Measures of Distributional Similarity

Lillian Lee
Cornell University

The Sparse Data Problem

Goal: estimate the probability of cooccurrences

- canonical application: speech recognition:

  "It's hard to wreck a nice beach", vs.

  "It's hard to recognize speech"

- other applications: parsing lexicalized formalisms, WSD, MT, ...

Problem: even in large training sets, reasonable lexical coocurrences do not appear.

"good ACL talk" – no hits on AltaVista

How do we estimate the probability of *unseen* events?

Similarity Information

Key idea: look at information provided by similar words:

$$\left.\begin{array}{l}\text{"great ACL talk"}\\ \text{"good ACL presentation"}\end{array}\right\} \Rightarrow \text{"good ACL talk" is likely}$$

We are interested in distributional similarity:

x and x' are similar means $P(Y|x) \approx P(Y|x')$

(Not semantic similarity: "You used me!" $\not\Rightarrow$ "You utilized me!")

What is the best way to measure distributional similarity?

Project Outline

- Empirically compare well-known similarity functions
- Find commonalities among best performers
- Develop better measures based on analysis

Similarity Functions

- Euclidean distance (L_2 norm)
- L_1 norm (Manhattan metric) [statistics, machine learning]
- cosine [information retrieval]
- Jaccard's coefficient [information retrieval] [Lin 98]

- Jensen-Shannon divergence [Dagan, Lee, and Pereira 99]
- Confusion probability [Sugawara et al. 85; Essen and Steinbiss 92; Grishman and Sterling 93]
- Kendall's tau [Hatzivassiloglou and McKeown, 93; Hatzivassiloglou, 96]

Not suitable: mutual info, Hamming distances, value differences, Dice, KL divergence

Evaluation Task

Data: 600K verb-object pairs from 1988 AP newswire; five disjoint test sets.

Test instances: $\{(n, v_1), (n, v_2)\}$

- both pairs unseen
- $P(v_1) \approx P(v_2)$ (eliminate frequency factor)

Question: which pair is more likely?

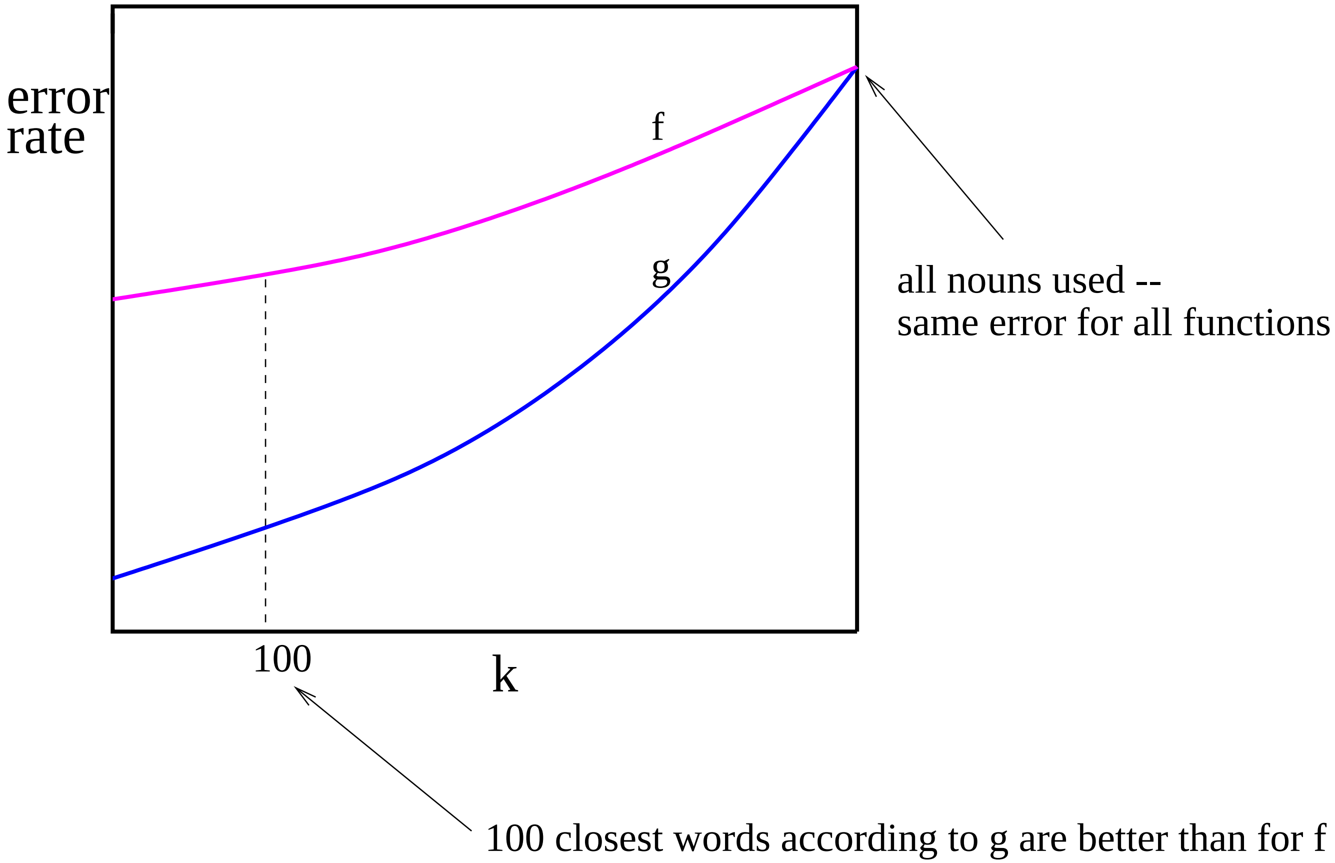
Defn: noun m prefers v_1 if $P(v_1|m) > P(v_2|m)$

Decision rule: choose pair preferred by majority of n's k nearest neighbors.

$\triangleright$ eliminates parametrization, scaling problems ...

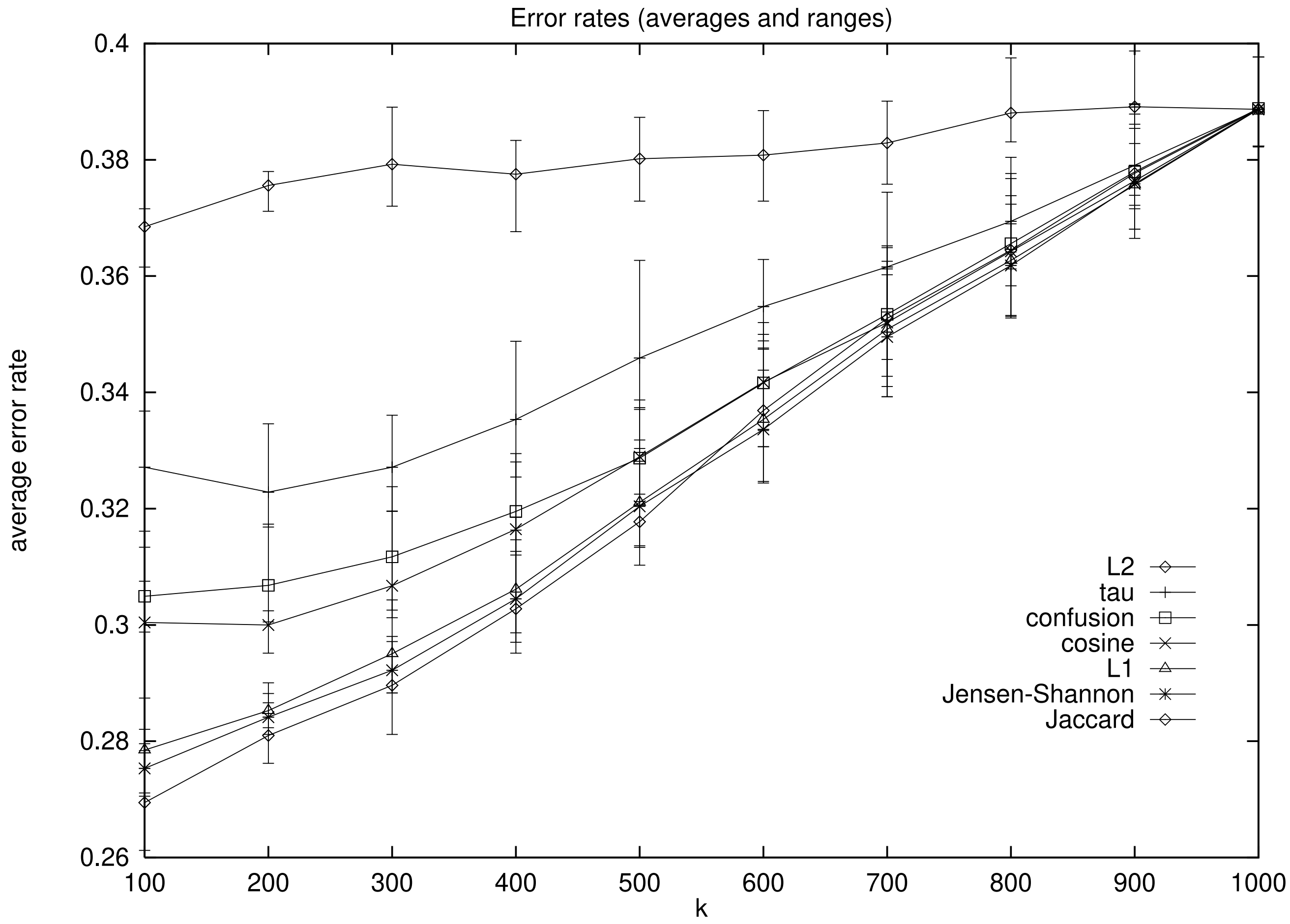
Evaluating Similarity Functions

Recall: choose pair preferred by majority of n's k nearest neighbors.



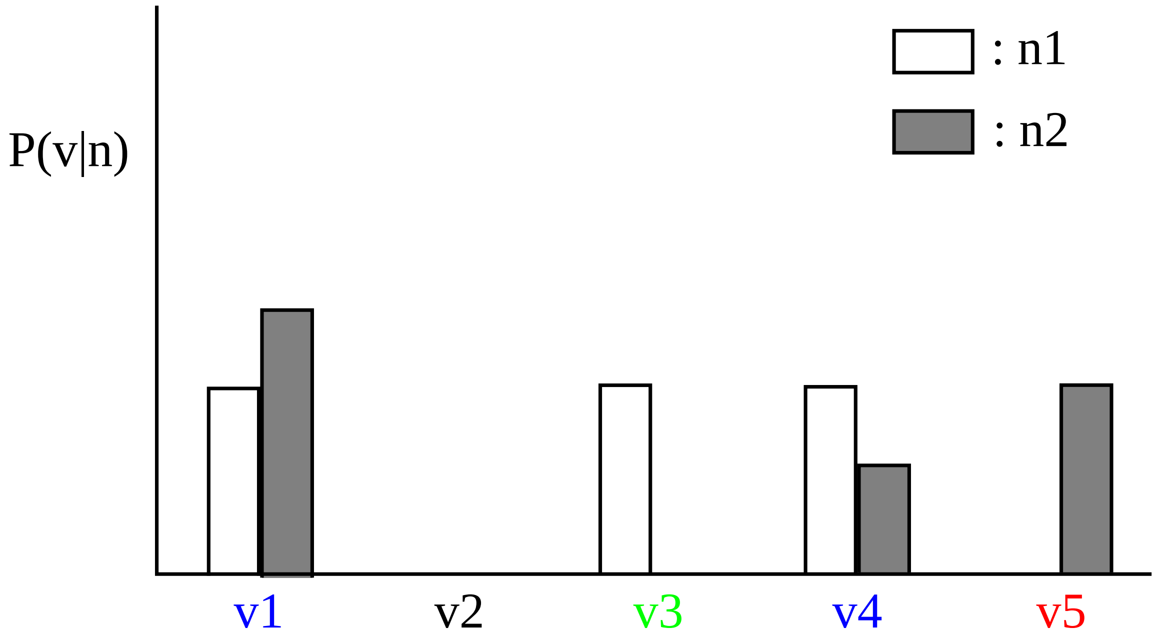
A steeper slope means an inherently better similarity function.

Comparison Results

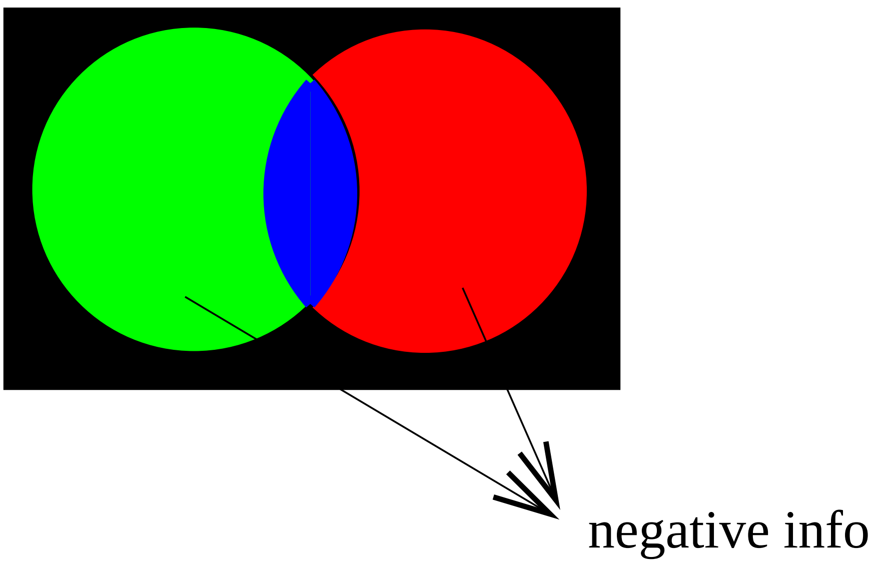


Can we find coherence among the four groups?

Histograms Partition Verbs

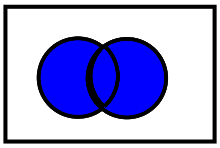


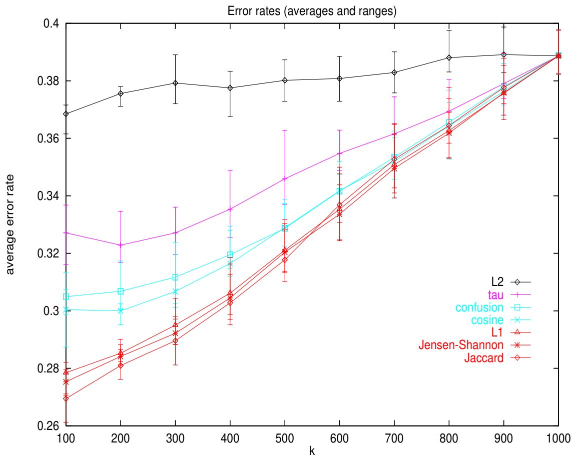
verb space:



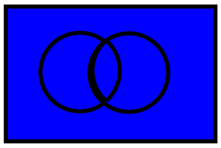
Verb Information Usage

L2

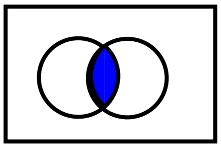




Tau

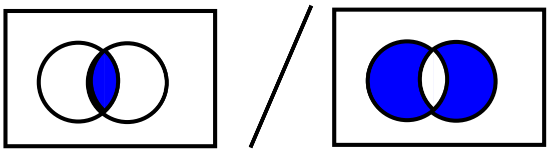


Confusion

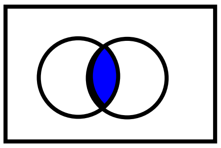


+ unigram probs

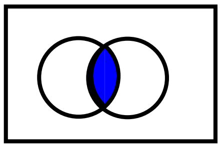
Cosine



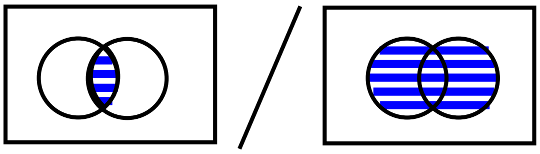
L1



Jensen-Shannon



Jaccard



(magnitude only)

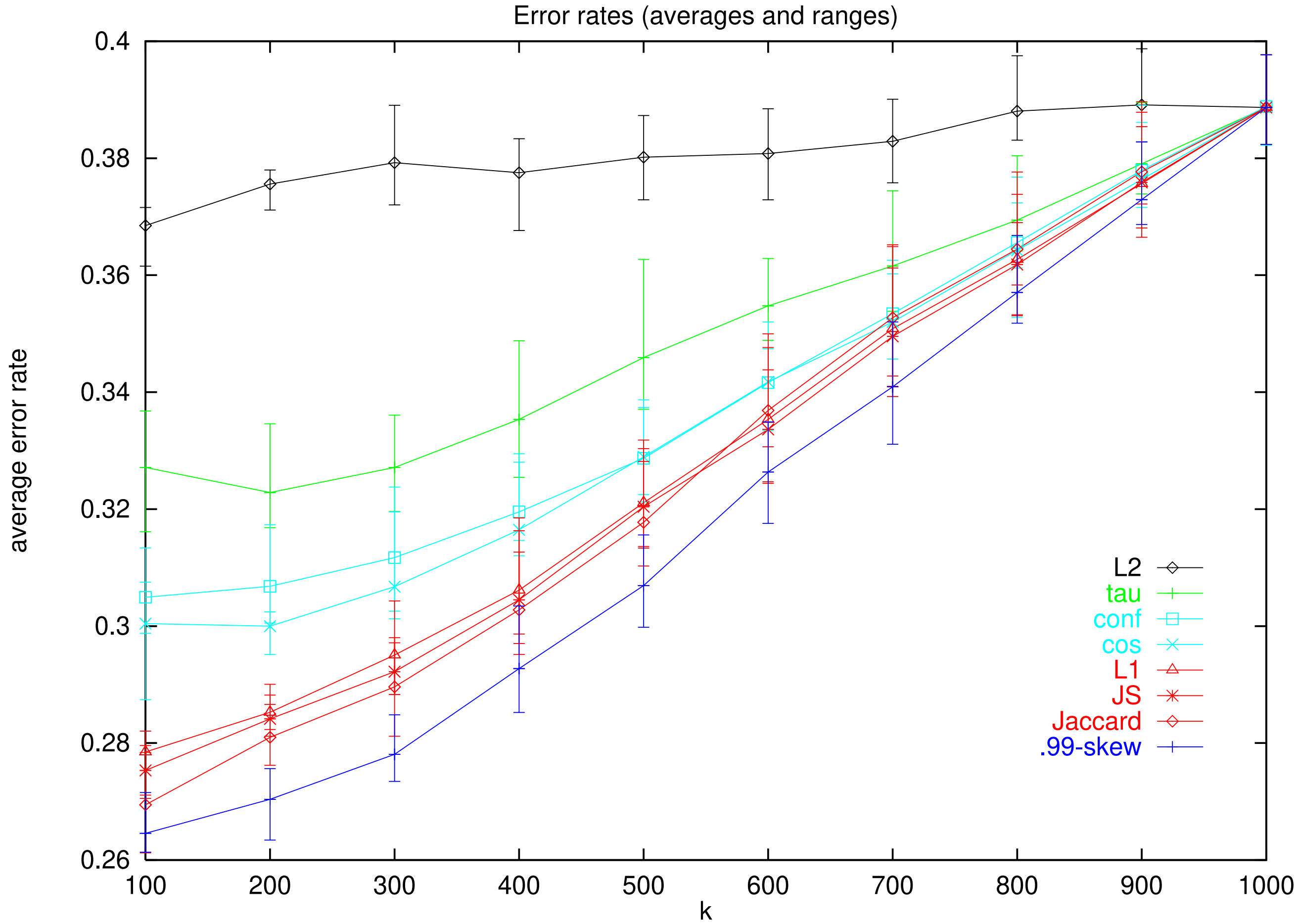
The Skew Divergence

The *skew divergence*: weighted version of the KL divergence $D(p\|q)$:

$$s(p,q,\alpha) = D(p \,\|\, \alpha \cdot q + (1-\alpha) \cdot p)$$

- relates to information-theoretic distance (coding inefficiency) between distributions
- based only on the “intersection” info
- asymmetric
- α: confidence in the empirical distribution of q
  - ▷ data doesn’t require smoothing (unless $\alpha = 1$)

.99-Skew Results



All differences (except $k = 1000$) statistically significant

Summary

- We can compare different functions despite scaling and parametrization issues.
- "Intersection" info helps account for similarity measure performance.
  - ▷ Is explicit negative info important?
- The skew divergence shows promise.

Future work:

Further examination of the skew divergence

▷ cf. weighted Shannon divergences [J. Lin 91]

Look at performance on application data

▷ PP attachment, ngrams