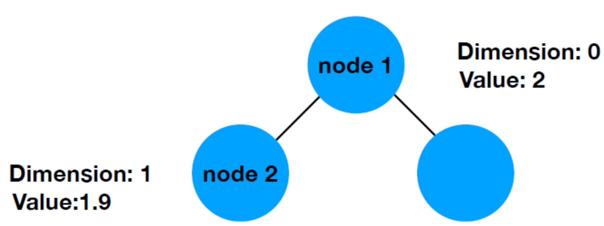
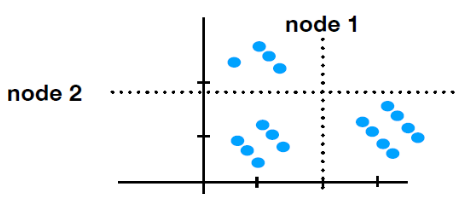
**String similarity**

- Why useful?
  - deal with out of vocab
  - correcting user input mistakes
  - propose alternative but similar possibilities
- Jaccard similarity
  - # similar letters / # total letters
- Jaccard distance
  - 1 - Jaccard similarity
  - not ideal for stings not ordered
    - Panmpi similar to mapping?
  - easy and fast to compute
- Edit distance
  - not fast (recursive)
  - can use BK-trees to reduce search space
  - Perhaps more intuitive for NLP

$$C(i,j) = \begin{cases} C(i-1,j-1) & \textbf{if } a_j = b_i \\ min(ins_{i,j}, del_{i,j}, sub_{i,j}) & \textbf{if } a_j \neq b_i \end{cases}$$

  - [Fill in edit distance table]()
- K nearest neighbors
  - Don't need to calculate distance for every document in corpus?**
    - Unlike nearest neighbor?
- Euclidean distance/weighted distance
- Cosine similarity
  - Choose whether to normalize for length of document
    - Ask yourself: is document length relevant to the problem?
- KD tree




  - Split along widest dimension
  - Split at value = median of the column
  - Stop when less than M points in node

Often wise to use different distances for different features

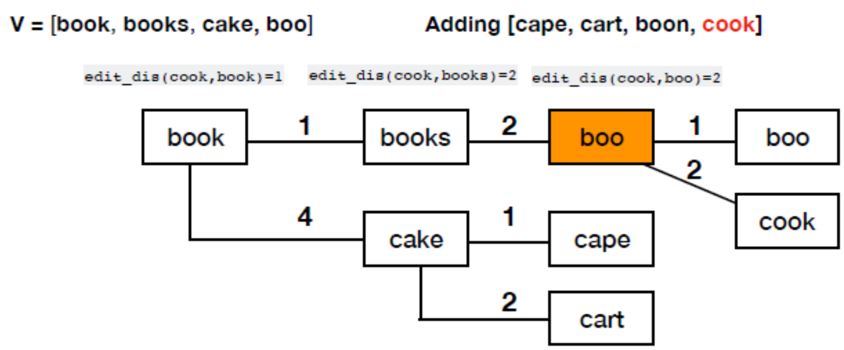
**BK trees**

Useful to minimize search space when searching for most similar word (for out of vocab word)

Allows efficient use of a single core

Don't need to search through all N words to find most similar word

Select any word to be root node, add all remaining words




**Vector representations for documents**

word_to_vec > bag of words

word_to_vec: ordered dictionary; keys are words and values are indices in the feature count vector (bag of words, for example)

bag of words can be a count or a boolean (indicating presence/absence of word)

TFIDF

Emphasize most relevant words in the documents

assignment "importance" or weight to each word

unique words (idf, in corpus) and more frequent (tf, in a document) are weighted more

TFIDF = TF * idf

TF: term frequency (document level)

IDF: inverse document frequency (corpus level)

TF and IDF are both vectors

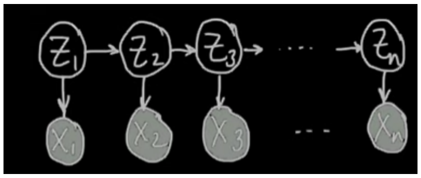
| | |
|---|---|
| raw count | $f_{t,d}$ |
| term frequency | $f_{t,d} \Big/ \sum_{t' \in d} f_{t',d}$ |

$$\text{idf}(w; X) = log\left(\frac{|X|}{1 + |X_w|}\right)$$

**Hidden Markov models**

"Go to"/baseline model for sequential models: easy to compute

Trellis diagram:




Parameters: initial, transition, and emission probabilities

Initial probabilities

- p(hidden_state_1 = z)
- Vector of length M (M is # of hidden states)
- Example: init_tag:det

Transition probabilities

- p(hidden_state_1 to hidden_state_2)
- M x M stochastic matrix
- Example: prev_tag:det::adj

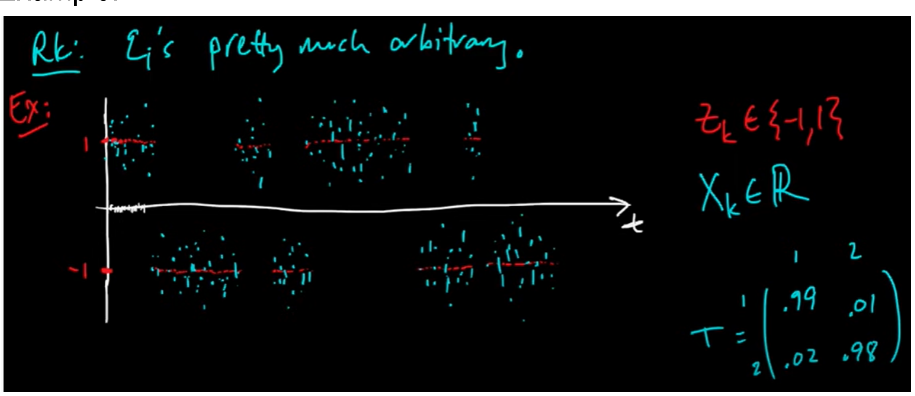
Emission probability

- p(observed_state_i given hidden_state_i)
- Vector of length M (for each observed value in sequence)
- Example: id:The::det

Joint probability:

$$p(x_1,\dots,x_n,z_1,\dots,z_n) = p(z_1)p(x_1|z_1)\prod_{k=2}^{n} p(z_k|z_{k-1})p(x_k|z_k)$$

Example:




Forward-Backward Algorithm

- Textbook example of dynamic programming
- Goal: compute p(z_k given all x)

Notation:

$\underline{F/B}$: Compute $p(z_k|x)$.

$\underline{F \text{ alg}}$: Compute $p(z_k, x_{1:k})$ $\forall k=1,\dots,n$.

$\underline{B \text{ alg}}$: Compute $p(x_{k+1:n}|z_k)$ $\forall k=1,\dots,n$.

$$p(z_k|x) \underset{z_k}{\propto} p(z_k,x) = \overbrace{p(x_{k+1:n}|z_k)}^{B}\,\overbrace{p(z_k,x_{1:k})}^{F}$$

Then divide by normalizing constant

Now we can do the following:

What you can do:

- Inference: $\nearrow$ $P(z_k \neq z_{k+1}|x)$ "change detection"
- Estimate params ("Baum-Welch")
- Sampling from posterior dist: $z|x$

difference between viterbi and posterior

how/when forward and backward algorithms used?

how to fit an HMM

**Structured Perceptron**

[Structured Prediction for Natural Language Processing]

**Language models (how to compute probabilities)**

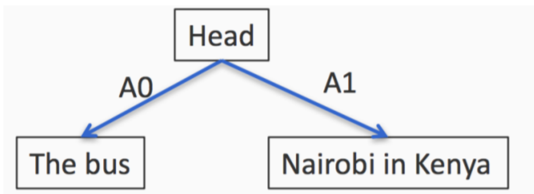
n-grams

## Structured output is…

❖ A predefine structure

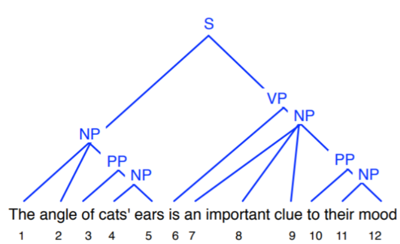
| Predicate | A0 | A1 | Location |
|---|---|---|---|
| Head | The bus | Nairobi in Kenya | - |

❖ Can be represented as a graph




NLP application

Problem: find compositional phrases from the whole sentence down to the words




**<u>Hints/previous exam questions</u>**

Structured prediction: it needs the label of the class, if not it can't be structured prediction

What should be the output of a neural system in a classification model? softmax or log_softmax → not sigmoids cause is not normalizing

- Sigmoid: every value between 0 and 1 (for classification is good, not for multiclass classification)
- Softmax / Log softmax: one high value for a class and the rest close to 0

Given this vector what is the softmax of this vector?

$$P(y=j \mid \mathbf{x}) = \frac{e^{\mathbf{x}^\mathsf{T}\mathbf{w}_j}}{\sum_{k=1}^{K} e^{\mathbf{x}^\mathsf{T}\mathbf{w}_k}}$$

Convert vector into probabilities

Exponential deals with negative values
Can make it weighted

Does idf change for 2 documents? NO
‘And’ : [0.9]
Does tf change for 2 documents? YES
‘And’ : [0.1, 0.2, 0.03, 0.07] ; 4 documents

Log sum exp trick

In machine learning, arithmetic underflow can become a problem when multiplying together many small probabilities. In many models it can be useful to calculate the log sum of exponentials.

$$\log \sum_{i=1}^{n} \exp(x_i)$$

If $x_i$ is sufficiently large or small, this will result in an arithmetic overflow/underflow. To avoid this we can use a common trick called the Log Sum Exponential trick.

$$\begin{aligned}\log \sum_{i=1}^{n} \exp(x_i) &= \log \exp(b) \sum_{i=1}^{n} \exp(x_i - b) \\ &= b + \log \sum_{i=1}^{n} \exp(x_i - b)\end{aligned}$$

Where $b$ is $\max(x)$.

forward quantity definition

$$f(1,x,c) := P_{\text{init}}(c \mid \text{start}) \times P_{\text{emiss}}(x_1 \mid c)$$

$$f(i,x,c) := P(X_1 = x_1, \ldots, X_i = x_i, Y_i = c)$$

$$b(N,x,c) := P_{\text{final}}(\text{stop} \mid c)$$

$$b(i,x,c) := P(X_{i+1} = x_{i+1}, \ldots, X_N = x_N \mid Y_i = c)$$

compute edit distance

| | * | k | n | i | t | t | i | n | g |
|---|---|---|---|---|---|---|---|---|---|
| * | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| k | 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| i | 2 | 1 | 1 | 1 | 2 | 3 | 4 | 5 | 6 |
| t | 3 | 2 | 2 | 2 | 1 | 2 | 3 | 4 | 5 |
| t | 4 | 3 | 3 | 3 | 2 | 1 | 2 | 3 | 4 |
| e | 5 | 4 | 4 | 4 | 3 | 2 | 2 | 3 | 4 |
| n | 6 | 5 | 4 | 5 | 4 | 3 | 3 | 2 | 3 |

Tupu tamadre - revélate quien eres wey
Rosa Melano
Elverga Larga