POLITÉCNICA
"Ingeniamos el futuro"

CAMPUS
DE EXCELENCIA
INTERNACIONAL

Text Mining 5

for real, now!

Information Extraction

Madrid Summer School 2014
Advanced Statistics and Data Mining

Florian Leitner
florian.leitner@upm.es

Incentive and Applications

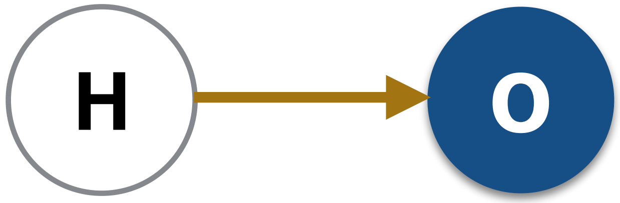
➡ Statistical analyses of text **sequences**

- Text Segmentation
  - word & sentence boundaries (see Text Mining #3)
- Part-of-Speech (PoS) Tagging & Chunking
  - noun, verb, adjective, … & noun/verb/preposition/… phrases
- Named Entity Recognition (NER)
  - organizations, persons, places, genes, chemicals, …
- Information Extraction
  - locations/times, constants, formulas, entity linking, entity-relationships, …

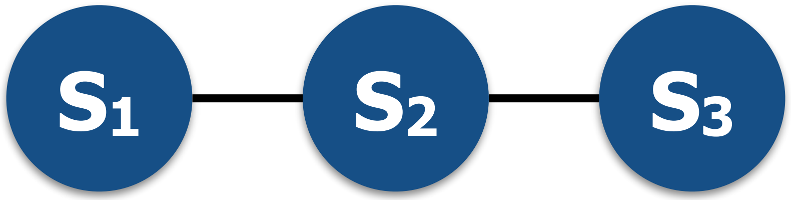
Probabilistic Graphical Models (PGM)

- Graphs of hidden (blank) and **observed** (shaded) **variables** (vertices/nodes).
- The edges depict **dependencies**, and if directed, show the **causal relationship** between nodes.

directed ➡ Bayesian Network (BN)



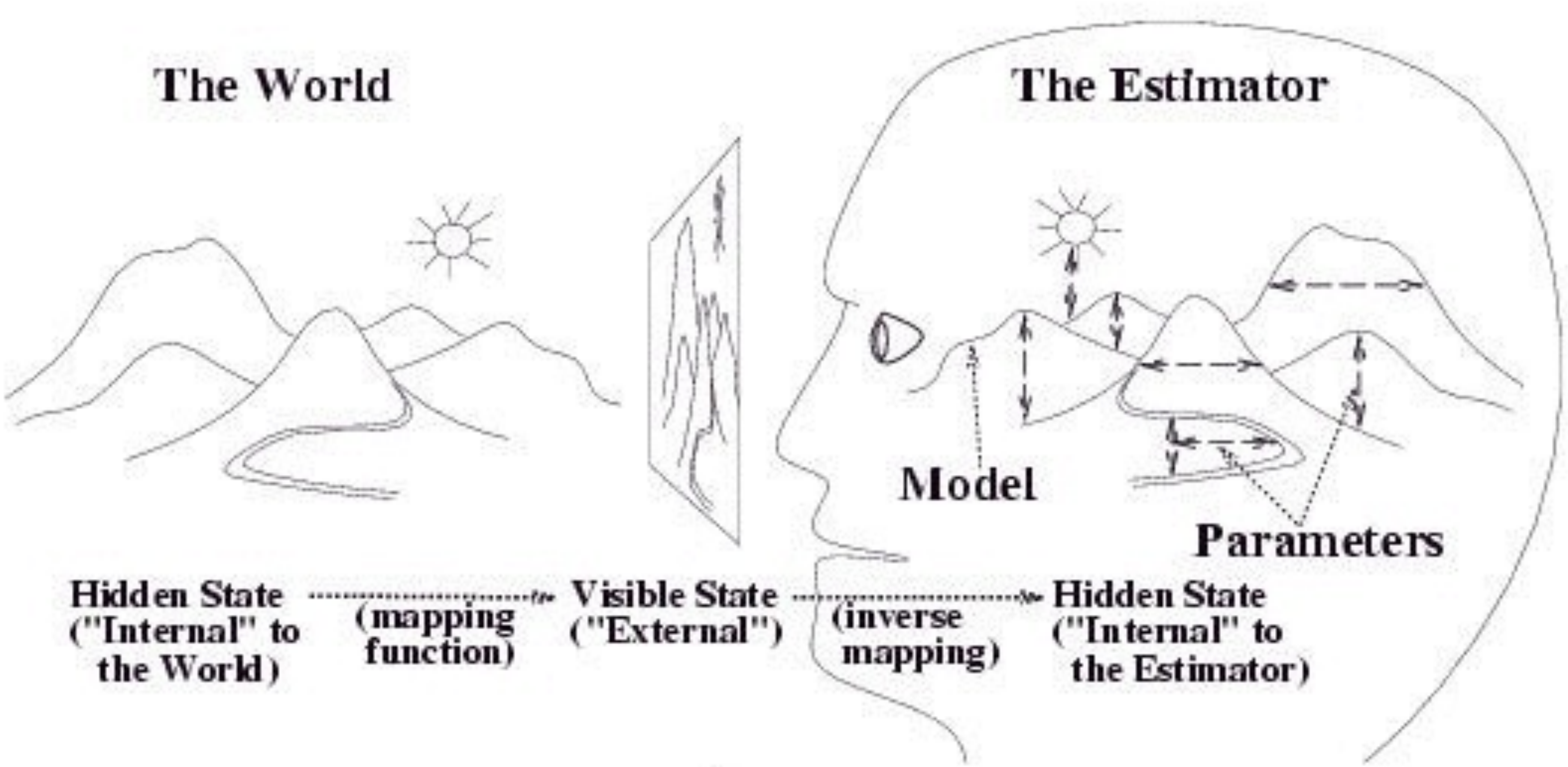
undirected ➡ Markov Random Field (MRF)



mixed ➡ Mixture Models

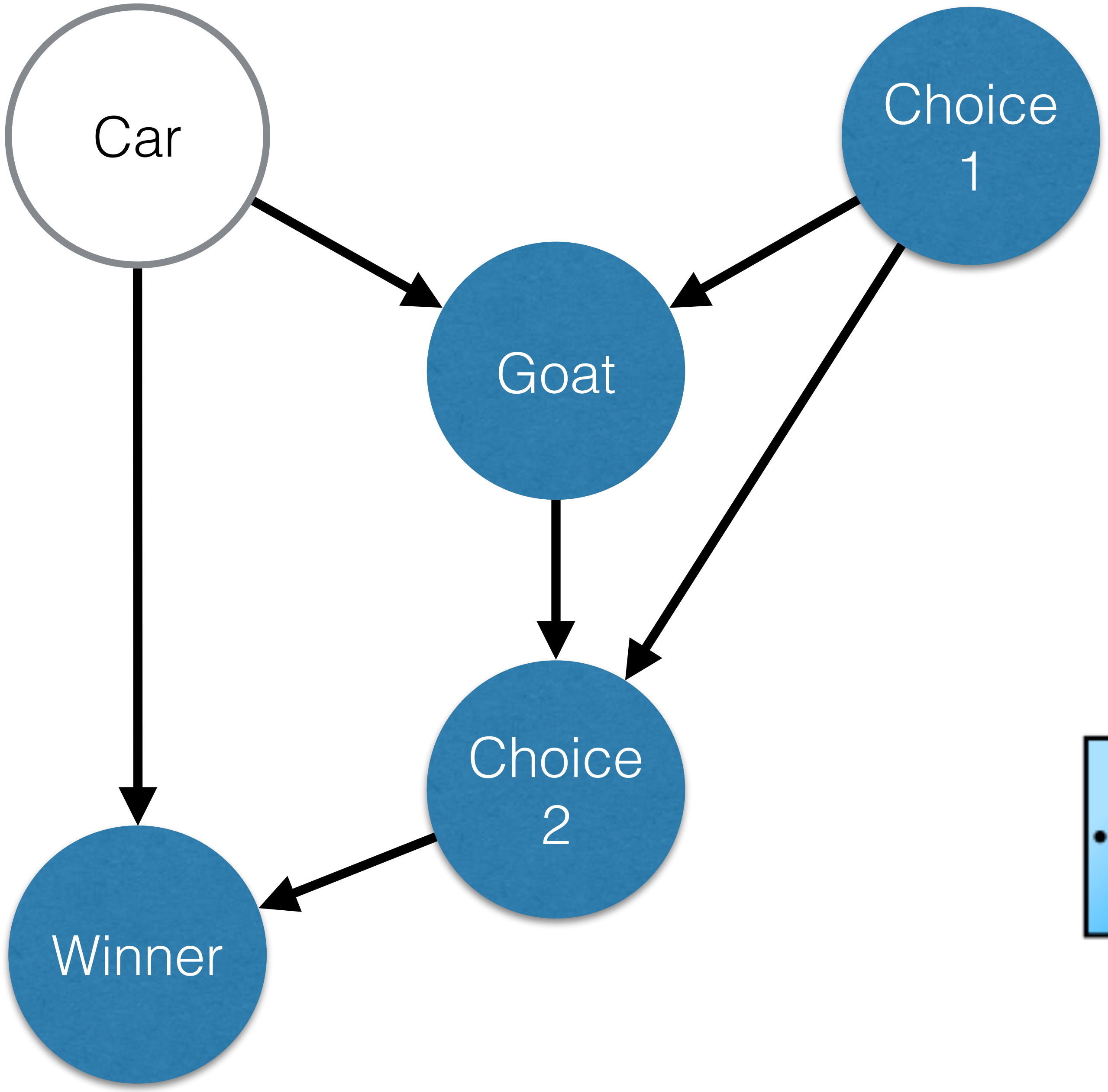
Koller & Friedman. Probabilistic Graphical Models. 2009

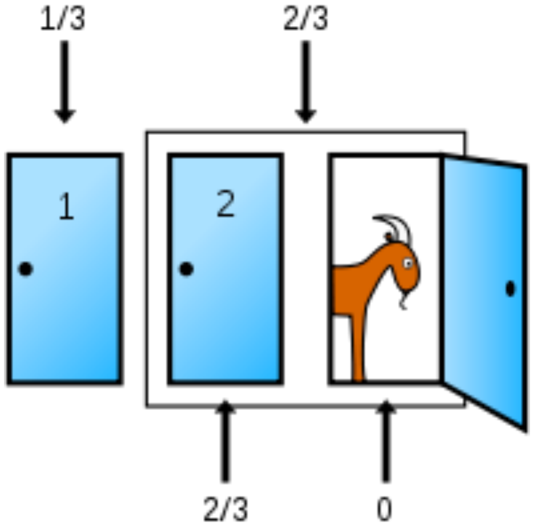
Hidden vs. Observed State and Statistical Parameters


The World
The Estimator
Model
Parameters
Hidden State ("Internal" to the World)
(mapping function)
Visible State ("External")
(inverse mapping)
Hidden State ("Internal" to the Estimator)

Rao. A Kalman Filter Model of the Visual Cortex. Neural Computation 1997

A Bayesian Network for the Monty Hall Problem



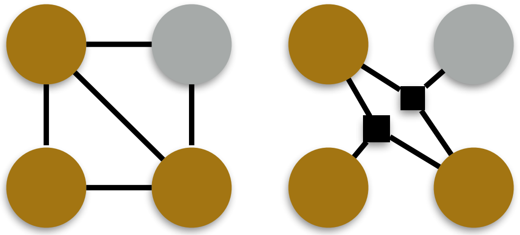


Markov Random Field

History class: Ising developed a linear field to model (binary) atomic spin states (Ising, 1924); the 2-dim. model problem was solved by Onsager in 1944.

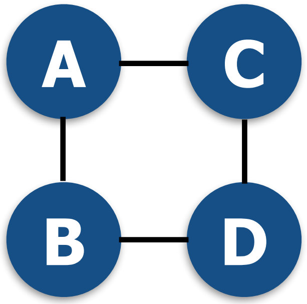
$$P(X = \vec{x}) = \frac{\prod_{cl \in \vec{x}} \overbrace{\phi_{cl}(cl)}^{\text{factor (clique potential)}}}{\underbrace{\sum_{\vec{x} \in X} \prod_{cl \in \vec{x}} \phi_{cl}(cl)}_{\text{normalizing constant (partition function } \mathbf{Z}\text{)}}}$$

cl … [**maximal**] **clique**; a subset of nodes in the graph where every pair of nodes is connected



θ(A, B)			θ(B, C)			θ(C, D)			θ(D, A)		
a	b	30	b	c	100	c	d	1	d	a	100
a	b	5	b	c	1	c	d	100	d	a	1
a	b	1	b	c	1	c	d	100	d	a	1
a	b	10	b	c	100	c	d	1	d	a	100

factor table



factor graph

$P(a_1, b_1, c_0, d_1) =$
10·1·100·100 ÷
7’201’840 =
0.014

A First Look At Probabilistic Graphical Models

- Latent Dirichlet Allocation: LDA
  - Blei, Ng, and Jordan. Journal of Machine Learning Research 2003
  - For assigning "topics" to "documents"

i.e., Text Classification; last lesson (4)

Latent Dirichlet Allocation (LDA 1/3)

- Intuition for LDA
  - From: Edwin Chen. Introduction to LDA. 2011
  - "Document Collection"

- I like to eat broccoli and bananas.
- I ate a banana and spinach smoothie for breakfast.

➡ Topic A

- Chinchillas and kittens are cute.
- My sister adopted a kitten yesterday.

➡ Topic B

- Look at this cute hamster munching on a piece of broccoli.

➡ Topic 0.6A + 0.4B

Topic A: 30% broccoli, 15% bananas, 10% breakfast, 10% munching, …

Topic B: 20% chinchillas, 20% kittens, 20% cute, 15% hamster, …

The Dirichlet Process

a Dirichlet Process is like drawing from an
(infinite) "bag of dice" (with finite faces)

- A Dirichlet is a [possibly continuos] **distribution over** [discrete/multinomial] **distributions** (probability **masses**).

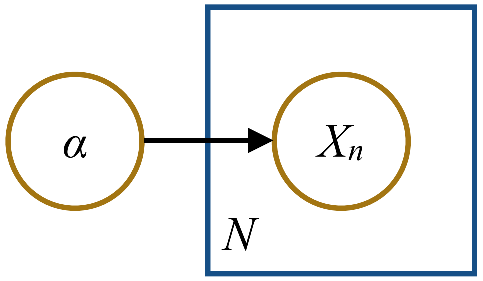
Gamma function -> a
"continuous" factorial [!]

$$D(\boldsymbol{\theta}, \boldsymbol{\alpha}) = \frac{\Gamma(\sum \alpha_i)}{\prod \Gamma(\alpha_i)} \prod \theta_i^{\alpha_i - 1}$$

α Dirichlet prior: $\forall\ \alpha_i \in \alpha : \alpha_i > 0$

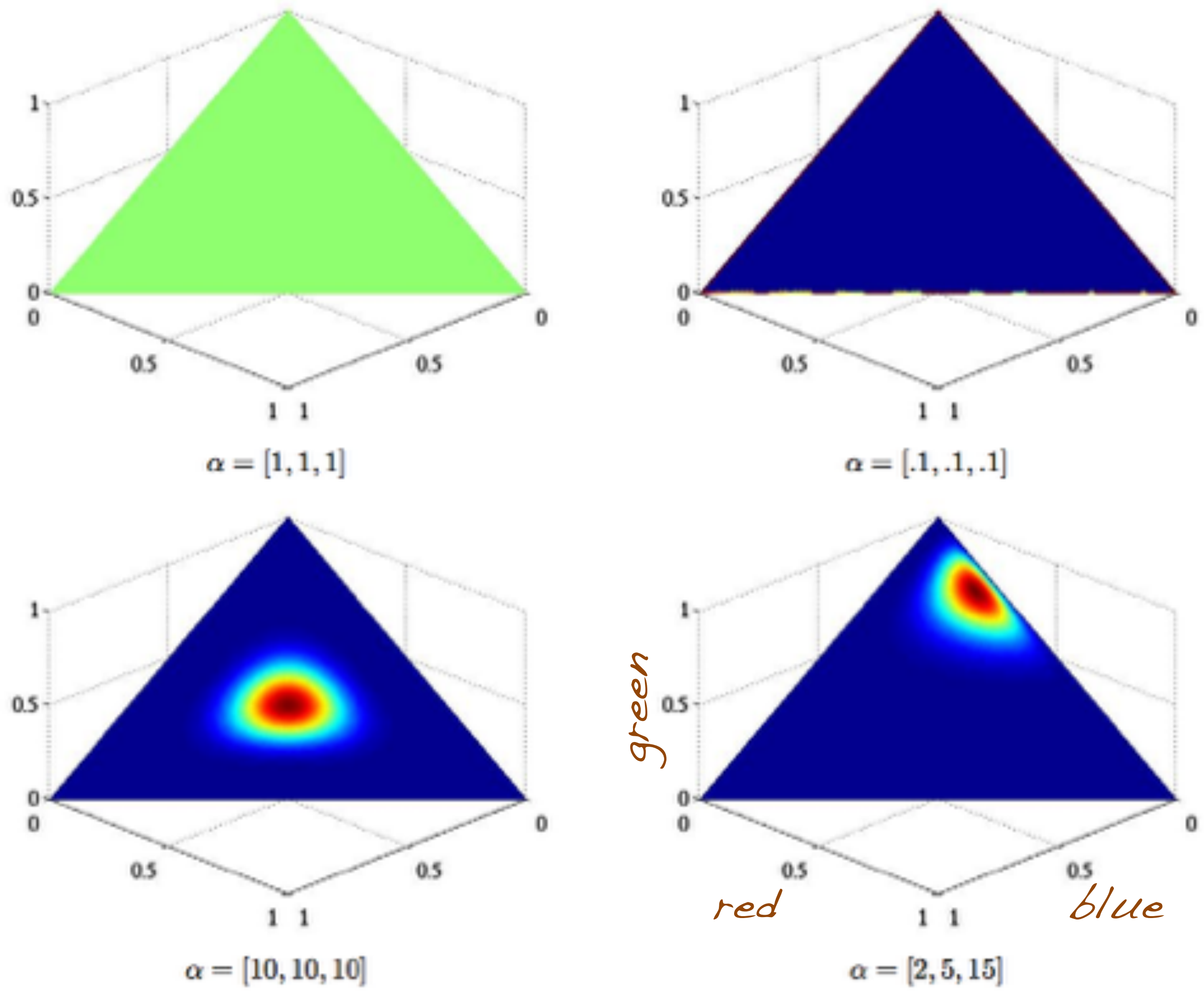
$\sum \theta_i = 1$; a PMF

- The **Dirichlet Process samples** multiple independent, discrete **distributions** θ_i with repetition from $\boldsymbol{\theta}$ ("statistical clustering").



1. Draw a new distribution X from $D(\boldsymbol{\theta}, \boldsymbol{\alpha})$
2. With probability $\alpha \div (\alpha + n - 1)$ draw a new X
   With probability $n \div (\alpha + n - 1)$, (re-)sample an X_i from X

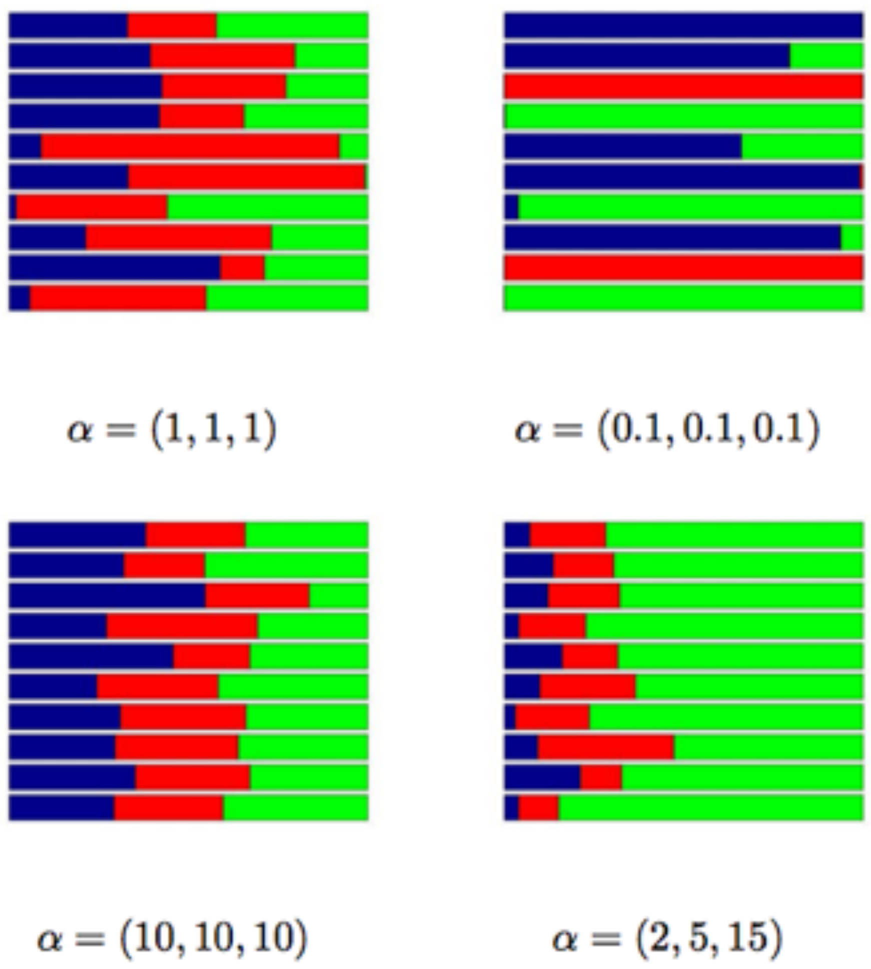
The Dirichlet Prior α



$\alpha = [1, 1, 1]$ $\alpha = [.1, .1, .1]$

$\alpha = [10, 10, 10]$ $\alpha = [2, 5, 15]$

Documents and Topic distributions (N=3)



$\alpha = (1, 1, 1)$ $\alpha = (0.1, 0.1, 0.1)$

$\alpha = (10, 10, 10)$ $\alpha = (2, 5, 15)$

↝ equal, =1 ➡ uniform distribution

↝ equal, <1 ➡ marginal distrib. ("choose few")

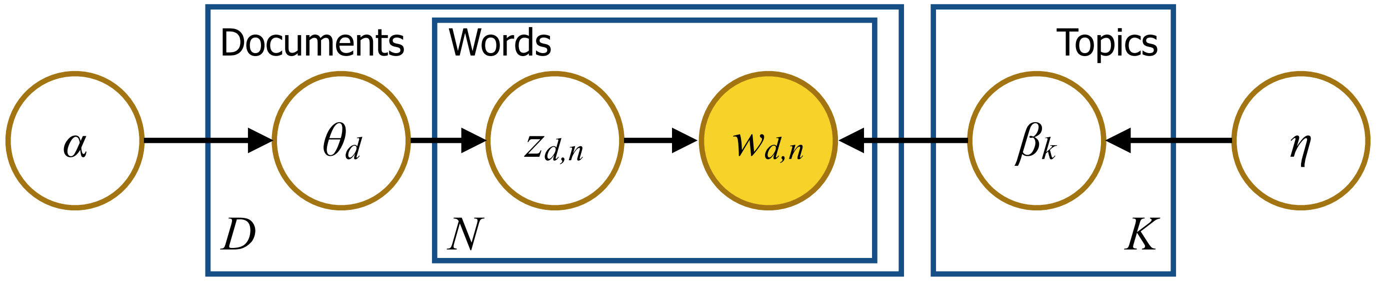
↝ equal, >1 ➡ symmetric, mono-modal distrib.

↝ not equal, >1 ➡ non-symmetric distribution

Frigyik et al. Introduction to the Dirichlet Distribution and Related Processes. 2010

Latent Dirichlet Allocation (LDA 2/3)

Posterior: P(Topic | Word) ☛ P(B, Θ, Z | W) = P(B, Θ, Z, W) ÷ P(W)



Joint Probability

P(Document-T.) P(Word | Topics, Word-T.)

$$P(B,\Theta,Z,W) = \left(\prod_k^K P(\beta_k|\eta)\right)\left(\prod_d^D P(\theta_d|\alpha)\prod_n^N P(z_{d,n}|\theta_d)P(w_{d,n}|\beta_{1:K},z_{d,n})\right)$$

P(Topics) P(Word-T. | Document-T.)

- α - per-document Dirichlet prior
- θ_d - topic distribution of document d
- $z_{d,n}$ - word-topic assignments
- $w_{d,n}$ - **observed** words
- β_k - word distrib. of topic k
- η - per-topic Dirichlet prior

dampens the topic-specific score of terms assigned to many topics

$$termscore_{k,n} = \hat{\beta}_{k,n} \boxed{log \frac{\hat{\beta}_{k,n}}{\left(\prod_j^K \hat{\beta}_{j,n}\right)^{1/K}}}$$

What Topics is a Word assigned to?

Latent Dirichlet Allocation (LDA 3/3)

- LDA inference in a nutshell
  - **Calculate the probability that Topic t generated Word w.**
  - Initialization: Choose K and randomly assign one out of the K Topics to each of the N Words in each of the D Documents.
    - The **same word** can have different Topics **at different positions** in the Document.
  - Then, for each Topic: And for each Word in each Document:

1. Compute P(Word-Topic | Document): the proportion of [Words assigned to] Topic t in Document d
2. Compute P(Word | Topics, Word-Topic): the probability a Word w is assigned a Topic t (using the general distribution of Topics and the Document-specific distribution of [Word-] Topics)
   - Note that a Word can be assigned a different Topic each time it appears in a Document.
3. Given the prior probabilities of a Document's Topics and that of Topics in general, reassign P(Topic | Word) = P(Word-Topic | Document) * P(Word | Topics, Word-Topic)

- Repeat until P(Topic | Word) stabilizes (e.g., MCMC Gibbs sampling, Course 04)

MCMC in Python: PyMC or PySTAN

Retrospective

We have seen how to…

- Design generative **models** of natural **language**.
- **Segment**, **tokenize**, and **compare** text and/or sentences.
- [Multi-] **labeling** whole documents or chunks of text.

Some open questions…

- **How to assign labels to individual tokens in a stream?**
  - **without using a dictionary/gazetteer**
- How to detect **semantic relationships** between tokens?
  - dependency & phrase structure parsing *not in this class* 😢

Probabilistic Models for Sequential Data

- a.k.a. **Temporal** or **Dynamic Bayesian Networks**
  - **static** process w/ **constant** model ➡ **temporal** process w/ **dynamic** model
  - model structure and parameters are [still] **constant**
  - the topology within a [constant] "**time slice**" is depicted

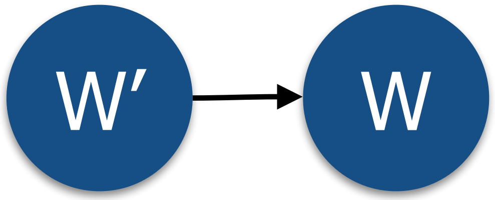
generative

- **Markov** Chain (MC; Markov. 1906)
- Hidden **Markov** Model (HMM; Baum et al. 1970)

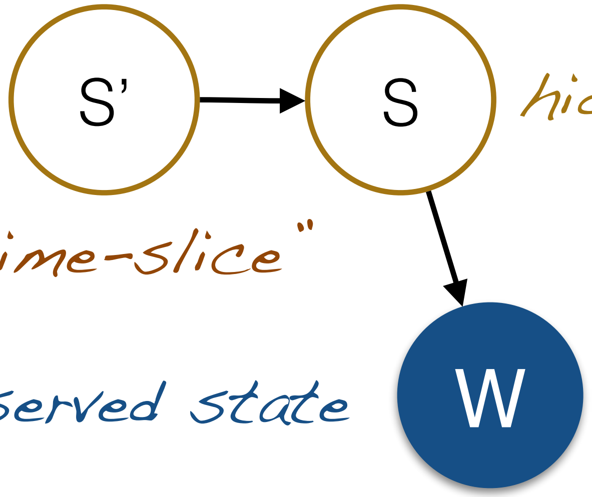
discriminative

- MaxEnt **Markov** Model (MEMM; McCallum et al. 2000)
- [**Markov**] Conditional Random Field (CRF; Lafferty et al. 2001)
  - Naming: all four models make the **Markov assumption** (Text Mining 2)

From A Markov Chain to a Hidden Markov Model



P(W | W')

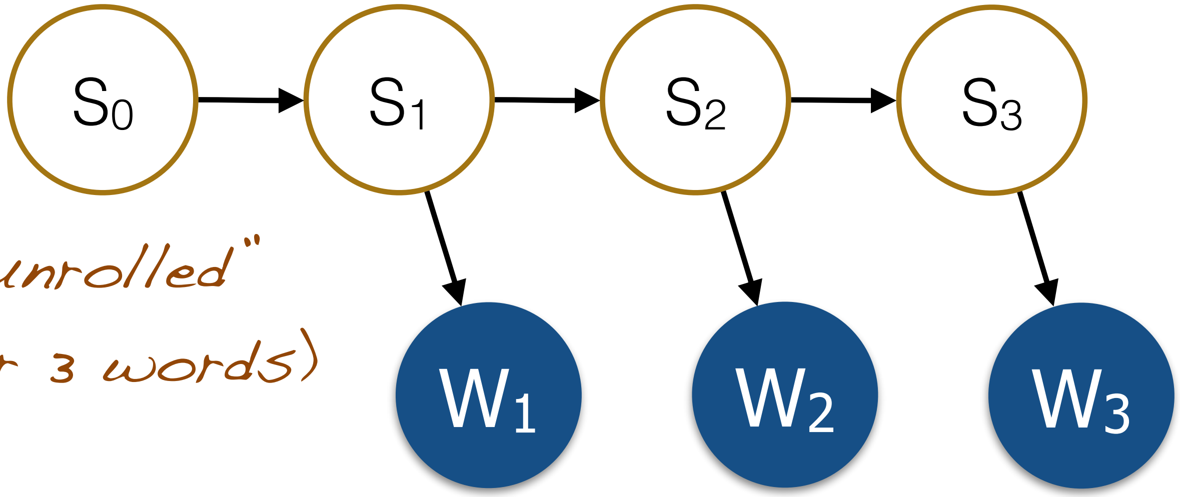


hidden states

"time-slice"

observed state

P(W | S)
P(S | S')

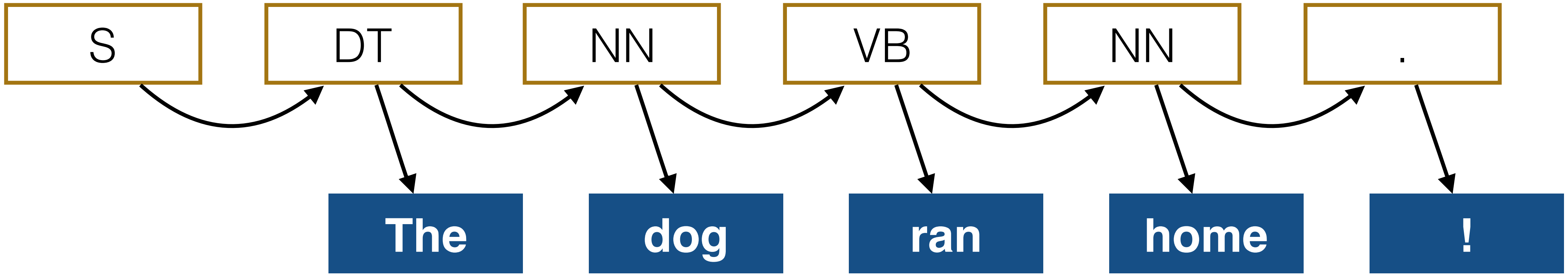


"unrolled"
(for 3 words)

W depends on S
and S in turn
only depends on
S'

A Language-Based Intuition for Hidden Markov Models

- A Markov Chain: $P(W) = \prod P(w \mid w')$
  - assumes the observed words are in and of themselves the cause of the observed sequence.
- A Hidden Markov Model $P(S, W) = \prod P(s \mid s') \, P(w \mid s)$
  - assumes the observed words are emitted by a hidden (not observable) sequence, for example the chain of part-of-speech-states.



again, this is the "unrolled" model that does not depict the conditional dependencies

The two Matrices of a Hidden Markov Model

a.k.a. "CPDs": Conditional Probability Distributions

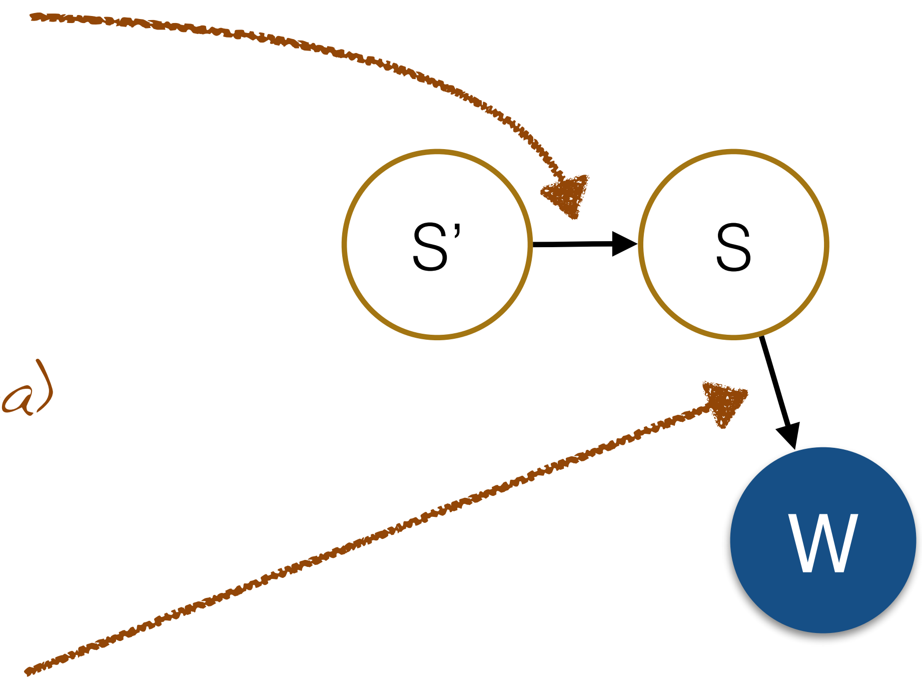
P(s I s’)	DT	NN	VB	…
DT	0.03	0.7	0	
NN	0	0	0.5	
VB	0	0.5	0.2	
…				

(as measured from annotated PoS corpora)

P(w I s)	word	word	word	…
DT	0.3	0	0	
NN	0.0001	0.002	0	
VB	0	0	0.001	
…				

underflow danger ➡ use "log Ps"!

Transition Matrix



Observation Matrix

very sparse (W is large)

➡ Smoothing!

Three Problems Solved by HMMs

Evaluation: Given a HMM, **infer** the P of the observed sequence (because a HMM is a generative model).

in Bioinformatics: Likelihood of a particular DNA element

Solution: **Forward Algorithm**

Decoding: Given a HMM and an observed sequence, **predict** the hidden states that lead to this observation.

in Statistical NLP: PoS annotation

Solution: **Viterbi Algorithm**

Training: Given only the graphical model and an observation sequence, **learn** the best [smoothed] parameters.

Solution: **Baum-Welch Algorithm**

all three algorithms are are implemented using dynamic programming

Three Limitations of HMMs

CRF MEMM "unsolved"

Markov assumption: The next state only depends on the current state.

Example issue: trigrams *(long-range dependencies!)*

Output assumption: The output (observed value) is independent of all previous outputs (given the current state).

Example issue: word morphology *(inflection, declension!)*

Stationary assumption: Transition probabilities are independent of the actual time when they take place.

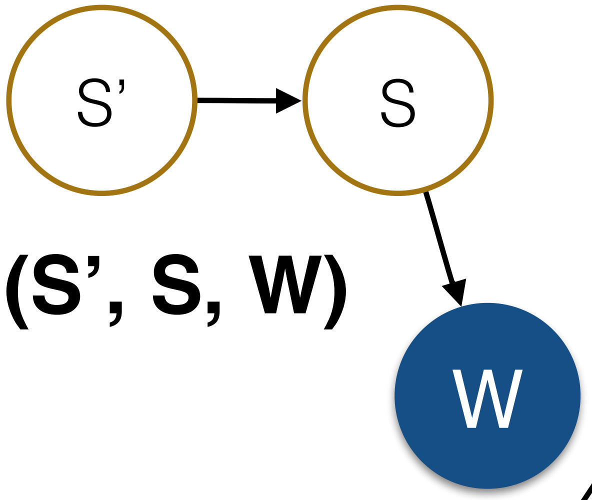
Example issue: position in sentence *(label bias* problem!)*

(CRF)

From Generative to Discriminative Markov Models

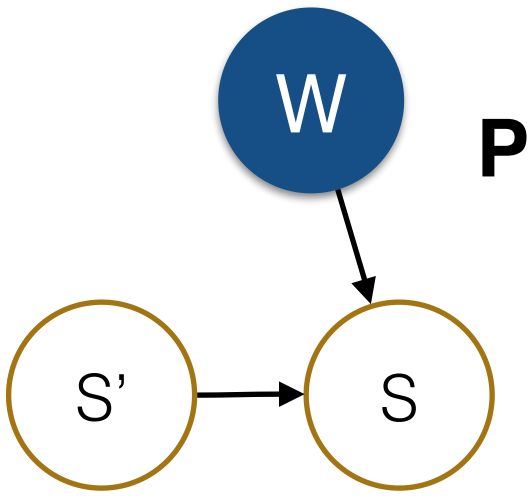
Hidden Markov Model

(first oder version)



P(S’, S, W)

generative model; lower bias is beneficial for small training sets



P(S I S’, W)

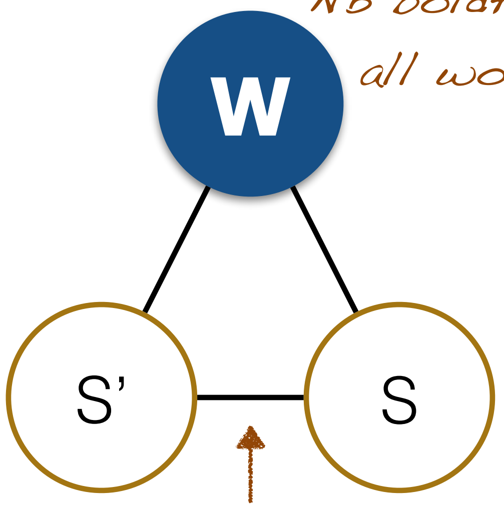
Maximum Entropy Markov Model

(first oder version)

Conditional Random Field

(linear chain version)

NB boldface **W**: all words!



this "clique" makes CRFs expensive to compute

Maximum Entropy (MaxEnt 3/3)

- In summary, MaxEnt is about selecting the “maximal” model p*:

$$p^* = \underset{p \in P}{argmax} \; - \sum_{x \in X} p(x) \; \sum_{y \in Y} p(y|x) \; log_2 \; p(y|x)$$

select some model that maximizes the conditional entropy...

- That obeys the following conditional equality constraint:

$$\sum_{x \in X} P(x) \; \sum_{y \in Y} P(y|x) \; f(x,y) = \sum_{x \in X, y \in Y} P(x,y) \; f(x,y)$$

...using a conditional model that matches the (observed) joint probabilities

REMINDER

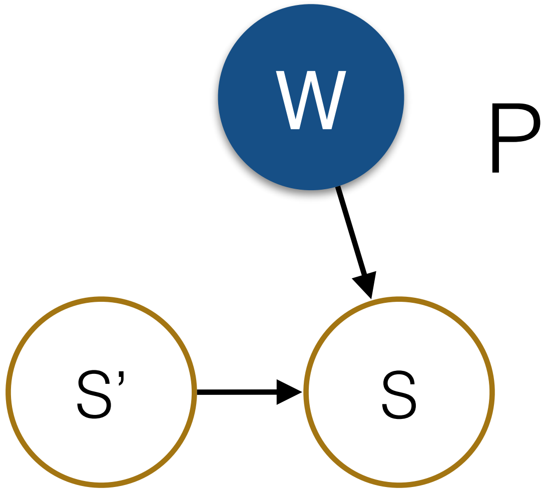
- Using, e.g., **Langrange multipliers**, one can establish the optimal λ weights of the model that maximize the entropy of this probability:

$$p^*(y|x) = \frac{exp(\sum \lambda_i f_i(x,y))}{\sum_{y \in Y} exp(\sum \lambda_i f_i(x,y))}$$

“ Exponential Model ”

Maximum Entropy Markov Models (MEMM)

Simplify training of P(s | s’, w) by splitting the model into |S| separate transition functions $P_{s'}(s \mid w)$ for each s’



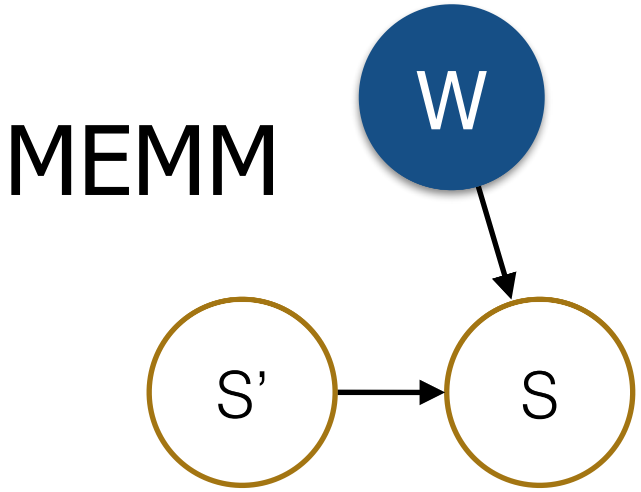
P(S | S’, W)

$$P(s \mid s', w) = P_{s'}(s \mid w)$$

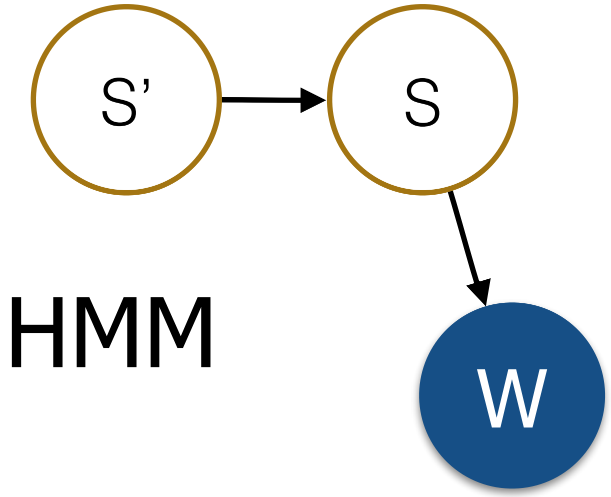
$$P_{s'}(s|w) = \frac{exp\left(\sum \lambda_i f_i(w,s)\right)}{\sum_{s^* \in S} exp\left(\sum \lambda_i f_i(w,s^*)\right)}$$

MaxEnt used {x, y} which in the sequence model now are {w, s}

The Label Bias Problem of Directional Markov Models



because of their directionality constraints, MEMMs & HMMs suffer from the label bias problem



The robot wheels Fred around.

DT NN **VB** NN RB

The robot wheels were broken.

DT NN **NN** VB JJ

The robot wheels are round.

DT NN **??** *yet unseen!*

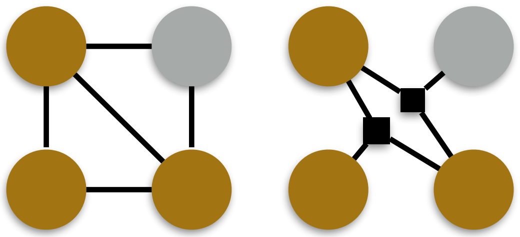
Wallach. Efficient Training of CRFs. MSc 2002

Markov Random Field

$$P(X = \vec{x}) = \frac{\prod_{cl \in \vec{x}} \overbrace{\phi_{cl}(cl)}^{\text{factor (clique potential)}}}{\underbrace{\sum_{\vec{x} \in X} \prod_{cl \in \vec{x}} \phi_{cl}(cl)}_{\text{normalizing constant (partition function – Z)}}}$$

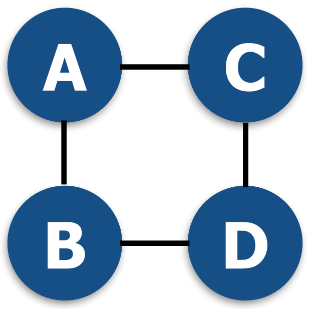
REMINDER

cl … [**maximal**] **clique**; a subset of nodes in the graph where every pair of nodes is connected



θ(A, B)			θ(B, C)			θ(C, D)			θ(D, A)		
a	b	30	b	c	100	c	d	1	d	a	100
a	b	5	b	c	1	c	d	100	d	a	1
a	b	1	b	c	1	c	d	100	d	a	1
a	b	10	b	c	100	c	d	1	d	a	100

factor table



factor graph

$P(a_1, b_1, c_0, d_1) =$
$10 \cdot 1 \cdot 100 \cdot 100 \div$
$7'201'840 =$
0.014

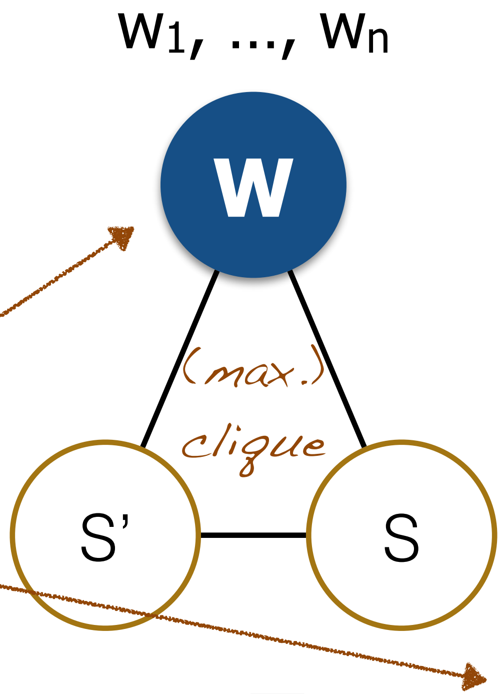
Conditional Random Field

MRF:

$$P(X=\vec{x}) = \frac{\prod_{cl \in \vec{x}} \phi_{cl}(cl)}{\sum_{\vec{x} \in X} \prod_{cl \in \vec{x}} \phi_{cl}(cl)}$$

CRF:

$$P(Y=\vec{y}\,|X=\vec{x}) = \frac{\prod_{y \in \vec{y}} \phi_{cl}(y', y, \vec{x})}{\sum_{\vec{y} \in Y} \prod_{y \in \vec{y}} \phi_{cl}(y', y, \vec{x})}$$



Label bias "solved" by conditioning the MRF Y-Y' on the (possibly) entire observed sequence (feature function dependent).

note W (upper-case/bold), not w (lower-case)

$$P(s|W) = \frac{exp\left(\sum \lambda_i f_i(W, s', s)\right)}{\sum_{s^* \in S} exp\left(\sum \lambda_i f_i(W, s'^*, s^*)\right)}$$

$$\text{"}\ \frac{P(s, s', W)}{P(W)}\ \text{"}$$

s' WENT MISSING

Models a per-state **exponential function** of joint probability over the **entire** observed **sequence** W.

Parameter Estimation and L2-Regularization of CRFs

(regularization reduces the effects of overfitting)

- For training $\{Y^{(n)}, X^{(n)}\}^{N}_{n=1}$ sequence pairs with K features
- Parameter estimation using **conditional log-likelihood**

$$\ell(\Lambda) = \underset{\lambda \in \Lambda}{argmax} \sum_{n}^{N} log\ P(Y^{(n)}|X^{(n)};\lambda)$$

- Substitute log $P(Y^{(n)} \mid X^{(n)})$ with log **exponential model**

$$\ell(\lambda) = \sum_{n=1}^{N} \sum_{y \in Y^{(n)}} \sum_{i=1}^{K} \lambda_i f_i(y', y, X^{(n)}) - \sum_{n=1}^{N} log\ Z(X^{(n)})$$

- Add a penalty for parameters with a to high **L2-norm**

$$\ell(\lambda) = \sum_{n=1}^{N} \sum_{y \in Y^{(n)}} \sum_{i=1}^{K} \lambda_i f_i(y', y, X^{(n)}) - \sum_{n=1}^{N} log\ Z(X^{(n)}) - \sum_{i=1}^{K} \frac{\lambda_i^2}{2\sigma^2}$$

free parameter

Model Summary: HMM, MEMM, CRF

- A **HMM**
  - generative model
  - **efficient** to learn and deploy
  - trains with **little data**
  - generalizes well (**low bias**)
- A **MEMM**
  - better labeling **performance**
  - modeling of **features**
  - **label bias** problem

- **CRF**
  - conditioned on **entire observation**
  - **complex features** over full input
  - training time **scales exponentially**
    - O(NTM^2G)
    - N: # of sequence pairs;
      T: E[sequence length];
      M: # of (hidden) states;
      G: # of gradient computations for parameter estimation

    a PoS model w/ 45 states and 1 M words can take a week to train...

Sutton & McCallum. An Introduction to CRFs for Relational Learning. 2006

Example Applications of Sequence Models

The Parts of Speech

I	ate	the	pizza	with	green	peppers	.
PRP	VB	DT	NN	IN	JJ	NN	.

"PoS tags"

REMINDER

- The Parts of Speech:
  - noun: NN, verb: VB, adjective: JJ, adverb: RB, preposition: IN, personal pronoun: PRP, …
  - e.g. the full **Penn Treebank PoS tagset** contains 48 tags:
  - 34 grammatical tags (i.e., "real" parts-of-speech) for words
  - one for cardinal numbers ("CD"; i.e., a series of digits)
  - 13 for [mathematical] "SYM" and currency "$" symbols, various types of punctuation, as well as for opening/closing parenthesis and quotes

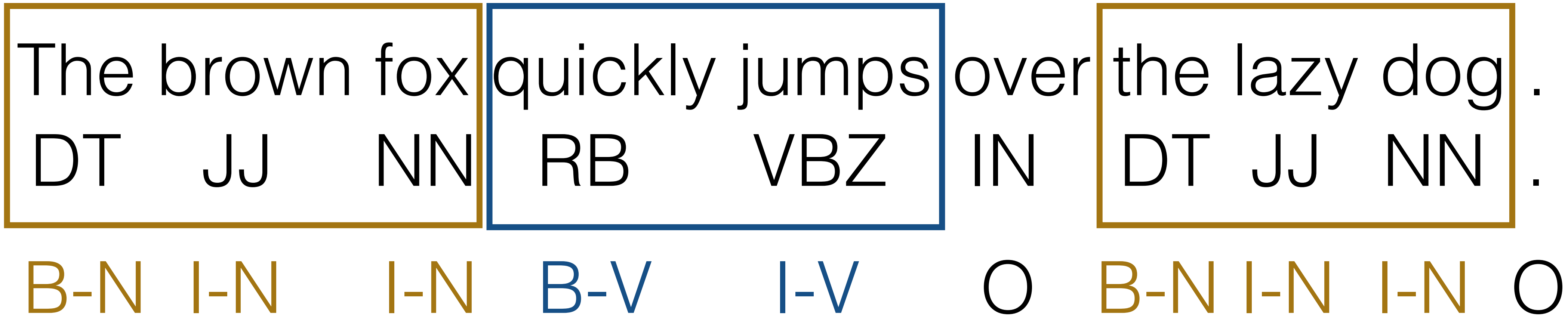
The Parts of Speech

I	ate	the	pizza	with	green	peppers	.
PRP	VB	DT	NN	IN	JJ	NN	.

- **Corpora** for the [supervised] **training** of PoS taggers
  - **Brown** Corpus (AE from ~1961)
  - British National Corpus: **BNC** (20th century British)
  - American National Corpus: **ANC** (AE from the 90s)
  - Lancaster Corpus of Mandarin Chinese: **LCMC** (Books in Mandarin)
  - The **GENIA** corpus (Biomedical abstracts from PubMed)
  - **NEGR@** (German Newswire from the 90s)
  - Spanish and Arabian corpora should be (commercially…) available… ???

Noun/Verb Phrase Chunking and BIO-Labels

a pangram (hint: check the letters)



The	brown	fox	quickly	jumps	over	the	lazy	dog	.
DT	JJ	NN	RB	VBZ	IN	DT	JJ	NN	.
B-N	I-N	I-N	B-V	I-V	O	B-N	I-N	I-N	O

Performance (2nd order CRF) ~ 94%
Main Problem: embedded & chained NPs (N of N and N)

Chunking is "more robust to the highly diverse corpus of text on the Web" and [exponentially] faster than parsing.

Banko et al. Open Information Extraction from the Web. IJCAI 2007 a paper with over 700 citations

Wermter et al. Recognizing noun phrases in biomedical text. SMBM 2005 error sources

Word Sense Disambiguation (WSD)

Note the PoS-tag "dependency": otherwise, the two examples would have even more senses!

- Basic Example: **hard** [JJ]
  - physically hard (a hard stone)
  - difficult [task] (a hard task)
  - strong/severe (a hard wind)
  - dispassionate [personality] (a hard bargainer)
- Entity Disambig.: **bank** [NN]
  - finance ("bank account")
  - terrain ("river bank")
  - aeronautics ("went into bank")
  - grouping ("a bank of …")

- **SensEval**
  - http://www.senseval.org/
  - SensEval/SemEval Challenges
  - provides corpora where every word is tagged with its sense
- **WordNet**
  - http://wordnet.princeton.edu/
  - a labeled graph of word senses
- **Applications**
  - Named Entity Recognition
  - Machine Translation
  - Language Understanding

Word Representations: Unsupervised WSD

- Idea 1: Words with similar meaning have similar **environments**.
- Use a word vector to count a word’s **surrounding words**.
- Similar words now will have similar word vectors.
  - see Text Mining 4, Cosine Similarity
  - visualization: Principal Component Analysis

Turian et al. Word representations. ACL 2010

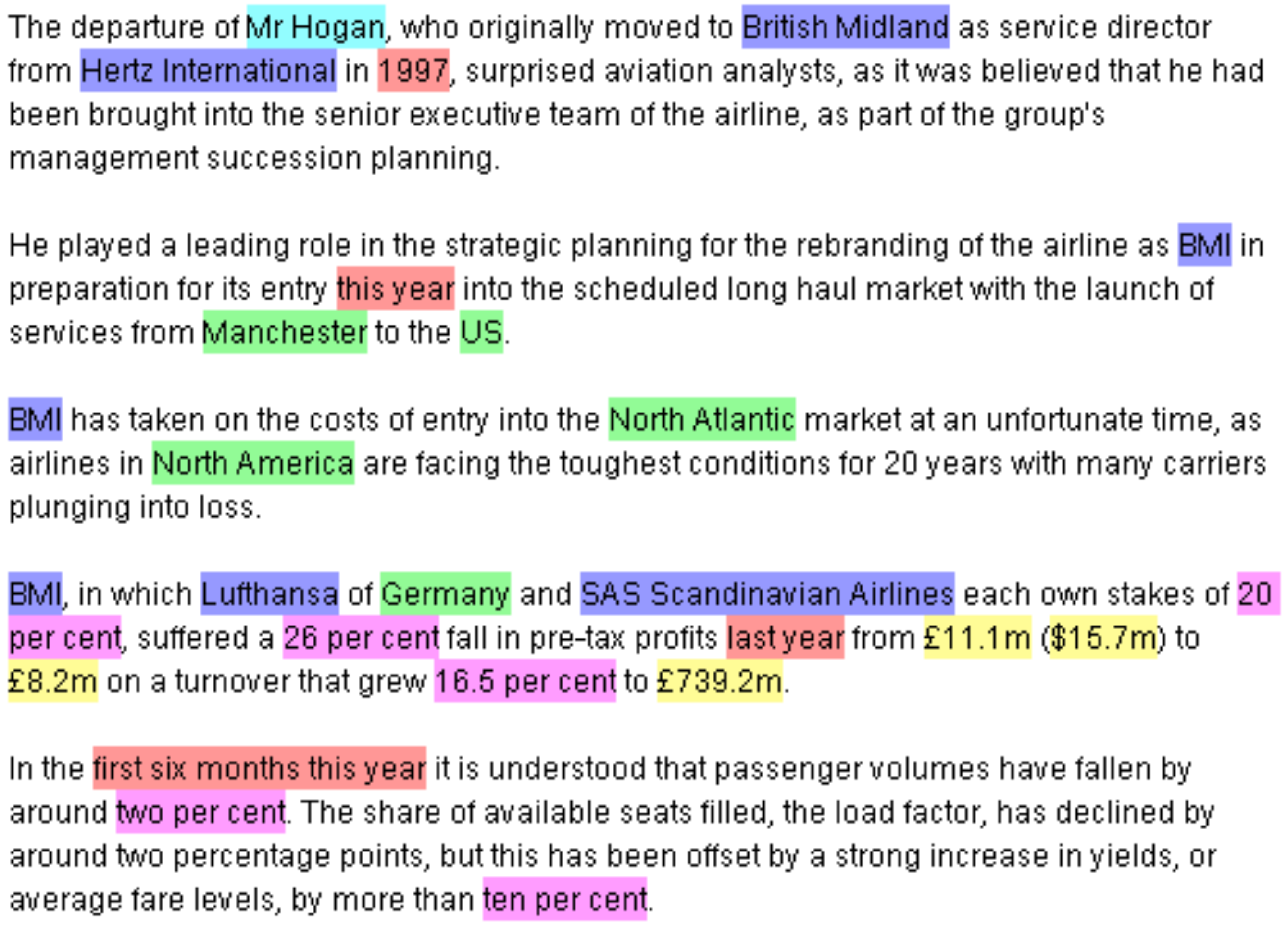
- Idea 2: Words with similar meaning have similar **environments**.
- Use the surrounding of **unseen words** to “smoothen” language models (i.e., the correlation between word w_i and its context c_j).
  - see Text Mining 4: TF-IDF weighting, Cosine similarity and point-wise MI
  - Levy & Goldberg. Linguistic Regularities in Sparse and Explicit Word Representations. CoNLL **2014**

Levy & Goldberg took two words on each side to “beat” a neural network model with a four word window!

Named Entity Recognition (NER)

How much training data do I need?
"corpora list"

Image Source: v@s3k [a GATE session; http://vas3k.ru/blog/354/]



The departure of Mr Hogan, who originally moved to British Midland as service director from Hertz International in 1997, surprised aviation analysts, as it was believed that he had been brought into the senior executive team of the airline, as part of the group's management succession planning.

He played a leading role in the strategic planning for the rebranding of the airline as BMI in preparation for its entry this year into the scheduled long haul market with the launch of services from Manchester to the US.

BMI has taken on the costs of entry into the North Atlantic market at an unfortunate time, as airlines in North America are facing the toughest conditions for 20 years with many carriers plunging into loss.

BMI, in which Lufthansa of Germany and SAS Scandinavian Airlines each own stakes of 20 per cent, suffered a 26 per cent fall in pre-tax profits last year from £11.1m ($15.7m) to £8.2m on a turnover that grew 16.5 per cent to £739.2m.

In the first six months this year it is understood that passenger volumes have fallen by around two per cent. The share of available seats filled, the load factor, has declined by around two percentage points, but this has been offset by a strong increase in yields, or average fare levels, by more than ten per cent.

Date
Location
Money
Organization
Percentage
Person

➡ Conditional Random Field

➡ Ensemble Methods; +SVM, HMM, MEMM, ... ➡ pyensemble

NB these are corpus-based approaches (supervised)

CoNLL03: http://www.cnts.ua.ac.be/conll2003/ner/

The Next Step: Relationship Extraction

Given this piece of text:

The fourth Wells account moving to another agency is the packaged paper-products division of Georgia-Pacific Corp., which arrived at Wells only last fall. Like Hertz and the History Channel, it is also leaving for an Omnicom-owned agency, the BBDO South unit of BBDO Worldwide. BBDO South **in** Atlanta, which handles corporate advertising for Georgia-Pacific, will assume additional duties for brands like Angel Soft toilet tissue and Sparkle paper towels, said Ken Haldin, a spokesman for Georgia-Pacific **in** Atlanta.

Which organizations operate in Atlanta? (BBDO S., G-P)

Four Approaches to Relationship Extraction

Co-mention window

- e.g.: if ORG and LOC entity within same sentence and no more than x tokens in between, treat the pair as a hit.
- Low precision, high recall; trivial, many false positives.

Dependency Parsing

- if the shortest a path containing certain nodes (e.g. the "in/IN") connecting the two entities, extract the pair.
- Balanced precision and recall, computationally EXPENSIVE.

Pattern Extraction

preposition (PoS)

- e.g.: <ORG>+ <IN> <LOC>+
- High precision, low recall; cumbersome, but very common
- pattern **learning** can help

Machine Learning

- features for sentences with entities and some classifier (e.g., SVM, MaxEnt, ANN, …)

token-distance, #tokens between the entities, tokens before/after them, etc.)

- very variable milages

but much more fun than anything else in the speaker's opinion…

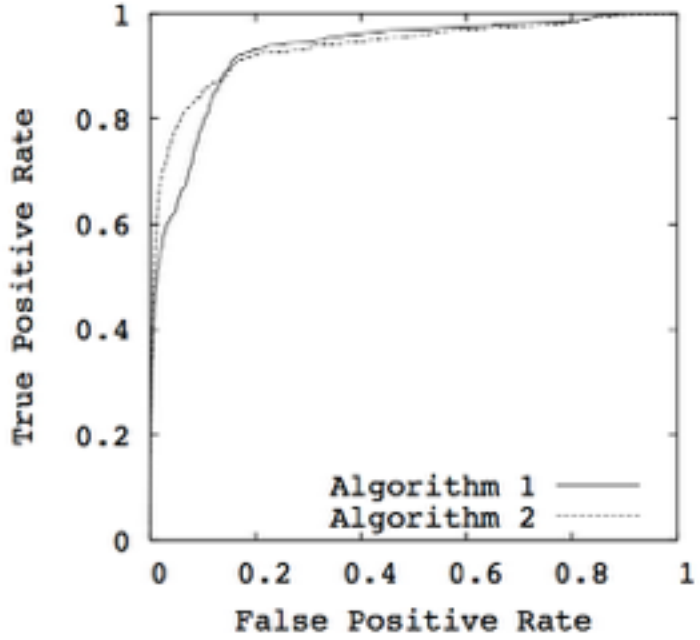
Advanced Evaluation Metrics: AUC ROC and PR

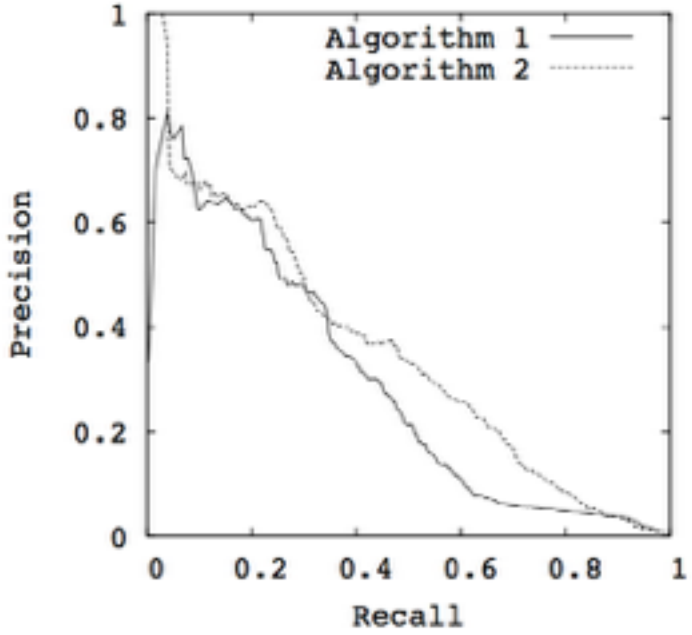
Area Under the Curve
Receiver-Operator Characteristic
Precision-Recall

TPR / Recall
TP ÷ (TP + FN)

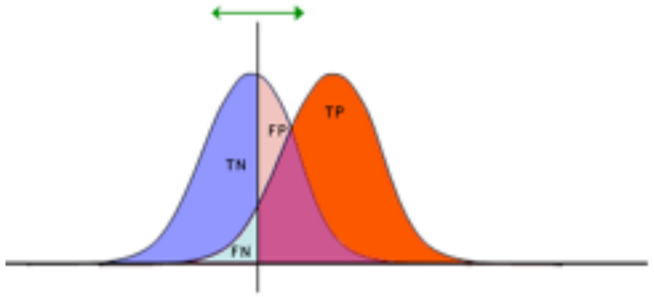
FPR
FP ÷ (FP + TN)

Precision
TP ÷ (TP + FP)





i.e., ROC is affected by the same class-imbalance issues as accuracy (Lesson #4)!



TP	FP
FN	TN
1	1

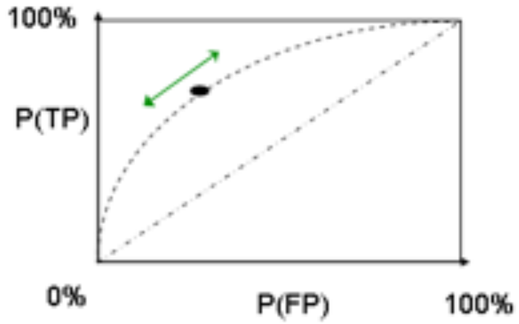


Image Source: WikiMedia Commons, kku (“kakau”, eddie)

“an algorithm which optimizes the area under the ROC curve is not guaranteed to optimize the area under the PR curve”

Davis & Goadrich, 2006

- Davis & Goadrich. The Relationship Between PR and ROC Curves. ICML 2006
- Landgrebe et al. Precision-recall operating characteristic (P-ROC) curves in imprecise environments. Pattern Recognition 2006
- Hanczar et al. Small-Sample Precision of ROC-Related Estimates. Bioinformatics 2010

Practical: Implement a Shallow Parser and NER

Implement a Python class and command-line tool that will allow you to do shallow parsing and NER on “standard” English text.

“The first principle is that you must not fool yourself
- and you are the easiest person to fool.”

Richard Feynman, 1974