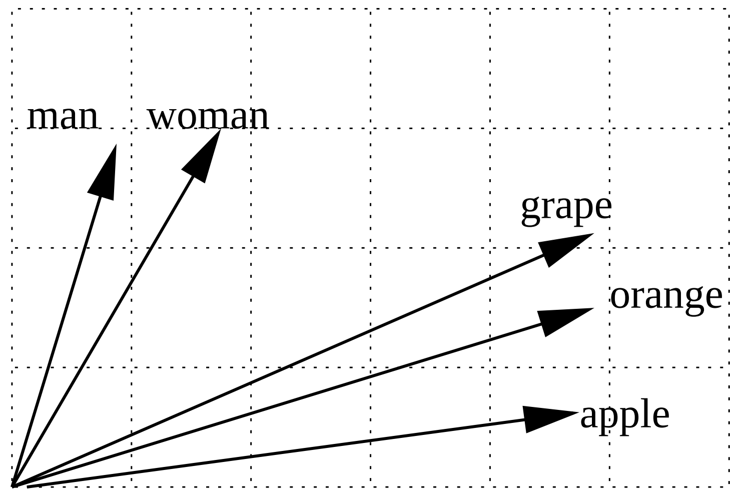
Lexical Semantics

Regina Barzilay

MIT

October, 5766

Last Time: Vector-Based Similarity Measures

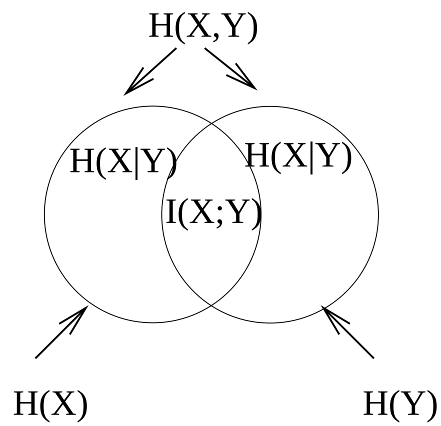


- Euclidian: $|\vec{x}, \vec{y}| = |\vec{x} - \vec{y}| = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$
- Cosine: $cos(\vec{x}, \vec{y}) = \frac{\vec{x} * \vec{y}}{|\vec{x}||\vec{y}|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \sqrt{\sum_{i=1}^{n} y_i^2}}$

Last Time: Probabilistic Similarity Measures

(Pointwise) Mutual Information: $I(x;y) = \log \frac{P(x,y)}{P(x)P(y)}$

- Mutual Information: $I(X;Y) = E_{p(x,y)} \log \frac{p(X,Y)}{p(X)p(Y)}$



Example: Computing MI

$I(w_1, w_2)$	$C(w_1)$	$C(w_2)$	$C(w_1, w_2)$	w_1	w_2
16.31	30	117	20	Agatha	Christie
15.94	77	59	20	videocassette	recorder
15.19	24	320	20	unsalted	butter
1.09	14907	9017	20	first	made
0.29	15019	15629	20	time	last

Example: Computing MI

$I(w_1, w_2)$	$C(w_1)$	$C(w_2)$	$C(w_1, w_2)$	w_1	w_2
15.02	1	19	1	fewest	visits
12.00	5	31	1	Indonesian	pieces
9.21	13	82	20	marijuana	growing

Last Time: Probabilistic Similarity Measures

Kullback Leibler Distance: $D(p||q) = \sum p(x) log \frac{p(x)}{q(x)}$

- Closely related to mutual information

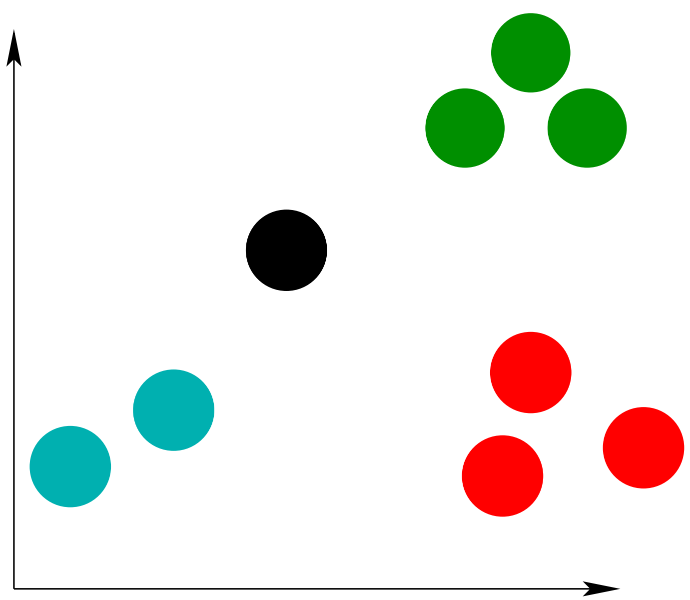
$$I(X;Y) = D(p(x,y)||p(x)p(y))$$

- Related measure : Jensen-Shannon divergence:

$$D_{JS(p,q)} = \frac{1}{2} D(p||\frac{p+q}{2}) + \frac{1}{2} D(q||\frac{p+q}{2})$$

Beyond Pairwise Similarity

- Clustering is "The art of finding groups in data"(Kaufmann and Rousseeu)
- Clustering algorithms divide a data set into homogeneous groups (clusters), based on their similarity under the given representation.



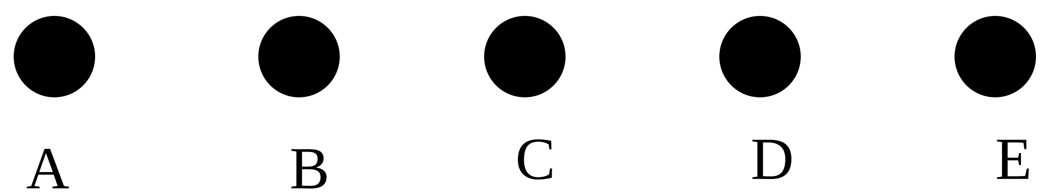
Hierarchical Clustering

Greedy, bottom-up version:

- Initialization: Create a separate cluster for each object
- Each iteration: Find two most similar clusters and merge them
- Termination: All the objects are in the same cluster

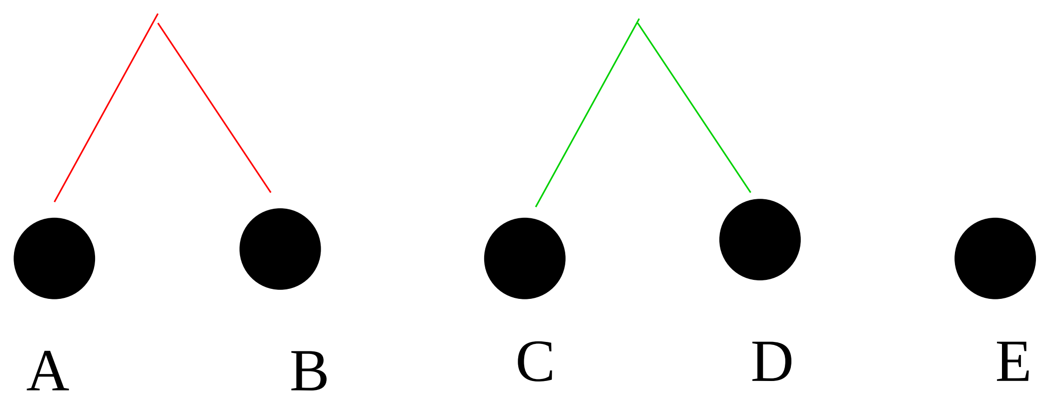
Agglomerative Clustering

	E	D	C	B
A	0.1	0.2	0.2	0.8
B	0.1	0.1	0.2	
C	0.0	0.7		
D	0.6			



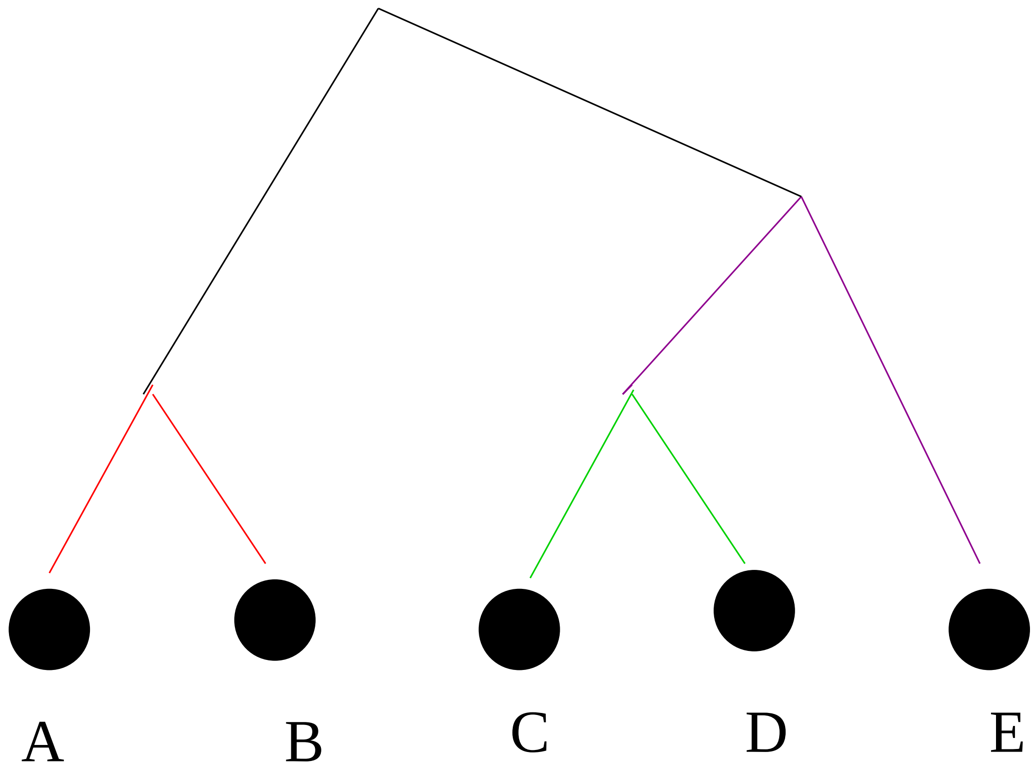
Agglomerative Clustering

	E	D	C	B
A	0.1	0.2	0.2	0.8
B	0.1	0.1	0.2	
C	0.0	0.7		
D	0.6			



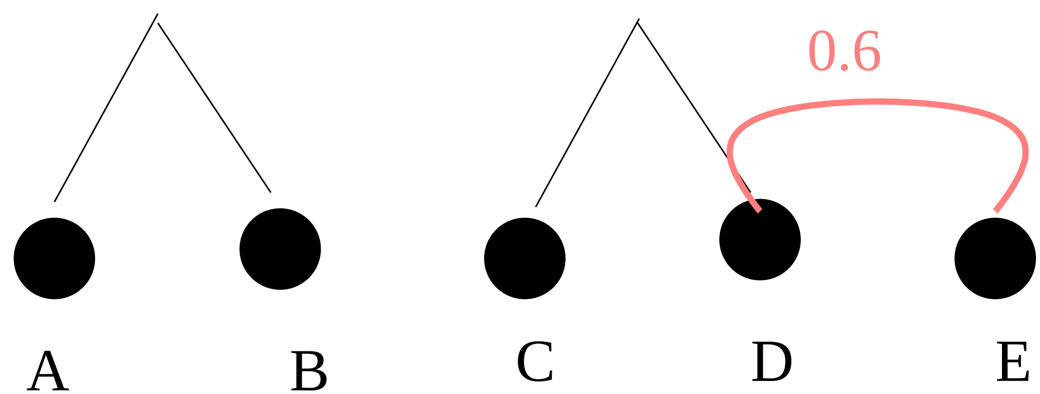
Agglomerative Clustering

	E	D	C	B
A	0.1	0.2	0.2	0.8
B	0.1	0.1	0.2	
C	0.0	0.7		
D	0.6			



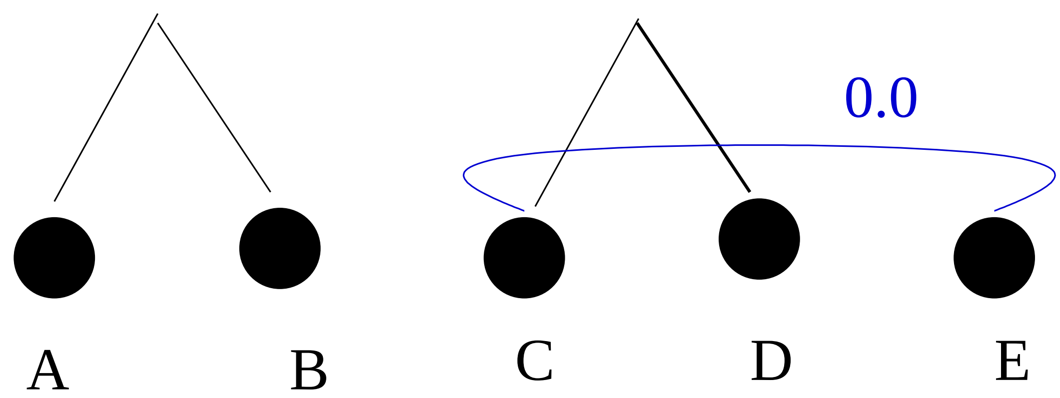
Clustering Function

	E	D	C	B
A	0.1	0.2	0.2	0.8
B	0.1	0.1	0.2	
C	0.0	0.7		
D	0.6			



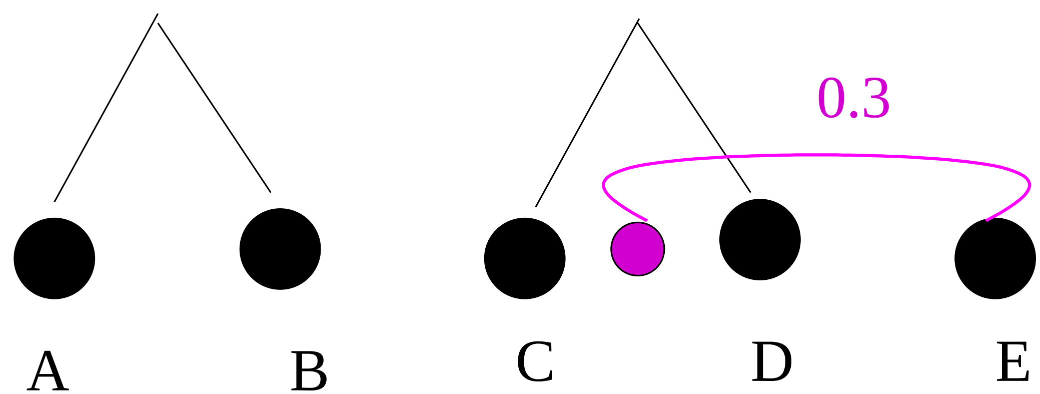
Clustering Function

	E	D	C	B
A	0.1	0.2	0.2	0.8
B	0.1	0.1	0.2	
C	0.0	0.7		
D	0.6			



Clustering Function

	E	D	C	B
A	0.1	0.2	0.2	0.8
B	0.1	0.1	0.2	
C	0.0	0.7		
D	0.6			



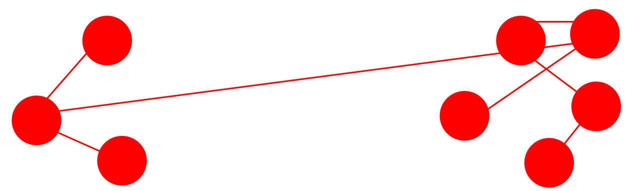
Clustering Function

- Single-link: Similarity of two most similar members
- Complete-link: Similarity of two least similar members
- Group-average: Average similarity between members

Single-Link Clustering

- Achieves Local Coherence
- Complexity $O(n^2)$
- Fails when clusters are not well separated

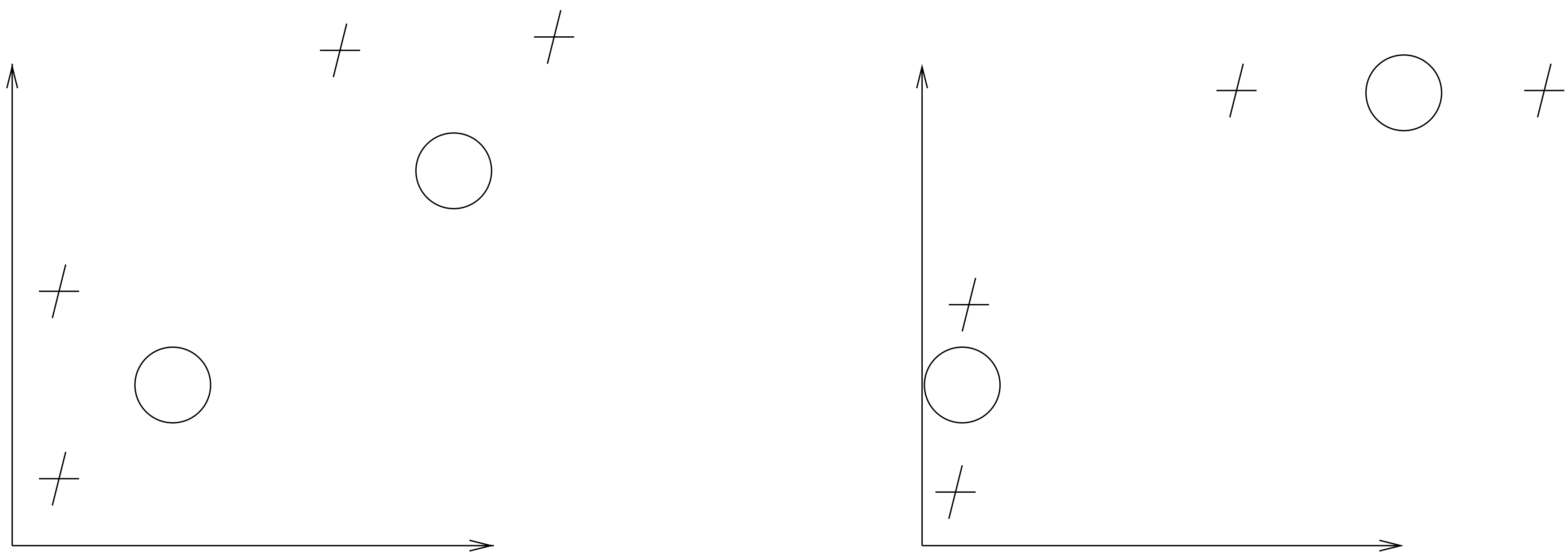
Complete-Link Clustering



- Achieves Global Coherence
- Complexity $O(n^2 \log n)$
- Fails when clusters aren't spherical, or of uniform size

K-Means Algorithm: Example

Iterative, hard, flat clustering algorithm based on Euclidean distance



K-Means Algorithm

1. Choose k points at random as cluster centers
2. Assign each instance to its closest cluster center
3. Calculate the centroid (mean) for each cluster, use it as a new cluster center
4. Iterate steps 2 and 3 until the cluster centers don't change anymore

K-Means Algorithm: Hard EM

1. Guess initial parameters
2. Use model to make the best guess of c_i (E-step)
3. Use the new complete data to learn better model (M-step)
4. Iterate (2-3) until convergence

Evaluating Clustering Methods

- Perform task-based evaluation
- Test the resulting clusters intuitively, i.e., inspect them and see if they make sense. Not advisable.
- Have an expert generate clusters manually, and test the automatically generated ones against them.
- Test the clusters against a predefined classification if there is one

Comparing Clustering Methods

(Meila, 2002)

n	total # of points
n_k	# of points in cluster C_k
K	# of nonempty clusters
N_{11}	# of pairs that are in the same cluster under C and C'
N_{00}	# of pairs that are in different clusters under C and C'
N_{10}	# of pairs that are in the same cluster under C but not C'
N_{01}	# of pairs that are in the same cluster under C' but not C

Comparing by Counting Pairs

- Wallace criteria

$$W_1(C, C') = \frac{N_{11}}{\sum_k n_k(n_k - 1)/2}$$

$$W_2(C, C') = \frac{N_{11}}{\sum_{k'} n_{k'}({n'}_{k'} - 1)/2}$$

- Fowles-Mallows criterion

$$F(C, C') = \sqrt{W_1(C, C')W_2(C, C')}$$

Problems: ?

Comparing Clustering by Set Matching

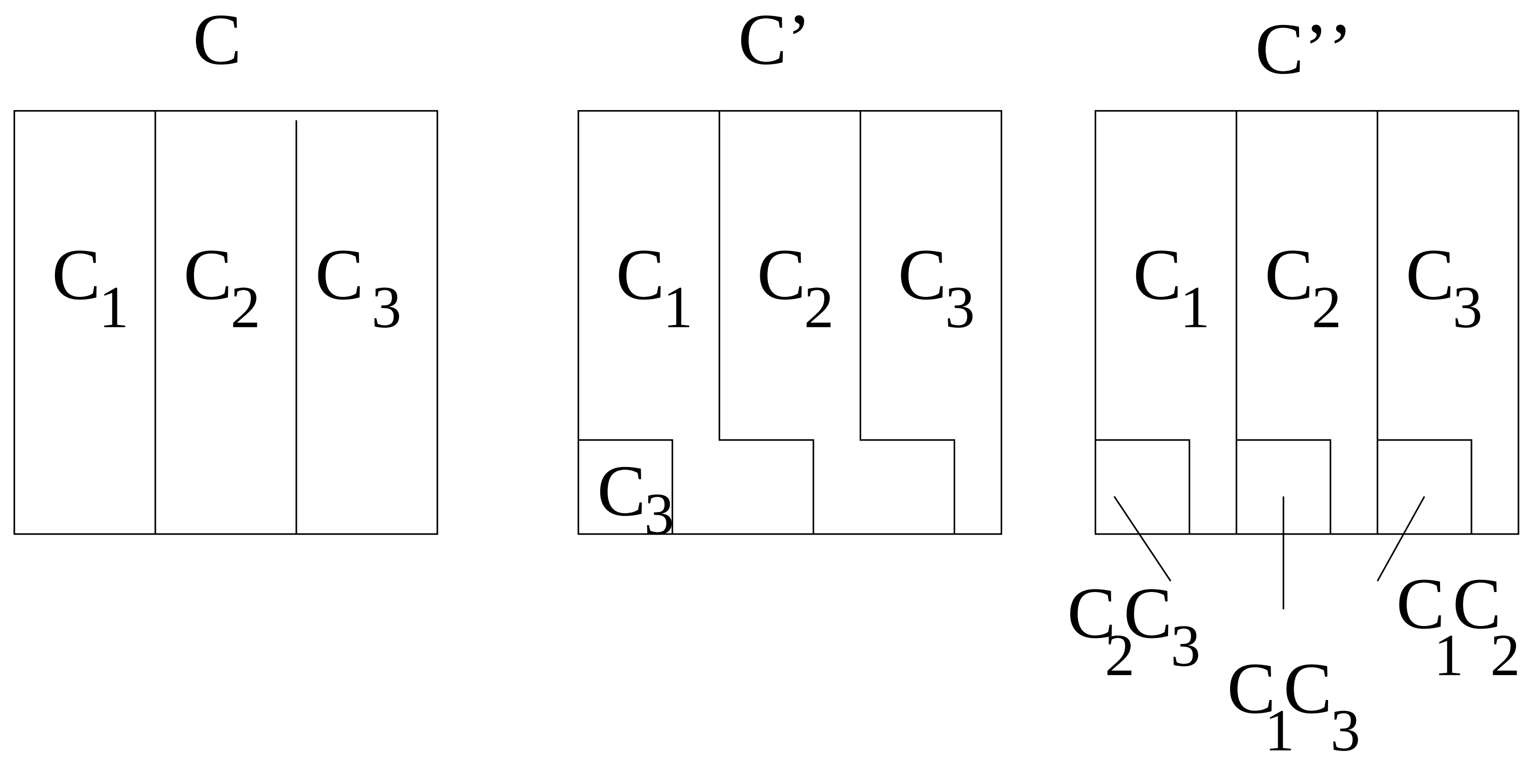
Contingency table M is a $K \times K$ matrix, whose kk' element is the number of points in the intersection of clusters C_k and $C'_{k'}$

$$L(C, C') = \frac{1}{K} \sum_k \max_{k'} \frac{2m_{kk'}}{n_k + n'_k}$$

Problems: ?

Comparing Clustering by Set Matching

$$L(C, C') = \frac{1}{K} \sum_k \max_{k'} \frac{2m_{kk'}}{n_k + n'_k}$$



Distributional Syntax

Sequences of word clusters and their contexts (Klein, 2005)

Tag	Top Context by Frequency
DT	(IN-NN), (IN-JJ), (IN-NNP), (VB-NN)
JJ	(DT-NN), (IN-NNS), (IN-NN), (JJ-NN),(DT-NNS)
MD	(NN-VB), (PRP-VB), (NNS-VB), (NNP-VB), (WDT-VB)
NN	(DT-IN), (JJ-IN), (DT-NN), (NN-IN), (NN-.)
VB	(TO-DT), (TO-IN), (MD-DT), (MD-VBN),(TO-JJ)

Distributional Syntax

The most similar POS pairs and POS sequence pairs based on D_{JS} of their context

Rank	Tag pairs	Sequence Pairs
1	(VBZ,VBD)	(NNP NNP, NNP NNP NNP)
2	(DT,PRP$)	(DT JJ NN IN, DT NN IN)
3	(NN,NNS)	(NNP NNP NNP NNP, NNP NNP NNP)
4	(WDT,WP)	(DT NNP NNP, DT NNP)
5	(VBG,VBN)	(IN DT JJ NN, IN DT NN)
14	(JJS, JJR)	(NN IN DT, NN DT)

Linear vs. Hierarchical Context

The left (right) context of x is the left(right) sibling of the lowest ancestor of x

Rank	Linear	Hierarchical
1	(NN NNS, JJ NNS)	(NN NNS, JJ NNS)
2	(IN NN, IN DT NN)	(IN NN, IN DT NN)
3	(DT JJ NN, DT NN)	(IN DT JJ NN, IN JJ NNS)
4	(DT JJ NN, DT NN NN)	(VBZ VBN, VBD VBN)
5	(IN DT JJ NN, IN DT NN)	(NN NNS, JJ NN NNS)

Grammar Induction

- Task: Unsupervised learning of a language's syntax from a corpus of observed sentences

The cat stalked the mouse.
The mouse quivered.
The cat smiled.

- A tree induction system is not forced to learn all aspects of language (semantics, discourse)

Motivation

- Linguistic motivation:
  - Empirical argument against the poverty of the stimulus (Chomsky, 1965)
  - Empirical investigation of syntax modularity (Fodor, 1983; Jackendoff, 1996)
- Engineering motivation:
  - No need in training data

Evaluation and Baselines

- Evaluation:
  - Compare grammars
  - Compare trees
- Baselines:
  - Random Trees
  - Left- and Right-Branching Trees

Structure Search Experiment

- Structure search
  - Add production to context free grammar
  - Select HMM topology
- Parameter search
  - Determine parameters for a fixed PCFG

Finding Topology

Stolcke&Omohundro, 1994: Bayesian model merging

- **Data incorporation:** Given a body of data X, build an initial model M_0 by explicitly accommodating each data point individually such that M_0 maximizes the likelihood $P(X|M)$.
- **Generalization:** Build a sequence of new models, obtaining M_{i+1} from M_i by applying a *merging* operator m that coalesces substructures in M_i, $M_{i+1} = m(M_i), i = 0, 1$
- **Optimization:** Maximize posterior probability
- **Search strategy:** Greedy or beam search through the space of possible merges

HMM Topology Induction

- **Data incorporation:** For each observed sample create a unique path between the initial and final states by assigning a new state to each symbol token in the sample
- **Generalization:** Two HMM states are replaced by a single new state, which inherits the union of the transitions and emissions from the old states.

HMM Topology Induction

- **Prior distribution:** Choose uninformative priors for a model M with topology M_s and parameters θ_M.

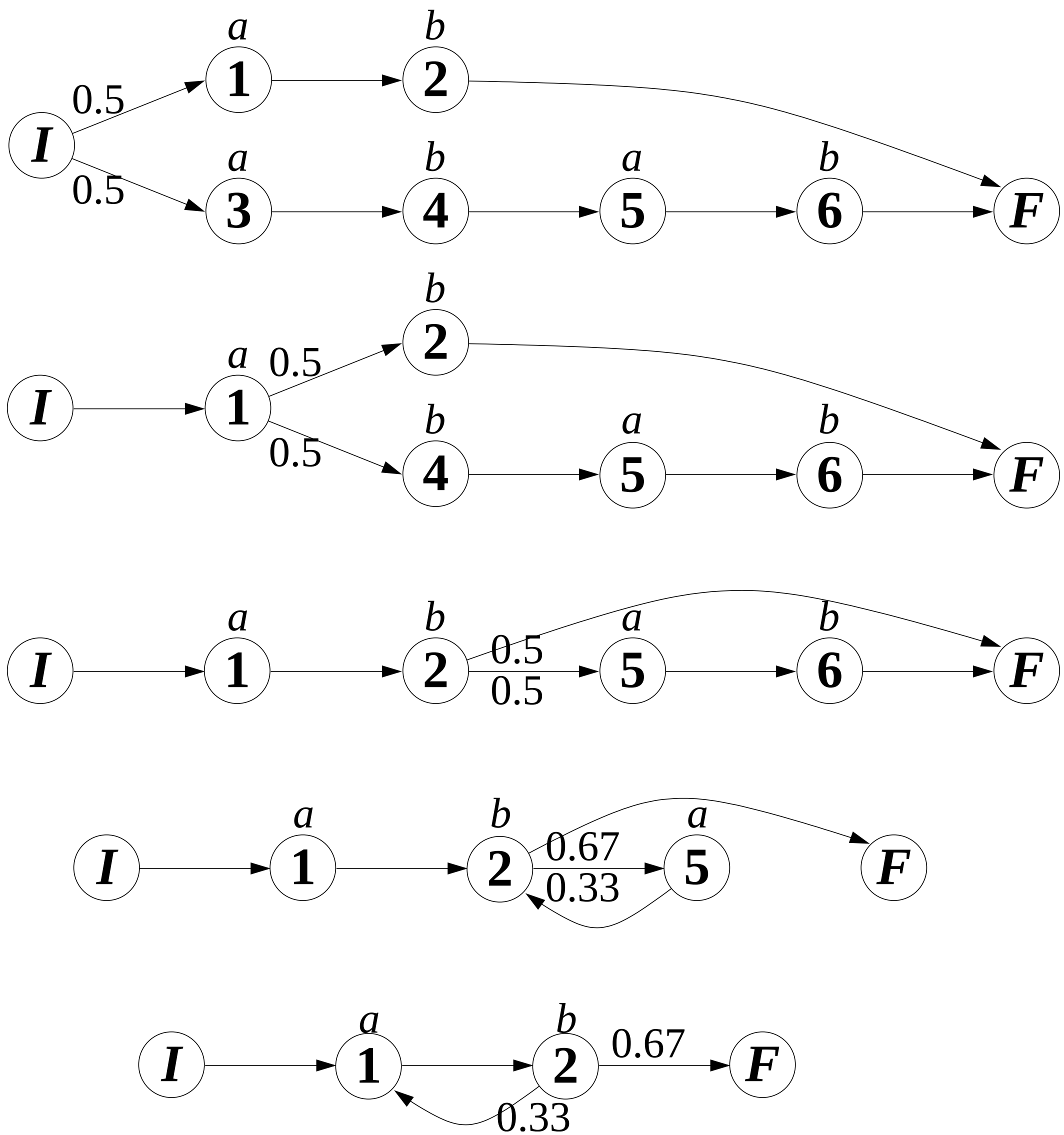
$$P(M) = P(M_s)P(\theta_M|M_s)$$

$$P(M_s) \propto \exp(-l(M_s))$$

where $l(M_s)$ is the number of bits required to encode M_s.

- **Search:** Greedy merging strategy.

Example



a
b
0.5
I
1
2
0.5
a
b
a
b
3
4
5
6
F
b
I
a
0.5
1
2
0.5
b
a
b
4
5
6
F
I
a
1
b
2
0.5
0.5
a
5
b
6
F
I
a
1
b
2
0.67
0.33
a
5
F
I
a
1
b
2
0.67
0.33
F

PCFG Induction

- **Data Incorporation:** Add a top-level production that covers the sample precisely. Create one nonterminal for each observed terminal.
- **Merging and Chunking:** During merging, two nonterminals are replaced by a single new state. Chunking takes a given sequence of nonterminals and abbreviates it using a newly created nonterminal.
- **Prior distribution:** Similar to HMM.
- **Search:** Beam search.

Example

Input: {ab,aabb,aaabbb}

	S -> A B -> A A B B -> A A A B B B A -> a B -> b
Chunk(AB)->X	S -> X -> A X B -> A A X B B X -> A B
Chunk(AXB)->Y	S -> X -> Y -> A Y B X -> A B Y -> A X B
Merge S,Y	S -> X -> A S B X -> A B
Merge S,X	S -> A B -> A S B

Results for PCFGS

- Formal language experiments
  - Successfully learned simple grammars

Language	Sample no.	Grammar	Search
Parentheses	8	$S \longrightarrow ()\|(S)\|SS$	BF
a^{2n}	5	$S \longrightarrow aa\|SS$	BF
$(ab)^n$	5	$S \longrightarrow ab\|aSb$	BF
wcw^R, $w \in \{a, b\}^\star$	7	$S \longrightarrow c\|aSa\|bSb$	BS (3)
Addition strings	23	$S \longrightarrow a\|b\|(S)\|S + S$	BS(4)

- Natural Language syntax
  - Mixed results (issues related to data sparseness)

Example of Learned Grammar

Target Grammar	Learned Grammar
$S \rightarrow NP\, VP$	$S \rightarrow NP\, VP$
$VP \rightarrow Verb\, NP$	$VP \rightarrow V\ NP$
$NP \rightarrow Det\, Noun$	$NP \rightarrow DetN$
$NP \rightarrow Det\, Noun\, RC$	$NP \rightarrow NP\, RC$
$RC \rightarrow Rel\, VP$	$RC \rightarrow REL\, VP$
$Verb \rightarrow saw \vert heard$	$V \rightarrow saw \vert heard$
$Noun \rightarrow cat \vert dog \vert mouse$	$N \rightarrow cat \vert dog \vert mouse$
$Det \rightarrow a \vert the$	$Det \rightarrow a \vert the$
$Rel \rightarrow that$	$Rel \rightarrow that$

Example

Input: {ab,aabb,aaabbb}

S −> A B
−> A A B B
−> A A A B B B
A −> a
B −> b

Chunk(AB)−>X

S −> X
−> A X B
−> A A X B B
X −> A B

Chunk(AXB)−>Y

S −> X
−> Y
−> A Y B
X −> A B
Y −> A X B

Merge S,Y

S −> X
−> A S B
X −> A B

Merge S,X

S −> A B
−> A S B

Issue with Chunk/Merge Systems

- Hard to recover from initial choices
- Hard to make local decision which will interact with each other (e.g., group verb preposition and preposition-determiner, both wrong and non consistent)
- Good local heuristics often don't have well formed objectives that can be evaluated for the target grammar

Learn PCFGs with EM

- (Lari&Young 1990): Learning PCFGs with EM
  - Full binary grammar over n symbols
  - Parse randomly at first
  - Re-estimate rule probabilities of parses
  - Repeat

Grammar Format

- Lari&Young, 1990: Satisfactory grammar learning requires more nonterminals than are theoretically needed to describe a language at hand
- There is no guarantee that the nonterminals that the algorithm learns will have any resemblance to nonterminals motivated in linguistic analysis
- Constraints on the grammar format may simplify the reestimation procedure
  - Carroll&Charniak, 1992: Specify constraints on non-terminals that may appear together on the right-hand side of the rule

Partially Unsupervised Learning

Pereira&Schabes 1992

- Idea: Encourage the probabilities into a good region of the parameter space
- Implementation: modify Inside-Outside algorithm to consider only parses that do not cross provided bracketing
- Experiments: 15 non terminals over 45 POS tags
  The algorithm uses Treebank bracketing, but ignores the labels
- Evaluation Measure: fraction of nodes in gold trees correctly posited in proposed trees (unlabeled recall)

- Results:
  - Constrained and unconstrained grammars have similar cross-entropy
  - But very different bracketing accuracy: 37% vs. 90%

Current Performance

- Constituency recall:

Random Baseline	39.4
Klein'2005	88.0
Supervised PCFG	92.8

- Why it works?
  - Combination of simple models
  - Representations designed for unsupervised learning