_1, \ldots, x_m) = \prod_{j=1}^{m} p(x_i \mid \mathrm{pa}_i).$$

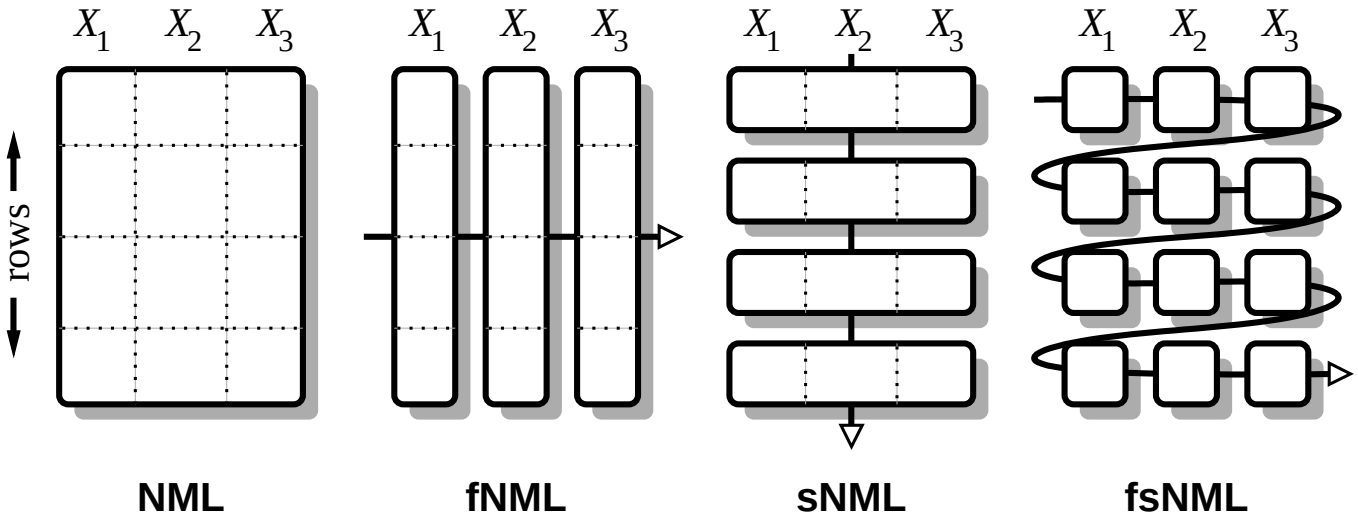
Bayesian Networks

Problem with NML for Bayesian networks is the summation over all possible data-sets: exponential complexity.

Bayesian Networks

Problem with NML for Bayesian networks is the summation over all possible data-sets: exponential complexity.

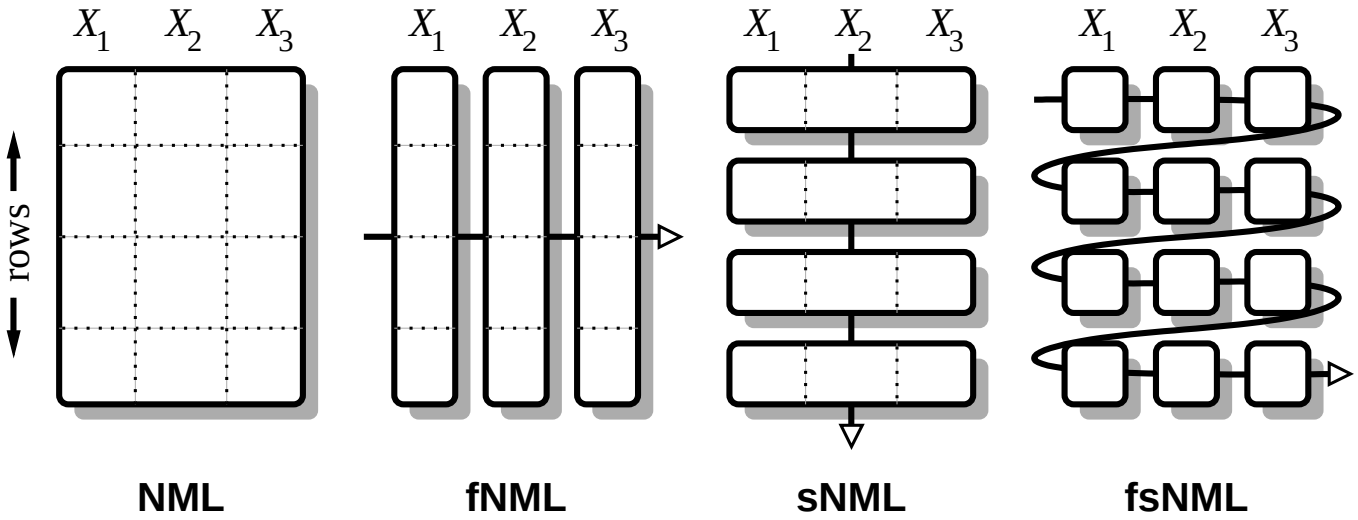
Approximation: normalization over smaller blocks at a time.



Bayesian Networks

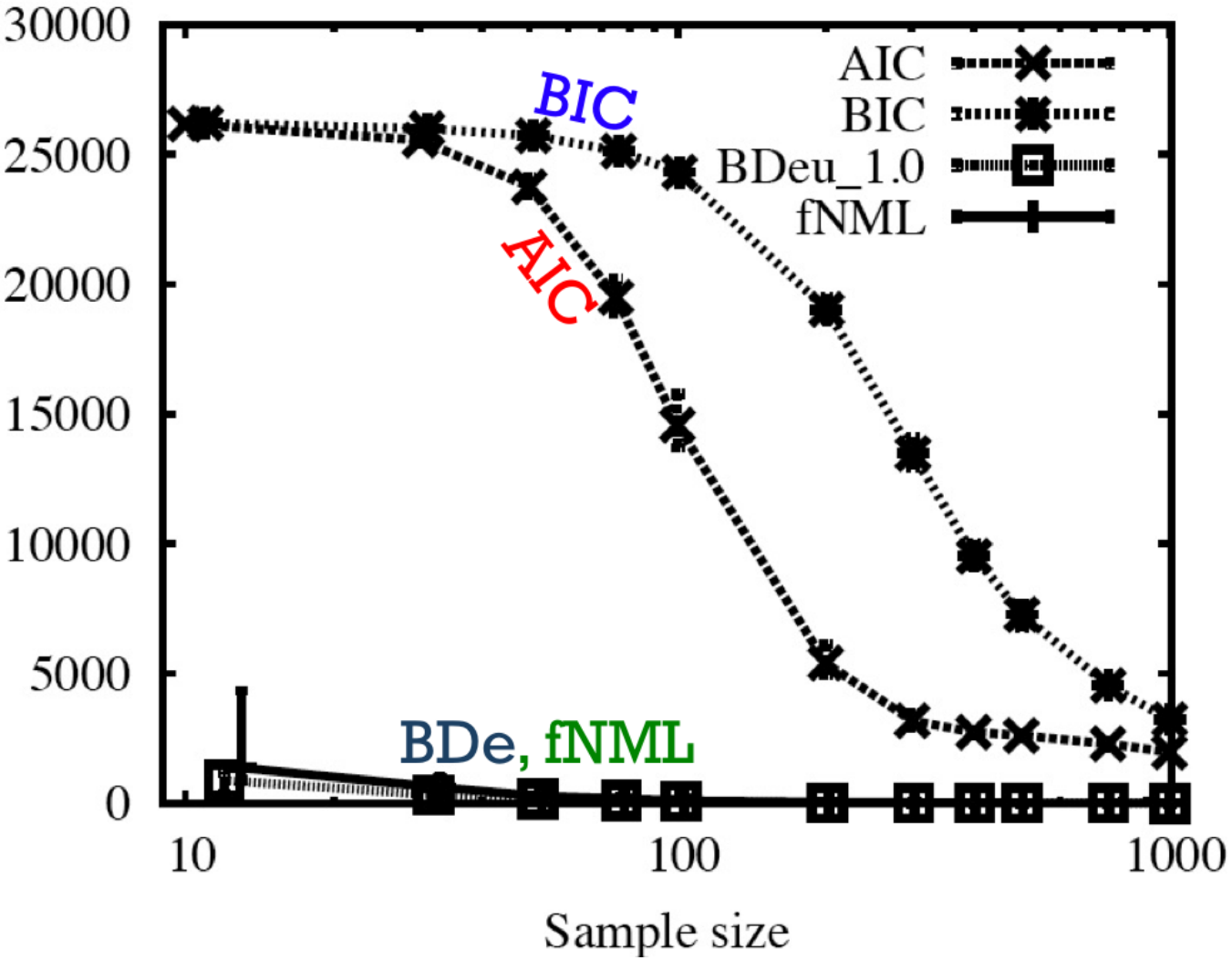
Problem with NML for Bayesian networks is the summation over all possible data-sets: exponential complexity.

Approximation: normalization over smaller blocks at a time.

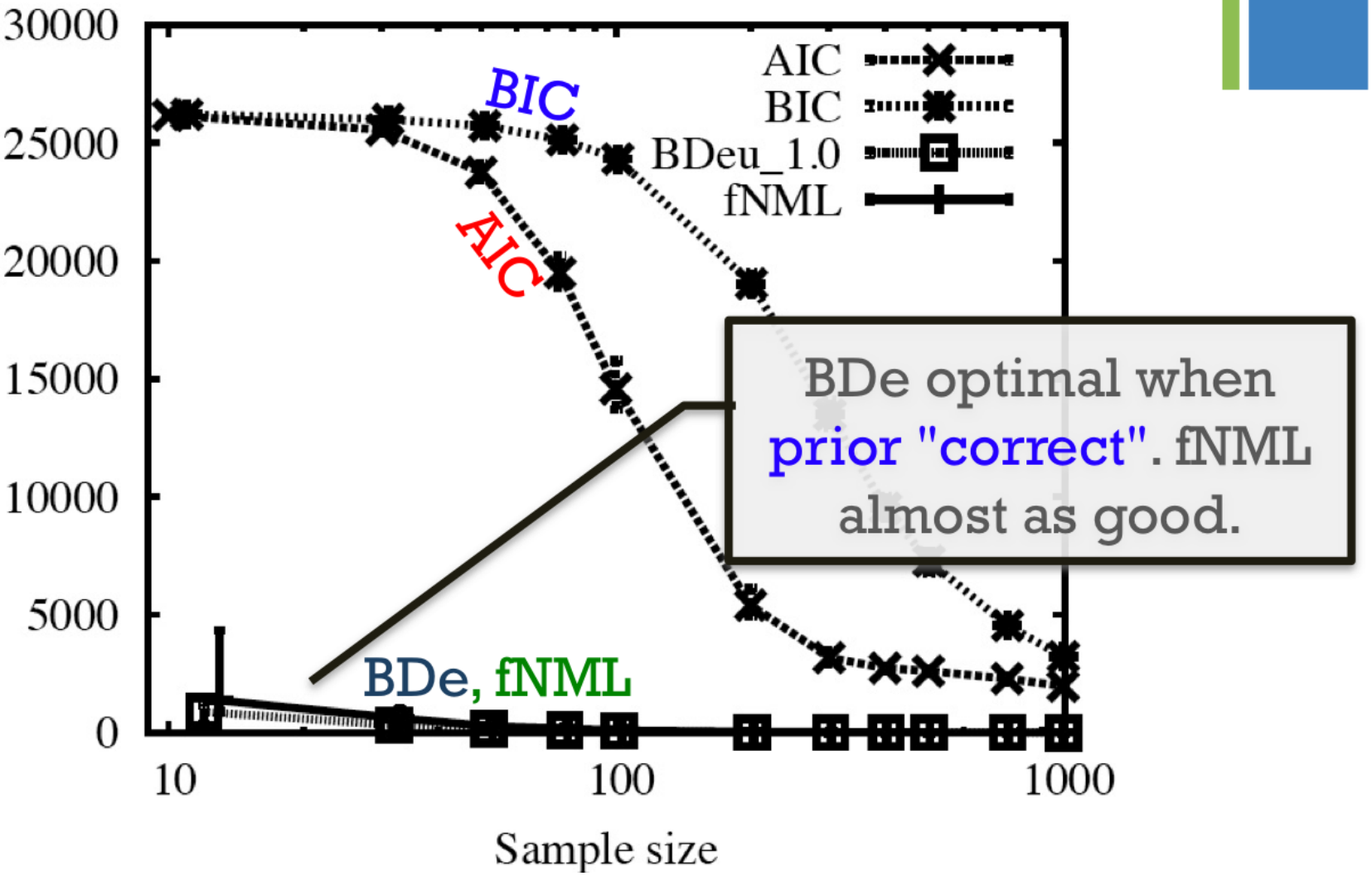


$\Rightarrow$ **Factorized NML**

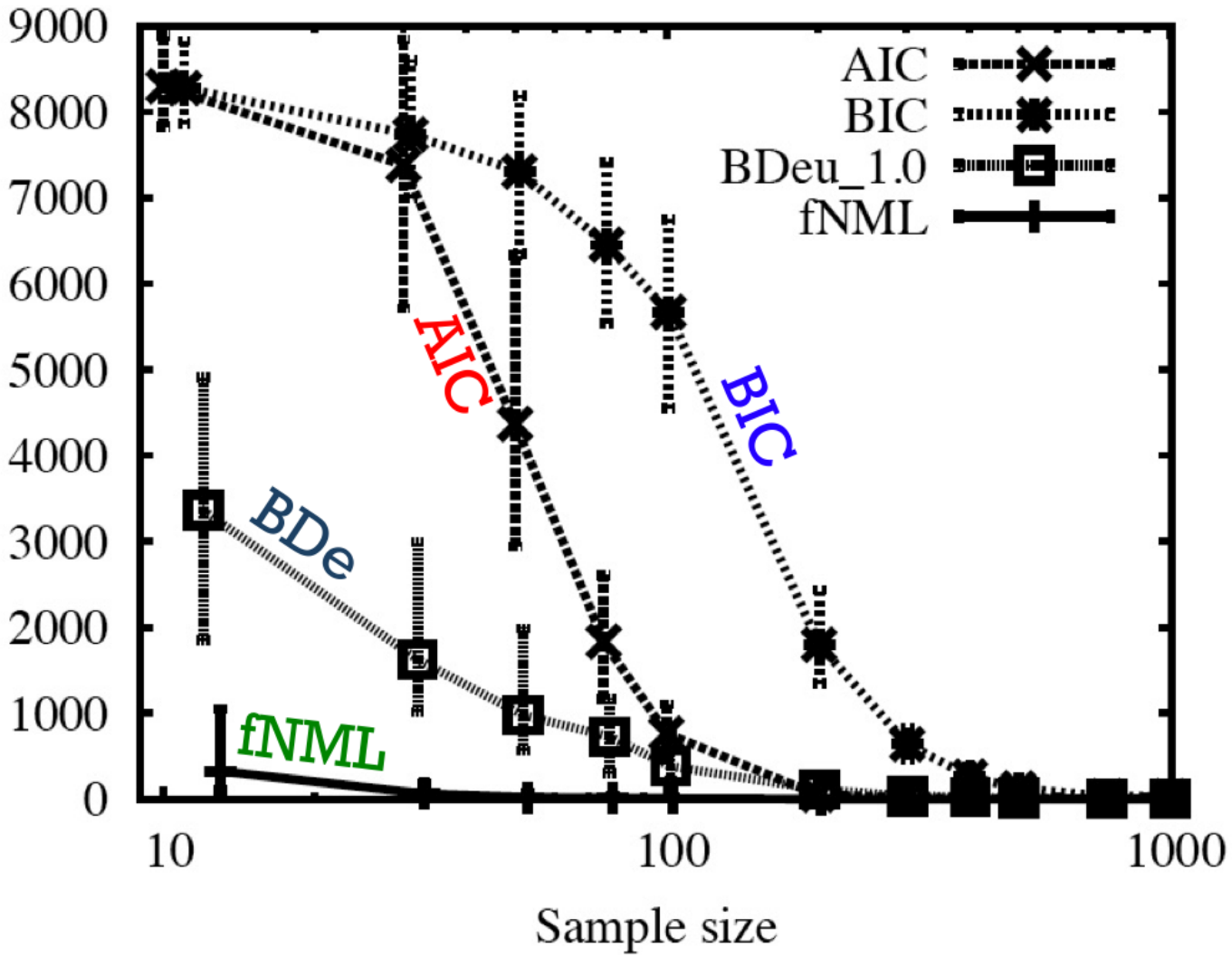
Bayesian Networks



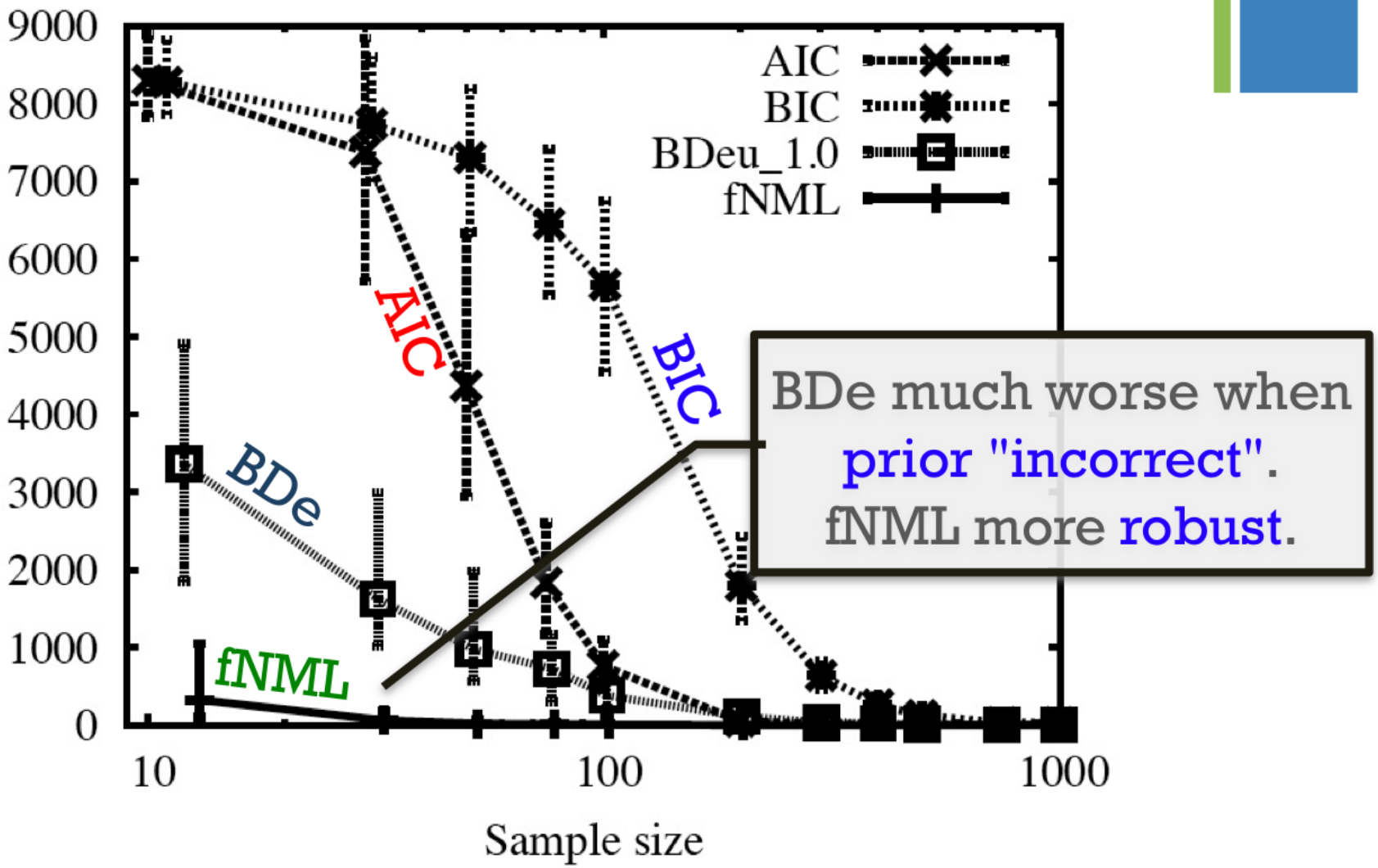
Bayesian Networks



Bayesian Networks



Bayesian Networks



How to Encode Continuous Data?

In order to encode data using, say, the Gaussian density we face the question: How to encode continuous data?

How to Encode Continuous Data?

In order to encode data using, say, the Gaussian density we face the question: How to encode continuous data?

We already know how to encode using models with continuous *parameters*:

How to Encode Continuous Data?

In order to encode data using, say, the Gaussian density we face the question: How to encode continuous data?

We already know how to encode using models with continuous *parameters*:

- two-part with optimal quantization $\left(\approx \frac{k}{2} \log_2 n\right)$,

How to Encode Continuous Data?

In order to encode data using, say, the Gaussian density we face the question: How to encode continuous data?

We already know how to encode using models with continuous *parameters*:

- two-part with optimal quantization $\left(\approx \frac{k}{2} \log_2 n\right)$,
- mixture code,

How to Encode Continuous Data?

In order to encode data using, say, the Gaussian density we face the question: How to encode continuous data?

We already know how to encode using models with continuous *parameters*:

- two-part with optimal quantization $\left(\approx \frac{k}{2} \log_2 n\right)$,
- mixture code,
- NML.

How to Encode Continuous Data?

In order to encode data using, say, the Gaussian density we face the question: How to encode continuous data?

We already know how to encode using models with continuous *parameters*:

- two-part with optimal quantization $\left(\approx \frac{k}{2} \log_2 n\right)$,
- mixture code,
- NML.

Obviously not possible to encode data with infinite precision. Have to **discretize**: encode x only up to precision δ.

Gaussian Density

Recall the Gaussian density function:

$$\phi_{\mu,\sigma^2}(x_1,\ldots,x_n) \stackrel{(i.i.d.)}{=} \left(2\pi\sigma^2\right)^{-n/2} e^{-\frac{\sum_{i=1}^{n}(x_i-\mu)^2}{2\sigma^2}}.$$

Gaussian Density

Recall the Gaussian density function:

$$\phi_{\mu,\sigma^2}(x_1,\ldots,x_n) \stackrel{(i.i.d.)}{=} \left(2\pi\sigma^2\right)^{-n/2} e^{-\frac{\sum_{i=1}^n (x_i-\mu)^2}{2\sigma^2}}.$$

The code-length is then

$$\frac{n}{2}\log_2(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^n (x_i-\mu)^2.$$

Gaussian Density

Ok, we have our Gaussian code-length formula:

$$\frac{n}{2}\log_2(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2.$$

Gaussian Density

Ok, we have our Gaussian code-length formula:

$$\frac{n}{2}\log_2(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2.$$

Let's use the two-part code and plug in the maximum likelihood parameters:

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i, \quad \hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i-\hat{\mu})^2.$$

Gaussian Density

Ok, we have our Gaussian code-length formula:

$$\frac{n}{2}\log_2(2\pi\hat{\sigma}^2) - \frac{1}{2\hat{\sigma}^2}\sum_{i=1}^{n}(x_i - \hat{\mu})^2.$$

Let's use the two-part code and plug in the maximum likelihood parameters:

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i, \quad \hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{\mu})^2.$$

Gaussian Density

Ok, we have our Gaussian code-length formula:

$$\frac{n}{2}\log_2(2\pi\hat{\sigma}^2) - \frac{1}{2\hat{\sigma}^2}\sum_{i=1}^{n}(x_i - \hat{\mu})^2.$$

Let's use the two-part code and plug in the maximum likelihood parameters:

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i, \quad \hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{\mu})^2.$$

Gaussian Density

Ok, we have our Gaussian code-length formula:

$$\frac{n}{2}\log_2(2\pi\hat{\sigma}^2) - \frac{n}{2}.$$

Let's use the two-part code and plug in the maximum likelihood parameters:

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i, \quad \hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{\mu})^2.$$

Gaussian Density

Ok, we have our Gaussian code-length formula:

$$\frac{n}{2}\log_2 \hat{\sigma}^2 + \textit{constant}.$$

Let's use the two-part code and plug in the maximum likelihood parameters:

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i, \quad \hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{\mu})^2.$$

Gaussian Density

We get the total (two-part) code-length formula:

$$\frac{n}{2}\log_2 \hat{\sigma}^2 + \frac{k}{2}\log_2 n + \textit{constant}.$$

Gaussian Density

We get the total (two-part) code-length formula:

$$\frac{n}{2}\log_2\hat{\sigma}^2 + \frac{k}{2}\log_2 n + \textit{constant}.$$

Since we have two parameters, μ and σ^2, we let $k = 2$.

Gaussian Density

We get the total (two-part) code-length formula:

$$\frac{n}{2}\log_2 \hat{\sigma}^2 + \frac{2}{2}\log_2 n + \textit{constant}.$$

Since we have two parameters, μ and σ^2, we let $k = 2$.

Linear Regression

A similar treatment can be given to *linear regression models*.

Linear Regression

A similar treatment can be given to *linear regression models*.

The model includes a set of regressor variables $x_1, \ldots, x_p \in \mathbb{R}$, and a set of coefficients $\beta_1, \ldots, \beta_p$.

Linear Regression

A similar treatment can be given to *linear regression models*.

The model includes a set of regressor variables $x_1, \ldots, x_p \in \mathbb{R}$, and a set of coefficients $\beta_1, \ldots, \beta_p$.

The dependent variable, Y, is assumed to be Gaussian:

- the mean μ is given as a linear combination of the regressors:

$$\mu = \beta_1 x_1 + \cdots + \beta_p x_p = \beta' x,$$

- variance is some parameter σ^2.

Linear Regression

For a sample of size n, the matrix notation is convenient:

$$Y = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix} \quad X = \begin{pmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{pmatrix} \quad \beta = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_p \end{pmatrix} \quad \epsilon = \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{pmatrix}$$

Linear Regression

For a sample of size n, the matrix notation is convenient:

$$Y = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix} \quad X = \begin{pmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{pmatrix} \quad \beta = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_p \end{pmatrix} \quad \epsilon = \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{pmatrix}$$

Then the model can be written as

$$Y = X\beta + \epsilon,$$

where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$.

Linear Regression

The maximum likelihood estimators are now

$$\hat{\beta} = (X'X)^{-1}X'Y, \quad \hat{\sigma}^2 = \frac{1}{n}\|Y - X\hat{\beta}\|_2^2 = \frac{\mathrm{RSS}}{n},$$

where RSS is the "residual sum of squares".

Linear Regression

The maximum likelihood estimators are now

$$\hat{\beta} = (X'X)^{-1}X'Y, \quad \hat{\sigma}^2 = \frac{1}{n}\|Y - X\hat{\beta}\|_2^2 = \frac{\mathrm{RSS}}{n},$$

where RSS is the "residual sum of squares".

Since the errors are assumed Gaussian, our code-length formula applies:

$$\frac{n}{2}\log_2 \hat{\sigma}^2 + \frac{k}{2}\log_2 n + \text{constant}.$$

Linear Regression

The maximum likelihood estimators are now

$$\hat{\beta} = (X'X)^{-1}X'Y, \quad \hat{\sigma}^2 = \frac{1}{n}\|Y - X\hat{\beta}\|_2^2 = \frac{\mathrm{RSS}}{n},$$

where RSS is the "residual sum of squares".

Since the errors are assumed Gaussian, our code-length formula applies:

$$\frac{n}{2}\log_2 \mathrm{RSS} + \frac{k}{2}\log_2 n + \mathrm{constant}.$$

Linear Regression

The maximum likelihood estimators are now

$$\hat{\beta} = (X'X)^{-1}X'Y, \quad \hat{\sigma}^2 = \frac{1}{n}\|Y - X\hat{\beta}\|_2^2 = \frac{\mathrm{RSS}}{n},$$

where RSS is the "residual sum of squares".

Since the errors are assumed Gaussian, our code-length formula applies:

$$\frac{n}{2}\log_2 \mathrm{RSS} + \frac{k}{2}\log_2 n + \mathrm{constant}.$$

The number of parameters is now $p+1$ (p of the βs and σ^2), so we get...

Linear Regression

The maximum likelihood estimators are now

$$\hat{\beta} = (X'X)^{-1}X'Y, \quad \hat{\sigma}^2 = \frac{1}{n}\|Y - X\hat{\beta}\|_2^2 = \frac{\mathrm{RSS}}{n},$$

where RSS is the "residual sum of squares".

Since the errors are assumed Gaussian, our code-length formula applies:

$$\frac{n}{2}\log_2 \mathrm{RSS} + \frac{p+1}{2}\log_2 n + \mathrm{constant}.$$

The number of parameters is now $p+1$ (p of the βs and σ^2), so we get...

Subset Selection Problem

Often we have a large set of potential regressors, some of which may be irrelevant.

Subset Selection Problem

Often we have a large set of potential regressors, some of which may be irrelevant.

The MDL principle can be used to select a subset of them by comparing the total code-lengths:

$$\min_S \left[\frac{n}{2} \log_2 \mathrm{RSS}_S + \frac{|S|+1}{2} \log_2 n \right],$$

where RSS_S is the RSS obtained by using subset S of the regressors.

Subset Selection Problem

Often we have a large set of potential regressors, some of which may be irrelevant.

The MDL principle can be used to select a subset of them by comparing the total code-lengths:

$$\min_S \left[\frac{n}{2} \log_2 \mathrm{RSS}_S + \frac{|S|+1}{2} \log_2 n \right],$$

where RSS_S is the RSS obtained by using subset S of the regressors.

$\Rightarrow$ Exercise 10.

Denoising

$$
\begin{aligned}
\textit{Complexity} &= \textit{Information} + \textit{Noise} \\
&= \textit{Regularity} + \textit{Randomness} \\
&= \textit{Algorithm} + \textit{Compressed file}
\end{aligned}
$$

Denoising

$$
\begin{aligned}
\textit{Complexity} &= \textit{Information} + \textit{Noise} \\
&= \textit{Regularity} + \textit{Randomness} \\
&= \textit{Algorithm} + \textit{Compressed file}
\end{aligned}
$$

Denoising means the process of removing noise from a signal.

Denoising

$$
\begin{aligned}
\textit{Complexity} &= \textit{Information} + \textit{Noise} \\
&= \textit{Regularity} + \textit{Randomness} \\
&= \textit{Algorithm} + \textit{Compressed file}
\end{aligned}
$$

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.

Denoising

$$
\begin{aligned}
\textit{Complexity} &= \textit{Information} + \textit{Noise} \\
&= \textit{Regularity} + \textit{Randomness} \\
&= \textit{Algorithm} + \textit{Compressed file}
\end{aligned}
$$

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).

Wavelet Denoising

One particularly useful way to obtain the regressor (design) matrix is to use **wavelets**.

Wavelet Denoising

One particularly useful way to obtain the regressor (design) matrix is to use **wavelets**.

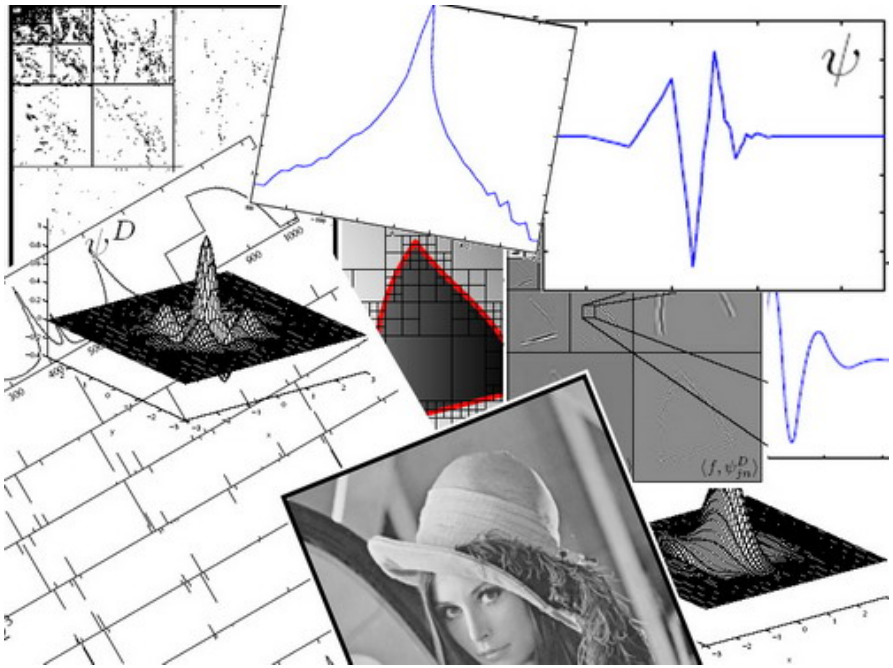


Image by Gabriel Peyré

Wavelet Denoising

IEEE TRANS. SIGNAL PROCESSING, VOL. ?, NO. ?, 2009

MDL Denoising Revisited

Teemu Roos *Member*, Petri Myllymäki, and Jorma Rissanen *Fellow*

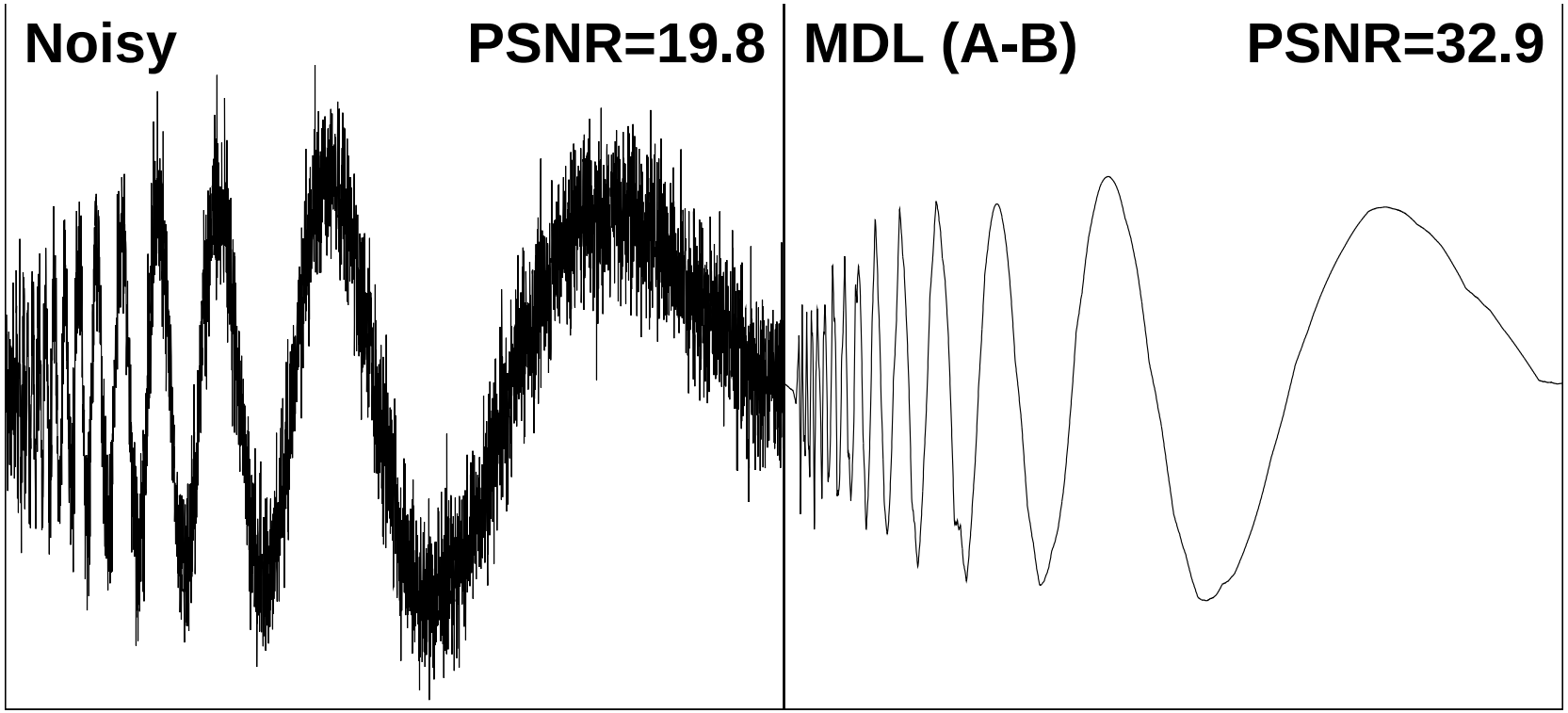
***Abstract*— We refine and extend an earlier minimum description length (MDL) denoising criterion for wavelet-based denoising. We start by showing that the denoising problem can be reformulated as a clustering problem, where the goal is to obtain separate clusters for informative and non-informative wavelet coefficients, respectively. This suggests two refinements, adding a code-length for the model index, and extending the model in order to account for subband-dependent coefficient distributions. A third refinement is the derivation of soft thresholding inspired by predictive universal coding with weighted mixtures. We propose a practical method incorporating all three refinements, which is shown to achieve good performance and robustness in denoising both artificial and natural signals.**

***Index Terms*— Minimum description length (MDL) principle, wavelets, denoising.**

(both of which include the Gaussian and d densities as special cases).

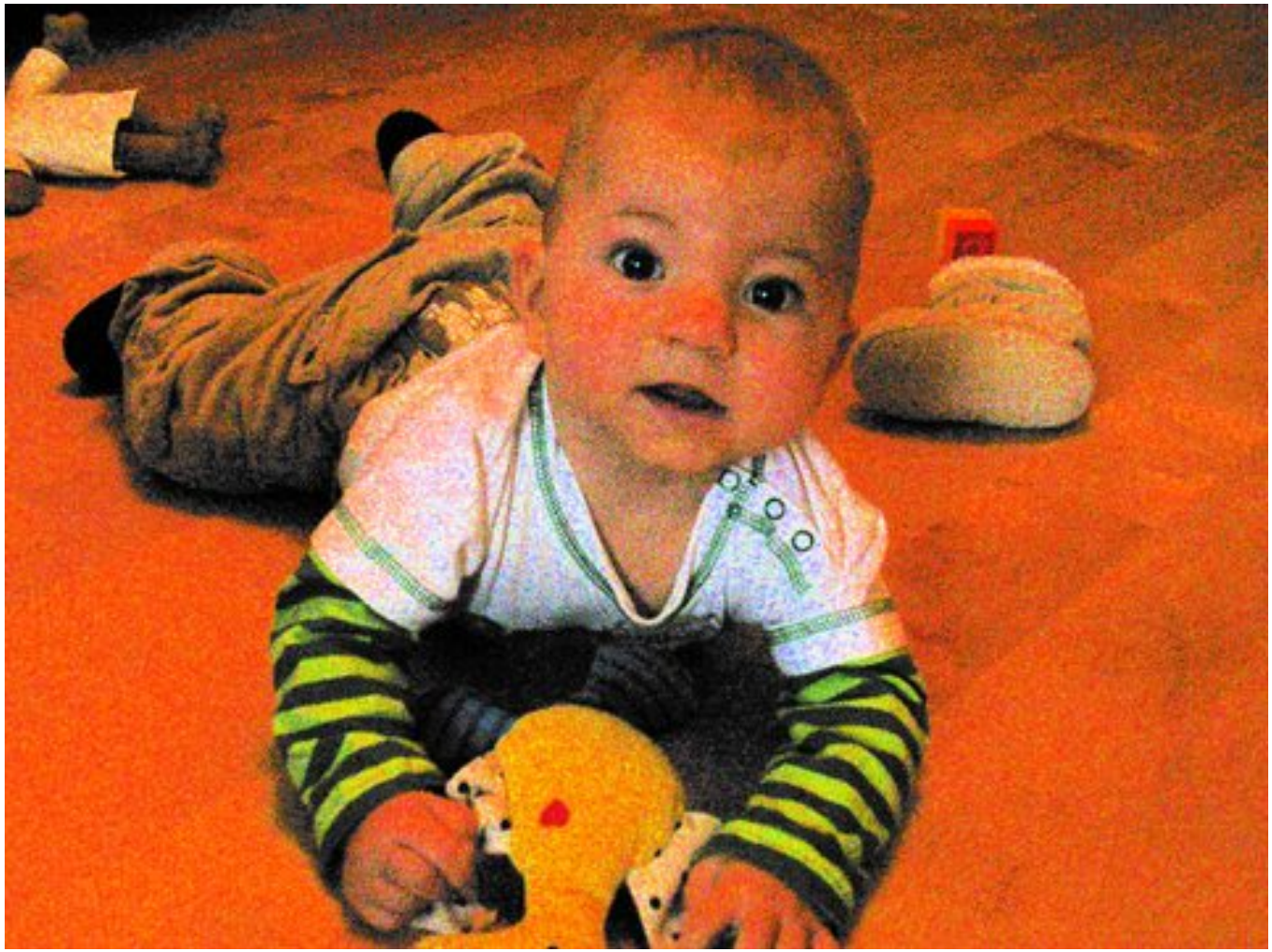
A third approach to denoising is based description length (MDL) principle [16]–[2 ent MDL denoising methods have been su [21]–[25]. We focus on what we consider MDL approach, namely that of Rissanen [24 is two-fold: First, as an immediate result extending the earlier MDL denoising met new practical method with greatly impro and robustness. Secondly, the denoising p to illustrate theoretical issues related to the involving the problem of unbounded paran and the necessity of encoding the model cl

Example: Denoising

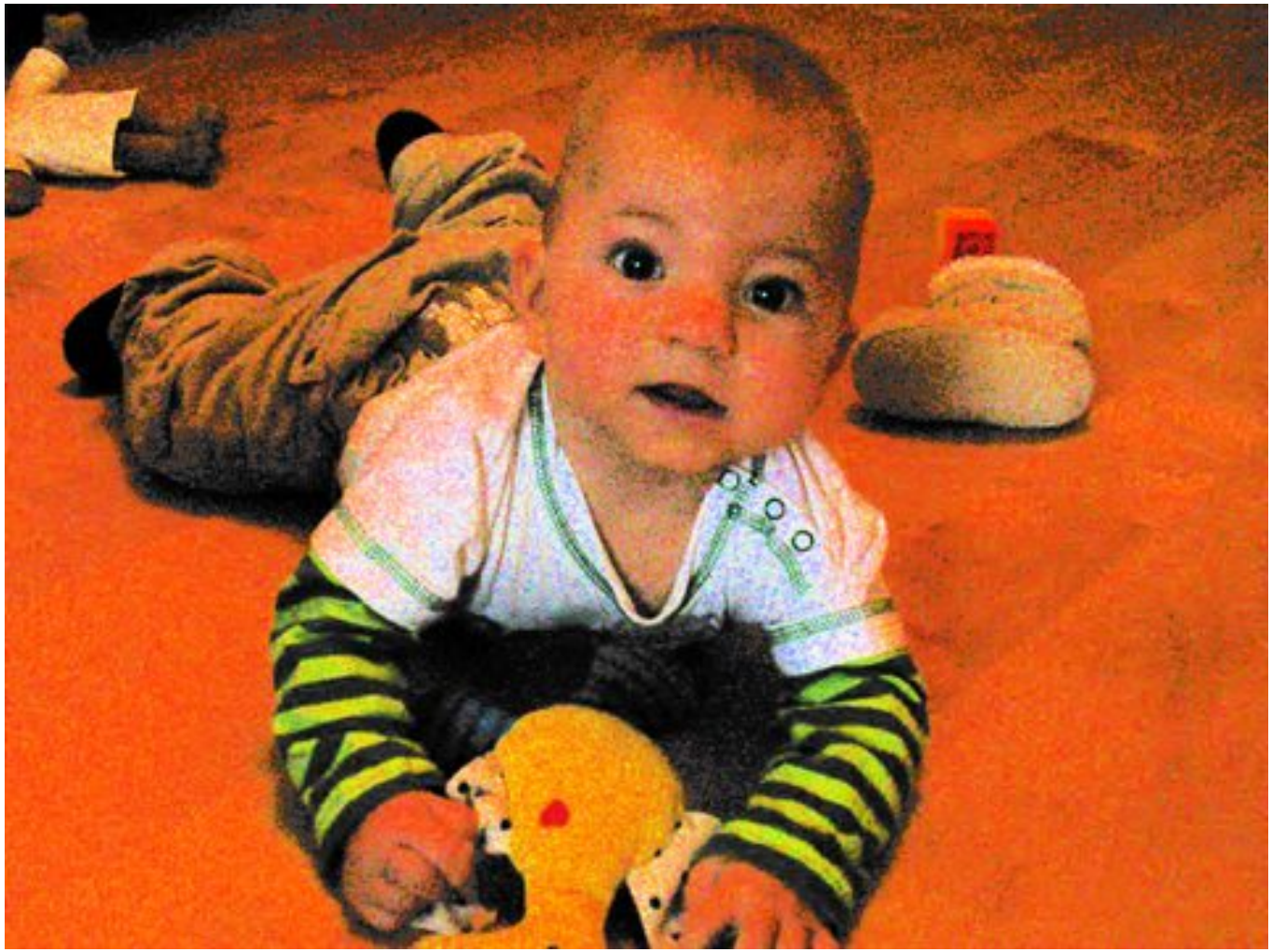


Example: Denoising

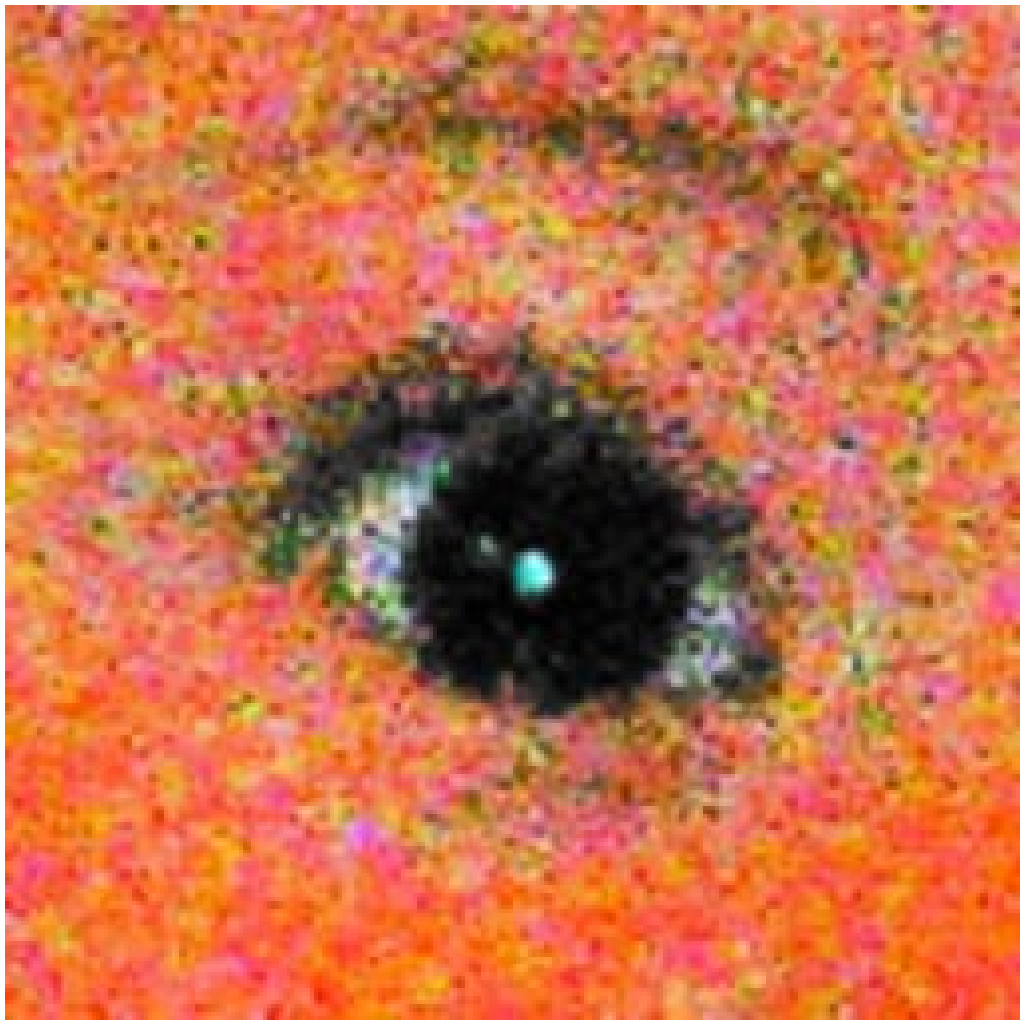
Example: Denoising



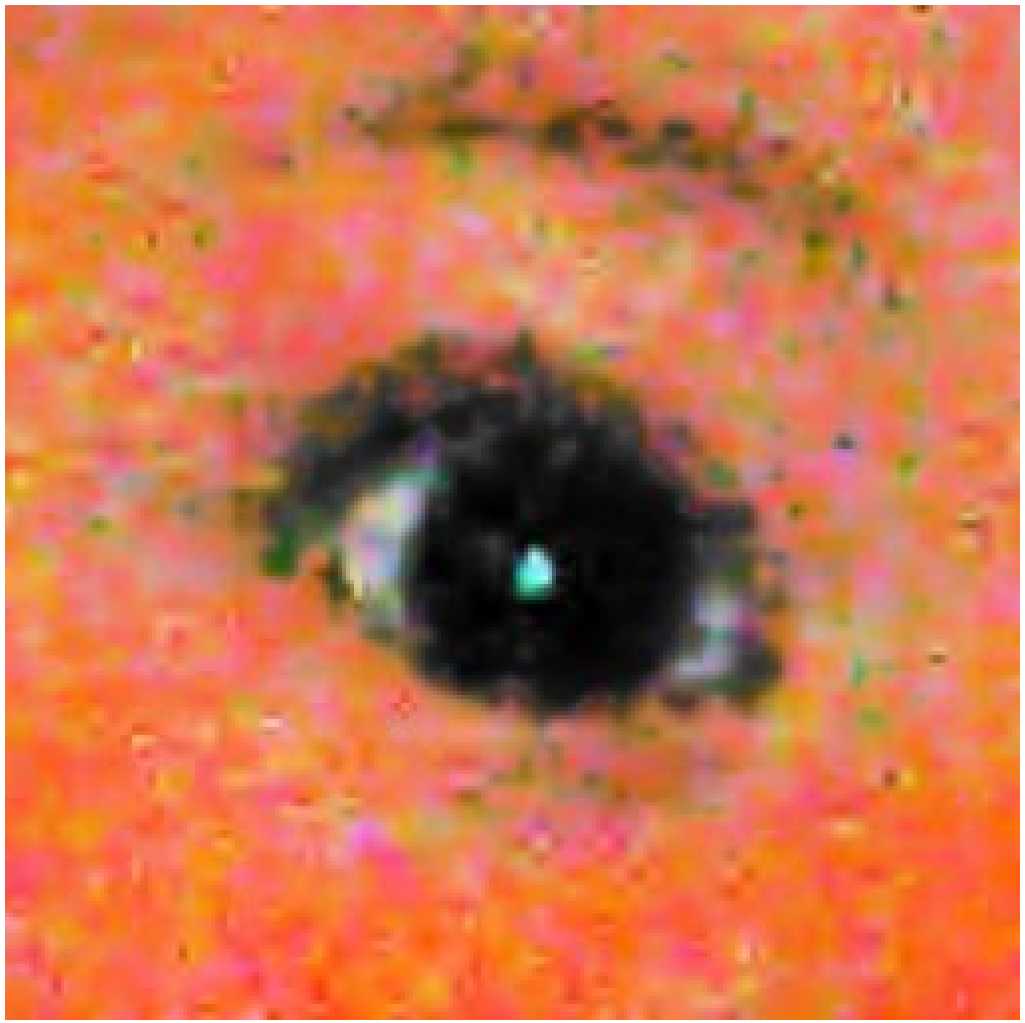
Example: Denoising



Example: Denoising



Example: Denoising



Last Slide

Thanks for listening!