http://www.youtube.com/watch?v=MvRTALJp8DM

http://www.youtube.com/watch?v=geqip_0Vjec

Sampling from Bayes Nets

Paper reviews

- Should be useful feedback for the authors
- A critique of the paper
- No paper is perfect!
  - if you don't understand it, state it
- Technically sound vs. convinced
- Give explicit examples, the more the better
  - cite sections, paragraphs, tables, figures, equations, etc.
- Make different sections clear
  - many conference reviews will have a similar format

Asking questions about distributions

- We want to be able to ask questions about these probability distributions
- Given *n* variables, a query splits the variables into three sets:
  - query variable(s)
  - known/evidence variables
  - unknown/hidden variables
- P(query | evidence)
  - if we had no hidden variables, we could just multiply all the values in the different CPTs
  - to answer this, we need to sum over the hiden variables!

Two approaches

- Enumeration
  - top-down, multiply probabilities and sum out the hidden variables

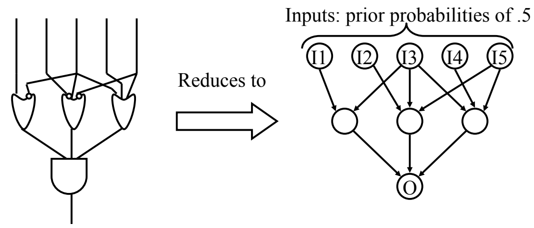
$$p(FO \mid hb,lo) = \alpha\, p(FO)\, p(lo \mid FO) \sum_{bp} p(bp) \sum_{do} p(do \mid FO,bp)\, p(hb \mid do)$$

- Variable elimination
  - avoids repeated work
  - bottom-up (right to left)
  - two operations: point-wise product of factors and summing out hidden variables

$$p(FO \mid hb,lo) = f_1(fo) f_2(lo,fo) \sum_{bp} f_3(bp) \sum_{do} f_4(do,fo,bp) f_5(hb,do)$$

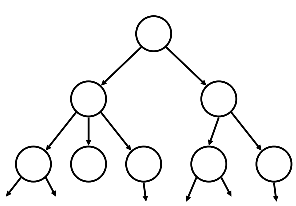
So is VE any better than Enumeration?

- Yes and No…
  - For singly-connected networks (poly-trees), YES
  - In general, NO
    - The problem is NP-Hard



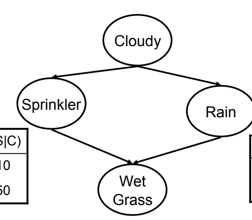
Bayesian Network Inference

- **But**…inference is still tractable in some cases.
- Special case: trees (each node has one parent)
- VE is LINEAR in this case



So, what about all those graphs with cycles?

P(C)
.50



C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

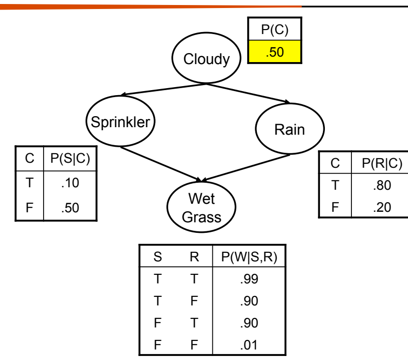
Approximate Inference!

Approximate Inference by Stochastic Simulation

- Recall when we wanted to find out the underlying distribution (of say a coin or die) we used sampling to estimate it

- Basic Idea:
  - Draw N samples from the distribution
  - Compute an approximate probability P
  - Eventually, for large samples sizes this converges to the true probability P

Sampling Basics: Sampling from an empty network



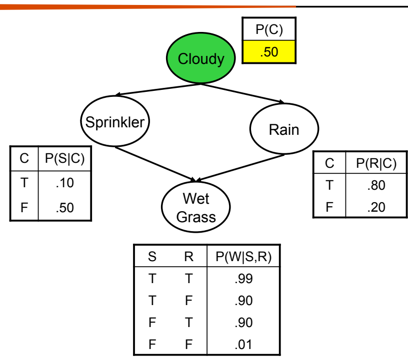
P(C)
.50

C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

Sampling Basics: Sampling from an empty network



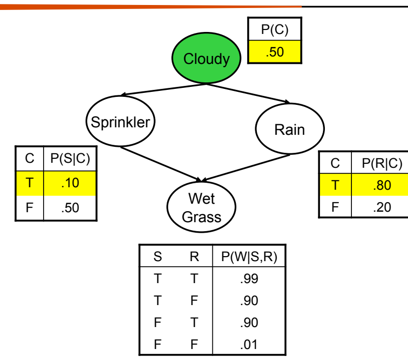
P(C)
.50

C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

Sampling Basics: Sampling from an empty network



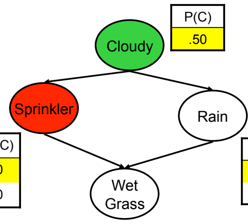
P(C)
.50

C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

Sampling Basics: Sampling from an empty network



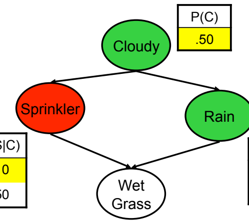
P(C)
.50

C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

Sampling Basics: Sampling from an empty network



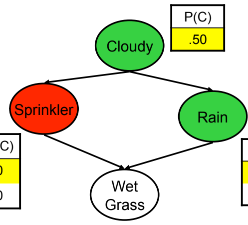
P(C)
.50

C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

Sampling Basics: Sampling from an empty network



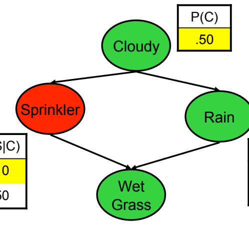
P(C)
.50

C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

Sampling Basics: Sampling from an empty network



P(C)
.50

C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

Sample: [T, F, T, T]

Calculating probabilities

- If we do this a number of times, then we can approximate answers to queries

[C, S, R, W]
[T, T, F, T]
[F, F, F, F]
[F, T, F, T]
[F, F, T, T]
[T, F, F, F]
[T, T, F, T]
[F, T, F, T]
[T, F, F, F]
[F, T, T, F]
[T, T, F, F]

What is the probability of rain?

Calculating probabilities

- If we do this a number of times, then we can approximate answers to queries

[C, S, R, W]
[T, T, F, T]
[F, F, F, F]
[F, T, F, T]
[F, F, T, T]
[T, F, F, F]
[T, T, F, T]
[F, T, F, T]
[T, F, F, F]
[F, T, T, F]
[T, T, F, F]

$$p(rain) = \frac{\text{num with rain}}{\text{total samples}} = \frac{2}{10} = 0.2$$

Rejection sampling

- What if we want to know the probability conditioned on some evidence?
  - p(rain | wet_grass)

[C, S, R, W]
[T, T, F, T]
[F, F, F, F]
[F, T, F, T]
[F, F, T, T]
[T, F, F, F]
[T, T, F, T]
[F, T, F, T]
[T, F, F, F]
[F, T, T, F]
[T, T, F, F]

Adding Evidence: Rejection Sampling

$\hat{P}(X \mid \mathbf{e})$ estimated from samples agreeing with e

E.g. Estimate P(R|s)
Samples ([C, S, R, W]):
[T, T, F, T]
[F, F, F, F]
[F, T, F, T]
[F, F, T, T]
[T, F, F, F]
[T, T, F, T]
[F, T, F, T]
[T, F, F, F]
[F, T, T, F]
[T, T, F, F]

$$p(rain \mid wet\_grass) = \frac{\text{num with rain and wet\_grass}}{\text{num with wet\_grass}}$$

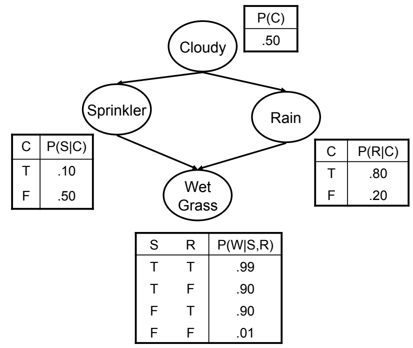
$$= \frac{1}{5} = 0.2$$

Problem with Rejection Sampling?

Likelihood weighting

- The problem with rejection sampling is that we may have to generate a lot of samples
  - low probability/rare events
  - large networks
- Likelihood weighting
  - rather than randomly sampling over all of the variables, only randomly pick values for the query variables and hidden variables
  - for those, the evidence variables weight the examples based on the likelihood of obtaining their fixed value

Likelihood weighting: p(rain | cloudy, wet_grass)



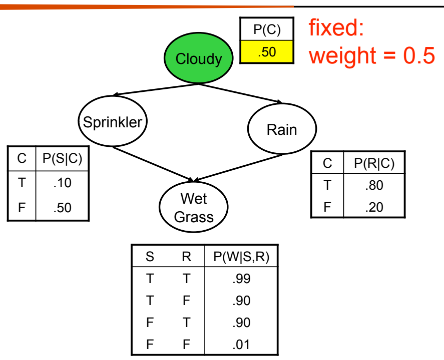
P(C)
.50

C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

Likelihood weighting: p(rain | cloudy, wet_grass)



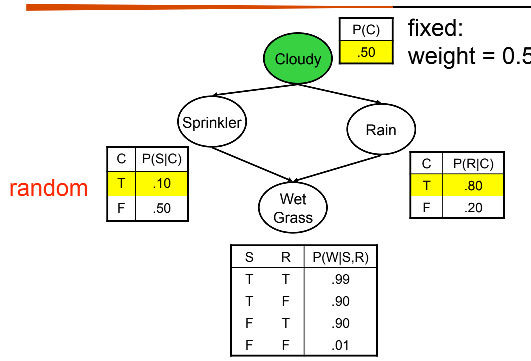
P(C)
.50

C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

Likelihood weighting: p(rain | cloudy, wet_grass)



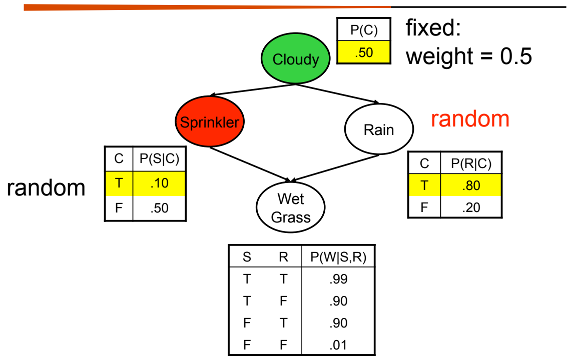
P(C)
.50

C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

Likelihood weighting: p(rain | cloudy, wet_grass)



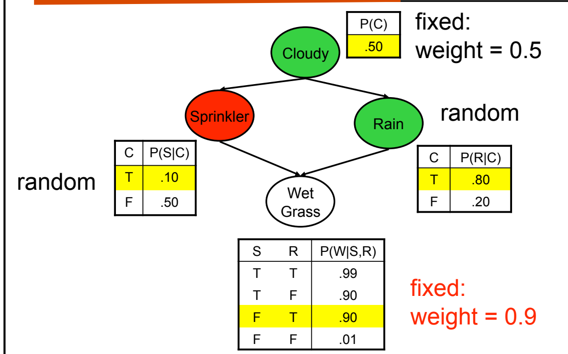
P(C)
.50

C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

Likelihood weighting: p(rain | cloudy, wet_grass)



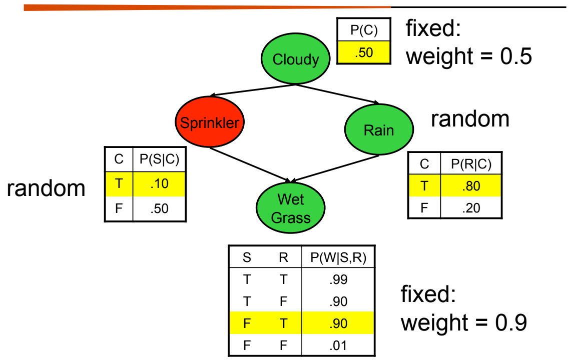
P(C)
.50

C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

Likelihood weighting: p(rain | cloudy, wet_grass)



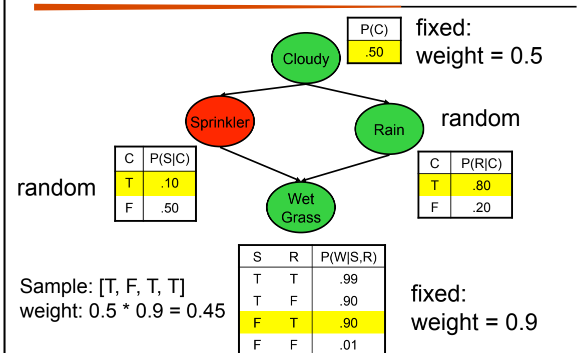
P(C)
.50

C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

Likelihood weighting: p(rain | cloudy, wet_grass)



P(C)
.50

C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

Likelihood weighting: p(rain | cloudy, wet_grass)

[C, S, R, W]	weight
[T, F, T, T]	0.45
[T, F, T, T]	0.45
[T, T, F, T]	0.45
[T, F, T, T]	0.45
[T, F, F, T]	0.005
[T, T, T, T]	0.495
[T, T, T, T]	0.495
[T, F, T, T]	0.45
[T, T, T, T]	0.495
[T, T, F, T]	0.45

$$= \frac{\text{weighted sum with rain,cloudy,wet\_grass}}{\text{weighted sum with cloudy,wet\_grass}}$$

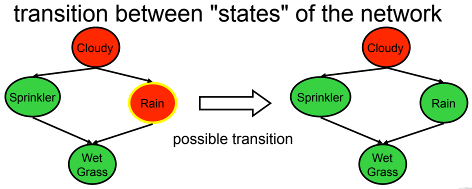
Problem with likelihood weighting?

Problems with likelihood weighting

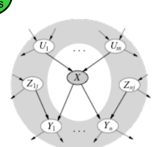
- As number of variables increased, weights will be very small
  - similar to rejection sampling, will only be a small number of higher probability ones that will actually effect the outcome
- If evidence variables are late in the ordering (BN), simulations will be not be influenced by evidence and so samples will not look much like reality

Approximate Inference using MCMC

- MCMC = Markov chain Monte Carlo
- Idea: Rather than generate individual samples, transition between "states" of the network



Choose one variable and sample it given its Markov Blanket

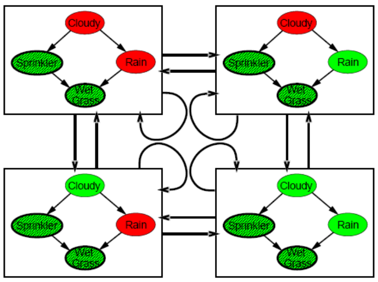


MCMC Sampling

- Start in some valid configuration of the variables
- Repeat the following steps:
  - pick a non-evidence variable
  - randomly sample given its markov blanket
  - count this new state as a sample
- If the process visits 20 states where Rain is true and 60 states where Rain is false,
  - Then the answer to the query is <20/80, 60/80> = <0.25, 0.75>

MCMC

If you know Sprinkler=T and Wet Grass=T, there are 4 network states



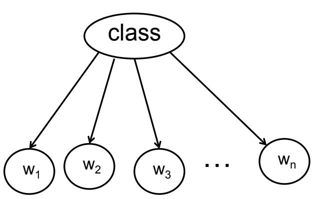
Wander for awhile, average what you see

Document classification

- Naïve Bayes classifier works surprisingly well for its simplicity
- We can do better!

(Big Boy models)

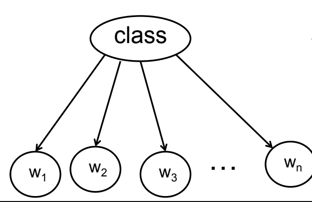
Revisiting the Naïve Bayes model



n words in our vocabulary

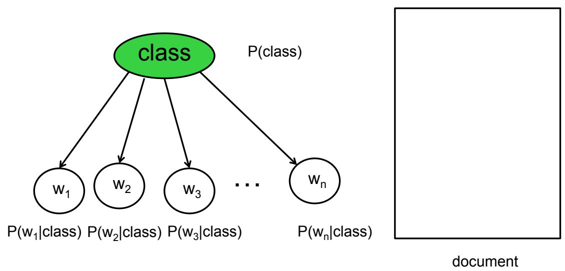
"Generating" a document

- The *generative story* of a model describes how the model would generate a sample (document)
- It can help understand the independences and how the model works
- As before, we can generate a random sample from the BN

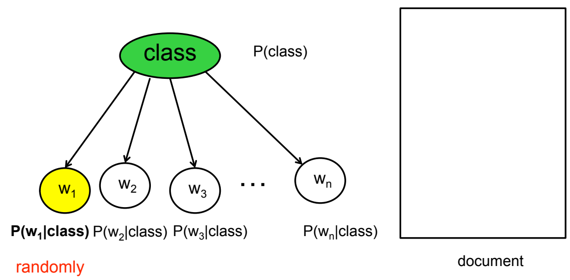


How does that work for Naïve Bayes? How would we generate a positive document?

Sampling

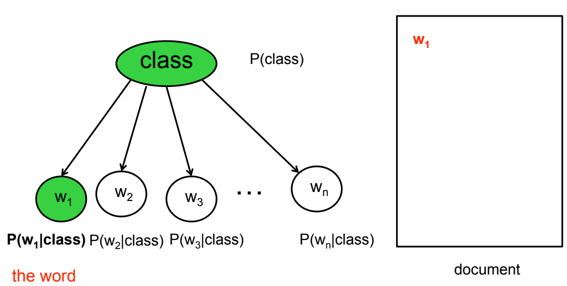


Sampling



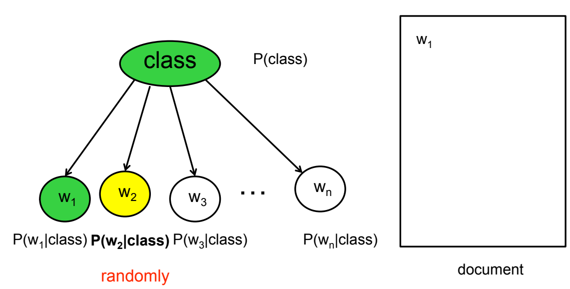
randomly
sample

Sampling



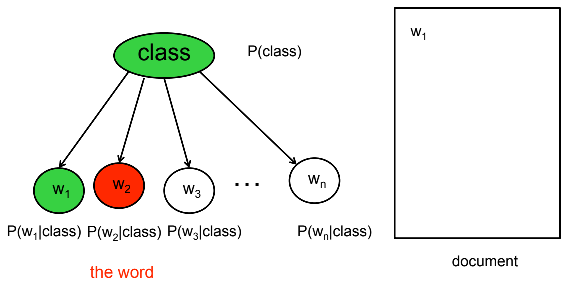
the word
occurs in the
document

Sampling



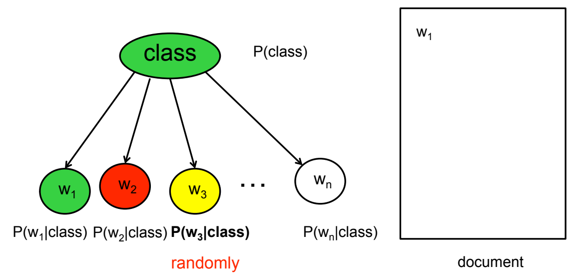
randomly
sample

Sampling



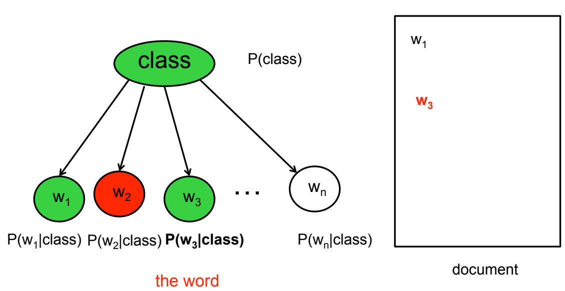
the word
does NOT
occur in the
document

Sampling



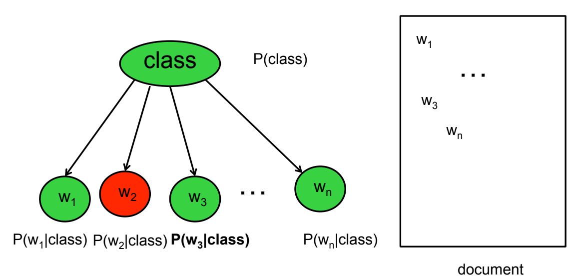
randomly
sample

Sampling



the word
occurs in the
document

Sampling



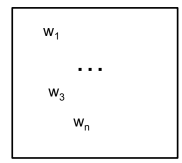
How are we simplifying the data?

Bag of words representation

- Notice that there is no ordering in the model
  - "I ate a banana" is viewed as the same as "ate I banana a"
- Called the "bag of words" representation

NB model

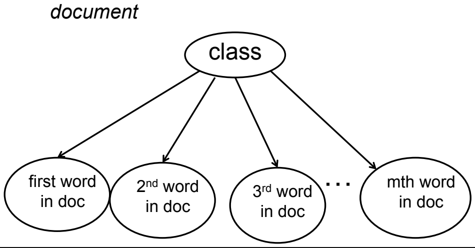
- A word either occurs or doesn't occur
  - no frequency information
- Word occurrences are independent, given the class
  - when we sample, the only thing we condition on is the class



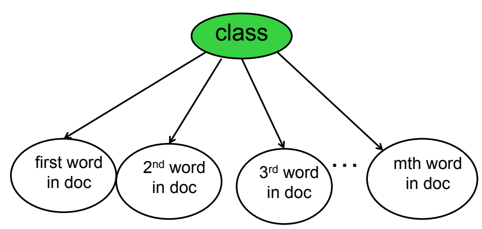
document

Incorporating frequency

- Multinomial model:
  - rather than picking whether or not a word occurs, pick **what** each word in the document will be
  - now rather than having boolean random variables, our random variables space is the *number of words in the document*

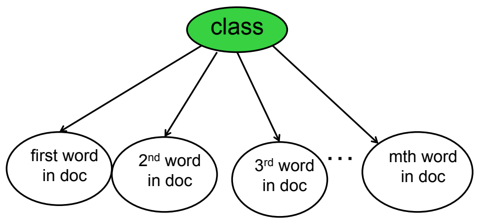


Sampling



What will the conditional probability tables look like?

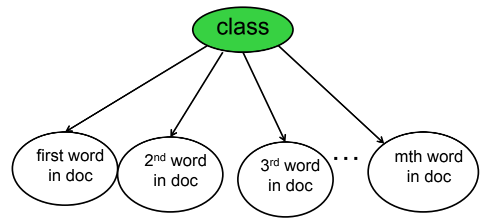
Sampling



class word p(word|class)
A w_1
A w_2
A w_3
A w_4
…
B w_1
…

Each position in the document has a distribution over all words and each class

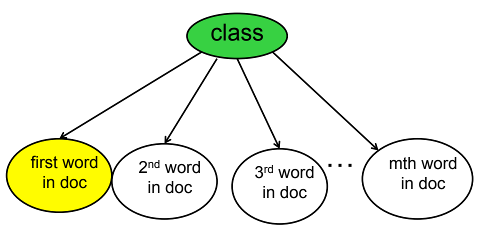
Sampling



class word p(word|class)
A w_1
A w_2
A w_3
A w_4
…
B w_1
…

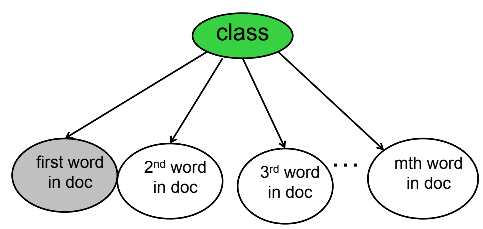
In practice, we use the same distribution for all word positions!

Sampling



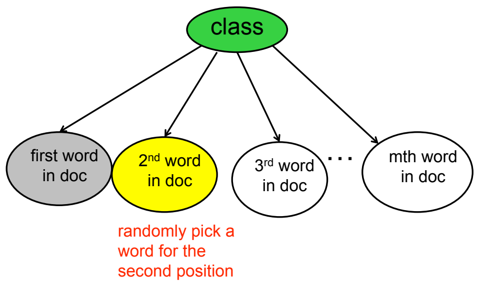
randomly pick a word for the first position

Sampling



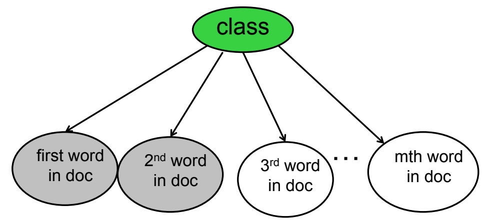
The

Sampling



The

Sampling



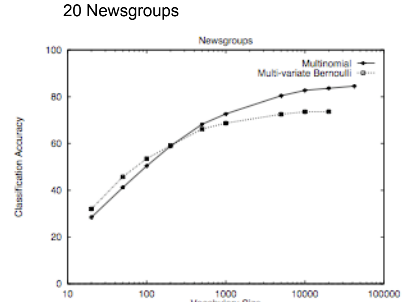
The a …

Multinomial model

- Called a multinomial model because the word frequencies drawn for a document of length m, follow a multinomial distribution
  - sampling with replacement from a fixed distribution
- Word occurrences are still independent!
  - doesn't matter what other words we've drawn
- Although technically the position is specified, doesn't really give us positional information
- Still a naïve Bayes model!

Boolean NB vs. Multinomial NB

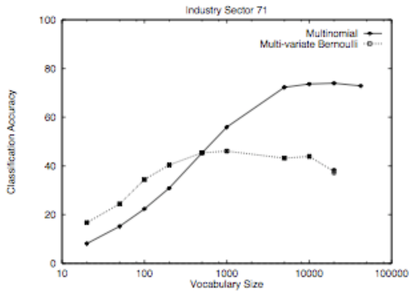
20 Newsgroups



http://www.cs.cmu.edu/~knigam/papers/multinomial-aaaiws98.pdf

Boolean NB vs. Multinomial NB

Industry Sector data (71 classes, web pages)



http://www.cs.cmu.edu/~knigam/papers/multinomial-aaaiws98.pdf

Plate notation

- It can be tedious to write out all of the children in a BN
- When they're all the same type, we can use "plate" notation
  - A plate represents a set of variables
  - We specify repetition by putting a number in the lower right corner
  - edges crossing plate boundaries are considered to be multiple edges

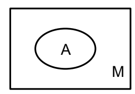
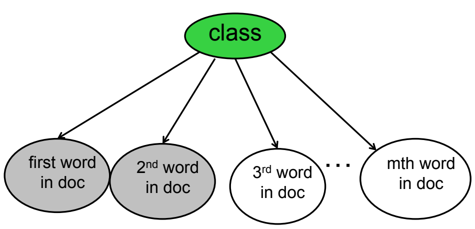
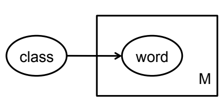
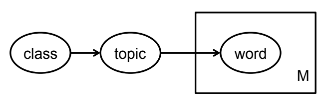


Plate notation





Dirichlet Compound Multinomial (DCM)



- To generate a document
  - pick a class
  - based on that class, draw a multinomial representing a topic
    - p(topic | class) is represented by a Dirichlet distribution
    - Gives us a distribution *over* multinomials
  - Based on this multinomial, sample as before

DCM

- Key problem with NB multinomial: words tend to be "bursty"
  - if a word occurs once, it's likely to occur again
  - particularly content words, e.g. Bush
- DCM model allows us to model burstiness by picking multinomials for a given document that have a higher probability of ocurring

For those that like math ☺

$$p(\mathbf{x} \mid \alpha) = \int_\theta \frac{|\mathbf{x}|!}{\prod_{w=1}^{W} x_w!} \left(\prod_{w=1}^{W} \theta_w^{\,x_w} \right) \frac{\Gamma\left(\sum_{w=1}^{W} \alpha_w\right)}{\prod_{w=1}^{W} \Gamma(\alpha_w)} \prod_{w=1}^{W} \theta_w^{\,\alpha_w - 1} d\theta$$

$$= \frac{|\mathbf{x}|!}{\prod_{w=1}^{W} x_w!} \frac{\Gamma\left(\sum_{w=1}^{W} \alpha_w\right)}{\prod_{w=1}^{W} \Gamma(\alpha_w)} \int_\theta \prod_{w=1}^{W} \theta_w^{\,\alpha_w + x_w - 1} d\theta$$

$$= \frac{|\mathbf{x}|!}{\prod_{w=1}^{W} x_w!} \frac{\Gamma\left(\sum_{w=1}^{W} \alpha_w\right)}{\prod_{w=1}^{W} \Gamma(\alpha_w)} \prod_{w=1}^{W} \frac{\Gamma(x_w + \alpha_w)}{\Gamma(\alpha_w)}$$

DCM vs. Multinomial

	Industry	20 Newsgroups
Multinomial	0.600	0.853
DCM	**0.806**	**0.890**

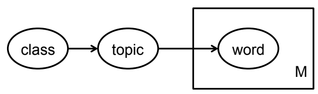
http://www.cs.pomona.edu/~dkauchak/papers/kauchak05modeling.pdf

Topic models

- Often a document isn't just about one idea/topic
- Topic models view documents as a blend of "topics"

topic 1 +
topic 2 +
topic 3

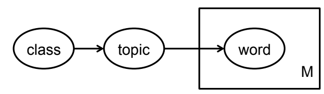
Topic models



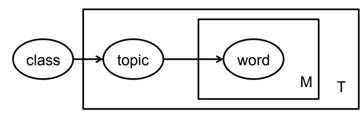
DCM model

How might we model this as a Bayes net?

Topic models

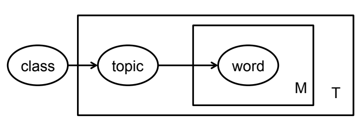


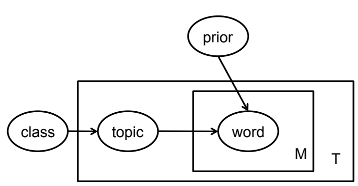
DCM model



LDA model (Latent Dirichlet Allocation)

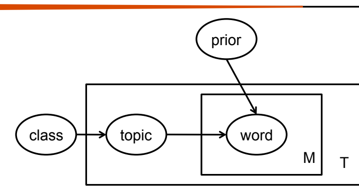
LDA





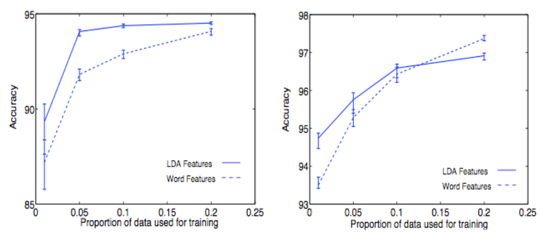
LDA model (Latent Dirichlet Allocation)

LDA



- To generate a document
  - for each word in the document:
    - pick a topic given the class
    - pick a word given the topic (and the prior)
- Key paper:
  - http://www.cs.princeton.edu/~blei/papers/BleiNgJordan2003.pdf

LDA



- Two binary tasks from industry sector
- Used LDA to extract features
- Then used SVMs (support vector machines) for classification

Document classification

- "Generative models"
  - represent underlying probability distribution
  - can be used for classification, but also other tasks
  - models:
    - Bernoulli (boolean) naïve Bayes
    - Multinomial naïve Bayes
    - Dirichlet Compound Multinomial
    - Latent Dirichlet Allocation

  more expensive to train

- Discriminative models
  - support vector machines
  - markov random fields

  very good for classification only

Midterm

- Open book
  - still only 75 min, so don't rely on it too much
- Anything we've talked about in class or read about is fair game
- Written questions are a good place to start

Review

- Intro to AI
  - what is AI
  - goals
  - challenges
  - problem areas

Review

- Uninformed search
  - reasoning through search
  - agent paradigm (sensors, actuators, environment, etc.)
  - setting up problems as search
    - state space (starting state, next state function, goal state)
    - actions
    - costs
  - problem characteristics
    - observability
    - determinism
    - known/unknown state space
  - techniques
    - BFS
    - DFS
    - uniform cost search
    - depth limited search
    - Iterative deepening

Review

- Uninformed search cont.
  - things to know about search algorithms
    - time
    - space
    - completeness
    - optimality
    - when to use them
  - graph search vs. tree search
- Informed search
  - heuristic function
    - admissibility
    - combining functions
    - dominance
  - methods
    - greedy best-first search
    - A*

Review

- Adversarial search
  - game playing through search
    - ply
    - depth
    - branching factor
    - state space sizes
    - optimal play
  - game characteristics
    - observability
    - # of players
    - discrete vs. continuous
    - real-time vs. turn-based
    - determinism

Review

- Adversarial search cont
  - minimax algorithm
  - alpha-beta pruning
    - optimality, etc.
  - evalution functions (heuristics)
    - horizon effect
  - improvements
    - transposition table
    - history/end-game tables
  - dealing with chance/non-determinism
    - expected minimax
  - dealing with partially observable games

Review

- Local search
  - when to use/what types of problems
  - general formulation
  - hill-climbing
    - greedy
    - random restarts
    - randomness
    - simulated annealing
    - local beam search
    - taboo list
  - genetic algorithms

Review

- CSPs
  - problem formulation
    - variables
    - domain
    - constraints
  - why CSPs? applications?
  - constraint graph
  - CSP as search
    - backtracking algorithm
    - forward checking
    - arc consistency
  - heuristics
    - most constrained variable
    - least constrained value
    - ...

Review

- Basic probability
  - why probability (vs. say logic)?
  - vocabulary
    - experiment
    - sample
    - event
    - random variable
    - probability distribution
  - unconditional/prior probability
  - joint distribution
  - conditional probability
  - Bayes rule
  - estimating probabilities

Review

- Bayes nets
  - representation
  - dependencies/independencies
    - d-separation
    - Markov blanket
  - reasoning/querying
    - exact:
      - enumeration
      - variable elimination
    - sampling
      - basic
      - variable elimination
      - MCMC

Review

- Bayesian classification
  - problem formulation, argmax, etc.
  - NB model
  - Other models
    - multinomial, DCM, LDA
  - training, testing, evaluation
  - plate notation