GRAPHICAL MODELS

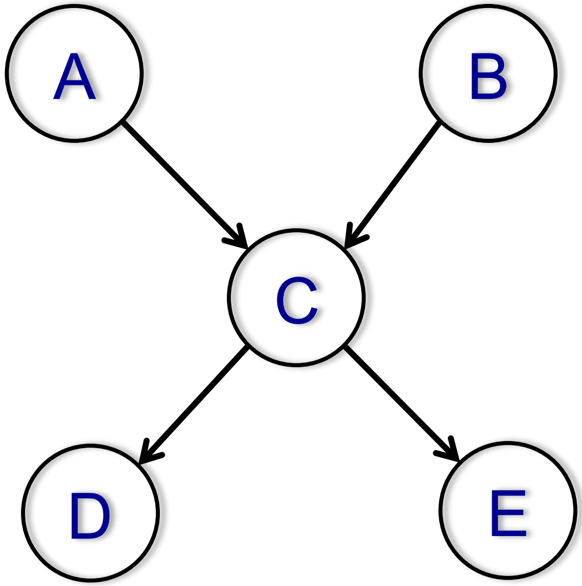
TRU CAO

HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY AND JOHN VON NEUMANN INSTITUTE

OUTLINE

- **Bayesian Networks (revisited)**
- **Naïve Bayes Classifier (revisited)**
- **Tree Augmented Naïve Bayes Model**
- **Hidden Markov Model**

BAYESIAN NETWORKS



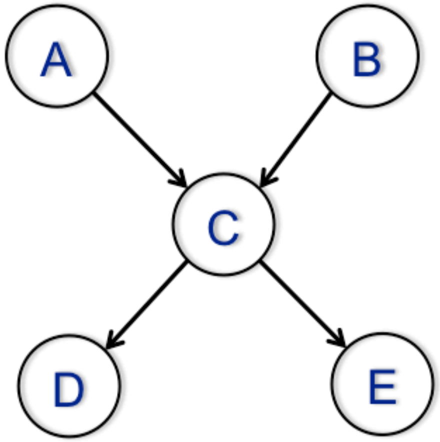
BAYESIAN NETWORKS

- **Advantages of graphical modeling:**
  - Conditional independence:

$$p(D \mid C, E, A, B) = p(D \mid C)$$

  - Factorization:

$$p(A, B, C, D, E) = p(D \mid C).p(E \mid C).p(C \mid A, B).p(A).p(B)$$

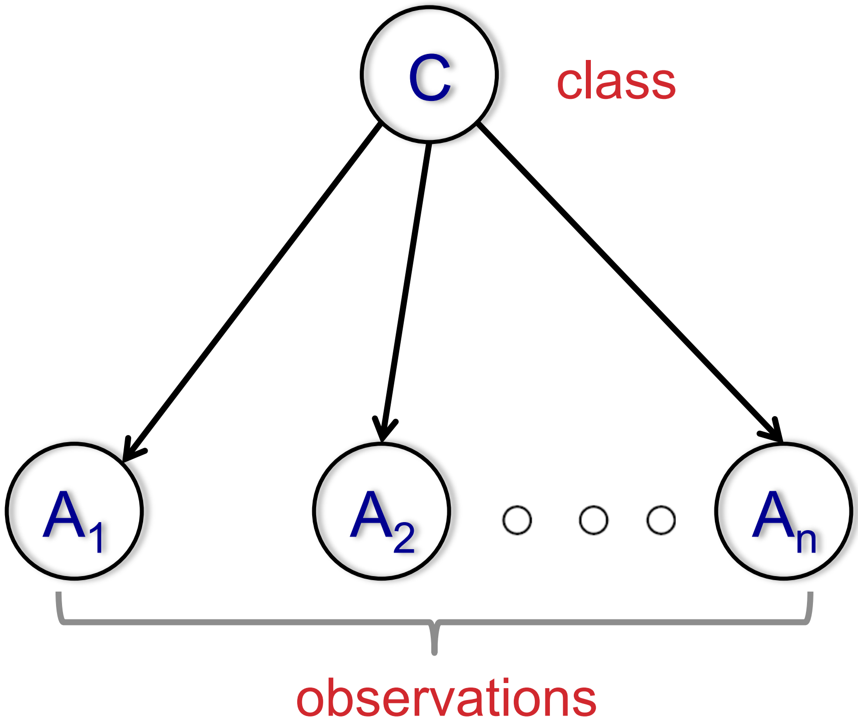


NAÏVE BAYES CLASSIFIER

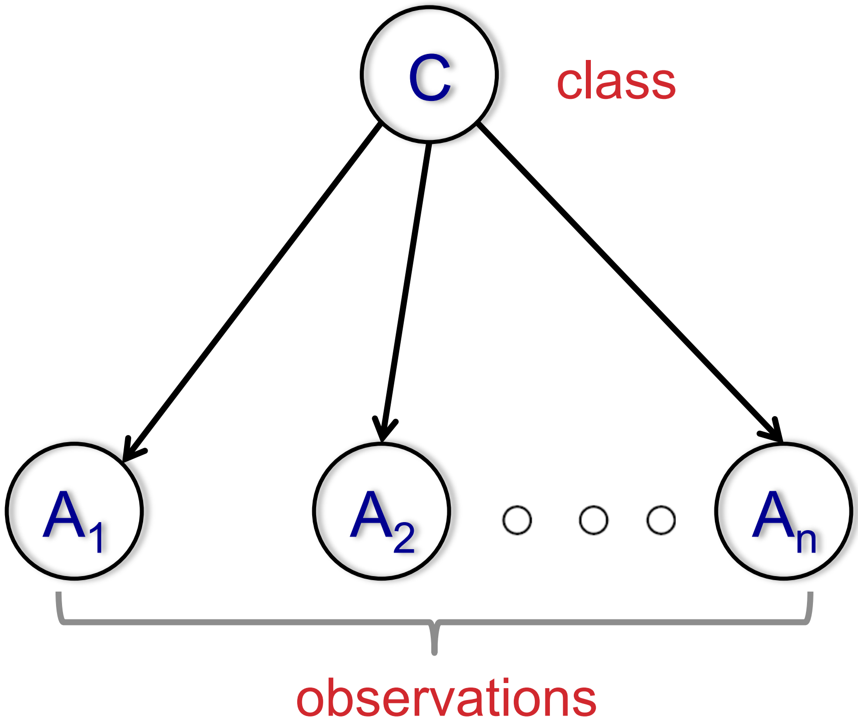
- Each instance x is described by a conjunction of attribute values $<a_1, a_2, \ldots, a_n>$.
- It is to assign the most probable class c to an instance.
- $c_{NB} = \text{argmax}_{c \in C}\ p(a_1, a_2, \ldots, a_n \mid c).p(c)$

$$= \text{argmax}_{c \in C} \prod_{i=1,n} p(a_i \mid c).p(c)$$

NAÏVE BAYES CLASSIFIER



NAÏVE BAYES CLASSIFIER



Joint distribution: $p(C, A_1, A_2, \ldots, A_n)$

NAÏVE BAYES CLASSIFIER

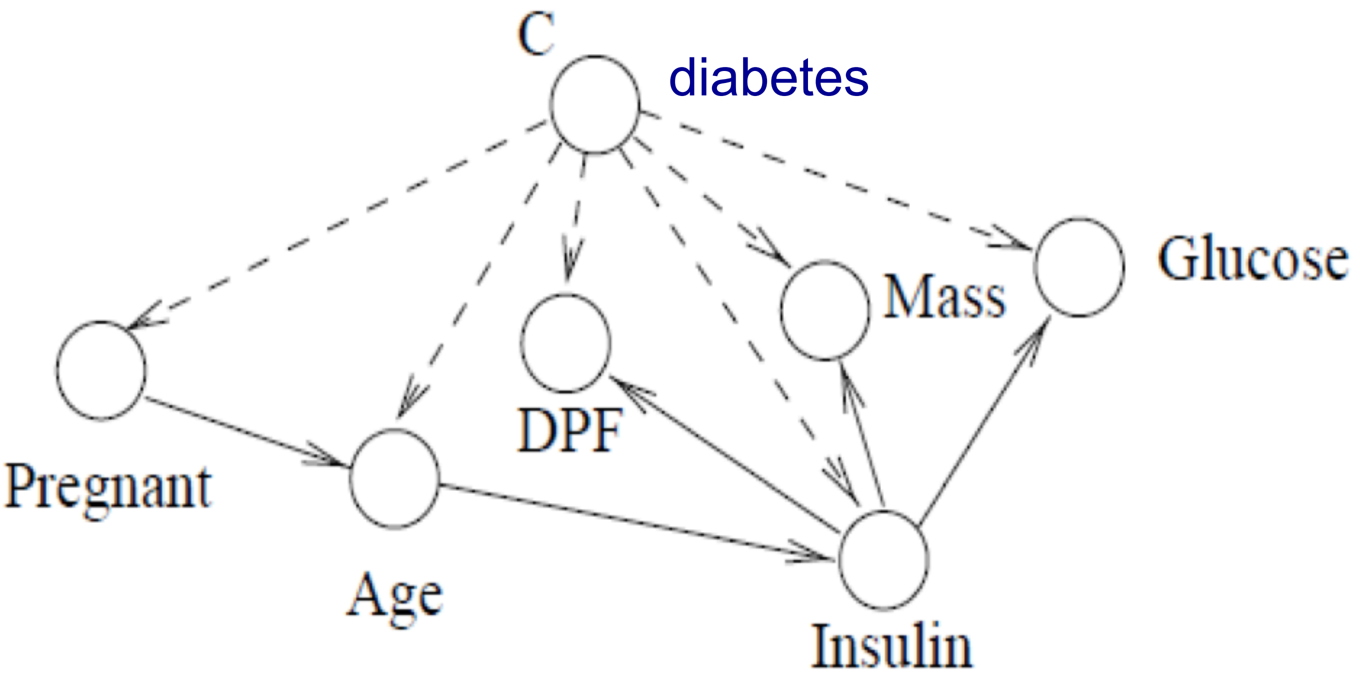
- **NB is a generative model:**
  - It models a joint distribution: p(C, A)
  - It can generate any distribution on C and A.

NAÏVE BAYES CLASSIFIER

- **NB is a generative model:**
  - It models a joint distribution: p(C, A)
  - It can generate any distribution on C and A.
- **In contrast to a discriminative model (e.g., CRF):**
  - Conditional distribution: p(C I A)
  - It discriminates C given A.

TREE AUGMENTED NB MODEL

- **An extension of NB with dependence between attributes/observations:**



HIDDEN MARKOV MODELS

- Introduction
- Example
- Independence assumptions
- Forward algorithm
- Viterbi algorithm
- Training
- Application to NER

HIDDEN MARKOV MODELS

- **One of the most popular graphical models.**
- **Dynamic extension of Bayesian networks.**
- **Sequential extension of NB classifier.**

HIDDEN MARKOV MODELS

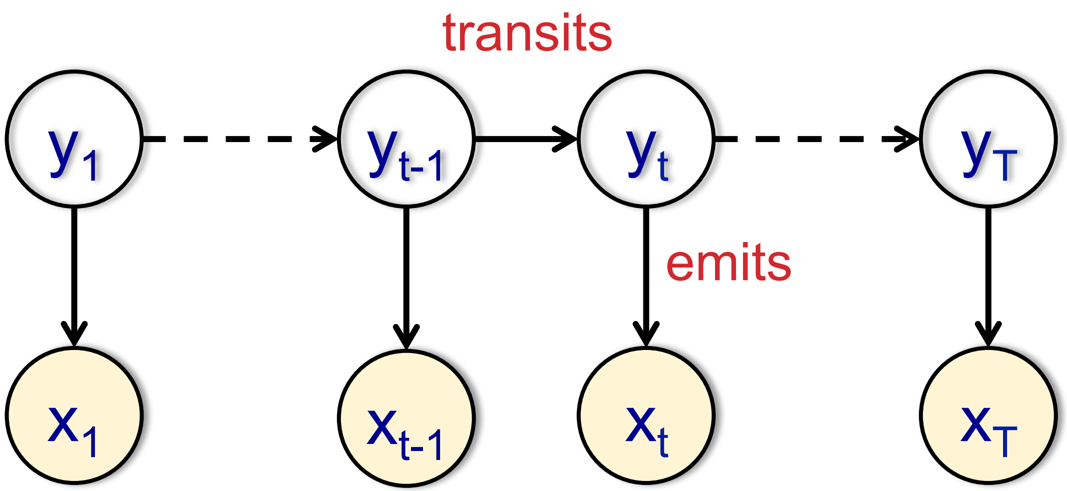
- **Example:**
  - Your possible looking prior to the exam = {**tired**, **hungover**, **scared**, **fine**}.
  - Your possible activity last night = {**TV**, **pub**, **party, study**}.
  - Given a sequence of observations of your looking, guess what you did in previous nights.

HIDDEN MARKOV MODELS

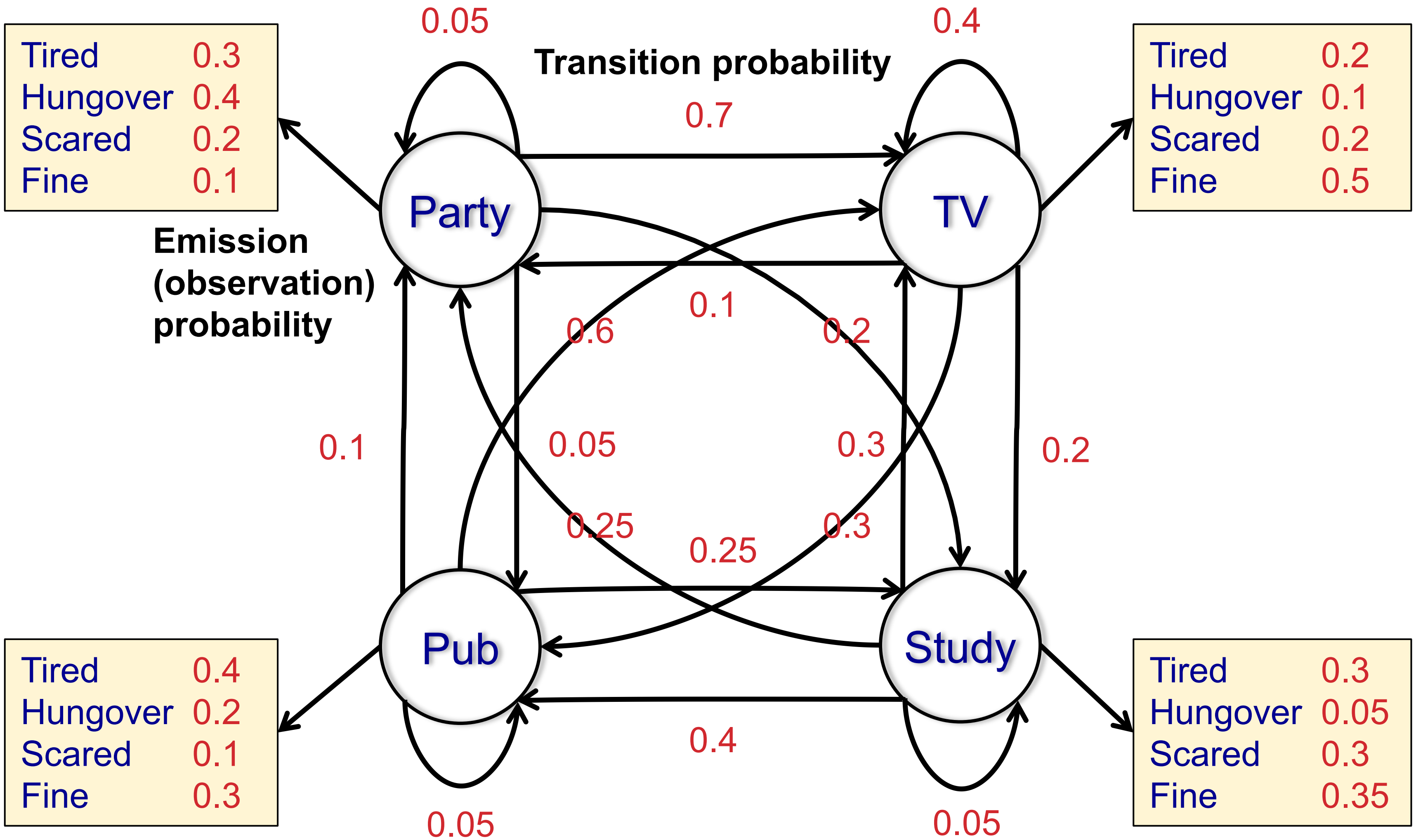
- **Example:**
  - Your possible looking prior to the exam = {**tired**, **hungover**, **scared**, **fine**}.
  - Your possible activity last night = {**TV**, **pub**, **party, study**}.
  - Given a sequence of observations of your looking, guess what you did in previous nights.
- **A model:**
  - Your looking depends on what you did in the night before.
  - Your activity in a night depends on what you did in some previous nights.

HIDDEN MARKOV MODELS

- A finite set of possible **observations**.
- A finite set of possible hidden **states**.
- To predict the most probable sequence of underlying states $\{y_1, y_2, \ldots, y_T\}$ for a given sequence of observations $\{x_1, x_2, \ldots, x_T\}$.

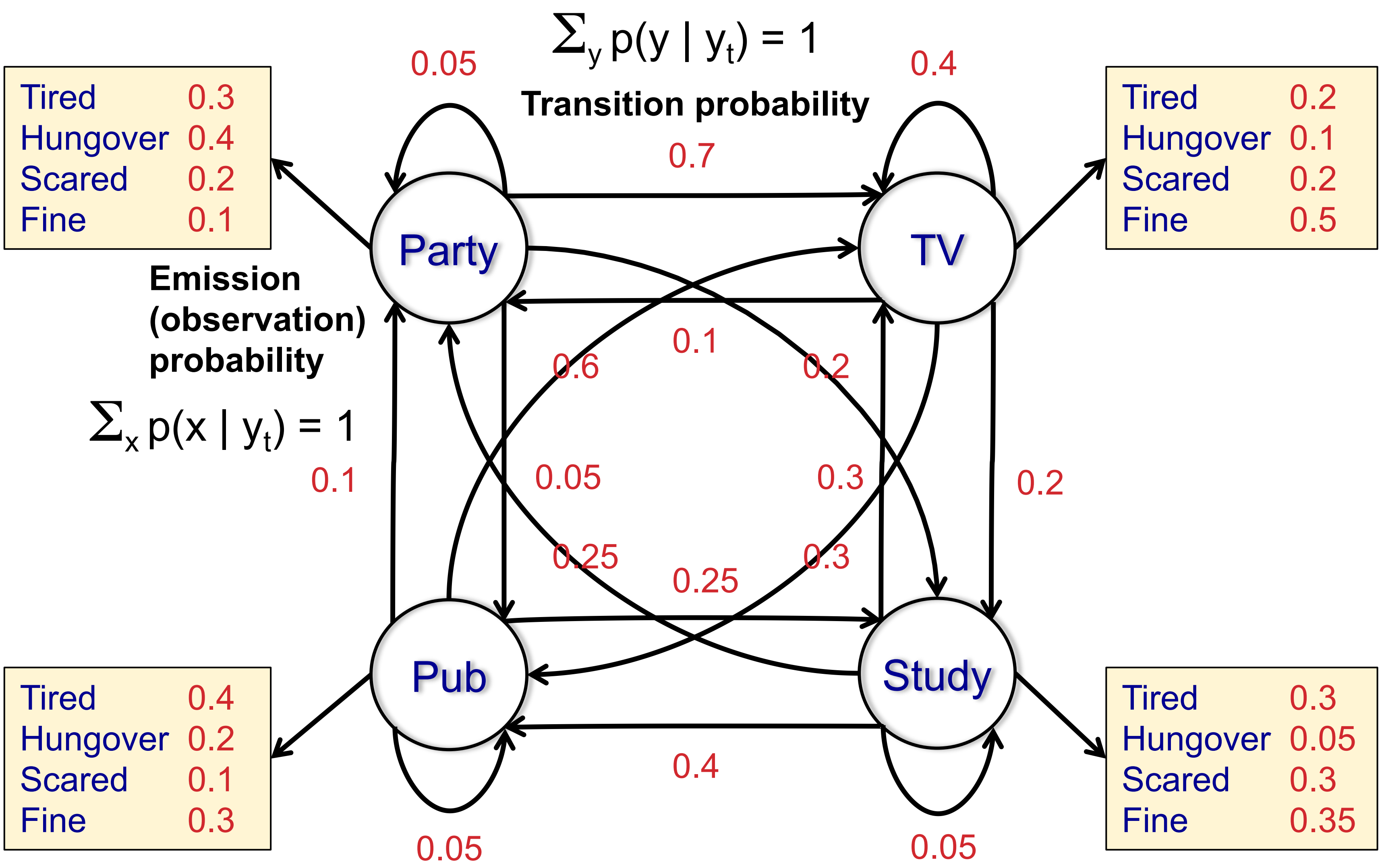


HIDDEN MARKOV MODELS



Marsland, S. (2009) Machine Learning: An Algorithmic Perspective.

HIDDEN MARKOV MODELS



Marsland, S. (2009) Machine Learning: An Algorithmic Perspective.

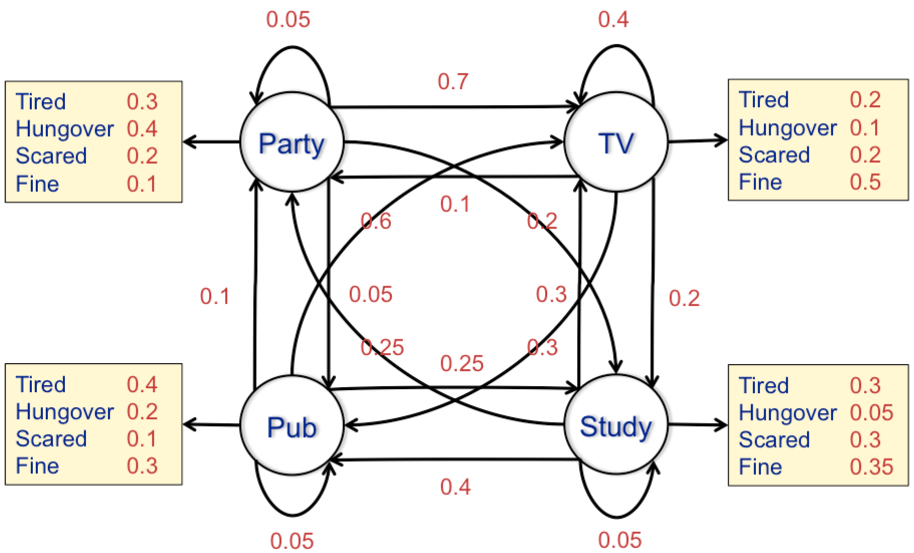
HIDDEN MARKOV MODELS

- **HMM conditional independence assumptions:**
  - State at time **t** depends only on state at time **t – 1**.

$$p(y_t \mid y_{t-1}, Z) = p(y_t \mid y_{t-1})$$

  - Observation at time **t** depends only on state at time **t**.

$$P(x_t \mid y_t, Z) = p(x_t \mid y_t)$$



HIDDEN MARKOV MODELS

- **HMM is a generative model:**
  - Joint distributions:

$$p(Y, X) = p(y_1, y_2, \ldots, y_T, x_1, x_2, \ldots, x_T) = \prod_{t=1,T} p(x_t \mid y_t).p(y_t \mid y_{t-1})$$

HIDDEN MARKOV MODELS

- **HMM is a generative model:**
  - Joint distributions:

$$p(Y, X) = p(y_1, y_2, \ldots, y_T, x_1, x_2, \ldots, x_T) = \prod_{t=1,T} p(x_t \mid y_t).p(y_t \mid y_{t-1})$$

$$p(y_1 \mid y_0) = p(y_1)$$

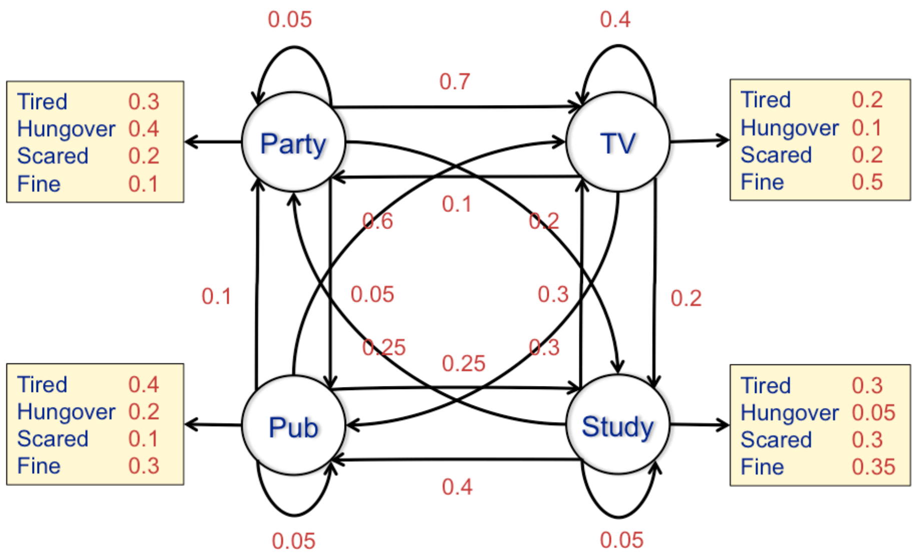
HIDDEN MARKOV MODELS

- **HMM is a generative model:**
  - Joint distributions:

$$p(Y, X) = p(y_1, y_2, \ldots, y_T, x_1, x_2, \ldots, x_T) = \prod_{t=1,T} p(y_t \mid y_{t-1}).p(x_t \mid y_t)$$

$$p(y_1 \mid y_0) = p(y_1)$$

  - It can generate any distribution on Y and X.



HIDDEN MARKOV MODELS

- **HMM is a generative model:**
  - Joint distributions:

    $$p(Y, X) = p(y_1, y_2,\ldots, y_T, x_1, x_2,\ldots, x_T) = \prod_{t=1,T} p(y_t \mid y_{t-1}).p(x_t \mid y_t)$$

    $$p(y_1 \mid y_0) = p(y_1)$$
  - It can generate any distribution on Y and X.
- **In contrast to a discriminative model (e.g., CRF):**
  - Conditional distributions: $p(Y \mid X)$
  - It discriminates Y given X.

HIDDEN MARKOV MODELS

- **Forward algorithm:**
  - To compute the joint probability of the state at time **t** being $\mathbf{y_t}$ and the sequence of observations in the first **t** steps being $\mathbf{\{x_1, x_2, \ldots, x_t\}}$:

$$\alpha_t(y_t) = p(y_t, x_1, x_2, \ldots, x_t)$$

HIDDEN MARKOV MODELS

- **Forward algorithm:**
  - To compute the joint probability of the state at time **t** being $\mathbf{y_t}$ and the sequence of observations in the first **t** steps being $\mathbf{\{x_1, x_2, \ldots, x_t\}}$:

$$\alpha_t(y_t) = p(y_t, x_1, x_2, \ldots, x_t)$$

  - Bayes' theorem gives:

$$p(y_t \mid x_1, x_2, \ldots, x_t)$$

$$= p(y_t, x_1, x_2, \ldots, x_t)/p(x_1, x_2, \ldots, x_t)$$

$$= \alpha_t(y_t)/p(x_1, x_2, \ldots, x_t)$$

HIDDEN MARKOV MODELS

- **Forward algorithm:**
  - To compute the joint probability of the state at time **t** being $\mathbf{y_t}$ and the sequence of observations in the first **t** steps being $\mathbf{\{x_1, x_2, \ldots, x_t\}}$:

$$\alpha_t(y_t) = p(y_t, x_1, x_2, \ldots, x_t)$$

  - Bayes' theorem gives:

$$p(y_t \mid x_1, x_2, \ldots, x_t)$$

$$= p(y_t, x_1, x_2, \ldots, x_t)/p(x_1, x_2, \ldots, x_t)$$

$$= \alpha_t(y_t)/p(x_1, x_2, \ldots, x_t)$$

  - The highest $\boldsymbol{\alpha_t(y_t)}$ is, the most likely $\mathbf{y_t}$ would be given the same $\mathbf{\{x_1, x_2, \ldots, x_t\}}$.

HIDDEN MARKOV MODELS

- **Forward algorithm:**

  $\alpha_t(y_t)$

  $= p(y_t, x_1, x_2, \ldots, x_t)$

  $= \Sigma_{y_{t-1}} p(y_t, y_{t-1}, x_1, x_2, \ldots, x_t)$

  $= \Sigma_{y_{t-1}} p(x_t \mid y_t, y_{t-1}, x_1, x_2, \ldots, x_{t-1}).p(y_t, y_{t-1}, x_1, x_2, \ldots, x_{t-1})$

  $= \Sigma_{y_{t-1}} p(x_t \mid y_t).p(y_t \mid y_{t-1}, x_1, x_2, \ldots, x_{t-1}).p(y_{t-1}, x_1, x_2, \ldots, x_{t-1})$

  $= \Sigma_{y_{t-1}} p(x_t \mid y_t).p(y_t \mid y_{t-1}).p(y_{t-1}, x_1, x_2, \ldots, x_{t-1})$

  $= p(x_t \mid y_t) \Sigma_{y_{t-1}} p(y_t \mid y_{t-1}).\alpha_{t-1}(y_{t-1})$

HIDDEN MARKOV MODELS

- **Forward algorithm:**

$\alpha_t(y_t)$

$$\boxed{\alpha_1(y_1) = p(y_1, x_1) = p(x_1 | y_1).\, p(y_1)}$$

$= p(y_t, x_1, x_2, \ldots, x_t)$

$= \Sigma_{y_{t-1}} p(y_t, y_{t-1}, x_1, x_2, \ldots, x_t)$

$= \Sigma_{y_{t-1}} p(x_t \,|\, y_t, y_{t-1}, x_1, x_2, \ldots, x_{t-1}).p(y_t, y_{t-1}, x_1, x_2, \ldots, x_{t-1})$

$= \Sigma_{y_{t-1}} p(x_t \,|\, y_t).p(y_t \,|\, y_{t-1}, x_1, x_2, \ldots, x_{t-1}).p(y_{t-1}, x_1, x_2, \ldots, x_{t-1})$

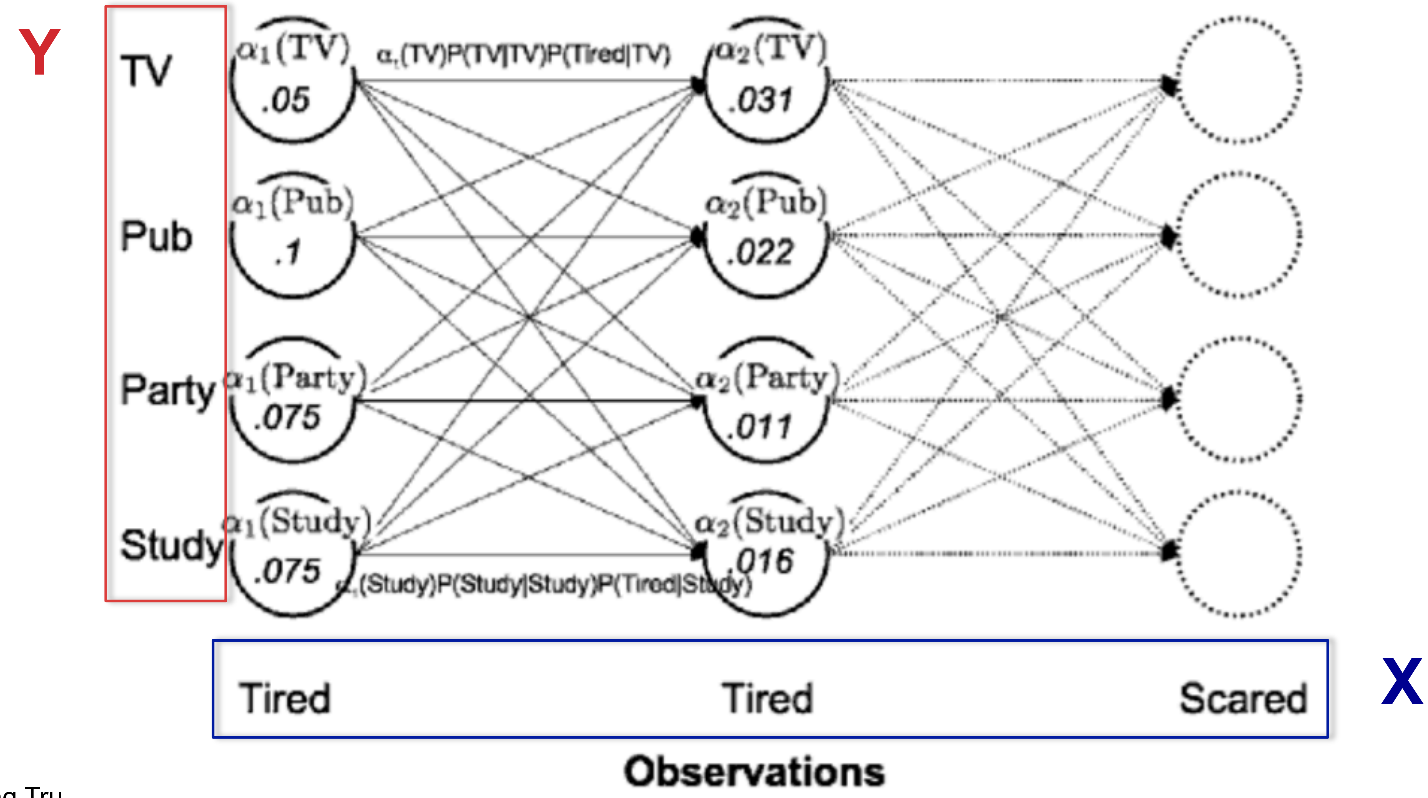
$= \Sigma_{y_{t-1}} p(x_t \,|\, y_t).p(y_t \,|\, y_{t-1}).p(y_{t-1}, x_1, x_2, \ldots, x_{t-1})$

$= p(x_t \,|\, y_t)\, \Sigma_{y_{t-1}}\, p(y_t \,|\, y_{t-1}).\alpha_{t-1}(y_{t-1})$

HIDDEN MARKOV MODELS

- **Forward algorithm:**

$$\alpha_t(y_t) = p(x_t \mid y_t) \sum_{y_{t-1}} p(y_t \mid y_{t-1}).\alpha_{t-1}(y_{t-1})$$



HIDDEN MARKOV MODELS

- **Viterbi algorithm:**
  - To compute the most probable sequence of states $\{y_1, y_2, \ldots, y_T\}$ given a sequence of observations $\{x_1, x_2, \ldots, x_T\}$:

$$Y^* = \mathrm{argmax}_Y \, p(Y \mid X) = \mathrm{argmax}_Y \, p(Y, X)$$

HIDDEN MARKOV MODELS

- **Viterbi algorithm:**
  - To compute the most probable sequence of states $\{y_1, y_2, \ldots, y_T\}$ given a sequence of observations $\{x_1, x_2, \ldots, x_T\}$:

$$Y^* = \text{argmax}_Y\ p(Y \mid X) = \text{argmax}_Y\ p(Y, X)$$

Andrew J. Viterbi[1]

Born	March 9, 1935 (age 80) Bergamo, Italy
Nationality	Italian, American
Education	Massachusetts Institute of Technology (BS, MS) University of Southern California (PhD)

HIDDEN MARKOV MODELS

- **Viterbi algorithm:**

$\max_{y_{1:T}} p(y_1, y_2, \ldots, y_T, x_1, x_2, \ldots, x_T)$

$= \max_{y_T} \max_{y_{1:T-1}} p(y_1, y_2, \ldots, y_T, x_1, x_2, \ldots, x_T)$

$= \max_{y_T} \max_{y_{1:T-1}} \{p(x_T \mid y_T).p(y_T \mid y_{T-1}).p(y_1, \ldots, y_{T-1}, x_1, \ldots, x_{T-1})\}$

$= \max_{y_T} \max_{y_{T-1}} \{p(x_T \mid y_T).p(y_T \mid y_{T-1}).\max_{y_{1:T-2}} p(y_1, \ldots, y_{T-1}, x_1, \ldots, x_{T-1})\}$

HIDDEN MARKOV MODELS

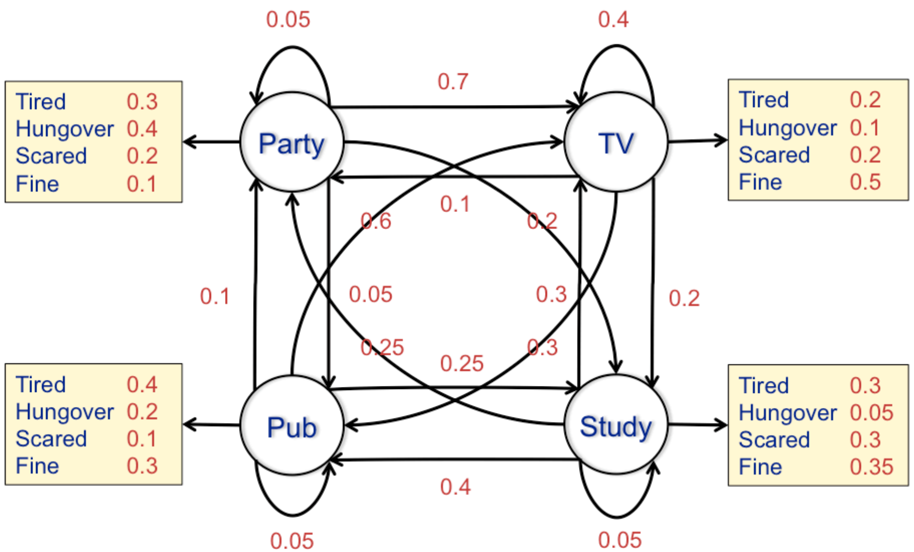
- **Viterbi algorithm:**

$\max_{y_{1:T}} p(y_1, y_2, \ldots, y_T, x_1, x_2, \ldots, x_T)$

$= \max_{y_T} \max_{y_{1:T-1}} p(y_1, y_2, \ldots, y_T, x_1, x_2, \ldots, x_T)$

$= \max_{y_T} \max_{y_{1:T-1}} \{p(x_T \mid y_T).p(y_T \mid y_{T-1}).p(y_1, \ldots, y_{T-1}, x_1, \ldots, x_{T-1})\}$

$= \max_{y_T} \max_{y_{T-1}} \{p(x_T \mid y_T).p(y_T \mid y_{T-1}).\max_{y_{1:T-2}} p(y_1, \ldots, y_{T-1}, x_1, \ldots, x_{T-1})\}$



HIDDEN MARKOV MODELS

- **Viterbi algorithm:**

  $\max_{y_{1:T}} p(y_1, y_2, \ldots, y_T, x_1, x_2, \ldots, x_T)$

  $= \max_{y_T} \max_{y_{1:T-1}} p(y_1, y_2, \ldots, y_T, x_1, x_2, \ldots, x_T)$

  $= \max_{y_T} \max_{y_{1:T-1}} \{p(x_T \mid y_T).p(y_T \mid y_{T-1}).p(y_1, \ldots, y_{T-1}, x_1, \ldots, x_{T-1})\}$

  $= \max_{y_T} \max_{y_{T-1}} \{p(x_T \mid y_T).p(y_T \mid y_{T-1}).\max_{y_{1:T-2}} p(y_1, \ldots, y_{T-1}, x_1, \ldots, x_{T-1})\}$

- **Dynamic programming:**
  - Compute

    $$\operatorname{argmax}_{y_1} p(y_1, x_1) = \operatorname{argmax}_{y_1} p(x_1 \mid y_1).p(y_1)$$

  - For each t from 2 to T, and for each state y_t, compute:

    $$\operatorname{argmax}_{y_{1:t-1}} p(y_1, y_2, \ldots, y_t, x_1, x_2, \ldots, x_t)$$

  - Select $\operatorname{argmax}_{y_{1:T}} p(y_1, y_2, \ldots, y_T, x_1, x_2, \ldots, x_T)$

HIDDEN MARKOV MODELS

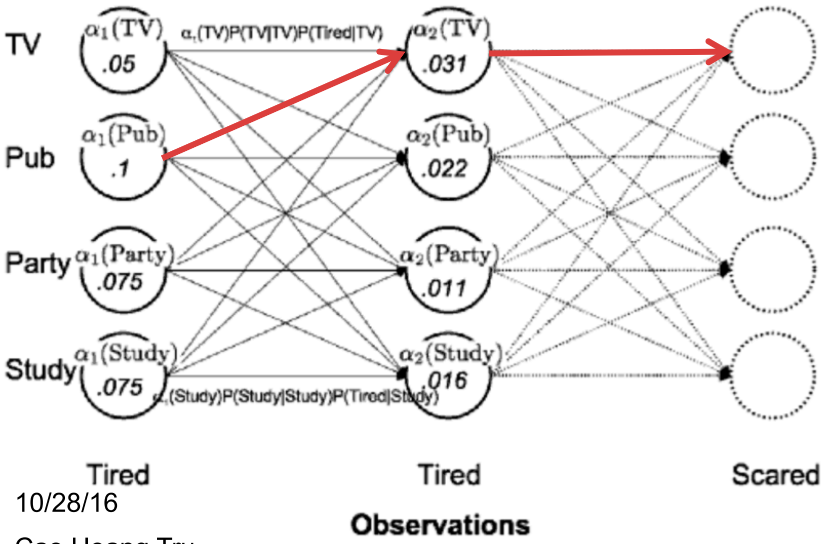
- **Dynamic programming:**
  - Compute

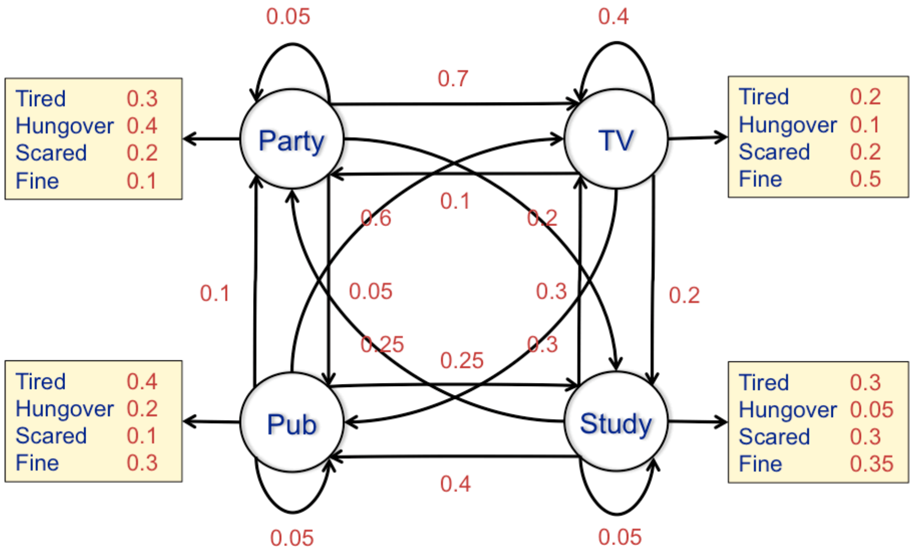
$$\text{argmax}_{y_1}\, p(y_1, x_1) = \text{argmax}_{y_1}\, p(x_1 \mid y_1).p(y_1)$$

  - For each t from 2 to T, and for each state y_t, compute:

$$\text{argmax}_{y_{1:t-1}}\, p(y_1, y_2, \ldots, y_t, x_1, x_2, \ldots, x_t)$$

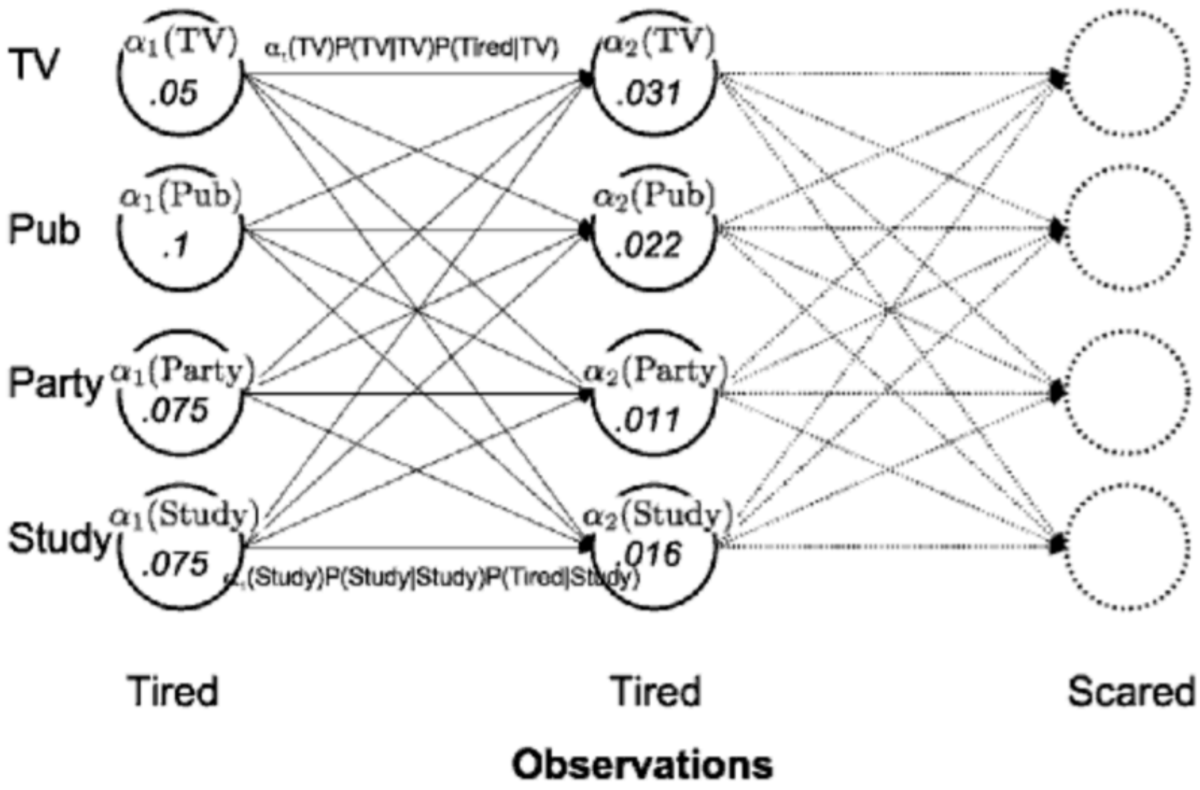
  - Select $\text{argmax}_{y_{1:T}}\, p(y_1, y_2, \ldots, y_T, x_1, x_2, \ldots, x_T)$



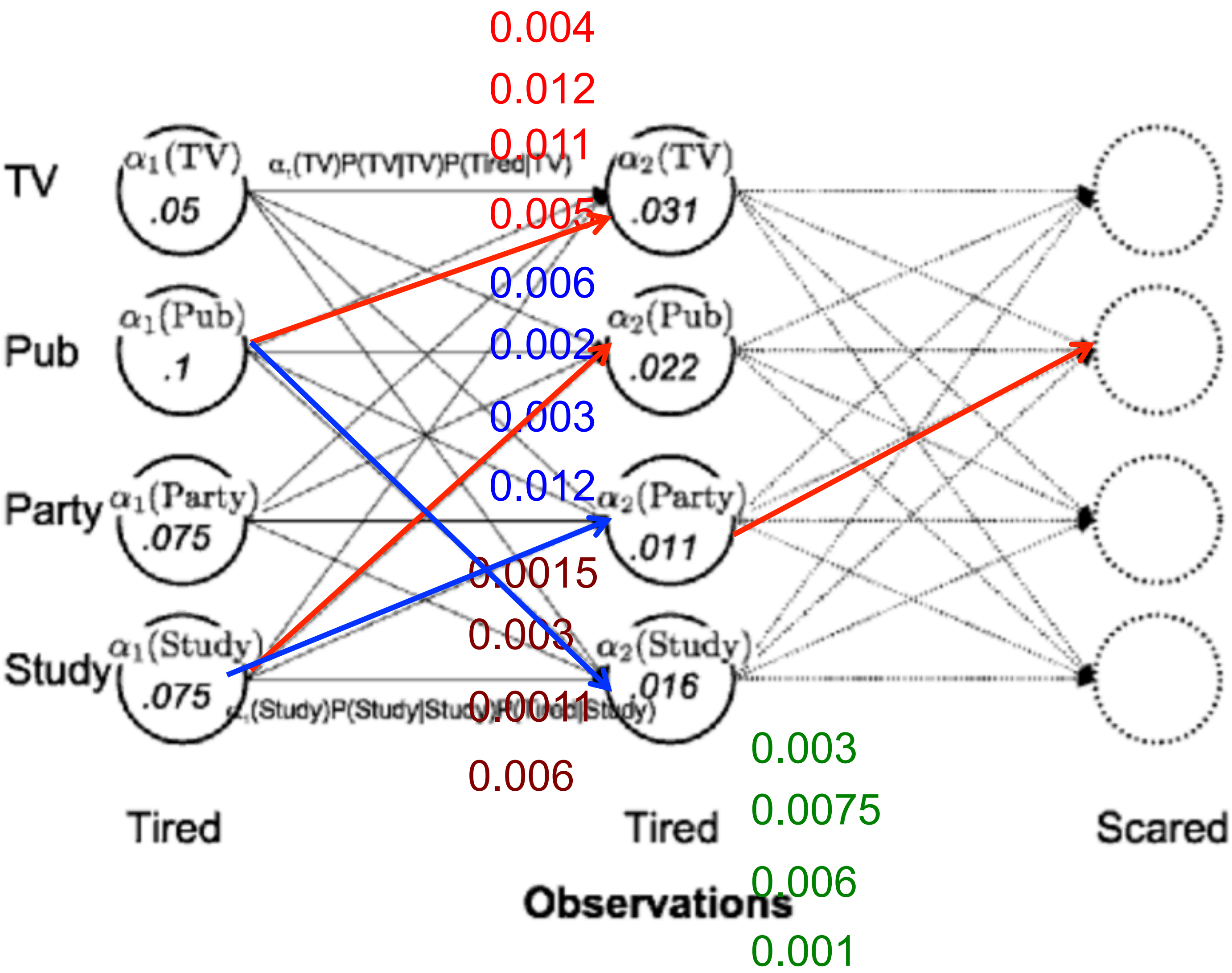


HIDDEN MARKOV MODELS

- **Could the results from the forward algorithm be used for Viterbi algorithm?**

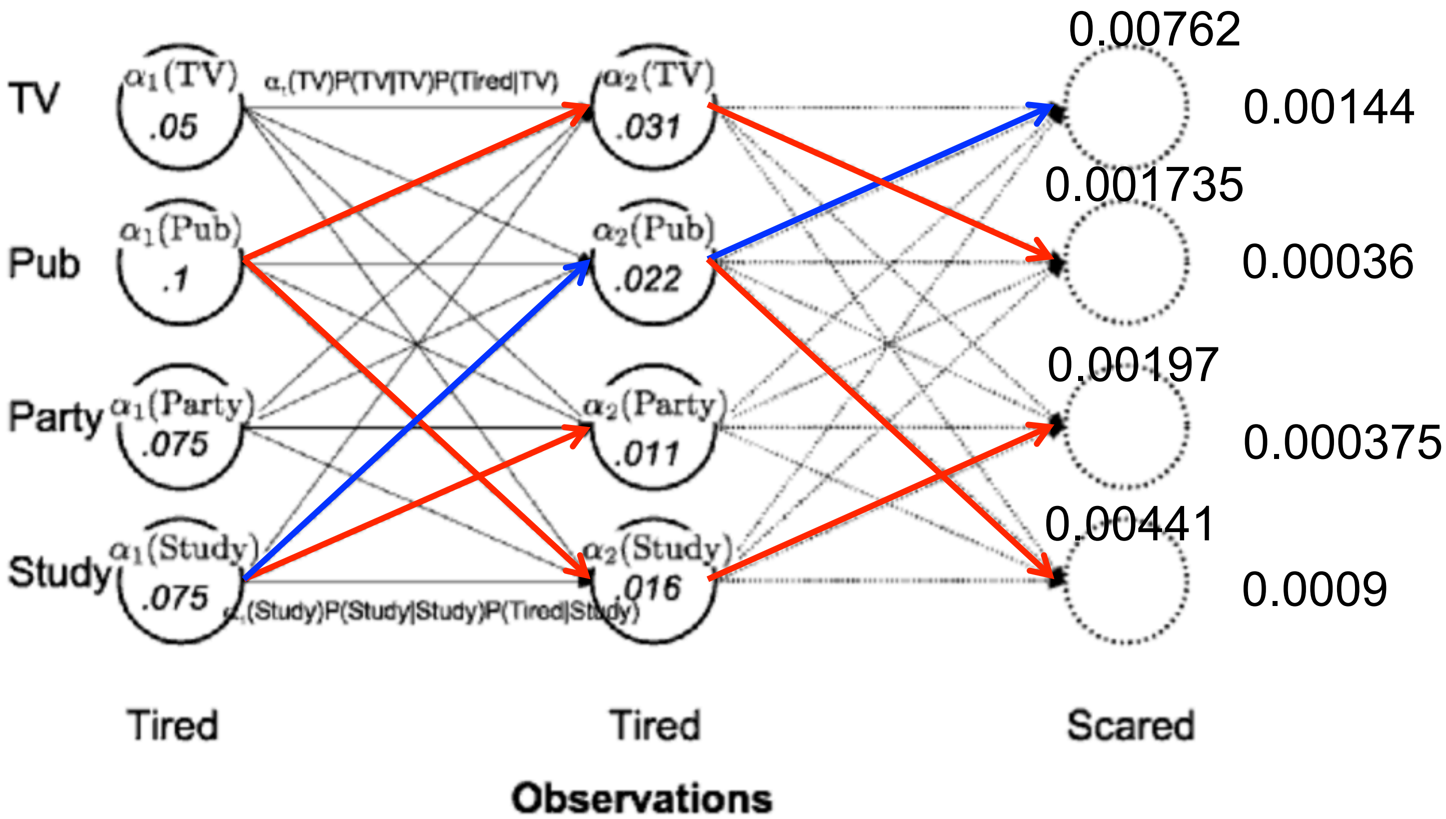


HIDDEN MARKOV MODELS



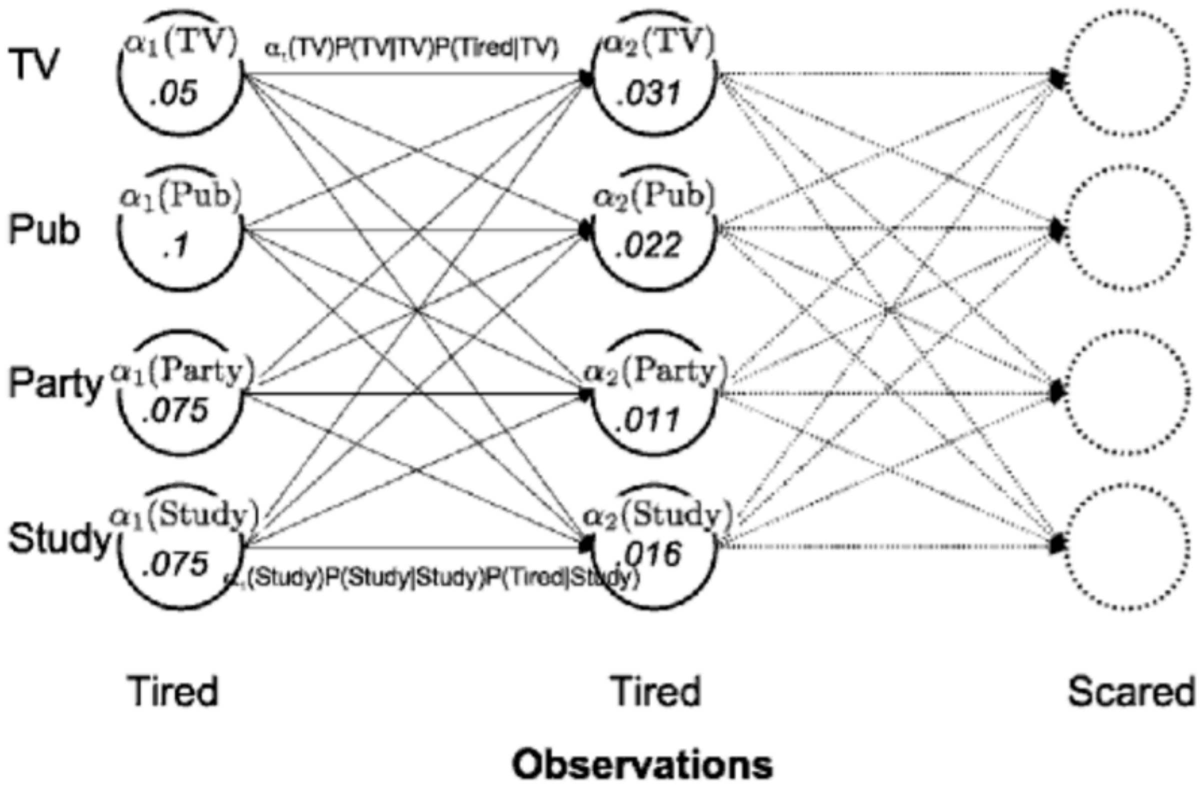
HIDDEN MARKOV MODELS

- **Could the results from the forward algorithm be used for Viterbi algorithm?**



HIDDEN MARKOV MODELS

- **Could the results from the forward algorithm be used for Viterbi algorithm?**



READING HOMEWORK 1

- **Marsland, S. (2009) Machine learning: An algorithmic perspective. Chapter 15 (graphical models).**

EXERCISES 1

- **Apply Viterbi algorithm to find the most probable 3-state sequence in the looking-activity example in the lecture.**

HIDDEN MARKOV MODELS

- **Where does an HMM come from?**

HIDDEN MARKOV MODELS

- **Training HMMs:**
  - Topology is designed beforehand.
  - Parameters to be learned: **emission** and **transition** probabilities.
  - Supervised or unsupervised training.

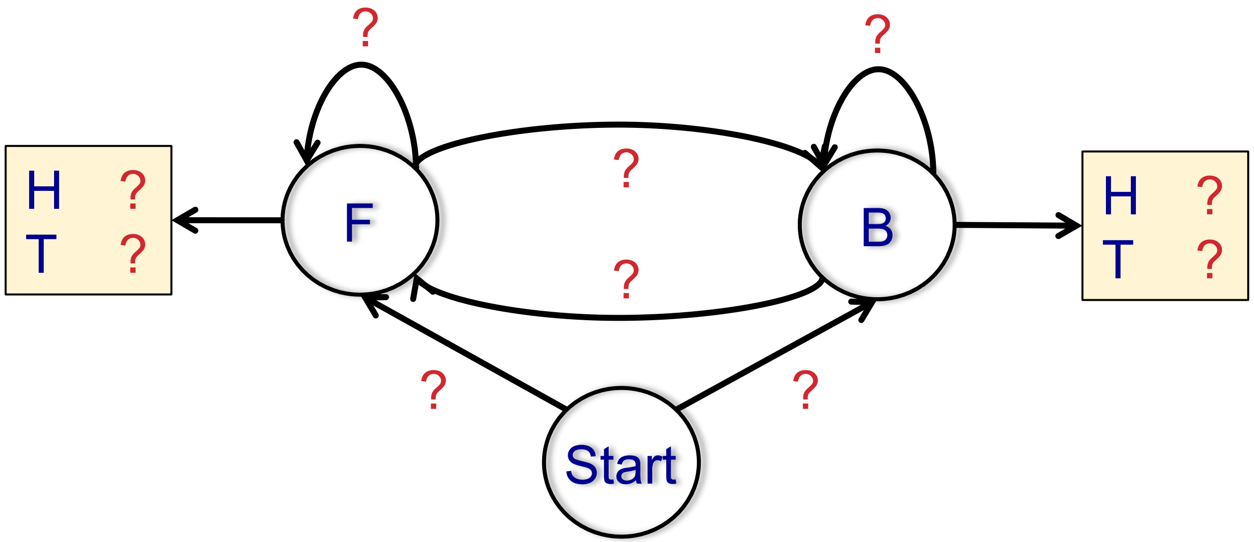
HIDDEN MARKOV MODELS

- **Supervised learning:**
  - Training data: paired sequences of states and observations $(y_1, y_2, \ldots, y_T, x_1, x_2, \ldots, x_T)$
  - $p(y_i)$ = num. of sequences starting with y_i/num. of all sequences.
  - $p(y_j \mid y_i)$ = number of (y_i, y_j)'s/number of all (y_i, y)'s.
  - $p(x_j \mid y_i)$ = number of (y_i, x_j)'s/number of all (y_i, x)'s.

HIDDEN MARKOV MODELS

- **Supervised learning example:**

FFFBFF	BFFBFF	FFBFFF	FFFFBF
HHTHTH	THTHTH	THHTTH	THTTTH
BFFFBF	FFFBBF	BFFFFF	BFBFFF
THHTHT	HHTHHT	HHTTHT	HTTTHH



HIDDEN MARKOV MODELS

- **Unsupervised learning:**
  - Only observation sequences are available.

~~FFFBFF~~	~~BFFBFF~~	~~FFBFFF~~	~~FFFFBF~~
HHTHTH	THTHTH	THHTTH	THTTTH
~~BFFFBF~~	~~FFFBBF~~	~~BFFFFF~~	~~BFBFFF~~
THHTHT	HHTHHT	HHTTHT	HTTTHH

HIDDEN MARKOV MODELS

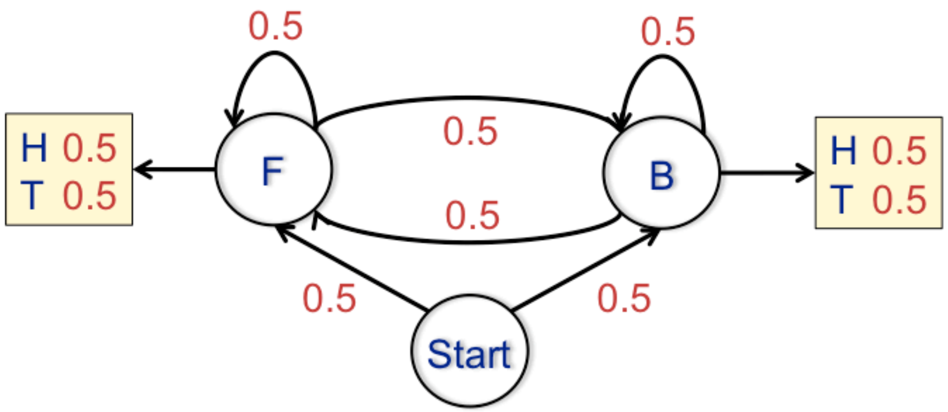
- **Unsupervised learning:**
  - Only observation sequences are available.

~~FFFBFF~~	~~BFFBFF~~	~~FFBFFF~~	~~FFFFBF~~
HHTHTH	THTHTH	THHTTH	THTTTH
~~BFFFBF~~	~~FFFBBF~~	~~BFFFFF~~	~~BFBFFF~~
THHTHT	HHTHHT	HHTTHT	HTTTHH

  - Iterative improvement of model parameters.
  - How?

HIDDEN MARKOV MODELS

- **Unsupervised learning:**
  - Initialize estimated parameters.



  - For each observation sequence, compute the most probable state sequence, using Viterbi algorithm.
  - Update the parameters using supervised learning on obtained paired state-observation sequences.
  - Repeat it until convergence.

HIDDEN MARKOV MODELS

- **Application to NER:**
  - Example: “**Facebook** CEO **Zuckerberg** visited **Vietnam**”.

Facebook	CEO	**Zuckerberg**	visited	**Vietnam**
ORG	NIL	**PER**	NIL	**LOC**

HIDDEN MARKOV MODELS

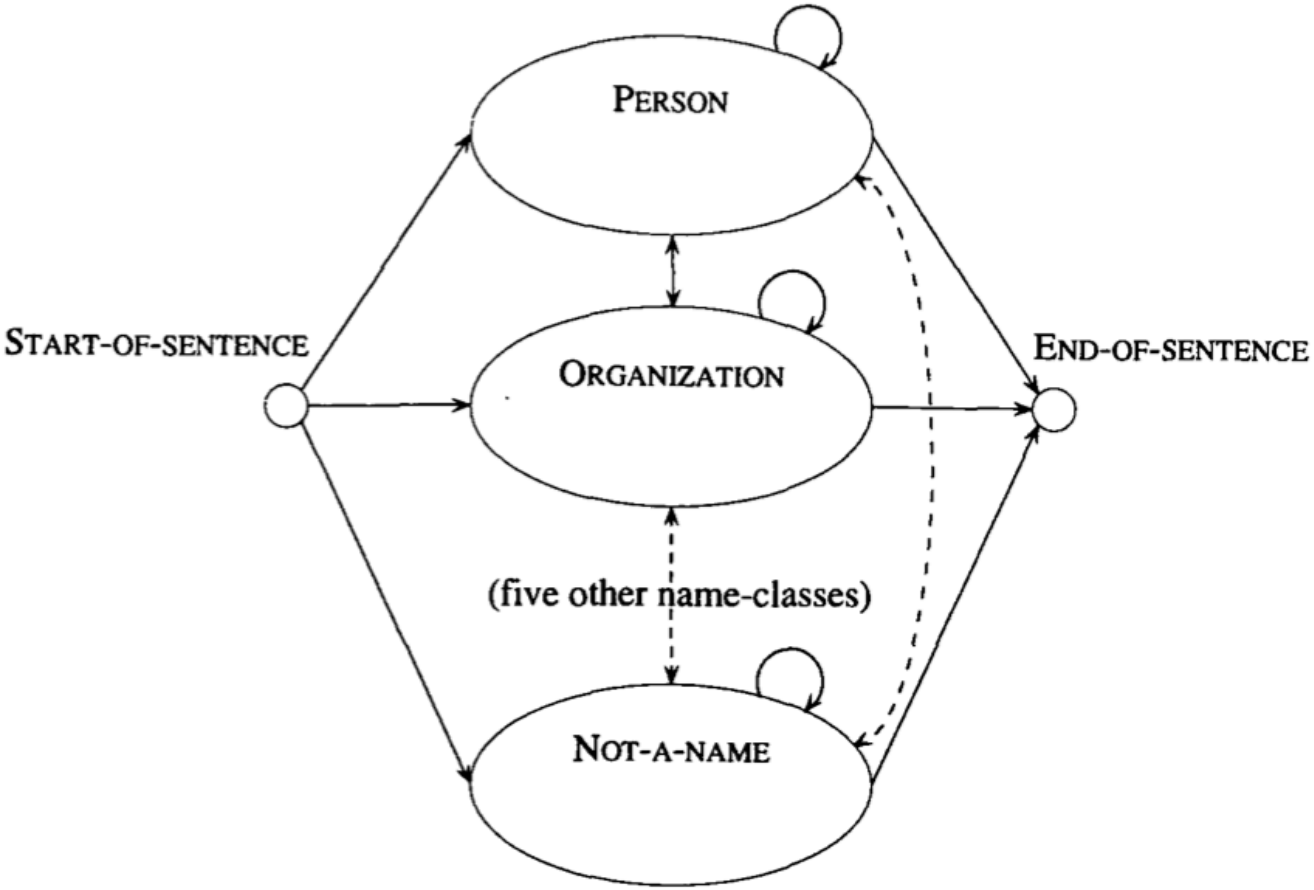
- **Application to NER:**
  - Example: “**Facebook** CEO **Zuckerberg** visited **Vietnam**”.

ORG	NIL	**PER**	NIL	**LOC**

  - States = Class labels
  - Observations = Words + Features

HIDDEN MARKOV MODELS

- **Application to NER:**



Bikel, D.M. (1997) A high-performance learning name-finder.

HIDDEN MARKOV MODELS

- **Application to NER:**
  - What if a name is a multi-word phrase?
  - Example: "… **John von Neumann** is …"

HIDDEN MARKOV MODELS

- **Application to NER:**
  - What if a name is a multi-word phrase?
  - Example: "… **John von Neumann** is …"

    B-PER **I-PER** **I-PER** **O**
  - BIO notation:
    {B-PER, **I-PER, B-ORG, I-ORG, B-LOC, I-LOC, B-MISC, I-MISC, O}**

READING HOMEWORK 2

- **Marsland, S. (2009) Machine learning: An algorithmic perspective. Chapter 15 (graphical models).**
- **Bikel, D.M. et al. (1997) Nymble: a high performance learning name-finder.**

EXERCISES 2

- **Write a program to carry out the unsupervised learning example for HMM in the lecture. Discuss on the result, in particular the convergence of the process.**