check-~~prep~~ appt scheduling . . .

11/18/14
lec 24

~~Recall our question:~~ how do you tell if two "things" ~~strings~~ are different ?

What is ~~language~~ are (statistical) language models for ? For the purposes of this class :

→ giving a ~~stati~~ compact representat'n of what the language is 'like' .

→ comparing two language sources . (ex: no country) .

~~We define~~ A language model is defined as a probability distribution over all strings w that we wish to consider "legal".

$\vec{\theta}$

~~some distributions aren't model-based per se, of Bo~~

[a good counterexample to keep in mind : the Poisson over strings of form "dude" : ~~Is~~ - finite # of params, no equivalent PCFG. (see Booth & Thompson '73)

ex: ~~or~~ HMMs (~~for now, no a let's~~)

- a finite non-empty set of states $q_1, \ldots, q_M$, q_b , q_e

distinguished ~~start~~ "begin" distinguished "end"

- ~~each~~ a finite vocabulary V ∪ {<~~start~~ begin>, <end>} not in V .

- each state ~~defn~~ has an associated emission distribution over V^* (all sequences ~~strings~~ of items from V, including the empty string)

this might be itself a (state-specific) L.M.

q_b emits only <begin>

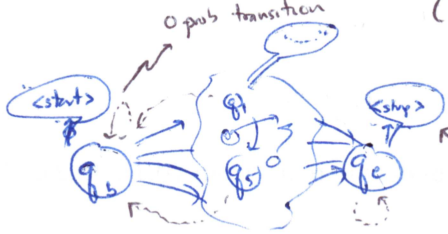
q_e " " <end>

q_i does not emit ~~e.g.~~ any sequence containing those special items

~~(conveniently kept in a matrix, all rows sum to 1)~~

- each state has a transition distribution over all other states , s.t.: (conveniently kept in a matrix , all rows summing to 1)

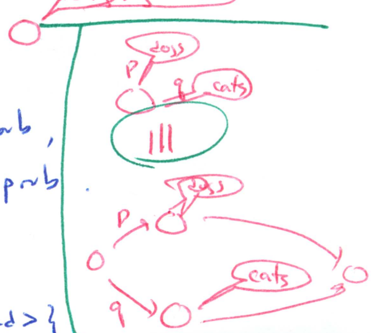
can ~~but bad idea if~~ altho' maybe a sparse rep would be better.



speech balloons

draw 1st
- drawing really encapsulated a lot

really big rats



will further assume every q_i reachable from q_b w/ non-zero prob ,

" q_i can reach q_e w/ non-zero prob .

→ these definitely seem like necessary

~~p(s) = for s ∈ {<begin>} V* {<end>} =~~

p(s) = sum of probs of ~~all~~ all "paths" generating s , s ∈ {<begin>} V^* {<end>}

↳ product of all the transition probs × prob of corresponding emissions (so you can think of the emissions as also ~~defin~~ a part of what defines a path .

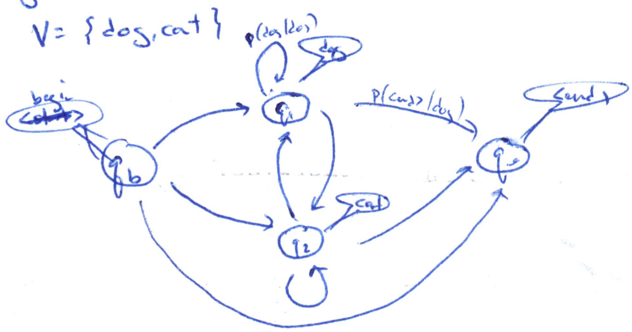
nb: should make clear diff. btwn termination ~~and properness (i.e., that~~ and the l.m. itself being proper .

what does this tell you?

- we have ~~broken~~ a distribution over ~~an~~ a potentially ∞ # of sequences to sthg definable by a finite set of parameters
  (indeed, you have a model!), so you're managing the complexity.
  - structure of this model gives you insight into what is "preferred".
- ~~store~~
  - $\exists$ learning algorithms for HMMs, either from labelled or unlabelled data, (altho' you have to set the # of states beforehand).
    need ~~to~~ specify M ahead of time.
  - ~~learnable~~ two training-data paradigms: { state-labeled data | unlabeled data }
    Baum-Welch/EM for max-likelihood estimation (tends to find saddle points or local maxima, b/c there a lot
    - REMEMBER: need constraints (o.w., just set all params to ∞!))
    - why are n-gram models so much more common than HMMs?
    - here, the data basically has state labels! Nothing is really "hidden".

ex: a "bigram lm" as an HMM:

$V = \{dog, cat\}$

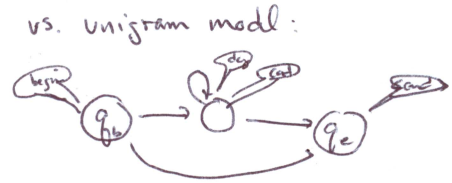


q_i = "just said v_i"

p(<start> dog dog <stop>)

P($\nearrow$) $\cdot$ 1 $\cdot$ P($\circlearrowleft$) ....

vs. unigram model:



- ~~tricky dependence of what is to be next generated~~
- ~~we can build more co~~
- ~~but it's not just finite state~~ .. we aren't just restricted to finite-state models

ex: probabilistic context-free grammars (PCFGs) — analogous defns for dependency grammars.
aka stochastic CFGs (SCFGs)

- each category type has a distribution over its decompositions, a finite set:

ex: P(Sent → NP VP)   P(Sent → Aux NP S-question)

declarative sentence   ~~que~~ question.

- ~~ideal~~.

$p_\theta(s)$ = sum of probs of ~~all~~ each trees generating s.
↳ product of each decomposition used in a tree

P(Sent → NP VP, NP → they, VP → left) = P(Sent → NP VP) P(NP → they) P(VP → left)

q: ~~does this~~ does every choice of valid 'decomposition distribution' induce a proper prob on all strings?
Turns out it's tricky.

- [Booth & Thompson '73] conditions under which you would get a proper distribution over all strings.

MS comments re: PCFGs.

[Chi & Geman '98]

ex: Sent → Sent Sent, prob p

Sent → ~~fish~~ police, prob $q = 1-p$.

For this $\theta = (p)$ <that's all we need>

$P_\theta(\text{"police"}) = q$

$P_\theta(\text{"police police"}) = p \cdot q \cdot q = pq^2$

$P_\theta(\text{"police police police"}) = [p \cdot p \cdot q^3] + [p \cdot q \cdot p \cdot q^2] = 2p^2q^3$

$\sum_s P_\theta(s) = \sum_{i=1}^{\infty} [\#\text{ of 'bracketings' of length-}i\text{ sequence}] \cdot q^i \cdot p^{i-1}$

q^i: generate i leaves

p^{i-1}: get i "parents" of those leaves.

$= \sum_{i=1}^{\infty} (\#\text{-brackets}(i)) \cdot q\,(qp)^{i-1}$

Catalan #:

$\sum_{i=1}^{\infty} \frac{1}{i}\binom{2(i-1)}{i-1} q\,(qp)^{i-1}$

<presumably thru some geometric sum>?

$\doteq q \sum_{i=1}^{\infty} \frac{1}{i} \frac{4^{(i-1)}i^{-2}}{\sqrt{\pi(i-1)}} (qp)^{i-1}$

<abandon this>

Let X_d = prob of all trees w/ ~~depth~~ ~~height~~ depth $\le d$ & fringe contains only words

$X_1 = q$

$X_2 = q + pq^2$

$X_3 = q + pq^2 + p(q+pq)^2(q+pq)^2$

$X_i = q + pX_{i-1}^2$

apparently converges to $\min(1, q/p)$

if ~~$p > 1/2$~~

if $p > 1/2$, this will not be proper (too much probability on the "expansion" as opposed to the "generation of words".

~~not~~

ex: content model: ~~learn~~ [Barzilay & Lee '04]. — code available (also Alexandre Passos's) — etc. state

→ (learn) how topics relate to each other w/in docs.

→ potentially relevant to projects: <probably have fixed set of topics>

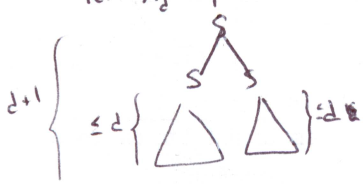
(etc. state)

talked about: LM on just top-most frequent words { stylistic; what do you do w/ the 'other' words to make them not dominate?

entropy → fixed-max-length strings.

2nd attempt, using Chi & Geman's approach:

let X_d = prob of all trees w/ depth $\le d$ (& fringe contains only words)



$\rightarrow p \cdot X_d \cdot X_d$ and then S → police $= q$.

$$X_{d+1} = q + pX_d^2.$$

~~given that $X_{d+1} \ge X_d \ge 0$ ... $\le q + pX_{d+1}^2$~~

~~$X_{d+1} - pX_{d+1}^2 \le q$~~

~~$X_{d+1}(1 - pX_{d+1}) \le q$~~

If we assume convergence, then we solve for $X = q + pX^2$, or $0 = pX^2 - X + q$.

~~By quadratic formula, roots are:~~

$\Rightarrow 0 = (px - (1-p))(x-1)$

$= (px - q)(x-1)$

solutions are $x = q/p$ or $x = 1$.

(thanks to Chenhao Tan for idea of assuming convergence)