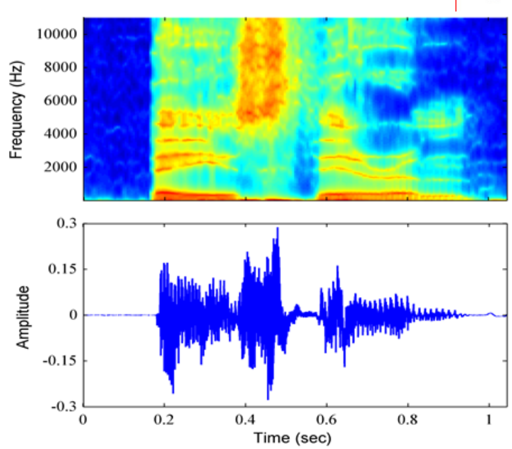
Sequential Data Modeling

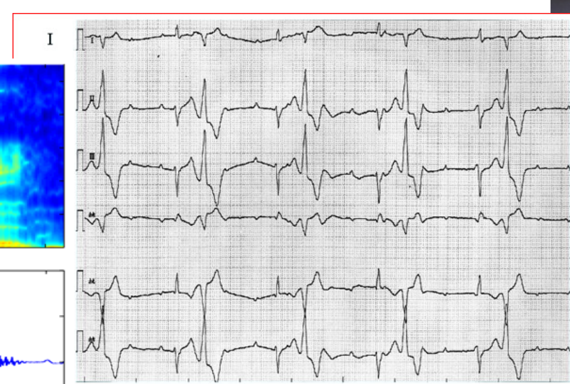
Tomoki Toda
Graham Neubig
Sakriani Sakti

Augmented Human Communication Laboratory
Graduate School of Information Science

Course Goals

The aim of this course is to **learn basic knowledge of sequential data modeling techniques** that can be applied to sequential data such as speech signals, biological signals, videos of moving objects, or natural language text. In particular, it will focus on deepening knowledge of methods based on **probabilistic models**, such as hidden Markov models or linear dynamical systems.





Credits and Grading

- 1 credit course
- Score will be graded by
  - Assignment report in every class
- Prerequisites
  - Fundamental Mathematics for Optimization（最適化数学基礎）
  - Calculus（微分積分学）
  - Basic Data Analysis（データ解析基礎）

Materials

- Textbook
  - There is no textbook for this course.

- Lecture slides
  - Handout will be distributed in each class.
  - PDF slides are availabe from http://ahclab.naist.jp/lecture/2016/sdm/index.html (internal access only)

- Reference materials
  - C.M. Bishop: Pattern Recognition and Machine Learning, Springer Science + Business Media, LLC, 2006
  - C.M. ビショップ（著）、元田、栗田、樋口、松本、村田（訳）: パターン認識と機械学習　上・下、シュプリンガー・ジャパン、2008

Office Hours

- NAIST Lecturers: Graham Neubig, Sakriani Sakti
  Augmented Human Communication Laboratory
- Office: B714
- Office hour: by appointment (send an email first)
- Email: neubig@is.naist.jp, ssakti@is.naist.jp

- Other Contact
  - Tomoki Toda
    - Email: tomoki@icts.nagoya-u.ac.jp

TA Members Email: **sdm2016@is.naist.jp**

- Rui Hiraoka hiraoka.rui.hj9@is.naist.jp
- Yoko Ishikawa ishikawa.yoko.io5@is.naist.jp

Hiraoka-kun

Ishikawa-san

Schedule

- 1st slot on every **Friday 9:20-10:50** in room **L1**

June

Sun	Mon	Tue	Wed	Thu	Fri	Sat
			1	2	$\boxed{3}$	4
5	6	7	8	9	$\boxed{10}$	11
12	13	14	15	16	$\boxed{17}$	18
19	20	21	22	23	$\boxed{24}$	25
26	27	28	29	30		

July/August

Sun	Mon	Tue	Wed	Thu	Fri	Sat
					$\boxed{1}$	2
3	4	5	6	7	8	9
10	11	12	13	14	$\boxed{15}$	16
17	18	19	20	21	22	23
24	25	26	27	28	$\boxed{29}$	30
31	$\boxed{\mathbf{1}}$					

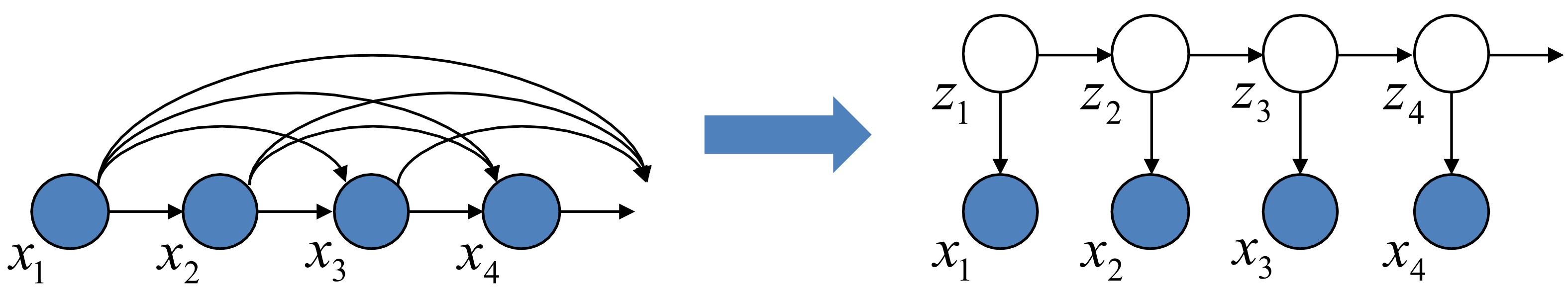
Syllabus

Date	Course description	Lecturer
6/03	Basics of sequential data modeling 1	Graham Neubig
6/10	Basics of sequential data modeling 2	Graham Neubig
6/17	Discrete latent variable models 1	Tomoki Toda
6/24	Discrete latent variable models 2	Tomoki Toda
7/1	Continuous latent variable models 1	Tomoki Toda
7/15	Discriminative models for sequential labeling 1	Sakriani Sakti
7/29	Continuous latent variable models 2	Tomoki Toda
8/1	Discriminative models for sequential labeling 2	Sakriani Sakti

1st and 2nd Classes (6/03 and 6/10)

- Lecturer: Graham Neubig

- Contents: Basics of sequential data modeling
    - Markov process
    - Latent variables
    - Mixture models
    - Expectation-maximization (EM) algorithm

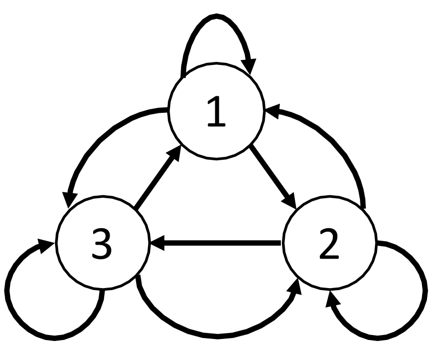


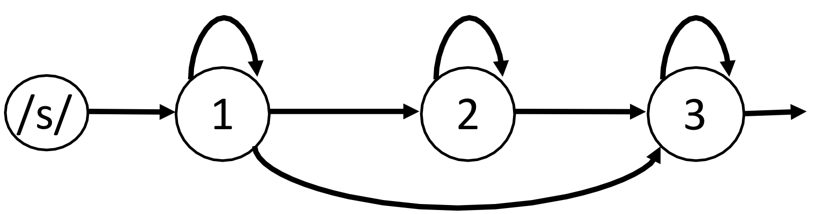
h

3rd and 4th Classes (6/17 and 6/24)

- Lecturer: Tomoki Toda

- Contents: Discrete latent variable models
  - Hidden Markov models
  - Forward-backward algorithm
  - Viterbi algorithm
  - Training algorithm

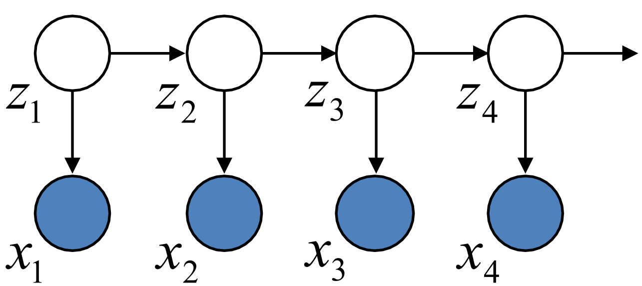


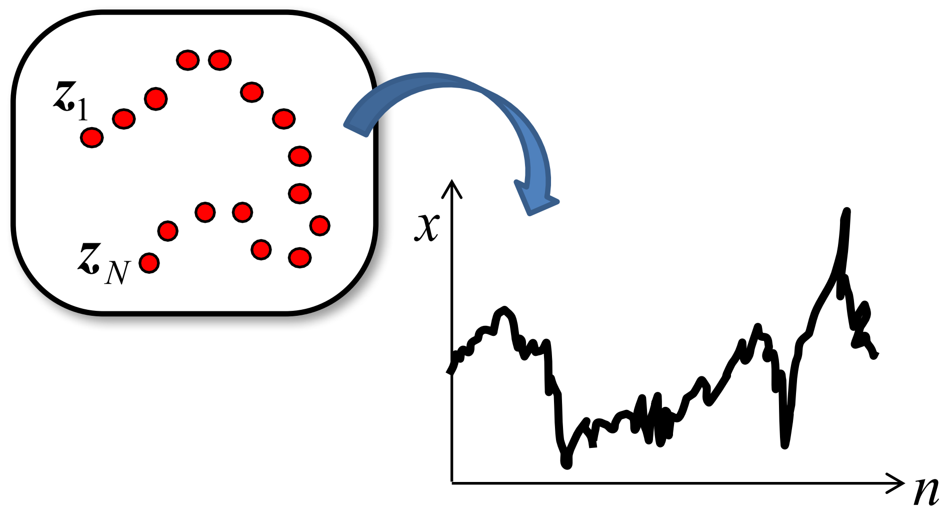


5th and 7th Classes (7/1 and 7/29)

- Lecturer: Tomoki Toda

- Contents: Continuous latent variable models
    - Factor analysis
    - Linear dynamical systems
    - Prediction and update
    - (Training algorithm)



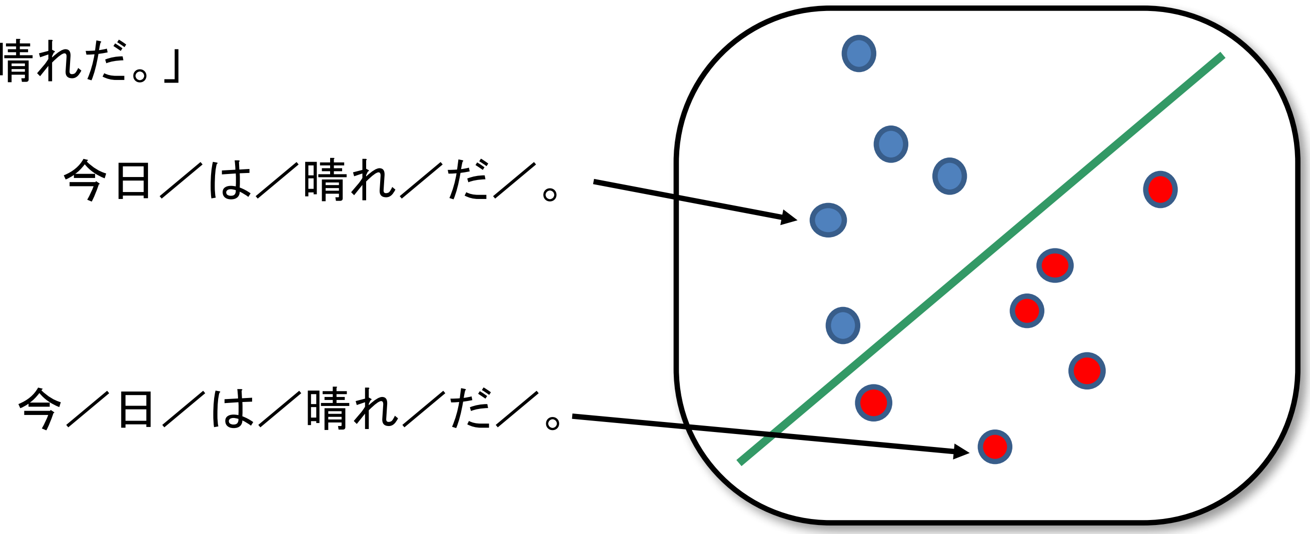


6th and 8th Classes (7/15 and 8/1)

- Lecturer: Sakriani Sakti

- Contents: Discriminative models for sequential labeling
    - Structured perceptron
    - Conditional Random Fields
    - Training algorithm

「今日は晴れだ。」



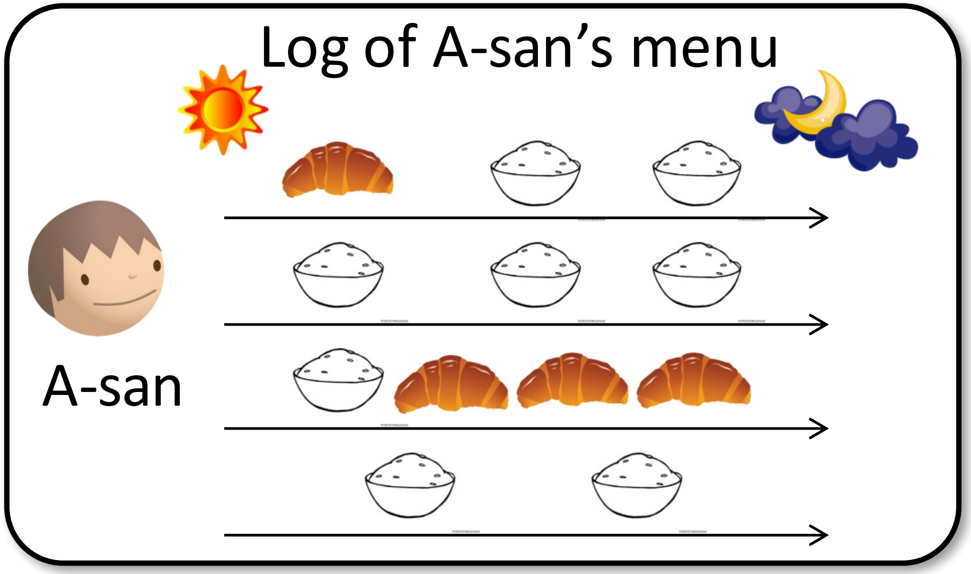
Sequential Data Modeling

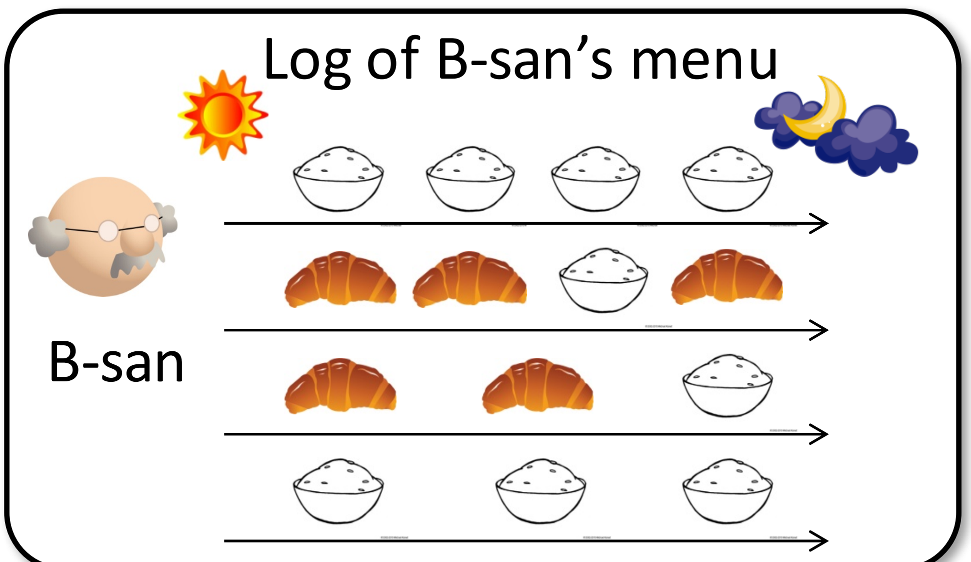
1st class
“Basics of sequential data modeling 1”

Graham Neubig

Augmented Human Communication Laboratory
Graduate School of Information Science

Question





One day, someone ate
the following menu.

Q1. A-san or B-san?

Q2. If this is A-san’s menu,
which is “?”, 

Q3. …

⋮

After this class, you can answer these questions!

Sequential Data

- Data examples
  - Time series (speech, actions, moving objects, exchange rates, …)
  - Character strings (word sequence, symbol string, …)
- Various lengths of data
  - E.g.,

    Data sample 1 (length = 5): { 1, 0, 1, 1, 0 }
    Data sample 2 (length = 8): { 1, 1, 1, 0, 1, 1, 0, 0 }
    Data sample 3 (length = 3): { 0, 0, 1 }
    Data sample 4 (length = 6): { 0, 1, 0, 1, 1, 0 }

- Probabilistic approach to modeling sequential data
  - Consistent framework for the quantification and manipulation of uncertainty
  - Effective for dealing with real data

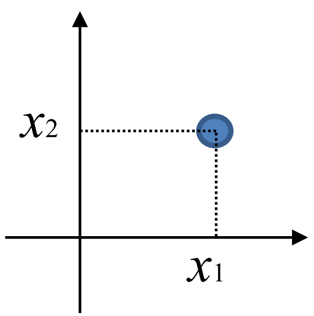
How to Represent Sequential Data?

- A sequential data sample is represented in a high-dimensional space ("# of dimensions" = "length of the sequential data sample").

Examples of sequential data:

Length = 2

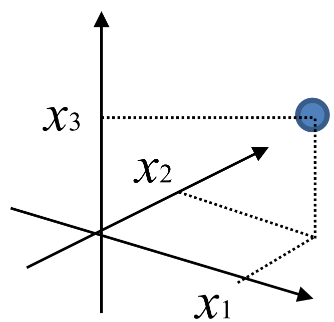
$$\left\{ \underset{n=1}{x_1} \quad \underset{n=2}{x_2} \right\}$$



Represented by
2-dimensional vector

Length = 3

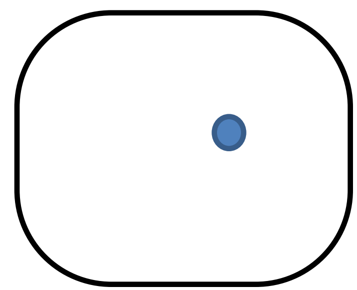
$$\left\{ \underset{n=1}{x_1} \quad \underset{n=2}{x_2} \quad \underset{n=3}{x_3} \right\}$$



Represented by
3-dimensional vector

Length = N

$$\left\{ \underset{n=1}{x_1} \quad \underset{n=2}{x_2} \quad \underset{n=3}{x_3} \quad \underset{\cdots}{\cdots} \quad \underset{n=N}{x_N} \right\}$$



Represented by
N-dimensional vector

We need to model probability distribution in these high-dimensional spaces!

Rules of Probability (1)

- Assume two random variables, X and Y
  - X : x_1 = "Bread", x_2 = "Rice", or x_3 = "Noodle"
  - Y : y_1 = "Home" or y_2 = "Restaurant"
- Assume the following data samples { X, Y }:

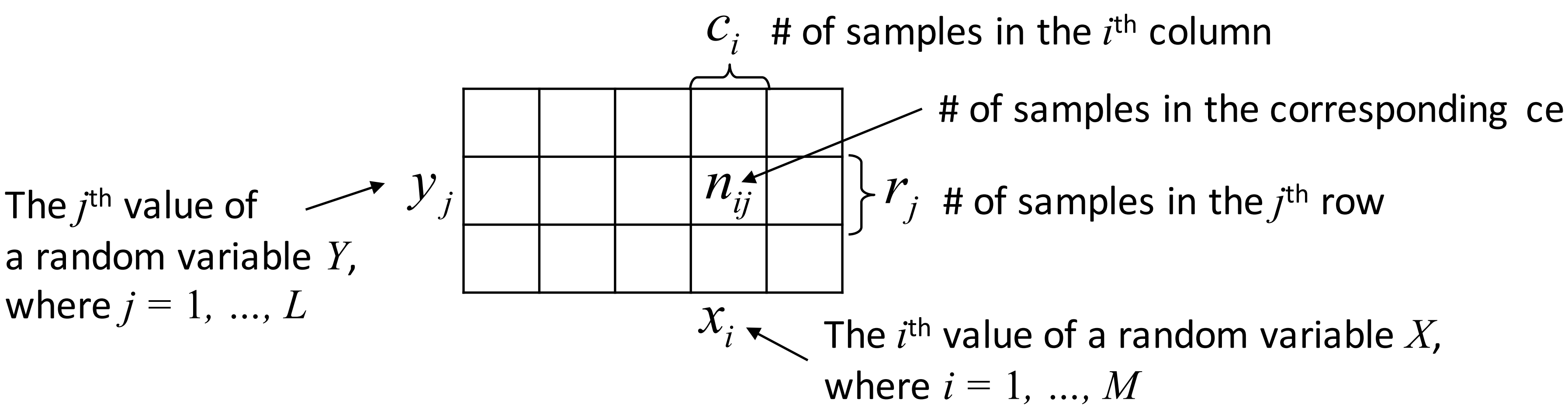
  {Bread, Home}, {Rice, Restaurant}, {Noodle, Home}, {Bread, Restaurant}, {Rice, Restaurant}, {Noodle, Home}, {Bread, Home}, {Rice, Home}, and {Bread, Home}
- Make the following table showing the number of samples

Number of samples

	Bread	Rice	Noodle
Home	**2**	**1**	**2**
Restaurant	**1**	**2**	**0**

\# of samples of {Noodle, Home}

Rules of Probability (2)



Joint probability : $p(X = x_i, Y = y_j) = \frac{n_{ij}}{N}$

Marginal probability : $p(X = x_i) = \frac{c_i}{N} = \frac{\sum_j n_{ij}}{N}$

Sum rule of probability : $p(X = x_i) = \sum_{j=1}^{L} p(X = x_i, Y = y_j)$

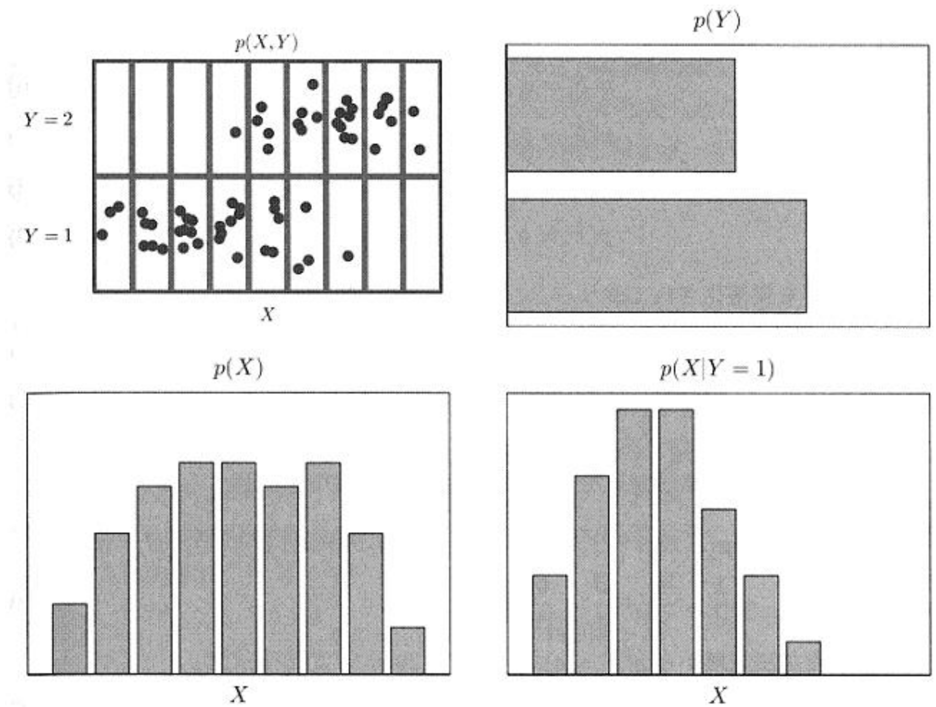
Conditional probability : $p(Y = y_j | X = x_i) = \frac{n_{ij}}{c_i}$

Product rule of probability:

$$\begin{aligned} p(X = x_i, Y = y_j) &= \frac{n_{ij}}{N} = \frac{n_{ij}}{c_i} \cdot \frac{c_i}{N} \\ &= p(Y = y_j | X = x_i) p(X = x_i) \end{aligned}$$

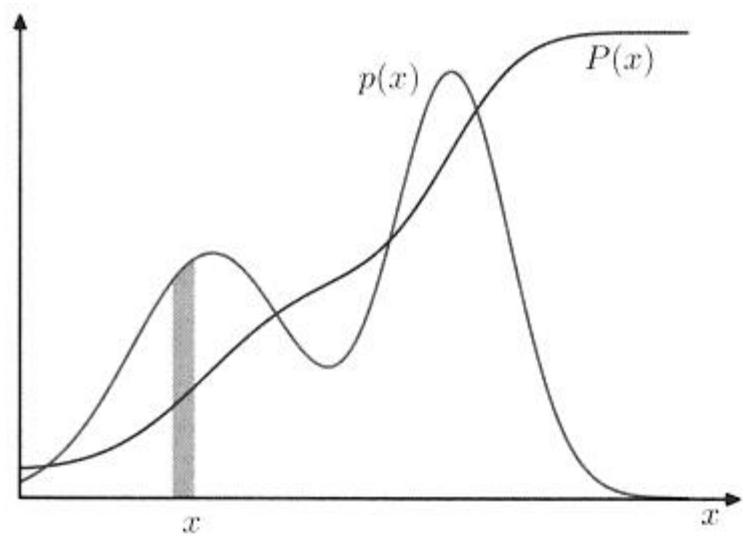
Rules of Probability (3)

- The rules of probability
  - **Sum rule** : $p(X) = \sum_Y p(X, Y)$
  - **Product rule** : $p(X, Y) = p(Y|X)p(X)$
- **Bayes' theorem** : $p(Y|X) = \dfrac{p(X|Y)p(Y)}{p(X)}$ $\longleftarrow$ $p(X) = \sum_Y p(X|Y)p(Y)$



Probability Densities

- Probabilities with respect to continuous variables
- Probability density over a real-valued variable x : $p(x)$
  - Probability that x will lie in an interval (a, b) : $p(x \in (a,b)) = \int_a^b p(x)\,\mathrm{d}x$
  - Conditions to be satisfied : $\begin{cases} p(x) \geq 0 \\ \int_{-\infty}^{\infty} p(x)\,\mathrm{d}x = 1 \end{cases}$
- Cumulative distribution function: $P(x)$
  - Probability that x lies in the interval $(-\infty, z)$: $P(z) = \int_{-\infty}^{z} p(x)\,\mathrm{d}x$

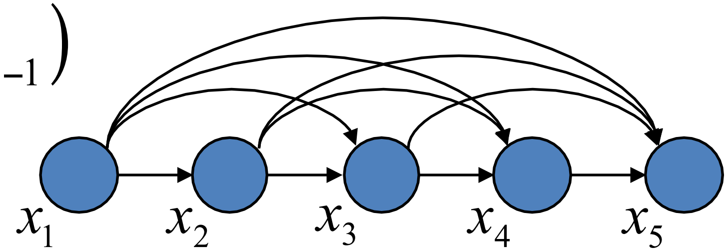


How to Model Joint Probability?

- Length of sequential data (# of data points over a sequence) varies…
  i.e., # of dimensions of joint probability distribution also varies…
- Joint probability distribution can be represented with conditional probability distributions of individual data points!
  i.e., # of distributions varies but # of dimensions of each distribution is fixed.
- However, conditional probability distribution of a present data point given all past data points needs to be modeled…

$$p(x_1, \cdots, x_N) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1, x_2) \cdots p(x_N \mid x_1, \cdots, x_{N-1})$$

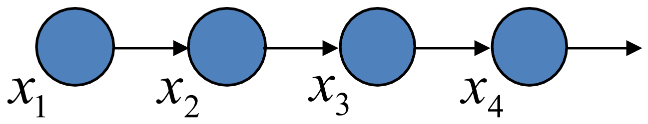
$$= p(x_1)\prod_{n=2}^{N} p(x_n \mid x_1, \cdots, x_{n-1})$$



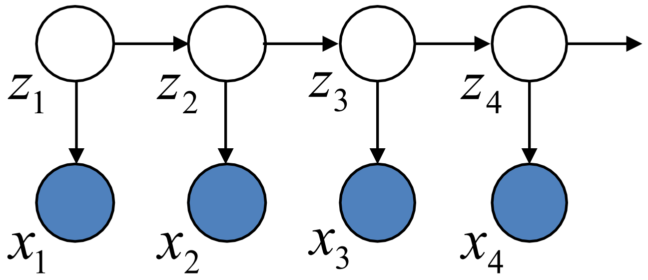
How can we effectively model joint probability distribution of sequential data?

Two Basic Approaches

- **Markov process**



- **Latent variables**



Markov Process

- Assume that the conditional probability distribution of the present states depends only on a few past states

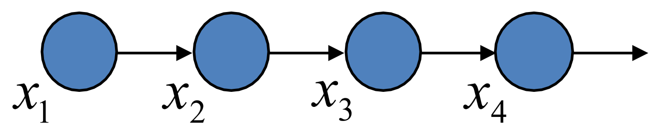
$$p(x_1, \cdots, x_N) = p(x_1) \prod_{n=2}^{N} p(x_n | x_1, \cdots, x_{n-1})$$

e.g., it depends on only one past state…

$$p(x_n | x_1, \cdots, x_{n-1}) = p(x_n | x_{n-1})$$

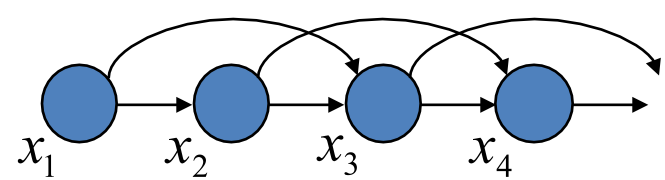
<u>1st order Markov chain</u>

$$p(x_1, \cdots, x_N) = p(x_1) \prod_{n=2}^{N} p(x_n | x_{n-1})$$



<u>2nd order Markov chain</u>

$$p(x_1, \cdots, x_N) = p(x_1) p(x_2 | x_1) \prod_{n=3}^{N} p(x_n | x_{n-1}, x_{n-2})$$



Example of 1st Order Markov Process

- How many probability distributions are needed if we model English text using the 1st order Markov process?

  If only using 27 characters including “space”,

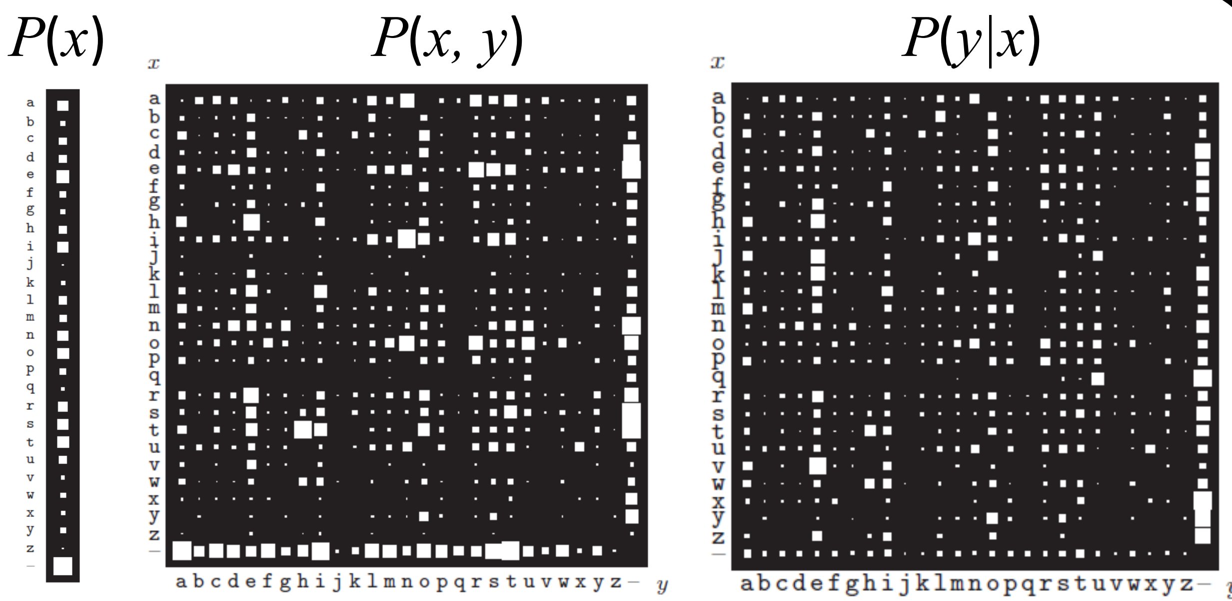
  P(“This sentence is represented by this …”)

  $= P(\text{T})\ P(\text{h}|\text{T})\ P(\text{i}|\text{h})\ P(\text{s}|\text{i})\ P(\text{-}|\text{s})\ P(\text{s}|\text{-})\ P(\text{e}|\text{s})\ P(\text{n}|\text{e})\ P(\text{t}|\text{n})\ P(\text{e}|\text{t})$ …

x : 1st letter

y : 2nd letter

Probability is shown by the areas of white squares.



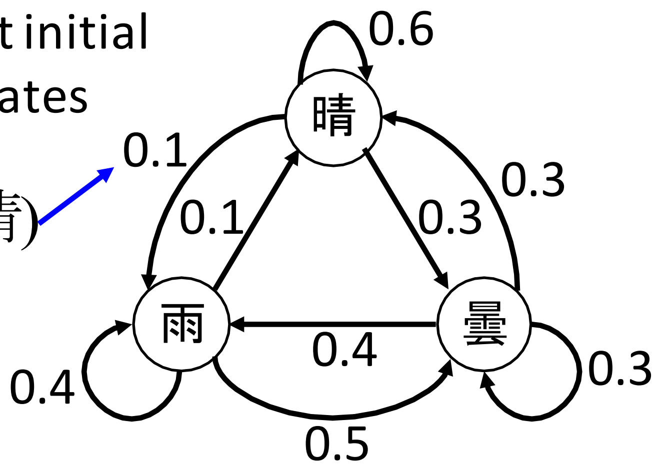
David J.C. MacKay, “Information Theory, Inference, and Learning Algorithms,” Cambridge University Press, pp. 22—24

State Transition Diagram/Matrix

State transition diagram

w/o explicit initial and final states

$p(\text{雨}|\text{晴})$



晴: fine
雨: rain
曇: could

State transition matrix

Present state

Past state

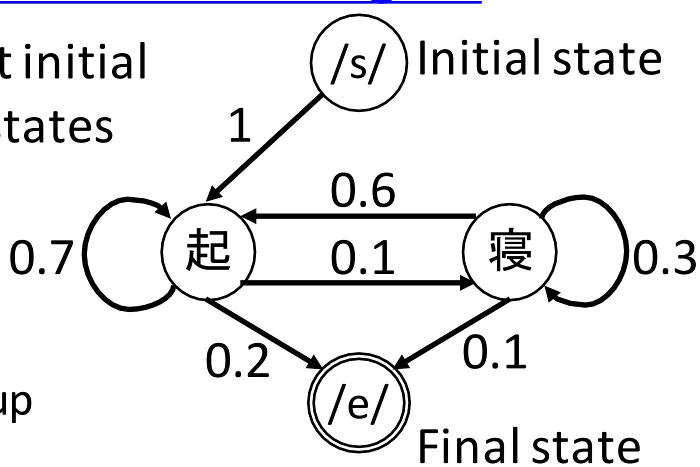
$p(\text{雨}|\text{晴})$

$$A = \begin{array}{c} \\ \text{晴} \\ \text{曇} \\ \text{雨} \end{array} \begin{array}{c} \begin{array}{ccc} \text{晴} & \text{曇} & \text{雨} \end{array} \\ \begin{bmatrix} 0.6 & 0.3 & \underline{0.1} \\ 0.3 & 0.3 & 0.4 \\ \boxed{0.1 \quad 0.5 \quad 0.4} \end{bmatrix} \end{array}$$

Sum becomes one.

State transition diagram

w/ explicit initial and final states



起: wake up
寝: sleep

State transition matrix

$$A = \begin{array}{c} \\ /s/ \\ \text{起} \\ \text{寝} \\ /e/ \end{array} \begin{array}{c} \begin{array}{cccc} /s/ & \text{起} & \text{寝} & /e/ \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 \\ & & & \\ & ? & & \\ 0 & 0 & 0 & \underline{1} \end{bmatrix} \end{array}$$

Final state transition

Example of Applications

- Language model (modeling discrete variables)
  - Modeling a word (morpheme) sequence with Markov model (n-gram)

    e.g.,「学校に行く」 ➡ /s/, 学校, に, 行, く, /e/

    Decompose into morphemes

    Modeling with bi-gram (2-gram)

$$p(s)p(\text{学校}|s)p(\text{に}|\text{学校})p(\text{行}|\text{に})p(\text{く}|\text{行})p(e|\text{く})$$

- Autoregressive model (modeling continuous variables)
  - Predicting the present data point from past M data points

$$p\left(x_n|x_{n-M},\cdots,x_{n-1}\right)=\mathcal{N}\left(x_n|\underline{a_M x_{n-M}+\cdots+a_1 x_{n-1}},\sigma^2\right)$$

Linear combination of past M data points

Model Training (Maximum Likelihood Estimation)

- Training of conditional probability distribution from sequential data samples given as training data $\left\{x_1^{(1)}, \cdots, x_{N_1}^{(1)}\right\}$ … $\left\{x_1^{(S)}, \cdots, x_{N_S}^{(S)}\right\}$

Likelihood function: $$\prod_{s=1}^{S} p\left(x_1^{(s)}, \cdots, x_{N_s}^{(s)} | \underline{\boldsymbol{\lambda}}\right)$$

Function of model parameters

Model parameter set

Determine the conditional probability distributions $p(w_c \mid w_p, \lambda)$ that maximizes the (log-scaled) likelihood function

$$\arg\max \sum_{s=1}^{S} \log p\left(x_1^{(s)} | \boldsymbol{\lambda}\right) + \sum_{s=1}^{S} \sum_{n=2}^{N_s} \log p\left(x_n^{(s)} | x_{n-1}^{(s)}, \boldsymbol{\lambda}\right)$$

subject to $\displaystyle\sum_{x \in \text{ all } w} p(x|w_p, \boldsymbol{\lambda}) = 1$

Constraint to normalize the estimates as probability

Maximum likelihood estimate :

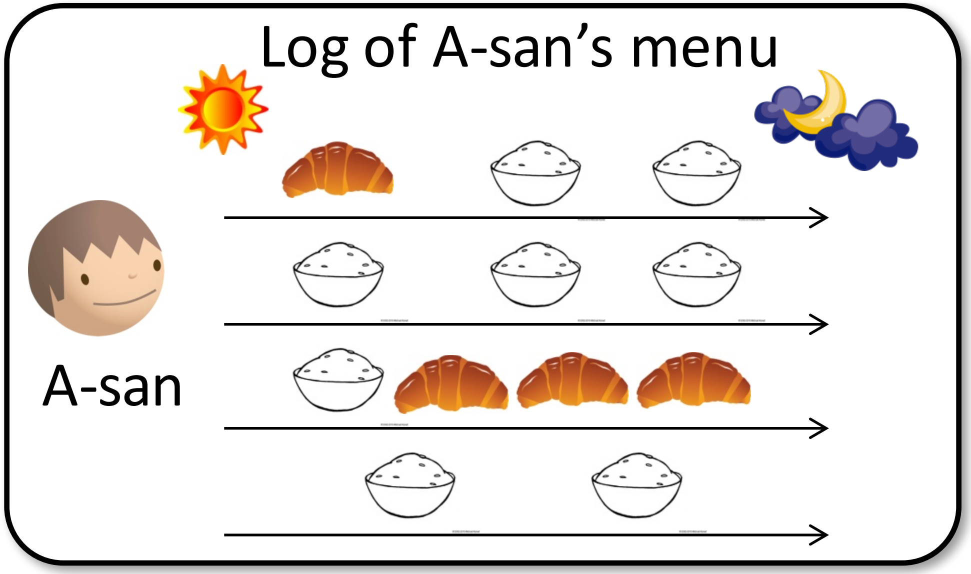
$$p(w_c|w_p, \boldsymbol{\lambda}) = \frac{\mathrm{Num}(w_p, w_c)}{\mathrm{Num}(w_p)}$$

# of samples $\{w_p, w_c\}$

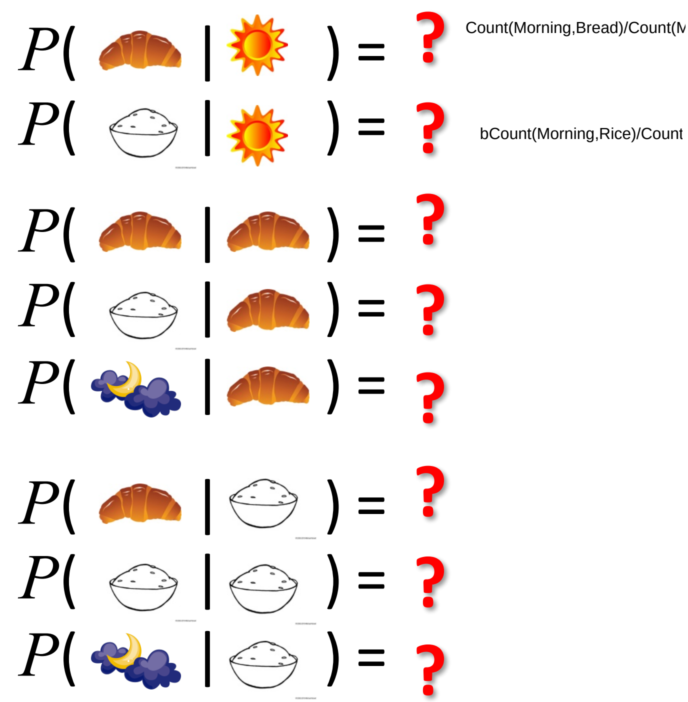
# of samples w_p

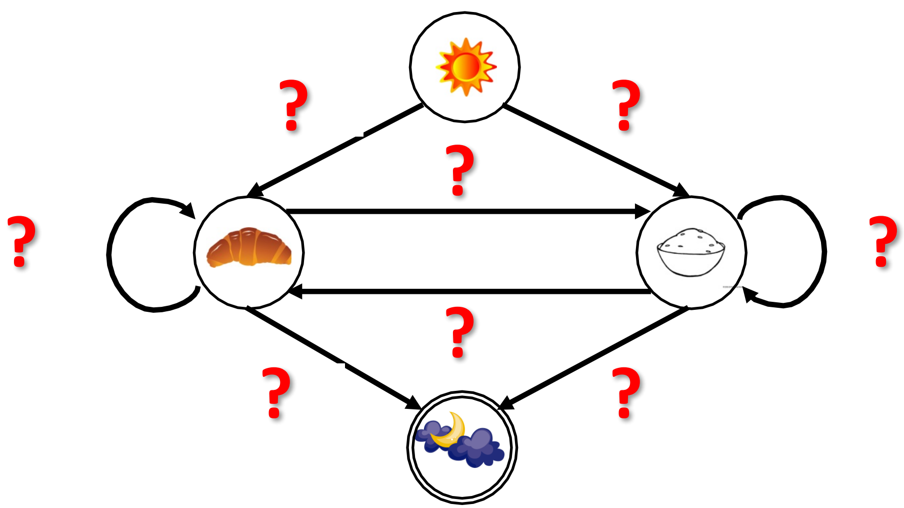
Example of MLE

morning : initial stater



MLE of conditional probabilities





State transition diagram

Methods for Evaluating Models

- Use of a test data set $\{w_1, \ldots, w_N\}$ not included in training data
- Evaluation metrics
  - Likelihood
    $$p(w_1, \cdots, w_N \mid \lambda) = \prod_{n=1}^{N} p(w_n \mid w_{n-1})$$
  - Log-scaled likelihood
    $$\log_2 p(w_1, \cdots, w_N \mid \lambda) = \sum_{n=1}^{N} \log_2 p(w_n \mid w_{n-1})$$
  - Entropy
    $$H = -\frac{1}{N} \log_2 p(w_1, \cdots, w_N \mid \lambda) = -\frac{1}{N} \sum_{n=1}^{N} \log_2 p(w_n \mid w_{n-1})$$
  - Perplexity
    $$PP = 2^H$$

A measure of effective "branching factor"

e.g., set a uniform distribution to all n-gram probabilities for M words

$$H = -\frac{1}{N} \sum_{n=1}^{N} \log_2 \frac{1}{M} = \frac{1}{N} \log_2 M^N = \log_2 M$$

$$PP = M$$

Classification/Inference/Generation

Classification

Multiple models $\{\boldsymbol{\lambda}_1, \cdots, \boldsymbol{\lambda}_I\}$

Sequential data $\{x_1, \cdots, x_N\}$

Model selection based on maximum posterior probability

Classification result

Inference (prediction)

Model $\boldsymbol{\lambda}$

Sequential data $\{x_1, \cdots, x_N\}$

Calculation of conditional probability

$$p\left(x_{N+1} | x_1, \cdots, x_N, \boldsymbol{\lambda}\right)$$

Inference (data or probability)

Generation

Model $\boldsymbol{\lambda}$

Data generation based on joint probability distribution

$$p(x_1, \cdots, x_N | \boldsymbol{\lambda})$$

Data $\{\hat{x}_1, \cdots, \hat{x}_N\}$

Classification w/ Maximum *A Posteriori*

- Select a model that maximizes the posterior probability

Likelihood function

Prior probability

Posterior probability:

$$p\left(\boldsymbol{\lambda}_i | x_1, \cdots, x_N\right) = \frac{p\left(x_1, \cdots, x_N | \boldsymbol{\lambda}_i\right) p\left(\boldsymbol{\lambda}_i\right)}{p\left(x_1, \cdots, x_N\right)}$$

Constant

The model is selected by

$$\begin{aligned} \hat{i} &= \arg\max_i p\left(\lambda_i \mid x_1, \cdots, x_N\right) \\ &= \arg\max_i p\left(x_1, \cdots, x_N \mid \lambda_i\right) p\left(\lambda_i\right) \end{aligned}$$

If prior probability is given by a uniform distribution,

$$\hat{i} = \arg\max_i p\left(x_1, \cdots, x_N \mid \lambda_i\right)$$

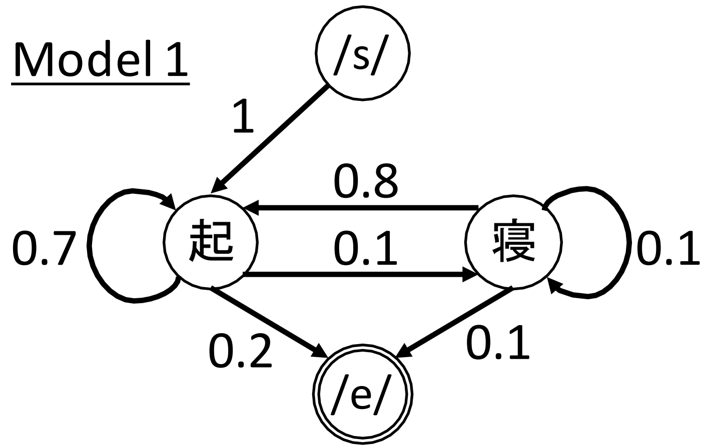
Classification

- Comparison of model likelihoods

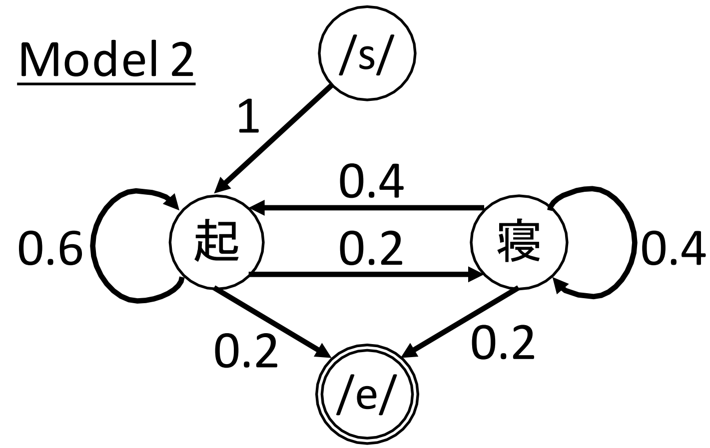
起: wake up
寝: sleep

Models :

Model 1



Model 2

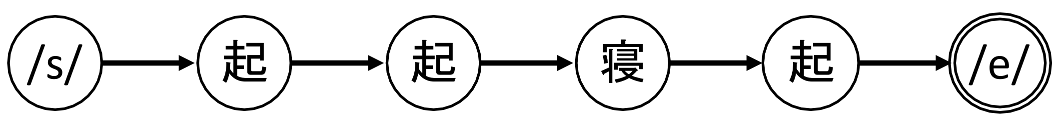


Observed data: /s/ 起 起 寝 起 /e/ Q. Which model is this data sample classified to?

Likelihood: $p(s)p(起|s)p(起|起)p(寝|起)p(起|寝)p(e|起)$

Trellis: Model1:P(s)=1P(|s)=1P(|)=0.7

Expansion of the state transition graph over time axis



Model 1: $1 \times 1 \times 0.7 \times 0.1 \times 0.8 \times 0.2 = 0.0112$

Model 2: ?

A. Classified to the model 1.

Marginalization for Unobserved Data

- Likelihood calculation with marginalization even if a part of data is not observed.

Observed data: /s/ ? 寝 ? /e/   Q. Which model is this data sample classified to?

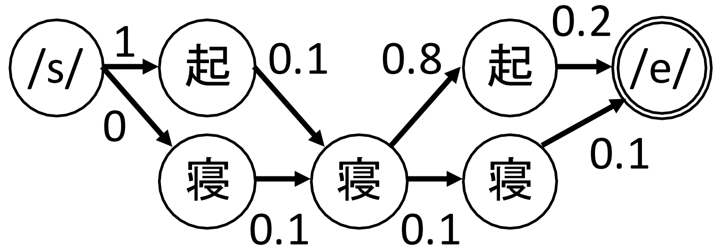
Likelihood : $\sum_{x_1, x_3 \in \{起, 寝\}} p(s) p(x_1 \mid s) p(寝 \mid x_1) p(x_3 \mid 寝) p(e \mid x_3)$

$$= p(s) \left\{ \sum_{x_1 \in \{起, 寝\}} p(x_1 \mid s) p(寝 \mid x_1) \right\} \left\{ \sum_{x_3 \in \{起, 寝\}} p(x_3 \mid 寝) p(e \mid x_3) \right\}$$

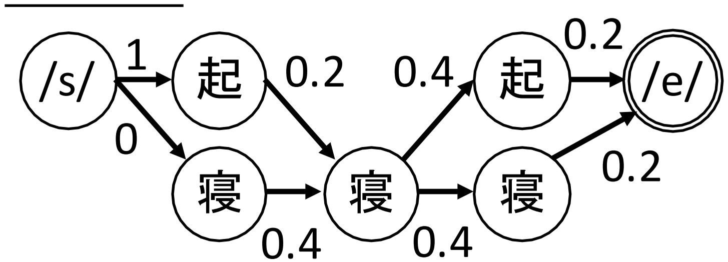
Consider all possible data samples

Model 1

Trellis:



Model 2



Data:/s/??/e/ (?:unobserved data),

$$1 \times (1 \times 0.1 + 0 \times 0.1)$$
$$\times (0.8 \times 0.2 + 0.1 \times 0.1) = 0.017$$

?

A. Classified to the model 2.

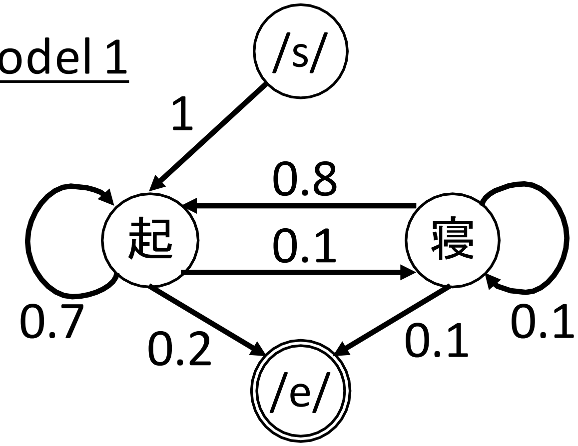
Inference

- Calculation of posterior probability

Observed data: /s/ 起 ? 起 /e/

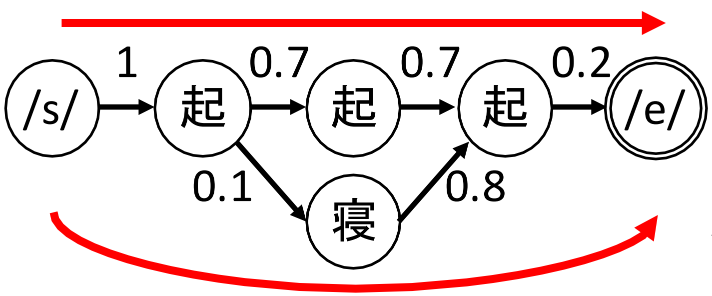
Q. Which “起” or “寢” is likely observed at “?” point?

Model 1



Posterior probability :

$$p(x_2 = \text{起} \mid x_1 = \text{起}, x_3 = \text{起}) = \frac{p(s, \text{起}, \text{起}, \text{起}, e)}{\sum_{x_2 \in \{\text{起}, \text{寢}\}} p(s, \text{起}, x_2, \text{起}, e)}$$



$$p(s, \text{起}, \text{起}, \text{起}, e) = 1 \times 1 \times 0.7 \times 0.7 \times 0.2 = 0.098$$

$$p(s, \text{起}, \text{寢}, \text{起}, e) = \text{?}$$

$$p(x_2 = \text{起} \mid x_1 = \text{起}, x_3 = \text{起}) = \frac{0.098}{0.098 + 0.016} \cong 0.860$$

$$p(x_2 = \text{寢} \mid x_1 = \text{起}, x_3 = \text{起}) = \text{?}$$

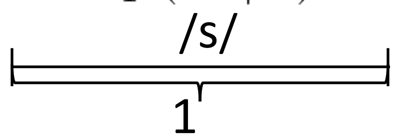
A. “?” is “起” with
86% of probability.

Generation

- Random generation of data samples from the model

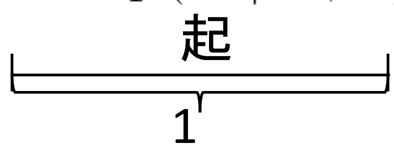
1. Sampling x_1 $\qquad \hat{x}_1 \sim p(x_1 | \boldsymbol{\lambda})$

/s/



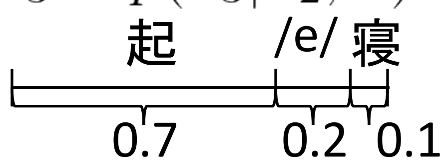
2. Sampling x_2 $\qquad \hat{x}_2 \sim p(x_2 | \hat{x}_1, \boldsymbol{\lambda})$

起



3. Sampling x_3 $\qquad \hat{x}_3 \sim p(x_3 | \hat{x}_2, \boldsymbol{\lambda})$

e.g., … 寝

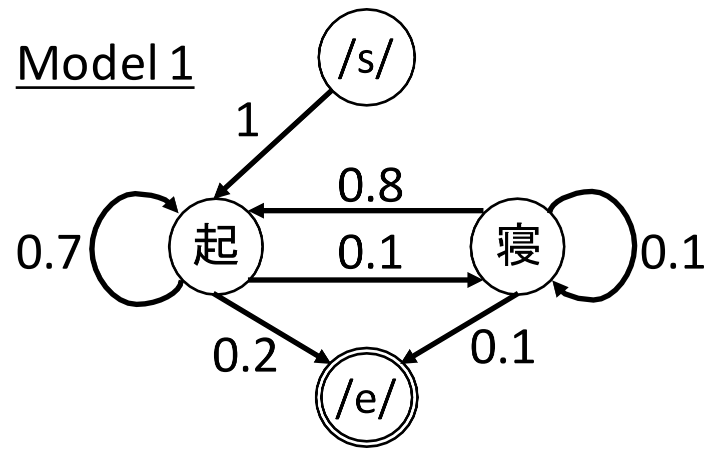


4. Sampling x_4 $\qquad \hat{x}_4 \sim p(x_4 | \hat{x}_3, \boldsymbol{\lambda})$

e.g., … 起

⋮

End if the final state /e/ is sampled

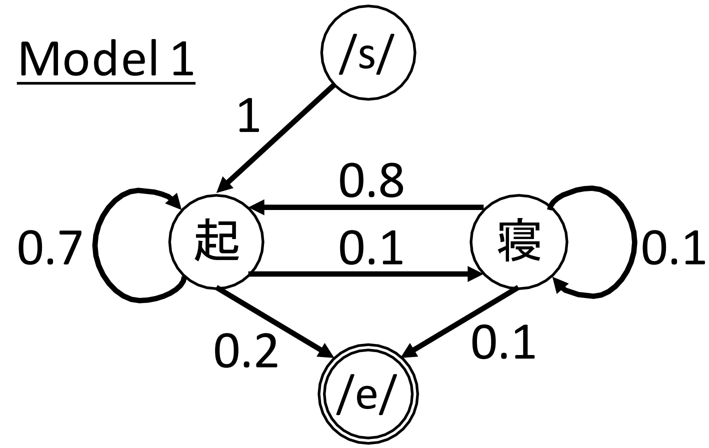


/s/ 起 寝 起 ・・・ /e/

Various lengths of data are generated.

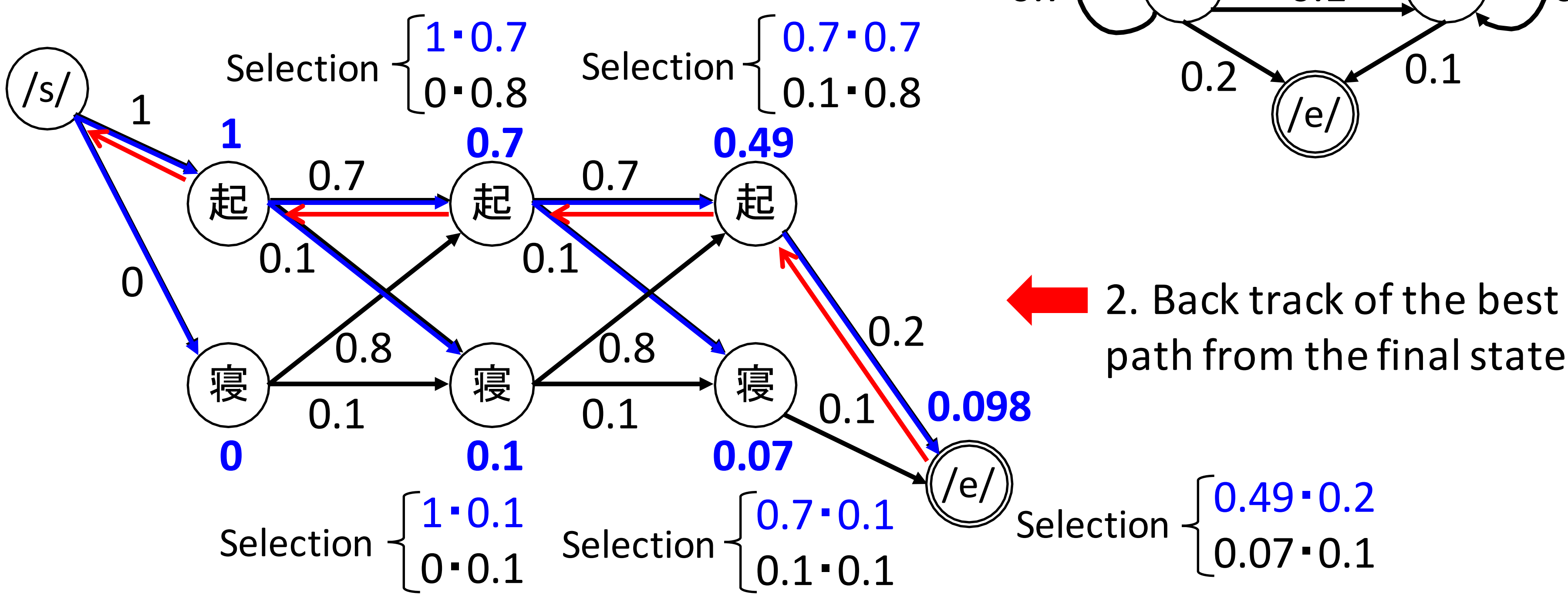
Maximum Likelihood Data Generation

- Data generation by maximizing likelihood under the condition that the length of data is given



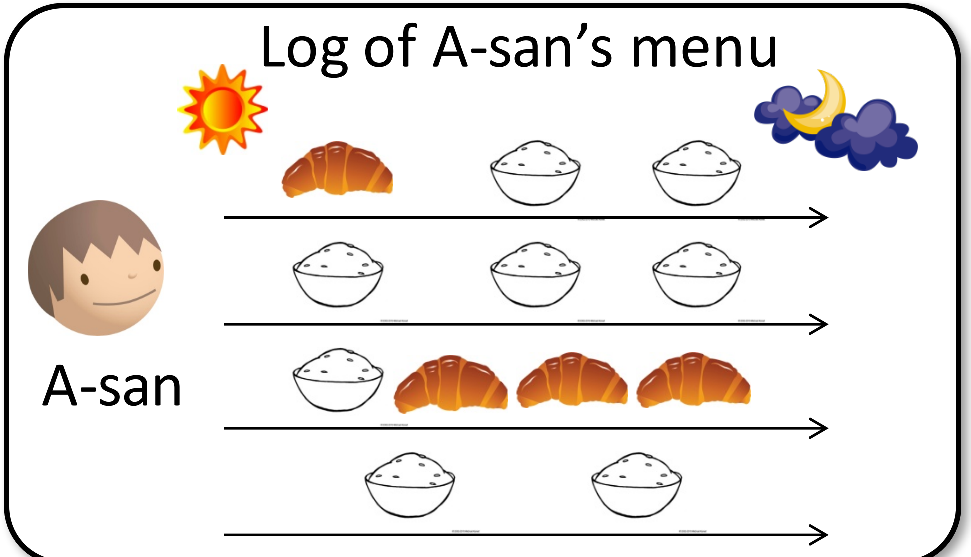
Dynamic programming

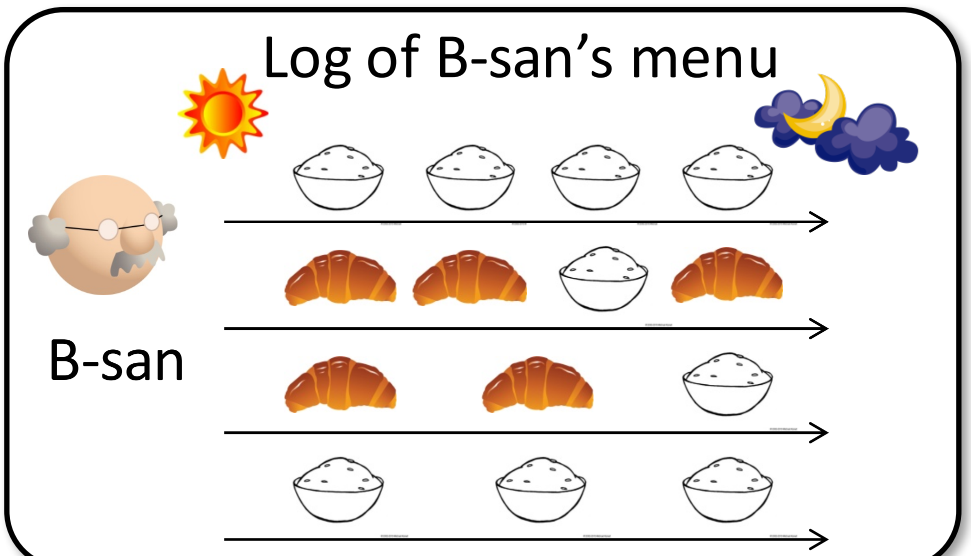
1. Store the best path at each state



/s/ 起 起 起 /e/ is generated if setting the data length to 3.

You Can Answer the Question!





One day, someone ate the following menu.

Q1. A-san or B-san?

Q2. If this is A-san’s menu, which is “?”, 

Q3. …

⋮

Let’s use Markov model to answer them!