Sentence Length

Gábor Borbély, **András Kornai**

Budapest University of Technology and Economics
Department of Algebra
https://hlt.bme.hu/
https://github.com/hlt-bme-hu/

2019.07.19.

Sentence length models

- Negative binomial (Yule, 1944)
  - Given number of failures in an sequence of independent and identically distributed Bernoulli trials

Sentence length models

- Negative binomial (Yule, 1944)
  - Given number of failures in an sequence of independent and identically distributed Bernoulli trials
- Log-normal (Williams, 1944) (Wake, 1957)
  - frequencies are normal on log-linear scale

Sentence length models

- Negative binomial (Yule, 1944)
  - Given number of failures in an sequence of independent and identically distributed Bernoulli trials
- Log-normal (Williams, 1944) (Wake, 1957)
  - frequencies are normal on log-linear scale
- Mixture of Poisson (Sichel, 1974)
  - mixture of continuous number of Poissons where the mixture distribution is parametrized

$$\phi(r) = \frac{\sqrt{1-\theta}^{\gamma}}{K_{\gamma}(\alpha\sqrt{1-\theta})} \frac{(\alpha\theta/2)^{r}}{r!} K_{r+\gamma}(\alpha)$$

Sentence length models

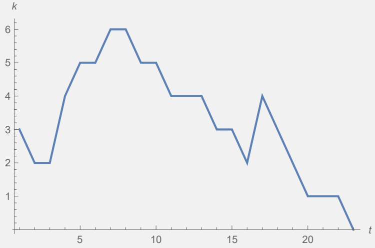
- Negative binomial (Yule, 1944)
  - Given number of failures in an sequence of independent and identically distributed Bernoulli trials
- Log-normal (Williams, 1944) (Wake, 1957)
  - frequencies are normal on log-linear scale
- Mixture of Poisson (Sichel, 1974)
  - mixture of continuous number of Poissons where the mixture distribution is parametrized

$$\phi(r) = \frac{\sqrt{1-\theta}^{\gamma}}{K_{\gamma}(\alpha\sqrt{1-\theta})} \frac{(\alpha\theta/2)^{r}}{r!} K_{r+\gamma}(\alpha)$$

- These models either don't fit the data or lack a clear genesis

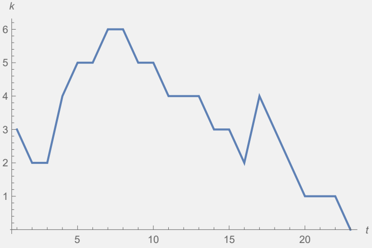
Random walk model

- y axis: valency



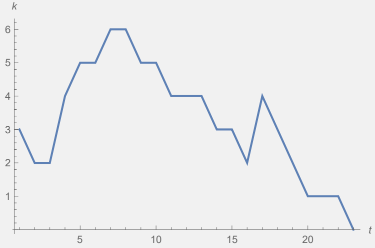
Random walk model

- y axis: valency
  - starting point is a parameter



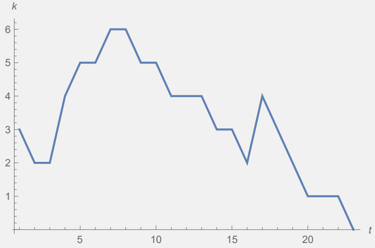
Random walk model

- y axis: valency
  - starting point is a parameter
- with probability p_2: two steps up



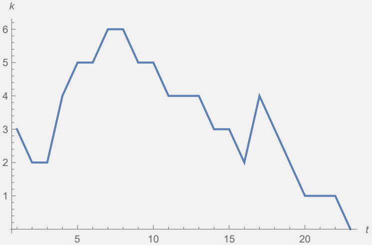
Random walk model

- y axis: valency
  - starting point is a parameter
- with probability p_2: two steps up
  - e.g. a transitive verb



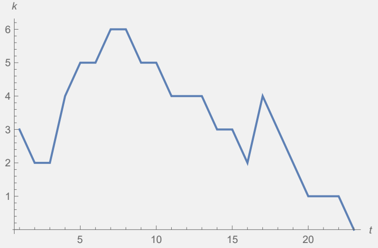
Random walk model

- y axis: valency
  - starting point is a parameter
- with probability p_2: two steps up
  - e.g. a transitive verb
- p_1: one step up



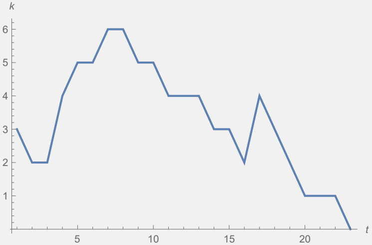
Random walk model

- y axis: valency
  - starting point is a parameter
- with probability p_2: two steps up
  - e.g. a transitive verb
- p_1: one step up
  - e.g. an intransitive verb or an adjective



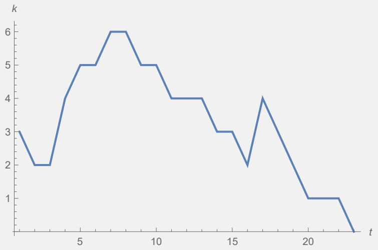
Random walk model

- y axis: valency
  - starting point is a parameter
- with probability p_2: two steps up
  - e.g. a transitive verb
- p_1: one step up
  - e.g. an intransitive verb or an adjective
- p_0: same height



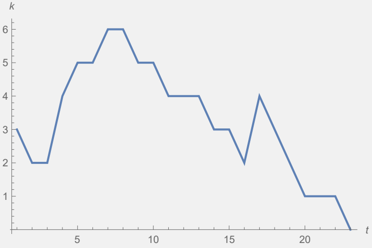
Random walk model

- y axis: valency
  - starting point is a parameter
- with probability p_2: two steps up
  - e.g. a transitive verb
- p_1: one step up
  - e.g. an intransitive verb or an adjective
- p_0: same height
  - e.g. an adverbial



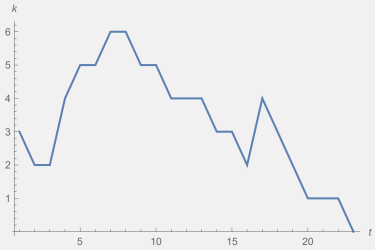
Random walk model

- y axis: valency
  - starting point is a parameter
- with probability p_2: two steps up
  - e.g. a transitive verb
- p_1: one step up
  - e.g. an intransitive verb or an adjective
- p_0: same height
  - e.g. an adverbial
- p_{-1}: one step down



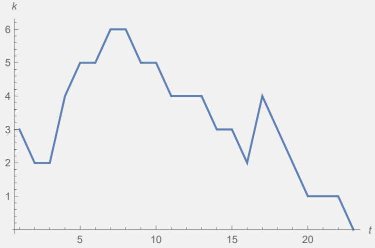
Random walk model

- y axis: valency
  - starting point is a parameter
- with probability p_2: two steps up
  - e.g. a transitive verb
- p_1: one step up
  - e.g. an intransitive verb or an adjective
- p_0: same height
  - e.g. an adverbial
- p_{-1}: one step down
  - e.g. a proper noun



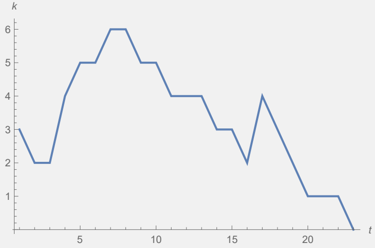
Random walk model

- y axis: valency
  - starting point is a parameter
- with probability p_2: two steps up
  - e.g. a transitive verb
- p_1: one step up
  - e.g. an intransitive verb or an adjective
- p_0: same height
  - e.g. an adverbial
- p_{-1}: one step down
  - e.g. a proper noun
- The predicted sentence length is the first time when the process reaches zero valency



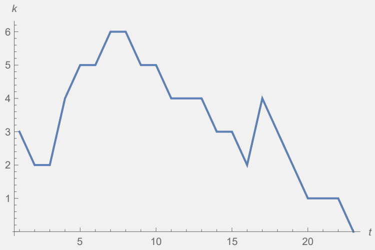
Random walk model

- y axis: valency
  - starting point is a parameter
- with probability p_2: two steps up
  - e.g. a transitive verb
- p_1: one step up
  - e.g. an intransitive verb or an adjective
- p_0: same height
  - e.g. an adverbial
- p_{-1}: one step down
  - e.g. a proper noun
- The predicted sentence length is the first time when the process reaches zero valency



Random walk model

- y axis: valency
  - starting point is a parameter
- with probability p_2: two steps up
  - e.g. a transitive verb
- p_1: one step up
  - e.g. an intransitive verb or an adjective
- p_0: same height
  - e.g. an adverbial
- p_{-1}: one step down
  - e.g. a proper noun
- The predicted sentence length is the first time when the process reaches zero valency



Some generalizations may complicate the model:

- order (upward steps)
- k-mixture
- auxiliary model

Model analysis

- $\tau_k = \min_{t \geq 0}\{t : X_k(t) = 0\}$

Model analysis

- $\tau_k = \min_{t \geq 0}\{t : X_k(t) = 0\}$
- τ_k is the sum of k independent copies of τ_1

Model analysis

- $\tau_k = \min_{t \geq 0}\{t : X_k(t) = 0\}$
- τ_k is the sum of k independent copies of τ_1
  - since going from $k \to 0$ requires k times going from 1 to 0.

Model analysis

- $\tau_k = \min_{t \geq 0}\{t : X_k(t) = 0\}$
- τ_k is the sum of k independent copies of τ_1
  - since going from $k \to 0$ requires k times going from 1 to 0.
  - this in not the case if $p_{-2} > 0$!

Model analysis

- $\tau_k = \min_{t \geq 0}\{t : X_k(t) = 0\}$
- τ_k is the sum of k independent copies of τ_1
  - since going from $k \to 0$ requires k times going from 1 to 0.
  - this in not the case if $p_{-2} > 0$!
- $f(x) := \mathbb{E}\left(x^{\tau_1}\right)$

Model analysis

- $\tau_k = \min_{t \geq 0}\{t : X_k(t) = 0\}$
- τ_k is the sum of k independent copies of τ_1
  - since going from $k \to 0$ requires k times going from 1 to 0.
  - this in not the case if $p_{-2} > 0$!
- $f(x) := \mathbb{E}\left(x^{\tau_1}\right)$
  - the probability generating function of τ_k is $f(x)^k$

Model analysis

- $\tau_k = \min_{t \geq 0}\{t : X_k(t) = 0\}$
- τ_k is the sum of k independent copies of τ_1
  - since going from $k \to 0$ requires k times going from 1 to 0.
  - this in not the case if $p_{-2} > 0$!
- $f(x) := \mathbb{E}\left(x^{\tau_1}\right)$
  - the probability generating function of τ_k is $f(x)^k$

Model analysis

- $\tau_k = \min_{t \geq 0}\{t : X_k(t) = 0\}$
- τ_k is the sum of k independent copies of τ_1
  - since going from $k \to 0$ requires k times going from 1 to 0.
  - this in not the case if $p_{-2} > 0$!
- $f(x) := \mathbb{E}\left(x^{\tau_1}\right)$
  - the probability generating function of τ_k is $f(x)^k$

$$
\begin{aligned}
f(x) = p_{-1} \cdot x + & \quad \text{finishing in one step} \\
p_0 \cdot x \cdot f(x) + & \quad \text{wait } \tau_1 \text{ again} \\
p_1 \cdot x \cdot f(x)^2 + & \quad \text{wait } \tau_1 \text{ two times} \\
p_2 \cdot x \cdot f(x)^3 & \quad \text{wait } \tau_1 \text{ three times}
\end{aligned}
$$

Model analysis

- $\tau_k = \min_{t \geq 0}\{t : X_k(t) = 0\}$
- τ_k is the sum of k independent copies of τ_1
  - since going from $k \to 0$ requires k times going from 1 to 0.
  - this in not the case if $p_{-2} > 0$!
- $f(x) := \mathbb{E}\left(x^{\tau_1}\right)$
  - the probability generating function of τ_k is $f(x)^k$

$f(x) = p_{-1} \cdot x +$	finishing in one step
$p_0 \cdot x \cdot f(x) +$	wait τ_1 again
$p_1 \cdot x \cdot f(x)^2 +$	wait τ_1 two times
$p_2 \cdot x \cdot f(x)^3$	wait τ_1 three times

f is the solution of the following equation:

$$p_{-1} \cdot x + (p_0 \cdot x - 1) \cdot f + p_1 \cdot x \cdot f^2 + p_2 \cdot x \cdot f^3 = 0$$

Model analysis II.

$$F(u) \coloneqq p_{-1} + p_0 \cdot u + p_1 \cdot u^2 + p_2 u^3$$
$$g(f) \coloneqq \frac{f}{F(f)}$$
$$x = g(f(x))$$

- so the solution is the inverse function of g

Model analysis II.

$$F(u) \coloneqq p_{-1} + p_0 \cdot u + p_1 \cdot u^2 + p_2 u^3$$

$$g(f) \coloneqq \frac{f}{F(f)}$$

$$x = g(f(x))$$

- so the solution is the inverse function of g
- we won't solve it explicitly

Model analysis II.

$$F(u) \coloneqq p_{-1} + p_0 \cdot u + p_1 \cdot u^2 + p_2 u^3$$
$$g(f) \coloneqq \frac{f}{F(f)}$$
$$x = g(f(x))$$

- so the solution is the inverse function of g
- we won't solve it explicitly
  - although it is theoretically possible up to 3 steps upwards (4^{th} order root formula exists)

Model analysis II.

$$F(u) \coloneqq p_{-1} + p_0 \cdot u + p_1 \cdot u^2 + p_2 u^3$$
$$g(f) \coloneqq \frac{f}{F(f)}$$
$$x = g(f(x))$$

- so the solution is the inverse function of g
- we won't solve it explicitly
  - although it is theoretically possible up to 3 steps upwards (4^{th} order root formula exists)
- rather find the Taylor expansion of f via Lagrange–Bürmann formula

Model analysis II.

$$
\begin{aligned}
F(u) &:= p_{-1} + p_0 \cdot u + p_1 \cdot u^2 + p_2 u^3 \\
g(f) &:= \frac{f}{F(f)} \\
x &= g(f(x))
\end{aligned}
$$

- so the solution is the inverse function of g
- we won't solve it explicitly
  - although it is theoretically possible up to 3 steps upwards (4^{th} order root formula exists)
- rather find the Taylor expansion of f via Lagrange–Bürmann formula
  - a version of Lagrange inversion theorem

Model analysis II.

$$F(u) \coloneqq p_{-1} + p_0 \cdot u + p_1 \cdot u^2 + p_2 u^3$$
$$g(f) \coloneqq \frac{f}{F(f)}$$
$$x = g(f(x))$$

- so the solution is the inverse function of g
- we won't solve it explicitly
  - although it is theoretically possible up to 3 steps upwards (4^{th} order root formula exists)
- rather find the Taylor expansion of f via Lagrange–Bürmann formula
  - a version of Lagrange inversion theorem

Model analysis II.

$$
\begin{aligned}
F(u) &\coloneqq p_{-1} + p_0 \cdot u + p_1 \cdot u^2 + p_2 u^3 \\
g(f) &\coloneqq \frac{f}{F(f)} \\
x &= g(f(x))
\end{aligned}
$$

- so the solution is the inverse function of g
- we won't solve it explicitly
  - although it is theoretically possible up to 3 steps upwards (4^{th} order root formula exists)
- rather find the Taylor expansion of f via Lagrange–Bürmann formula
  - a version of Lagrange inversion theorem

$$\mathbb{P}(\tau_k = i) = \frac{k}{i}\left[u^{i-k}\right](F(u))^i$$

involves calculating symbolic product of polynomials

Optimizing the parameters

- The task is to fit the parameters such that the resulted return time is close to the measured distribution in cross entropy

Optimizing the parameters

- The task is to fit the parameters such that the resulted return time is close to the measured distribution in cross entropy
- Also the $p_{-1}, p_0 \ldots$ parameters are constrained on a probabilistic simplex

Optimizing the parameters

- The task is to fit the parameters such that the resulted return time is close to the measured distribution in cross entropy
- Also the $p_{-1}, p_0 \ldots$ parameters are constrained on a probabilistic simplex

Optimizing the parameters

- The task is to fit the parameters such that the resulted return time is close to the measured distribution in cross entropy
- Also the $p_{-1}, p_0 \ldots$ parameters are constrained on a probabilistic simplex

$$F(u) := p_{-1} + p_0 \cdot u + p_1 \cdot u^2 + p_2 u^3$$

$$\mathbb{P}(\tau_k = i) = \frac{k}{i}[u^{i-k}]\,(F(u))^i$$

- The latter is differentiable in the model parameters

Optimizing the parameters

- The task is to fit the parameters such that the resulted return time is close to the measured distribution in cross entropy
- Also the $p_{-1}, p_0 \ldots$ parameters are constrained on a probabilistic simplex

$$F(u) := p_{-1} + p_0 \cdot u + p_1 \cdot u^2 + p_2 u^3$$
$$\mathbb{P}(\tau_k = i) = \frac{k}{i} [u^{i-k}] \left(F(u)\right)^i$$

- The latter is differentiable in the model parameters
  - one can perform gradient descent (or similar optimization techniques)

Optimizing the parameters

- The task is to fit the parameters such that the resulted return time is close to the measured distribution in cross entropy
- Also the $p_{-1}, p_0 \ldots$ parameters are constrained on a probabilistic simplex

$$F(u) := p_{-1} + p_0 \cdot u + p_1 \cdot u^2 + p_2 u^3$$

$$\mathbb{P}(\tau_k = i) = \frac{k}{i} [u^{i-k}] \left(F(u)\right)^i$$

- The latter is differentiable in the model parameters
  - one can perform gradient descent (or similar optimization techniques)
  - as long as the discrete parameters are fixed

Optimizing the parameters

- The task is to fit the parameters such that the resulted return time is close to the measured distribution in cross entropy
- Also the $p_{-1}, p_0 \ldots$ parameters are constrained on a probabilistic simplex

$$F(u) := p_{-1} + p_0 \cdot u + p_1 \cdot u^2 + p_2 u^3$$

$$\mathbb{P}(\tau_k = i) = \frac{k}{i}[u^{i-k}]\,(F(u))^i$$

- The latter is differentiable in the model parameters
  - one can perform gradient descent (or similar optimization techniques)
  - as long as the discrete parameters are fixed
- There are other (discrete) parameters

Optimizing the parameters

- The task is to fit the parameters such that the resulted return time is close to the measured distribution in cross entropy
- Also the $p_{-1}, p_0 \ldots$ parameters are constrained on a probabilistic simplex

$$F(u) := p_{-1} + p_0 \cdot u + p_1 \cdot u^2 + p_2 u^3$$

$$\mathbb{P}(\tau_k = i) = \frac{k}{i}[u^{i-k}]\,(F(u))^i$$

- The latter is differentiable in the model parameters
  - one can perform gradient descent (or similar optimization techniques)
  - as long as the discrete parameters are fixed
- There are other (discrete) parameters
  - starting valency

Optimizing the parameters

- The task is to fit the parameters such that the resulted return time is close to the measured distribution in cross entropy
- Also the $p_{-1}, p_0 \ldots$ parameters are constrained on a probabilistic simplex

$$F(u) := p_{-1} + p_0 \cdot u + p_1 \cdot u^2 + p_2 u^3$$

$$\mathbb{P}(\tau_k = i) = \frac{k}{i} [u^{i-k}] \left(F(u)\right)^i$$

- The latter is differentiable in the model parameters
  - one can perform gradient descent (or similar optimization techniques)
  - as long as the discrete parameters are fixed
- There are other (discrete) parameters
  - starting valency
  - maximum upward steps

Optimizing the parameters

- The task is to fit the parameters such that the resulted return time is close to the measured distribution in cross entropy
- Also the $p_{-1}, p_0 \ldots$ parameters are constrained on a probabilistic simplex

$$F(u) := p_{-1} + p_0 \cdot u + p_1 \cdot u^2 + p_2 u^3$$

$$\mathbb{P}(\tau_k = i) = \frac{k}{i} [u^{i-k}] \left(F(u)\right)^i$$

- The latter is differentiable in the model parameters
  - one can perform gradient descent (or similar optimization techniques)
  - as long as the discrete parameters are fixed
- There are other (discrete) parameters
  - starting valency
  - maximum upward steps
  - mixture components

Model comparison

- Let $\{n_x\}_{x \in X}$ be the measured frequencies of a data

Model comparison

- Let $\{n_x\}_{x \in X}$ be the measured frequencies of a data
- Let $\mathcal{H}_i$ be a model in a list of possible models

Model comparison

- Let $\{n_x\}_{x \in X}$ be the measured frequencies of a data
- Let $\mathcal{H}_i$ be a model in a list of possible models
- Within a model there can be other trained parameters

$$\mathbf{w}_i \in \mathcal{H}_i,\ \mathbb{Q}_{\mathbf{w}_i}(x) \coloneqq \mathbb{P}(x \mid \mathbf{w}_i, \mathcal{H}_i)$$

In our case $\mathcal{H}_i$ is the choice of the discrete parameters and $\mathbf{w}_i \in \mathcal{H}_i$ is trained by optimizing the continuous parameters of that model.

Model comparison

- Let $\{n_x\}_{x \in X}$ be the measured frequencies of a data
- Let $\mathcal{H}_i$ be a model in a list of possible models
- Within a model there can be other trained parameters

$$\mathbf{w}_i \in \mathcal{H}_i,\ \mathbb{Q}_{\mathbf{w}_i}(x) \coloneqq \mathbb{P}(x \mid \mathbf{w}_i, \mathcal{H}_i)$$

  In our case $\mathcal{H}_i$ is the choice of the discrete parameters and $\mathbf{w}_i \in \mathcal{H}_i$ is trained by optimizing the continuous parameters of that model.

- different models may have different dimensionality

Model comparison

- Let $\{n_x\}_{x \in X}$ be the measured frequencies of a data
- Let $\mathcal{H}_i$ be a model in a list of possible models
- Within a model there can be other trained parameters

$$\mathbf{w}_i \in \mathcal{H}_i,\ \mathbb{Q}_{\mathbf{w}_i}(x) \coloneqq \mathbb{P}(x \mid \mathbf{w}_i, \mathcal{H}_i)$$

  In our case $\mathcal{H}_i$ is the choice of the discrete parameters and $\mathbf{w}_i \in \mathcal{H}_i$ is trained by optimizing the continuous parameters of that model.

- different models may have different dimensionality
- Bayesian (evidence based) decision (MacKay, 2003):

$$\mathbb{P}(\mathcal{H}_i \mid \text{data}) \propto \mathbb{P}(\text{data} \mid \mathcal{H}_i) = \int_{\mathcal{H}_i} \underbrace{\mathbb{P}(\mathbf{w}_i \mid \mathcal{H}_i)}_{\text{uniform prior}} \prod_{x \in X} \left(\mathbb{Q}_{\mathbf{w}_i}(x)^{n_x}\right) \mathrm{d}\mathbf{w}_i$$

Estimating the evidence

$$\int_{\mathcal{H}_i} \frac{1}{\mathrm{Vol}(\mathcal{H}_i)} \prod_{x \in X} \left(\mathbb{Q}_{\mathbf{w}_i}(x)^{n_x}\right) \, \mathrm{d}\mathbf{w}_i =$$

Estimating the evidence

$$\int_{\mathcal{H}_i} \frac{1}{\mathrm{Vol}(\mathcal{H}_i)} \prod_{x \in X} \left(\mathbb{Q}_{\mathbf{w}_i}(x)^{n_x}\right) \, \mathrm{d}\mathbf{w}_i =$$

$$\frac{1}{\mathrm{Vol}(\mathcal{H}_i)} \int_{\mathcal{H}_i} \exp\left(\sum_{x \in X} n_x \cdot \ln \mathbb{Q}_{\mathbf{w}_i}(x)\right) \, \mathrm{d}\mathbf{w}_i$$

Estimating the evidence

$$\int_{\mathcal{H}_i} \frac{1}{\mathrm{Vol}(\mathcal{H}_i)} \prod_{x \in X} \left(\mathbb{Q}_{\mathbf{w}_i}(x)^{n_x}\right) \,\mathrm{d}\mathbf{w}_i =$$

$$\frac{1}{\mathrm{Vol}(\mathcal{H}_i)} \int_{\mathcal{H}_i} \exp\left(\sum_{x \in X} n_x \cdot \ln \mathbb{Q}_{\mathbf{w}_i}(x)\right) \,\mathrm{d}\mathbf{w}_i$$

$$f(\mathbf{w}_i) \coloneqq -\sum_{x \in X} \frac{n_x}{n} \ln \mathbb{Q}_{\mathbf{w}_i}(x)$$

Estimating the evidence

$$\int_{\mathcal{H}_i} \frac{1}{\mathrm{Vol}(\mathcal{H}_i)} \prod_{x \in X} \left(\mathbb{Q}_{\mathbf{w}_i}(x)^{n_x}\right) \,\mathrm{d}\mathbf{w}_i =$$

$$\frac{1}{\mathrm{Vol}(\mathcal{H}_i)} \int_{\mathcal{H}_i} \exp\left(\sum_{x \in X} n_x \cdot \ln \mathbb{Q}_{\mathbf{w}_i}(x)\right) \mathrm{d}\mathbf{w}_i$$

$$f(\mathbf{w}_i) \coloneqq -\sum_{x \in X} \frac{n_x}{n} \ln \mathbb{Q}_{\mathbf{w}_i}(x)$$

$$\frac{1}{\mathrm{Vol}(\mathcal{H}_i)} \int_{\mathcal{H}_i} e^{-n \cdot f(\mathbf{w}_i)} \,\mathrm{d}\mathbf{w}_i \approx \frac{1}{\mathrm{Vol}(\mathcal{H}_i)} \cdot e^{-n \cdot f(\mathbf{w}_i^*)} \cdot \frac{\left(\frac{2\pi}{n}\right)^{\frac{d}{2}}}{\sqrt{\det f''(\mathbf{w}_i^*)}}$$

Estimating the evidence

$$\int_{\mathcal{H}_i} \frac{1}{\mathrm{Vol}(\mathcal{H}_i)} \prod_{x \in X} \left(\mathbb{Q}_{\mathbf{w}_i}(x)^{n_x}\right) \mathrm{d}\mathbf{w}_i =$$

$$\frac{1}{\mathrm{Vol}(\mathcal{H}_i)} \int_{\mathcal{H}_i} \exp\left(\sum_{x \in X} n_x \cdot \ln \mathbb{Q}_{\mathbf{w}_i}(x)\right) \mathrm{d}\mathbf{w}_i$$

$$f(\mathbf{w}_i) := -\sum_{x \in X} \frac{n_x}{n} \ln \mathbb{Q}_{\mathbf{w}_i}(x)$$

$$\frac{1}{\mathrm{Vol}(\mathcal{H}_i)} \int_{\mathcal{H}_i} e^{-n \cdot f(\mathbf{w}_i)} \mathrm{d}\mathbf{w}_i \approx \frac{1}{\mathrm{Vol}(\mathcal{H}_i)} \cdot e^{-n \cdot f(\mathbf{w}_i^*)} \cdot \frac{\left(\frac{2\pi}{n}\right)^{\frac{d}{2}}}{\sqrt{\det f''(\mathbf{w}_i^*)}}$$

- $\mathbf{w}_i^* := \arg\min_{\mathbf{w}_i \in \mathcal{H}_i} f(\mathbf{w}_i)$

Estimating the evidence

$$\int_{\mathcal{H}_i} \frac{1}{\mathrm{Vol}(\mathcal{H}_i)} \prod_{x \in X} \left(\mathbb{Q}_{\mathbf{w}_i}(x)^{n_x} \right) \mathrm{d}\mathbf{w}_i =$$

$$\frac{1}{\mathrm{Vol}(\mathcal{H}_i)} \int_{\mathcal{H}_i} \exp\left(\sum_{x \in X} n_x \cdot \ln \mathbb{Q}_{\mathbf{w}_i}(x) \right) \mathrm{d}\mathbf{w}_i$$

$$f(\mathbf{w}_i) := -\sum_{x \in X} \frac{n_x}{n} \ln \mathbb{Q}_{\mathbf{w}_i}(x)$$

$$\frac{1}{\mathrm{Vol}(\mathcal{H}_i)} \int_{\mathcal{H}_i} e^{-n \cdot f(\mathbf{w}_i)} \mathrm{d}\mathbf{w}_i \approx \frac{1}{\mathrm{Vol}(\mathcal{H}_i)} \cdot e^{-n \cdot f(\mathbf{w}_i^*)} \cdot \frac{\left(\frac{2\pi}{n}\right)^{\frac{d}{2}}}{\sqrt{\det f''(\mathbf{w}_i^*)}}$$

- $\mathbf{w}_i^* := \arg\min_{\mathbf{w}_i \in \mathcal{H}_i} f(\mathbf{w}_i)$
- d is the dimension of $\mathcal{H}_i$ (number of free parameters)

Estimating the evidence

$$\int_{\mathcal{H}_i} \frac{1}{\mathrm{Vol}(\mathcal{H}_i)} \prod_{x \in X} \left(\mathbb{Q}_{\mathbf{w}_i}(x)^{n_x} \right) \, \mathrm{d}\mathbf{w}_i =$$

$$\frac{1}{\mathrm{Vol}(\mathcal{H}_i)} \int_{\mathcal{H}_i} \exp\left(\sum_{x \in X} n_x \cdot \ln \mathbb{Q}_{\mathbf{w}_i}(x) \right) \mathrm{d}\mathbf{w}_i$$

$$f(\mathbf{w}_i) := -\sum_{x \in X} \frac{n_x}{n} \ln \mathbb{Q}_{\mathbf{w}_i}(x)$$

$$\frac{1}{\mathrm{Vol}(\mathcal{H}_i)} \int_{\mathcal{H}_i} e^{-n \cdot f(\mathbf{w}_i)} \, \mathrm{d}\mathbf{w}_i \approx \frac{1}{\mathrm{Vol}(\mathcal{H}_i)} \cdot e^{-n \cdot f(\mathbf{w}_i^*)} \cdot \frac{\left(\frac{2\pi}{n}\right)^{\frac{d}{2}}}{\sqrt{\det f''(\mathbf{w}_i^*)}}$$

- $\mathbf{w}_i^* := \arg\min_{\mathbf{w}_i \in \mathcal{H}_i} f(\mathbf{w}_i)$
- d is the dimension of $\mathcal{H}_i$ (number of free parameters)
- f is cross entropy

Estimating the evidence

$$\int_{\mathcal{H}_i} \frac{1}{\mathrm{Vol}(\mathcal{H}_i)} \prod_{x \in X} \left(\mathbb{Q}_{\mathbf{w}_i}(x)^{n_x} \right) \, \mathrm{d}\mathbf{w}_i =$$

$$\frac{1}{\mathrm{Vol}(\mathcal{H}_i)} \int_{\mathcal{H}_i} \exp\left(\sum_{x \in X} n_x \cdot \ln \mathbb{Q}_{\mathbf{w}_i}(x) \right) \mathrm{d}\mathbf{w}_i$$

$$f(\mathbf{w}_i) := - \sum_{x \in X} \frac{n_x}{n} \ln \mathbb{Q}_{\mathbf{w}_i}(x)$$

$$\frac{1}{\mathrm{Vol}(\mathcal{H}_i)} \int_{\mathcal{H}_i} e^{-n \cdot f(\mathbf{w}_i)} \, \mathrm{d}\mathbf{w}_i \approx \frac{1}{\mathrm{Vol}(\mathcal{H}_i)} \cdot e^{-n \cdot f(\mathbf{w}_i^*)} \cdot \frac{\left(\frac{2\pi}{n}\right)^{\frac{d}{2}}}{\sqrt{\det f''(\mathbf{w}_i^*)}}$$

- $\mathbf{w}_i^* := \arg\min_{\mathbf{w}_i \in \mathcal{H}_i} f(\mathbf{w}_i)$
- d is the dimension of $\mathcal{H}_i$ (number of free parameters)
- f is cross entropy
- we take $-\frac{1}{n} \ln(\bullet)$ and also subtract the entropy of the data

Estimating the evidence

$$\int_{\mathcal{H}_i} \frac{1}{\mathrm{Vol}(\mathcal{H}_i)} \prod_{x \in X} \left(\mathbb{Q}_{\mathbf{w}_i}(x)^{n_x}\right) \mathrm{d}\mathbf{w}_i =$$

$$\frac{1}{\mathrm{Vol}(\mathcal{H}_i)} \int_{\mathcal{H}_i} \exp\left(\sum_{x \in X} n_x \cdot \ln \mathbb{Q}_{\mathbf{w}_i}(x)\right) \mathrm{d}\mathbf{w}_i$$

$$f(\mathbf{w}_i) := -\sum_{x \in X} \frac{n_x}{n} \ln \mathbb{Q}_{\mathbf{w}_i}(x)$$

$$\frac{1}{\mathrm{Vol}(\mathcal{H}_i)} \int_{\mathcal{H}_i} e^{-n \cdot f(\mathbf{w}_i)} \mathrm{d}\mathbf{w}_i \approx \frac{1}{\mathrm{Vol}(\mathcal{H}_i)} \cdot e^{-n \cdot f(\mathbf{w}_i^*)} \cdot \frac{\left(\frac{2\pi}{n}\right)^{\frac{d}{2}}}{\sqrt{\det f''(\mathbf{w}_i^*)}}$$

- $\mathbf{w}_i^* := \arg\min_{\mathbf{w}_i \in \mathcal{H}_i} f(\mathbf{w}_i)$
- d is the dimension of $\mathcal{H}_i$ (number of free parameters)
- f is cross entropy
- we take $-\frac{1}{n}\ln(\bullet)$ and also subtract the entropy of the data
    - none of which changes the relative order of the models

Estimating the evidence

$$\int_{\mathcal{H}_i} \frac{1}{\mathrm{Vol}(\mathcal{H}_i)} \prod_{x \in X} \left(\mathbb{Q}_{\mathbf{w}_i}(x)^{n_x}\right) \mathrm{d}\mathbf{w}_i =$$

$$\frac{1}{\mathrm{Vol}(\mathcal{H}_i)} \int_{\mathcal{H}_i} \exp\left(\sum_{x \in X} n_x \cdot \ln \mathbb{Q}_{\mathbf{w}_i}(x)\right) \mathrm{d}\mathbf{w}_i$$

$$f(\mathbf{w}_i) := -\sum_{x \in X} \frac{n_x}{n} \ln \mathbb{Q}_{\mathbf{w}_i}(x)$$

$$\frac{1}{\mathrm{Vol}(\mathcal{H}_i)} \int_{\mathcal{H}_i} e^{-n \cdot f(\mathbf{w}_i)} \mathrm{d}\mathbf{w}_i \approx \frac{1}{\mathrm{Vol}(\mathcal{H}_i)} \cdot e^{-n \cdot f(\mathbf{w}_i^*)} \cdot \frac{\left(\frac{2\pi}{n}\right)^{\frac{d}{2}}}{\sqrt{\det f''(\mathbf{w}_i^*)}}$$

- $\mathbf{w}_i^* := \arg\min_{\mathbf{w}_i \in \mathcal{H}_i} f(\mathbf{w}_i)$
- d is the dimension of $\mathcal{H}_i$ (number of free parameters)
- f is cross entropy
- we take $-\frac{1}{n}\ln(\bullet)$ and also subtract the entropy of the data
    - none of which changes the relative order of the models
    - this way the theoretical minimum is 0

Augmented model

$$\mathbb{P}(\text{data} \mid \mathcal{H}_i) = \int_{\mathcal{H}_i} \underbrace{\mathbb{P}(\mathbf{w}_i \mid \mathcal{H}_i)}_{\text{uniform prior}} \prod_{x \in X} \left(\mathbb{Q}_{\mathbf{w}_i}(x)^{n_x} \right) \mathrm{d}\mathbf{w}_i$$

- One can see that $\mathbb{Q}_{\mathbf{w}_i}(x) = 0$ is unacceptable

Augmented model

$$\mathbb{P}(\text{data} \mid \mathcal{H}_i) = \int_{\mathcal{H}_i} \underbrace{\mathbb{P}(\mathbf{w}_i \mid \mathcal{H}_i)}_{\text{uniform prior}} \prod_{x \in X} \left(\mathbb{Q}_{\mathbf{w}_i}(x)^{n_x}\right) \mathrm{d}\mathbf{w}_i$$

- One can see that $\mathbb{Q}_{\mathbf{w}_i}(x) = 0$ is unacceptable
- We introduced a dummy auxiliary model to capture the probabilities of the short sentences (shorter than the starting valency)

Augmented model

$$\mathbb{P}(\text{data} \mid \mathcal{H}_i) = \int_{\mathcal{H}_i} \underbrace{\mathbb{P}(\mathbf{w}_i \mid \mathcal{H}_i)}_{\text{uniform prior}} \prod_{x \in X} \left(\mathbb{Q}_{\mathbf{w}_i}(x)^{n_x}\right) \, \mathrm{d}\mathbf{w}_i$$

- One can see that $\mathbb{Q}_{\mathbf{w}_i}(x) = 0$ is unacceptable
- We introduced a dummy auxiliary model to capture the probabilities of the short sentences (shorter than the starting valency)
- $$\overline{\mathbb{Q}}_{\mathbf{w}_i,\mathbf{q}}(x) := \begin{cases} \lambda \cdot \mathbb{Q}_{\mathbf{w}_i}(x) & \text{if } \mathbb{Q}_{\mathbf{w}_i}(x) > 0 \\ (1-\lambda) \cdot q_x & \text{if } n_x > 0, \mathbb{Q}_{\mathbf{w}_i}(x) = 0 \end{cases}$$

  where q_x is also a trained parameter and

$$\lambda = \mathbb{P}(\mathbb{Q}_{\mathbf{w}_i} > 0) \qquad \text{covered probability}$$
$$1 - \lambda = \mathbb{P}(\mathbb{Q}_{\mathbf{w}_i} = 0) \qquad \text{uncovered probability}$$

Model comparison – final

$$-\lambda \cdot \ln \lambda + \overbrace{\sum_{x \in X \cap \mathrm{supp}(\mathcal{H}_i)} p_x \cdot \ln \frac{p_x}{\mathbb{Q}_{\mathbf{w}_i^*}(x)}} + \frac{d}{2n} \cdot \ln \frac{n}{2\pi} +$$

$$\frac{1}{n} \cdot \ln \left(\mathrm{Vol}(\mathcal{H}_i) \cdot \mathrm{Vol}(\text{aux. model}) \right) +$$

$$\frac{1}{2n} \cdot \ln \left(\det \left(\text{model Hessian} \right) \cdot \det \left(\text{aux. model Hessian} \right) \right)$$

Model comparison – final

$$-\lambda \cdot \ln \lambda + \overbrace{\sum_{x \in X \cap \mathrm{supp}(\mathcal{H}_i)} p_x \cdot \ln \frac{p_x}{\mathbb{Q}_{\mathbf{w}_i^*}(x)}} + \frac{d}{2n} \cdot \ln \frac{n}{2\pi} +$$

$$\frac{1}{n} \cdot \ln \left(\mathrm{Vol}(\mathcal{H}_i) \cdot \mathrm{Vol}(\text{aux. model})\right) +$$

$$\frac{1}{2n} \cdot \ln \left(\det\left(\text{model Hessian}\right) \cdot \det\left(\text{aux. model Hessian}\right)\right)$$

- λ is the covered probability

Model comparison – final

$$-\lambda \cdot \ln \lambda + \overbrace{\sum_{x \in X \cap \mathrm{supp}(\mathcal{H}_i)} p_x \cdot \ln \frac{p_x}{\mathbb{Q}_{\mathbf{w}_i^*}(x)}} + \frac{d}{2n} \cdot \ln \frac{n}{2\pi} +$$

$$\frac{1}{n} \cdot \ln \left(\mathrm{Vol}(\mathcal{H}_i) \cdot \mathrm{Vol}(\text{aux. model})\right) +$$

$$\frac{1}{2n} \cdot \ln \left(\det\left(\text{model Hessian}\right) \cdot \det\left(\text{aux. model Hessian}\right)\right)$$

- λ is the covered probability
- $\mathbf{w}_i^* \coloneqq \arg\min_{\mathbf{w}_i \in \mathcal{H}_i} KL(\mathbb{P} \parallel \mathbb{Q}_{\mathbf{w}_i})$

Model comparison – final

$$-\lambda \cdot \ln \lambda + \overbrace{\sum_{x \in X \cap \mathrm{supp}(\mathcal{H}_i)} p_x \cdot \ln \frac{p_x}{\mathbb{Q}_{\mathbf{w}_i^*}(x)}} + \frac{d}{2n} \cdot \ln \frac{n}{2\pi} +$$

$$\frac{1}{n} \cdot \ln \left(\mathrm{Vol}(\mathcal{H}_i) \cdot \mathrm{Vol}(\text{aux. model})\right) +$$

$$\frac{1}{2n} \cdot \ln \left(\det\left(\text{model Hessian}\right) \cdot \det\left(\text{aux. model Hessian}\right)\right)$$

- λ is the covered probability
- $\mathbf{w}_i^* \coloneqq \arg\min_{\mathbf{w}_i \in \mathcal{H}_i} KL(\mathbb{P} \parallel \mathbb{Q}_{\mathbf{w}_i})$
- n is the size of the dataset

Model comparison – final

$$-\lambda \cdot \ln \lambda + \overbrace{\sum_{x \in X \cap \mathrm{supp}(\mathcal{H}_i)} p_x \cdot \ln \frac{p_x}{\mathbb{Q}_{\mathbf{w}_i^*}(x)}} + \frac{d}{2n} \cdot \ln \frac{n}{2\pi} +$$
$$\frac{1}{n} \cdot \ln \left(\mathrm{Vol}(\mathcal{H}_i) \cdot \mathrm{Vol}(\text{aux. model})\right) +$$
$$\frac{1}{2n} \cdot \ln \left(\det\left(\text{model Hessian}\right) \cdot \det\left(\text{aux. model Hessian}\right)\right)$$

- λ is the covered probability
- $\mathbf{w}_i^* \coloneqq \arg\min_{\mathbf{w}_i \in \mathcal{H}_i} KL(\mathbb{P} \parallel \mathbb{Q}_{\mathbf{w}_i})$
- n is the size of the dataset
  - number of sentences

Model comparison – final

$$-\lambda \cdot \ln \lambda + \overbrace{\sum_{x \in X \cap \mathrm{supp}(\mathcal{H}_i)} p_x \cdot \ln \frac{p_x}{\mathbb{Q}_{\mathbf{w}_i^*}(x)}} + \frac{d}{2n} \cdot \ln \frac{n}{2\pi} +$$

$$\frac{1}{n} \cdot \ln \left(\mathrm{Vol}(\mathcal{H}_i) \cdot \mathrm{Vol}(\text{aux. model}) \right) +$$

$$\frac{1}{2n} \cdot \ln \left(\det \left(\text{model Hessian} \right) \cdot \det \left(\text{aux. model Hessian} \right) \right)$$

- λ is the covered probability
- $\mathbf{w}_i^* := \arg\min_{\mathbf{w}_i \in \mathcal{H}_i} KL(\mathbb{P} \parallel \mathbb{Q}_{\mathbf{w}_i})$
- n is the size of the dataset
  - number of sentences
- d is the number of model parameters (including auxiliary model)

Model comparison – final

$$-\lambda \cdot \ln \lambda + \overbrace{\sum_{x \in X \cap \mathrm{supp}(\mathcal{H}_i)} p_x \cdot \ln \frac{p_x}{\mathbb{Q}_{\mathbf{w}_i^*}(x)}} + \frac{d}{2n} \cdot \ln \frac{n}{2\pi} +$$

$$\frac{1}{n} \cdot \ln \left(\mathrm{Vol}(\mathcal{H}_i) \cdot \mathrm{Vol}(\text{aux. model})\right) +$$

$$\frac{1}{2n} \cdot \ln \left(\det\left(\text{model Hessian}\right) \cdot \det\left(\text{aux. model Hessian}\right)\right)$$

- λ is the covered probability
- $\mathbf{w}_i^* \coloneqq \arg\min_{\mathbf{w}_i \in \mathcal{H}_i} KL(\mathbb{P} \parallel \mathbb{Q}_{\mathbf{w}_i})$
- n is the size of the dataset
  - number of sentences
- d is the number of model parameters (including auxiliary model)
- the model volume is the volume of the parameter space

Model comparison – final

$$-\lambda \cdot \ln \lambda + \overbrace{\sum_{x \in X \cap \mathrm{supp}(\mathcal{H}_i)} p_x \cdot \ln \frac{p_x}{\mathbb{Q}_{\mathbf{w}_i^*}(x)}} + \frac{d}{2n} \cdot \ln \frac{n}{2\pi} +$$
$$\frac{1}{n} \cdot \ln \left(\mathrm{Vol}(\mathcal{H}_i) \cdot \mathrm{Vol}(\text{aux. model}) \right) +$$
$$\frac{1}{2n} \cdot \ln \left(\det(\text{model Hessian}) \cdot \det(\text{aux. model Hessian}) \right)$$

- λ is the covered probability
- $\mathbf{w}_i^* \coloneqq \arg\min_{\mathbf{w}_i \in \mathcal{H}_i} KL(\mathbb{P} \parallel \mathbb{Q}_{\mathbf{w}_i})$
- n is the size of the dataset
  - number of sentences
- d is the number of model parameters (including auxiliary model)
- the model volume is the volume of the parameter space
  - probabilistic simplexes

Model comparison – final

$$-\lambda \cdot \ln \lambda + \overbrace{\sum_{x \in X \cap \mathrm{supp}(\mathcal{H}_i)} p_x \cdot \ln \frac{p_x}{\mathbb{Q}_{\mathbf{w}_i^*}(x)}} + \frac{d}{2n} \cdot \ln \frac{n}{2\pi} +$$

$$\frac{1}{n} \cdot \ln \left(\mathrm{Vol}(\mathcal{H}_i) \cdot \mathrm{Vol}(\text{aux. model})\right) +$$

$$\frac{1}{2n} \cdot \ln \left(\det \left(\text{model Hessian}\right) \cdot \det \left(\text{aux. model Hessian}\right)\right)$$

- λ is the covered probability
- $\mathbf{w}_i^* := \arg\min_{\mathbf{w}_i \in \mathcal{H}_i} KL(\mathbb{P} \parallel \mathbb{Q}_{\mathbf{w}_i})$
- n is the size of the dataset
  - number of sentences
- d is the number of model parameters (including auxiliary model)
- the model volume is the volume of the parameter space
  - probabilistic simplexes
- the determinant of the Hessian can be considered as volume

Model comparison – beyond

- There are three type of terms in the final formula

Model comparison – beyond

- There are three type of terms in the final formula
  - constant in n

Model comparison – beyond

- There are three type of terms in the final formula
  - constant in n
  - proportional to $\frac{1}{n}$

Model comparison – beyond

- There are three type of terms in the final formula
  - constant in n
  - proportional to $\frac{1}{n}$
  - $\frac{\ln n}{n}$

Model comparison – beyond

- There are three type of terms in the final formula
  - constant in n
  - proportional to $\frac{1}{n}$
  - $\frac{\ln n}{n}$
- as $n \to \infty$ only the constant terms remain

Model comparison – beyond

- There are three type of terms in the final formula
  - constant in n
  - proportional to $\frac{1}{n}$
  - $\frac{\ln n}{n}$
- as $n \to \infty$ only the constant terms remain
  - and the model size is irrelevant

Model comparison – beyond

- There are three type of terms in the final formula
  - constant in n
  - proportional to $\frac{1}{n}$
  - $\frac{\ln n}{n}$
- as $n \to \infty$ only the constant terms remain
  - and the model size is irrelevant
  - this causes overfitting

Model comparison – beyond

- There are three type of terms in the final formula
  - constant in n
  - proportional to $\dfrac{1}{n}$
  - $\dfrac{\ln n}{n}$
- as $n \to \infty$ only the constant terms remain
  - and the model size is irrelevant
  - this causes overfitting
- if n is small then the Laplace integration doesn't even work

Model comparison – beyond

- There are three type of terms in the final formula
  - constant in n
  - proportional to $\frac{1}{n}$
  - $\frac{\ln n}{n}$
- as $n \to \infty$ only the constant terms remain
  - and the model size is irrelevant
  - this causes overfitting
- if n is small then the Laplace integration doesn't even work
  - also the data might be unreliable

Model comparison – beyond

- There are three type of terms in the final formula
  - constant in n
  - proportional to $\frac{1}{n}$
  - $\frac{\ln n}{n}$
- as $n \to \infty$ only the constant terms remain
  - and the model size is irrelevant
  - this causes overfitting
- if n is small then the Laplace integration doesn't even work
  - also the data might be unreliable
- we want to avoid optimizing for n

Model comparison – beyond

- There are three type of terms in the final formula
  - constant in n
  - proportional to $\frac{1}{n}$
  - $\frac{\ln n}{n}$
- as $n \to \infty$ only the constant terms remain
  - and the model size is irrelevant
  - this causes overfitting
- if n is small then the Laplace integration doesn't even work
  - also the data might be unreliable
- we want to avoid optimizing for n
  - “optimal corpus size”

Model comparison – beyond

- There are three type of terms in the final formula
  - constant in n
  - proportional to $\frac{1}{n}$
  - $\frac{\ln n}{n}$
- as $n \to \infty$ only the constant terms remain
  - and the model size is irrelevant
  - this causes overfitting
- if n is small then the Laplace integration doesn't even work
  - also the data might be unreliable
- we want to avoid optimizing for n
  - "optimal corpus size"
- we want stable result as $n \to \infty$

Inherent noise

- we defined the following quantity as a general measure of dissimilarity

Inherent noise

- we defined the following quantity as a general measure of dissimilarity
  - generalized Kullback–Leibler divergence

$$[-\lambda \cdot \ln \lambda]_{\lambda=\mathbb{P}(\mathrm{supp}(\mathbb{P})\cap \mathrm{supp}(\mathbb{Q}))} + \sum_{x\in \mathrm{supp}(\mathbb{P})\cap \mathrm{supp}(\mathbb{Q})} \mathbb{P}(x) \ln \frac{\mathbb{P}(x)}{\mathbb{Q}(x)}$$

Inherent noise

- we defined the following quantity as a general measure of dissimilarity
  - generalized Kullback–Leibler divergence

$$\left[-\lambda \cdot \ln \lambda\right]_{\lambda=\mathbb{P}(\operatorname{supp}(\mathbb{P})\cap\operatorname{supp}(\mathbb{Q}))} + \sum_{x\in\operatorname{supp}(\mathbb{P})\cap\operatorname{supp}(\mathbb{Q})} \mathbb{P}(x) \ln \frac{\mathbb{P}(x)}{\mathbb{Q}(x)}$$

- measured this quantity on two disjoint subset of the data (Kornai et al., 2013)
- modified the final evidence formula to tolerate for any error within inherent noise
  - if a model fits within inherent noise, then it is considered a perfect fit

Inherent noise

- we defined the following quantity as a general measure of dissimilarity
  - generalized Kullback–Leibler divergence

$$[-\lambda \cdot \ln \lambda]_{\lambda=\mathbb{P}(\mathrm{supp}(\mathbb{P})\cap\mathrm{supp}(\mathbb{Q}))} + \sum_{x\in\mathrm{supp}(\mathbb{P})\cap\mathrm{supp}(\mathbb{Q})} \mathbb{P}(x) \ln \frac{\mathbb{P}(x)}{\mathbb{Q}(x)}$$

- measured this quantity on two disjoint subset of the data (Kornai et al., 2013)
- modified the final evidence formula to tolerate for any error within inherent noise
  - if a model fits within inherent noise, then it is considered a perfect fit
- in this case the n-dependent terms can contribute in a meaningful way

Inherent noise

- we defined the following quantity as a general measure of dissimilarity
  - generalized Kullback–Leibler divergence

$$[-\lambda \cdot \ln \lambda]_{\lambda=\mathbb{P}(\mathrm{supp}(\mathbb{P})\cap\mathrm{supp}(\mathbb{Q}))} + \sum_{x\in\mathrm{supp}(\mathbb{P})\cap\mathrm{supp}(\mathbb{Q})} \mathbb{P}(x) \ln \frac{\mathbb{P}(x)}{\mathbb{Q}(x)}$$

- measured this quantity on two disjoint subset of the data (Kornai et al., 2013)
- modified the final evidence formula to tolerate for any error within inherent noise
  - if a model fits within inherent noise, then it is considered a perfect fit
- in this case the n-dependent terms can contribute in a meaningful way
- this solves the accuracy—complexity trade-off

Inherent noise

- we defined the following quantity as a general measure of dissimilarity
  - generalized Kullback–Leibler divergence

$$\left[-\lambda \cdot \ln \lambda\right]_{\lambda=\mathbb{P}(\mathrm{supp}(\mathbb{P})\cap\mathrm{supp}(\mathbb{Q}))} + \sum_{x\in\mathrm{supp}(\mathbb{P})\cap\mathrm{supp}(\mathbb{Q})} \mathbb{P}(x)\ln\frac{\mathbb{P}(x)}{\mathbb{Q}(x)}$$

- measured this quantity on two disjoint subset of the data (Kornai et al., 2013)
- modified the final evidence formula to tolerate for any error within inherent noise
  - if a model fits within inherent noise, then it is considered a perfect fit
- in this case the n-dependent terms can contribute in a meaningful way
- this solves the accuracy—complexity trade-off
- our method works for distributions with unequal support
  - the augmented model is actively contributing to the final decision

Experiments

- We have a fairly general formula for comparing various models

Experiments

- We have a fairly general formula for comparing various models
- Trained several models with several hyper-parameters

Experiments

- We have a fairly general formula for comparing various models
- Trained several models with several hyper-parameters
    - Random walk model

Experiments

- We have a fairly general formula for comparing various models
- Trained several models with several hyper-parameters
  - Random walk model
    - order: maximum allowed upward steps

Experiments

- We have a fairly general formula for comparing various models
- Trained several models with several hyper-parameters
  - Random walk model
    - order: maximum allowed upward steps
    - k-mixture: convex linear combination of several random walks starting from several different valency

Experiments

- We have a fairly general formula for comparing various models
- Trained several models with several hyper-parameters
  - Random walk model
    - order: maximum allowed upward steps
    - k-mixture: convex linear combination of several random walks starting from several different valency
    - including augmented model for the sentences which are shorter than the starting valency

Experiments

- We have a fairly general formula for comparing various models
- Trained several models with several hyper-parameters
  - Random walk model
    - order: maximum allowed upward steps
    - k-mixture: convex linear combination of several random walks starting from several different valency
    - including augmented model for the sentences which are shorter than the starting valency
  - Binned model

Experiments

- We have a fairly general formula for comparing various models
- Trained several models with several hyper-parameters
  - Random walk model
    - order: maximum allowed upward steps
    - k-mixture: convex linear combination of several random walks starting from several different valency
    - including augmented model for the sentences which are shorter than the starting valency
  - Binned model
    - take bins $[1, b_1), [b_1, b_2), [b_{m-1}, \infty)$

Experiments

- We have a fairly general formula for comparing various models
- Trained several models with several hyper-parameters
  - Random walk model
    - order: maximum allowed upward steps
    - k-mixture: convex linear combination of several random walks starting from several different valency
    - including augmented model for the sentences which are shorter than the starting valency
  - Binned model
    - take bins $[1, b_1), [b_1, b_2), [b_{m-1}, \infty)$
    - the modeled distribution is uniform within one bin

Experiments

- We have a fairly general formula for comparing various models
- Trained several models with several hyper-parameters
  - Random walk model
    - order: maximum allowed upward steps
    - k-mixture: convex linear combination of several random walks starting from several different valency
    - including augmented model for the sentences which are shorter than the starting valency
  - Binned model
    - take bins $[1, b_1), [b_1, b_2), [b_{m-1}, \infty)$
    - the modeled distribution is uniform within one bin
    - the probability of a bin is trained

Experiments

- We have a fairly general formula for comparing various models
- Trained several models with several hyper-parameters
  - Random walk model
    - order: maximum allowed upward steps
    - k-mixture: convex linear combination of several random walks starting from several different valency
    - including augmented model for the sentences which are shorter than the starting valency
  - Binned model
    - take bins $[1, b_1), [b_1, b_2), [b_{m-1}, \infty)$
    - the modeled distribution is uniform within one bin
    - the probability of a bin is trained
    - the bins themselves are the discrete parameters

Experiments

- We have a fairly general formula for comparing various models
- Trained several models with several hyper-parameters
  - Random walk model
    - order: maximum allowed upward steps
    - k-mixture: convex linear combination of several random walks starting from several different valency
    - including augmented model for the sentences which are shorter than the starting valency
  - Binned model
    - take bins $[1, b_1), [b_1, b_2), [b_{m-1}, \infty)$
    - the modeled distribution is uniform within one bin
    - the probability of a bin is trained
    - the bins themselves are the discrete parameters
  - Sichel model

Experiments

- We have a fairly general formula for comparing various models
- Trained several models with several hyper-parameters
  - Random walk model
    - order: maximum allowed upward steps
    - k-mixture: convex linear combination of several random walks starting from several different valency
    - including augmented model for the sentences which are shorter than the starting valency
  - Binned model
    - take bins $[1, b_1), [b_1, b_2), [b_{m-1}, \infty)$
    - the modeled distribution is uniform within one bin
    - the probability of a bin is trained
    - the bins themselves are the discrete parameters
  - Sichel model
    - we trained the parameters α and θ

Experiments

- We have a fairly general formula for comparing various models
- Trained several models with several hyper-parameters
  - Random walk model
    - order: maximum allowed upward steps
    - k-mixture: convex linear combination of several random walks starting from several different valency
    - including augmented model for the sentences which are shorter than the starting valency
  - Binned model
    - take bins $[1, b_1), [b_1, b_2), [b_{m-1}, \infty)$
    - the modeled distribution is uniform within one bin
    - the probability of a bin is trained
    - the bins themselves are the discrete parameters
  - Sichel model
    - we trained the parameters α and θ
    - γ was a model parameter (we couldn't back-propagate to the subscript parameter of the Bessel functions $K_\gamma(z)$)

Results

- The binned and Sichel models were rarely within inherent noise
  - the binned model fits well for many bins, but it has a lot more parameter than the parametric morels
  - the Sichel model fits only the binned data (when the datapoints are aggregated into 4-5 long bins)
  - this was actually mentioned by Sichel, although it was way better then its predecessors
- The random walk model always wins
- Never use more than one step upwards

Results

dataset	best parameters for various n values				
	1k	10k	100k	1M	1G
BNC-A	*o3.k1-5*	*o3.k2-5*	o1.k4.5	o1.k4.5	o1.k4.5
BNC-B	*o3.k1-5*	*o3.k1-5*	o1.k1.5	o1.k1.5	o1.k1.5
BNC-C	o3.k2-5	o3.k2-5	o3.k2-5	o1.k1.4	o1.k1.4
BNC-D	o3.k2.3.5	o3.k2.3.5	o3.k2.3.5	o1.k2	o1.k2
BNC-E	*o3.k1.3-5*	*o3.k1.3-5*	o1.k2.5	o1.k2.5	o1.k2.5
BNC-F	o3.k3.4.5	o3.k3.4.5	o3.k3.4.5	o1.k3	o1.k3
BNC-G	o3.k1-5	o3.k1-5	o1.k2.5	o1.k2.5	o1.k2.5
BNC-H	*o3.k2.4.5*	o3.k3.4.5	o1.k4	o1.k4	o1.k4
BNC-J	o3.k2.3.4	o3.k2.3.4	o3.k2.5	o1.k2	o1.k2
BNC-K	o3.k1-5	o3.k1-5	o1.k2	o1.k2	o1.k2
UMBC	*o3.k1.3-5*	*o3.k1.3-5*	o1.k2.5	o1.k2.5	o1.k2.5

Table: Best models I.

Results

dataset	best parameters for various n values				
	1k	10k	100k	1M	1G
Catalan	*o3.k2-5*	*o3.k2-5*	o1.k2.5	o1.k2.5	o1.k2.5
Croatian	o3.k3.4.5	o3.k3.4.5	o1.k2.5	o1.k2.5	o1.k2.5
Czech	*o3.k4.5*	o3.k1.3.5	o1.k2.5	o1.k2.5	o1.k2.5
Danish	*o3.k1-5*	o3.k1.3.5	o1.k2.5	o1.k2.5	o1.k2.5
Dutch	*o3.k1-5*	*o3.k3.4.5*	o1.k2.5	o1.k2.5	o1.k2.5
Finnish	*o3.k1.3.5*	o1.k2.4	o1.k2.4	o1.k2.4	o1.k2.4
Indonesian	o3.k1-5	o3.k1-5	o1.k2.5	o1.k2.5	o1.k2.5
Lithuanian	*o3.k2.3.4*	*o3.k2.3.4*	o1.k2.3	o1.k2.3	o1.k2.3
Bokmål	o3.k2.4.5	o3.k2.4.5	o1.k2.5	o1.k2.5	o1.k2.5
Nynorsk	*o3.k1-5*	o1.k2.5	o1.k2.5	o1.k2.5	o1.k2.5

Table: Best models II.

Results

dataset	best parameters for various n values				
	1k	10k	100k	1M	1G
Polish	o3.k2-5	o3.k2-5	o3.k2-5	o3.k2-5	o1.k2.5
Portuguese	o3.k2.3.5	o3.k2.3.5	o1.k2	o1.k2	o1.k2
Romanian	o3.k1.3-5	o3.k1.3-5	o1.k5	o1.k5	o1.k5
Serbian.sh	*o3.k1.2.4.5*	o3.k2.3.5	o1.k2.5	o1.k2.5	o1.k2.5
Serbian.sr	*o3.k2-5*	o3.k2.3.4	o1.k2.5	o1.k2.5	o1.k2.5
Slovak	*o3.k2.4.5*	o3.k2-5	o1.k2.5	o1.k2.5	o1.k2.5
Spanish	*o3.k2.4.5*	o1.k2.3	o1.k2.3	o1.k2.3	o1.k2.3
Swedish	o1.k2.4	o1.k2.4	o1.k2.4	o1.k2.4	o1.k2.4

Table: Best models III.

Results

dataset	best parameters for various n values				
	1k	10k	100k	1M	1G
BNC-A	*o3.k1-5*	o1.k4.5	o1.k4.5	o1.k1-5	o1.k1-5
BNC-B	*o3.k1-5*	o1.k2.3.5	o2.k4.5	o2.k4.5	o2.k4.5
BNC-C	o3.k2-5	o1.k2.4.5	o1.k2.4.5	o1.k2.4.5	o1.k2.4.5
BNC-D	o3.k3.4	o1.k2.5	o2.k2.5	o2.k2.5	o2.k2.5
BNC-E	*o3.k1.3-5*	o1.k4.5	o1.k4.5	o1.k4.5	o1.k4.5
BNC-F	o3.k3-5	o1.k2.4.5	o1.k2.4.5	o1.k2.4.5	o1.k2.4.5
BNC-G	o3.k1-5	o1.k4.5	o1.k2.4.5	o1.k2.4.5	o2.k2.4.5
BNC-H	o3.k3-5	o1.k4.5	o2.k2.4.5	o2.k2.4.5	o2.k2.4.5
BNC-J	o3.k1-5	o1.k2.4.5	o1.k2.4.5	o1.k2.4.5	o1.k2.4.5
BNC-K	o3.k2-5	o3.k2-5	o1.k2.4.5	o1.k2.4.5	o1.k2.4.5
UMBC	*o3.k1.3-5*	o1.k2.4	o1.k2.4.5	o1.k2.4.5	o1.k2.4.5

Table: Without tolerance for inherent noise I.

Results

dataset	best parameters for various n values				
	1k	10k	100k	1M	1G
Catalan	*o3.k2-5*	*o3.k2-5*	o1.k2.4	o1.k1.3-5	o1.k1.3-5
Croatian	o3.k3-5	o1.k2.3	o1.k2.3	o1.k3-5	o1.k3-5
Czech	o3.k2-5	o3.k3-5	o1.k2.3	o1.k1.3-5	o1.k1.3-5
Danish	*o3.k1-5*	o1.k2.3	o1.k1.2.4.5	o1.k1.2.4.5	o3.k2-5
Dutch	*o3.k1-5*	o1.k2.4	o1.k3.4	o1.k1-5	o1.k1-5
Finnish	*o3.k1.3.5*	o1.k1.3.4	o1.k1.3.4	o1.k1.3-5	o1.k1.3-5
Indonesian	o3.k1-5	o1.k3.5	o1.k3-5	o1.k3-5	o1.k3-5
Lithuanian	*o3.k2.3.4*	o1.k2.3	o1.k2-5	o1.k2-5	o1.k2-5
Bokmål	o3.k2.4.5	o3.k2.4.5	o1.k1.3-5	o1.k1.3-5	o1.k1.3-5
Nynorsk	*o3.k1-5*	o1.k2.4.5	o1.k1-5	o1.k1-5	o1.k1-5

Table: Without tolerance for inherent noise II.

Results

dataset	best parameters for various n values				
	1k	10k	100k	1M	1G
Polish	o3.k2-5	o3.k2-5	o1.k1.4.5	o1.k2-5	o1.k2-5
Portuguese	o3.k2.4.5	o1.k2.3	o1.k3.4	o1.k3.4	o1.k3.4
Romanian	o3.k2.4.5	o1.k2.4	o1.k2.3.4	o1.k2.3.4	o1.k2.3.4
Serbian.sh	*o3.k1.2.4.5*	o1.k2.4	o1.k3.4	o1.k2-5	o1.k2-5
Serbian.sr	*o3.k2-5*	o1.k4.5	o1.k4.5	o1.k4.5	o1.k4.5
Slovak	*o3.k2.4.5*	o1.k2.3	o1.k1.3-5	o1.k1.3-5	o1.k1.3-5
Spanish	*o3.k2.4.5*	o1.k2.3	o1.k1.3.5	o1.k1.3.5	o1.k1.3.5
Swedish	o1.k2.3	o1.k2.3	o1.k1-5	o1.k1-5	o1.k1-5

Table: Without tolerance for inherent noise III.

Visual fits

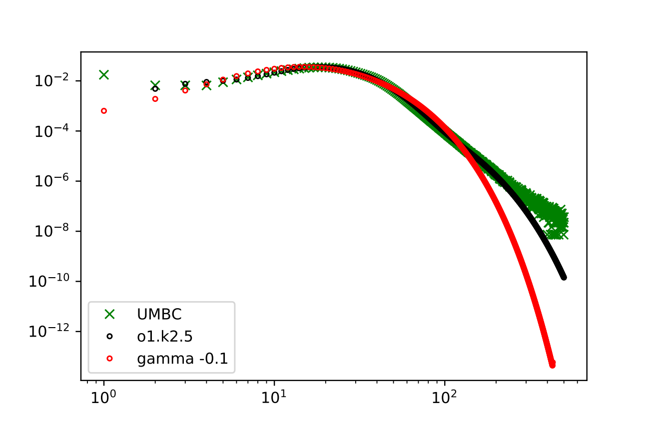


Figure: Random walk fits well, Sichel not

Visual fits

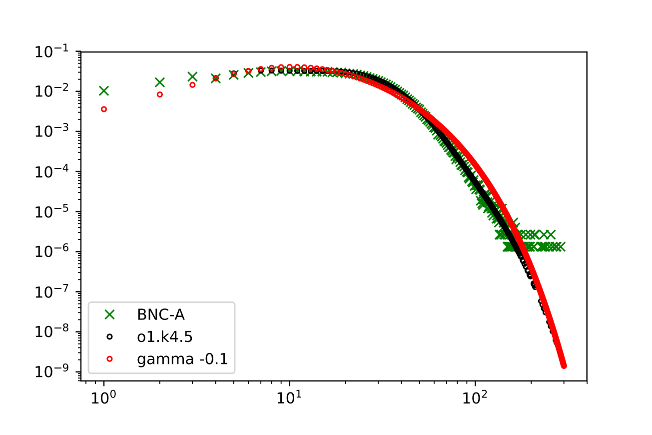


Figure: Random walk fits well, Sichel not

Visual fits

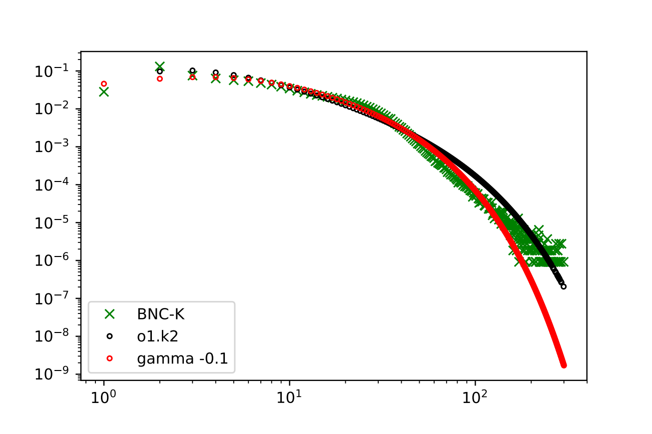


Figure: A rare case when Sichel fits within noise

Random walk fits best

dataset	Sichel	binned	random walk	inherent noise
BNC-A	*3.130e-2*	*1.489e-2*	4.409e-4	9.847e-4
BNC-B	*5.555e-2*	*1.274e-2*	7.215e-3	7.741e-3
BNC-C	*4.335e-2*	*1.431e-2*	6.989e-3	9.494e-3
BNC-D	*9.917e-2*	8.387e-2	5.945e-2	8.510e-2
BNC-E	*6.303e-2*	*2.251e-2*	4.353e-3	5.000e-3
BNC-F	*2.706e-2*	2.196e-2	2.270e-2	2.630e-2
BNC-G	*2.205e-2*	*1.495e-2*	5.762e-3	9.199e-3
BNC-H	*4.095e-2*	3.265e-2	3.106e-2	3.385e-2
BNC-J	2.665e-2	6.854e-2	2.946e-2	7.940e-2
BNC-K	6.525e-2	1.388e-1	3.899e-2	2.134e-1
UMBC	*6.320e-2*	*2.615e-2*	1.390e-3	2.442e-3

Table: Best of the models and their fit I.

Random walk fits best

dataset	Sichel	binned	random walk	inherent noise
Catalan	*1.227e-1*	*6.102e-2*	9.382e-4	1.751e-3
Croatian	*1.027e-1*	*4.604e-2*	2.063e-3	5.616e-3
Czech	*5.783e-2*	*3.687e-2*	2.563e-3	5.147e-3
Danish	*1.511e-1*	*3.072e-2*	2.772e-3	7.557e-3
Dutch	*1.844e-1*	*3.447e-2*	1.391e-3	2.408e-3
Finnish	*9.712e-2*	*2.830e-2*	1.659e-3	1.946e-3
Indonesian	*9.896e-2*	*5.017e-2*	1.390e-3	1.231e-2
Lithuanian	*1.617e-1*	*3.113e-2*	6.637e-4	1.184e-3
Bokmål	*1.028e-1*	*3.332e-2*	3.515e-3	3.564e-3
Nynorsk	*7.418e-2*	*2.830e-2*	3.757e-3	3.946e-3

Table: Best of the models and their fit II.

Random walk fits best

dataset	Sichel	binned	random walk	inherent noise
Polish	*1.675e-1*	*4.078e-2*	1.518e-3	8.508e-3
Portuguese	*6.421e-1*	*5.133e-2*	4.514e-2	4.973e-2
Romanian	*3.070e-2*	*6.539e-2*	1.579e-2	2.338e-2
Serbian.sh	*9.944e-2*	*4.676e-2*	1.346e-3	4.531e-3
Serbian.sr	*1.413845*	*1.389e-1*	6.971e-3	7.189e-3
Slovak	*5.507e-2*	*4.344e-2*	2.184e-3	2.572e-3
Spanish	*9.021e-2*	*6.501e-2*	7.718e-4	8.365e-4
Swedish	*2.225e-1*	*2.652e-2*	2.310e-3	2.526e-3

Table: Best of the models and their fit III.

Conclusions: the random walk model

- Fits notably better than earlier models
- Has clear genesis
- Opens a new way for checking statistical implications of grammatical observations

Acknowledgments

- NKFIH grant #120145: Deep Learning of Morphological Structure
- NKFIH grant #115288: Algebra and algorithms
- National Excellence Programme 2018-1.2.1-NKP-00008: Exploring the Mathematical Foundations of Artificial Intelligence
- A hardware grant from NVIDIA Corporation
- GNU parallel was used to run experiments (Tange, 2011)

References I

Kornai, A., Zséder, A., and Recski, G. (2013). Structure learning in weighted languages. In *Proceedings of the 13th Meeting on the Mathematics of Language (MoL 13)*, pages 72–82, Sofia, Bulgaria. Association for Computational Linguistics.

MacKay, D. J. (2003). *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press.

Sichel, H. (1974). On a distribution representing sentence length in written prose. *Journal of the Royal Statistical Society Series A*, 137(1):25–34.

Tange, O. (2011). Gnu parallel - the command-line power tool. *;login: The USENIX Magazine*, 36(1):42–47.

Wake, W. (1957). Sentence-length distributions of Greek authors. *Journal of the Royal Statistical Society Series A*, 120:331–346.

References II

Williams, C. (1944). A note on the statistical analysis of sentence-length as a criterion of literary style. *Biometrika*, 31:356–361.

Yule, G. U. (1944). *The Statistical Study of Literary Vocabulary*. Cambridge University Press.