Lexical Semantics

Natural Language Processing
CS 4120/6120—Spring 2017
Northeastern University

David Smith
some slides from
Jason Eisner & Richard Socher

Breaking News!

Breaking News!

- Words aren't just atomic symbols!
- People are bad at coming up with features for machine learning models!
- Using billions of features can be slow!

Overview

- Semantics so far: compositional semantics
  - How to put together propositions from atomic meanings (lexicon)?
- Now: lexical semantics
  - What are those atomic meanings?
  - Clustering words with similar senses
  - Sense disambiguation, functional clustering

Linguistic Objects in this Course

Linguistic Objects in this Course

- Trees (with strings at the nodes)
  - Syntax, semantics
  - **Algorithms:** Generation, parsing, inside-outside, build semantics

Linguistic Objects in this Course

- Trees (with strings at the nodes)
  - Syntax, semantics
  - **Algorithms:** Generation, parsing, inside-outside, build semantics
- Sequences (of strings)
  - n-grams, tag sequences
  - morpheme sequences, phoneme sequences
  - **Algorithms:** Finite-state, best-paths, forward-backward

Linguistic Objects in this Course

- Trees (with strings at the nodes)
  - Syntax, semantics
  - **Algorithms:** Generation, parsing, inside-outside, build semantics
- Sequences (of strings)
  - n-grams, tag sequences
  - morpheme sequences, phoneme sequences
  - **Algorithms:** Finite-state, best-paths, forward-backward
- "Atoms" (unanalyzed strings)
  - Words, morphemes
  - Represent by contexts – other words they occur with
  - **Algorithms:** Grouping similar words, splitting words into senses, mapping (senses of) words to continuous space (embedding)

Clustering

A Concordance for “party”

- thing. She was talking at a party thrown at Daphne's restaurant in
- have turned it into the hot dinner-party topic. The comedy is the
- selection for the World Cup party, which will be announced on May 1
- in the 1983 general election for a party which, when it could not bear to
- to attack the Scottish National Party, who look set to seize Perth and
- that had been passed to a second party who made a financial decision
- the by-pass there will be a street party. "Then," he says, "we are going
- number-crunchers within the Labour party, there now seems little doubt
- political tradition and the same party. They are both relatively Anglophilic
- he told Tony Blair's modernised party they must not retreat into "warm
- "Oh no, I'm just here for the party," they said. "I think it's terrible
- A future obliges each party to the contract to fulfil it by
- be signed by or on behalf of each party to the contract." Mr David N

What Good are Word Senses?

- thing. She was talking at a party thrown at Daphne's restaurant in
- have turned it into the hot dinner-party topic. The comedy is the
- selection for the World Cup party, which will be announced on May 1
- in the 1983 general election for a party which, when it could not bear to
- to attack the Scottish National Party, who look set to seize Perth and
- that had been passed to a second party who made a financial decision
- the by-pass there will be a street party. "Then," he says, "we are going
- number-crunchers within the Labour party, there now seems little doubt
- political tradition and the same party. They are both relatively Anglophilic
- he told Tony Blair's modernised party they must not retreat into "warm
- "Oh no, I'm just here for the party," they said. "I think it's terrible
- A future obliges each party to the contract to fulfil it by
- be signed by or on behalf of each party to the contract." Mr David N

What Good are Word Senses?

- thing. She was talking at a party thrown at Daphne's restaurant in
- have turned it into the hot dinner-party topic. The comedy is the
- selection for the World Cup party, which will be announced on May 1
- the by-pass there will be a street party. "Then," he says, "we are going
- "Oh no, I'm just here for the party," they said. "I think it's terrible

- in the 1983 general election for a party which, when it could not bear to
- to attack the Scottish National Party, who look set to seize Perth and
- number-crunchers within the Labour party, there now seems little doubt
- political tradition and the same party. They are both relatively Anglophilic
- he told Tony Blair's modernised party they must not retreat into "warm

- that had been passed to a second party who made a financial decision
- A future obliges each party to the contract to fulfil it by
- be signed by or on behalf of each party to the contract." Mr David N

What Good are Word Senses?

- John threw a “rain forest” party last December. His living room was full of plants and his box was playing Brazilian music …

What Good are Word Senses?

- Replace word w with sense s
  - Splits w into senses: distinguishes this token of w from tokens with sense t
  - Groups w with other words: groups this token of w with tokens of x that also have sense s

What Good are Word Senses?

- number-crunchers within the Labour party, there now seems little doubt
- political tradition and the same party. They are both relatively Anglophilic
- he told Tony Blair's modernised party they must not retreat into "warm
- thing. She was talking at a party thrown at Daphne's restaurant in
- have turned it into the hot dinner-party topic. The comedy is the
- selection for the World Cup party, which will be announced on May 1
- the by-pass there will be a street party. "Then," he says, "we are going
- "Oh no, I'm just here for the party," they said. "I think it's terrible

- an appearance at the annual awards bash , but feels in no fit state to
- -known families at a fundraising bash on Thursday night for Learning
- Who was paying for the bash? The only clue was the name Asprey,
- Mail, always hosted the annual bash for the Scottish Labour front-
- popular. Their method is to bash sense into criminals with a short,
- just cut off people's heads and bash their brains out over the floor,

What Good are Word Senses?

- number-crunchers within the Labour party, there now seems little doubt
- political tradition and the same party. They are both relatively Anglophilic
- he told Tony Blair's modernised party they must not retreat into "warm

- thing. She was talking at a party thrown at Daphne's restaurant in
- have turned it into the hot dinner-party topic. The comedy is the
- selection for the World Cup party, which will be announced on May 1
- the by-pass there will be a street party. "Then," he says, "we are going
- "Oh no, I'm just here for the party," they said. "I think it's terrible
- an appearance at the annual awards bash, but feels in no fit state to
- -known families at a fundraising bash on Thursday night for Learning
- Who was paying for the bash? The only clue was the name Asprey,
- Mail, always hosted the annual bash for the Scottish Labour front-

- popular. Their method is to bash sense into criminals with a short,
- just cut off people's heads and bash their brains out over the floor,

What Good are Word Senses?

What Good are Word Senses?

- Semantics / Text understanding
  - Axioms about TRANSFER apply to (some tokens of) `throw`
  - Axioms about BUILDING apply to (some tokens of) `bank`

What Good are Word Senses?

- Semantics / Text understanding
  - Axioms about TRANSFER apply to (some tokens of) `throw`
  - Axioms about BUILDING apply to (some tokens of) `bank`
- Machine translation

What Good are Word Senses?

- Semantics / Text understanding
  - Axioms about TRANSFER apply to (some tokens of) `throw`
  - Axioms about BUILDING apply to (some tokens of) `bank`
- Machine translation
- Info retrieval / Question answering / Text categ.
  - Query or pattern might not match document exactly

What Good are Word Senses?

- Semantics / Text understanding
  - Axioms about TRANSFER apply to (some tokens of) `throw`
  - Axioms about BUILDING apply to (some tokens of) `bank`
- Machine translation
- Info retrieval / Question answering / Text categ.
  - Query or pattern might not match document exactly
- Backoff for just about anything
  - what word comes next? (speech recognition, language ID, …)
    - trigrams are sparse but tri-meanings might not be
  - bilexical PCFGs: p(S[`devour`] → NP[`lion`] VP[`devour`] | S[`devour`])
    - approximate by p(S[EAT] → NP[`lion`] VP[EAT] | S[EAT])

What Good are Word Senses?

- Semantics / Text understanding
  - Axioms about TRANSFER apply to (some tokens of) `throw`
  - Axioms about BUILDING apply to (some tokens of) `bank`
- Machine translation
- Info retrieval / Question answering / Text categ.
  - Query or pattern might not match document exactly
- Backoff for just about anything
  - what word comes next? (speech recognition, language ID, …)
    - trigrams are sparse but tri-meanings might not be
  - bilexical PCFGs: p(S[`devour`] → NP[`lion`] VP[`devour`] | S[`devour`])
    - approximate by p(S[EAT] → NP[`lion`] VP[EAT] | S[EAT])
- Speaker's real intention is senses; words are a noisy channel

Cues to Word Sense

Cues to Word Sense

- Adjacent words (or their senses)

Cues to Word Sense

- Adjacent words (or their senses)
- Grammatically related words (subject, object, …)

Cues to Word Sense

- Adjacent words (or their senses)
- Grammatically related words (subject, object, …)
- Other nearby words

Cues to Word Sense

- Adjacent words (or their senses)
- Grammatically related words (subject, object, …)
- Other nearby words
- Topic of document

Cues to Word Sense

- Adjacent words (or their senses)
- Grammatically related words (subject, object, …)
- Other nearby words
- Topic of document
- Sense of other tokens of the word in the same document

Word Classes by Tagging

- Every tag is a kind of class
- Tagger assigns a class to each word token

Word Classes by Tagging

- Every tag is a kind of class
- Tagger assigns a class to each word token

Start PN Verb Det Noun Prep Noun Prep Noun **Stop**

Bill directed a cortege of autos throu

Word Classes by Tagging

- Every tag is a kind of class
- Tagger assigns a class to each word token

0.4 0.6

Start PN Verb Det Noun Prep Noun Prep Noun **Stop**

0.001

Bill directed a cortege of autos throu

Word Classes by Tagging

- Every tag is a kind of class
- Tagger assigns a class to each word token

probs
from tag
bigram
model

0.4 0.6

Start PN Verb Det Noun Prep Noun Prep Noun Stop

0.001

Bill directed a cortege of autos throu

Word Classes by Tagging

- Every tag is a kind of class
- Tagger assigns a class to each word token

probs from tag bigram model

0.4 0.6

Start PN Verb Det Noun Prep Noun Prep Noun Stop

0.001

probs from unigram replacement

Bill directed a cortege of autos throu

Word Classes by Tagging

- Every tag is a kind of class
- Tagger assigns a class to each word token
  - Simultaneously groups and splits words
  - “party” gets split into N and V senses
  - “bash” gets split into N and V senses
  - {party/N, bash/N} vs. {party/V, bash/V}
  - What good are these groupings?

Learning Word Classes

- Every tag is a kind of class
- Tagger assigns a class to each word token
  - {party/N, bash/N} vs. {party/V, bash/V}
  - What good are these groupings?
  - Good for predicting next word or its class!

- Role of forward-backward algorithm?
  - It adjusts classes etc. in order to predict sequence of words better (with lower perplexity)

Words as Vectors

- Represent each word **type** w by a point in k-dimensional space
  - e.g., k is size of vocabulary
  - the 17^{th} coordinate of w represents **strength** of w's association with vocabulary word 17

Words as Vectors

- Represent each word **type** w by a point in k-dimensional space

  = party

  - e.g., k is size of vocabulary
  - the 17th coordinate of w represents **strength** of w's association with vocabulary word 17

Words as Vectors

- Represent each word **type** w by a point in k-dimensional space
  - e.g., k is size of vocabulary
  - the 17th coordinate of w represents **strength** of w's association with vocabulary word 17

= party

(0, 0, 3, 1, 0, 7, . . . 1, 0)

Words as Vectors

- Represent each word **type** w by a point in k-dimensional space
  = party
  - e.g., k is size of vocabulary
  - the 17th coordinate of w represents **strength** of w's association with vocabulary word 17

aardvark	abacus	abandoned	abbot	abduct	above		zygote	zymurgy
(0,	0,	3,	1,	0,	7,	. . .	1,	0)

Words as Vectors

- Represent each word **type** w by a point in k-dimensional space
  = **party**
  - e.g., k is size of vocabulary
  - the 17th coordinate of w represents **strength** of w's association with vocabulary word 17

aardvark	abacus	abandoned	abbot	abduct	above		zygote	zymurgy
(0,	0,	3,	1,	0,	7,	. . .	1,	0)

From corpus:

Arlen Specter **abandoned** the Republican **party**.
There were lots of **abbots** and nuns dancing at that **party**.
The **party** **above** the art gallery was, **above** all, a laboratory for synthesizing **zygotes** and beer.

Words as Vectors

- Represent each word **type** w by a point in k-dimensional space
  - e.g., k is size of vocabulary
  - the 17^{th} coordinate of w represents **strength** of w's association with vocabulary word 17

= party

aardvark	abacus	abandoned	abbot	abduct	above		zygote	zymurgy
(0,	0,	3,	1,	0,	7,	. . .	1,	0)

count too high (too influential)

From corpus:

Arlen Specter **abandoned** the Republican **party**.
There were lots of **abbots** and nuns dancing at that **party**.
The **party** **above** the art gallery was, **above** all, a laboratory for synthesizing **zygotes** and beer.

Words as Vectors

- Represent each word **type** w by a point in k-dimensional space
  - e.g., k is size of vocabulary
  - the 17th coordinate of w represents **strength** of w's association with vocabulary word 17

= party

aardvark	abacus	abandoned	abbot	abduct	above		zygote	zymurgy
(0,	0,	3,	1,	0,	7,	. . .	1,	0)

count too high (too influential)

count too low

From corpus:

Arlen Specter **abandoned** the Republican **party**.
There were lots of **abbots** and nuns dancing at that **party**.
The **party** **above** the art gallery was, **above** all, a laboratory for synthesizing **zygotes** and beer.

Words as Vectors

- Represent each word **type** w by a point in k-dimensional space
  = party
  - e.g., k is size of vocabulary
  - the 17th coordinate of w represents **strength** of w's association with vocabulary word 17

aardvark	abacus	abandoned	abbot	abduct	above		zygote	zymurgy
(0,	0,	3,	1,	0,	7,	. . .	1,	0)

Words as Vectors

- Represent each word **type** w by a point in k-dimensional space
  - e.g., k is size of vocabulary
  - the 17th coordinate of w represents **strength** of w's association with vocabulary word 17

= party

how might you measure this?

aardvark	abacus	abandoned	abbot	abduct	above		zygote	zymurgy
(0,	0,	3,	1,	0,	7,	. . .	1,	0)

Words as Vectors

- Represent each word **type** w by a point in k-dimensional space
  = party
  - e.g., k is size of vocabulary
  - the 17th coordinate of w represents **strength** of w's association with vocabulary word 17

aardvark abacus abandoned abbot abduct above zygote zymurgy

(0, 0, 3, 1, 0, 7, . . . 1, 0)

how might you measure this?

- how often words appear next to each other

Words as Vectors

- Represent each word **type** w by a point in k-dimensional space
  = party
  - e.g., k is size of vocabulary
  - the 17th coordinate of w represents **strength** of w's association with vocabulary word 17

how might you measure this?

aardvark	abacus	abandoned	abbot	abduct	above		zygote	zymurgy
(0,	0,	3,	1,	0,	7,	. . .	1,	0)

- how often words appear next to each other
- how often words appear near each other

Words as Vectors

- Represent each word **type** w by a point in k-dimensional space
  = party
  - e.g., k is size of vocabulary
  - the 17th coordinate of w represents **strength** of w's association with vocabulary word 17

aardvark	abacus	abandoned	abbot	abduct	above		zygote	zymurgy
(0,	0,	3,	1,	0,	7,	. . .	1,	0)

how might you measure this?

- how often words appear next to each other
- how often words appear near each other
- how often words are syntactically linked

Words as Vectors

- Represent each word **type** w by a point in k-dimensional space
  = party
  - e.g., k is size of vocabulary
  - the 17th coordinate of w represents **strength** of w's association with vocabulary word 17

how might you measure this?

aardvark	abacus	abandoned	abbot	abduct	above		zygote	zymurgy
(0,	0,	3,	1,	0,	7,	. . .	1,	0)

- how often words appear next to each other
- how often words appear near each other
- how often words are syntactically linked
- should correct for commonness of word (e.g., "above")

Words as Vectors

- Represent each word **type** w by a point in k-dimensional space
  - e.g., k is size of vocabulary
  - the 17th coordinate of w represents **strength** of w's association with vocabulary word 17

aardvark abacus abandoned abbot abduct above zygote zymurgy

(0, 0, 3, 1, 0, 7, . . . 1, 0)

Words as Vectors

- Represent each word **type** w by a point in k-dimensional space
  - e.g., k is size of vocabulary
  - the 17^{th} coordinate of w represents **strength** of w's association with vocabulary word 17

aardvark abacus abandoned abbot abduct above zygote zymurgy

(0, 0, 3, 1, 0, 7, . . . 1, 0)

- Plot all word types in k-dimensional space

Words as Vectors

- Represent each word **type** w by a point in k-dimensional space
  - e.g., k is size of vocabulary
  - the 17th coordinate of w represents **strength** of w's association with vocabulary word 17

aardvark abacus abandoned abbot abduct above ... zygote zymurgy

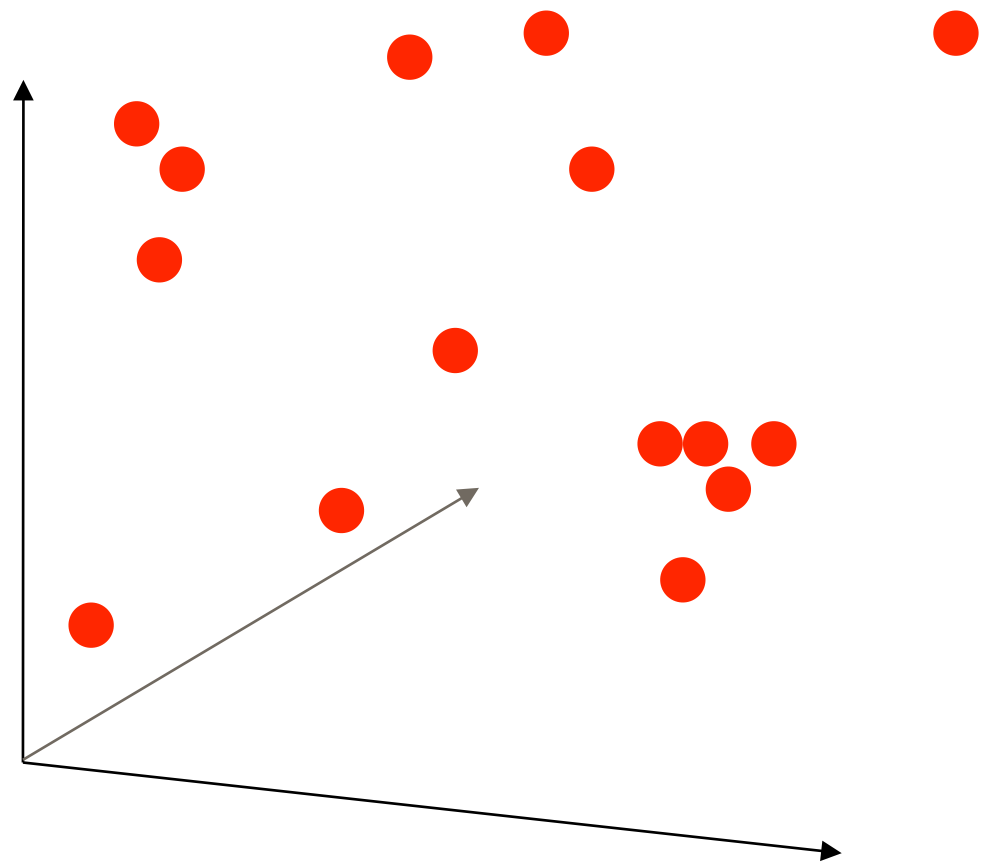
(0, 0, 3, 1, 0, 7, . . . 1, 0)

- Plot all word types in k-dimensional space
- Look for **clusters** of close-together types

Learning Classes by Clustering

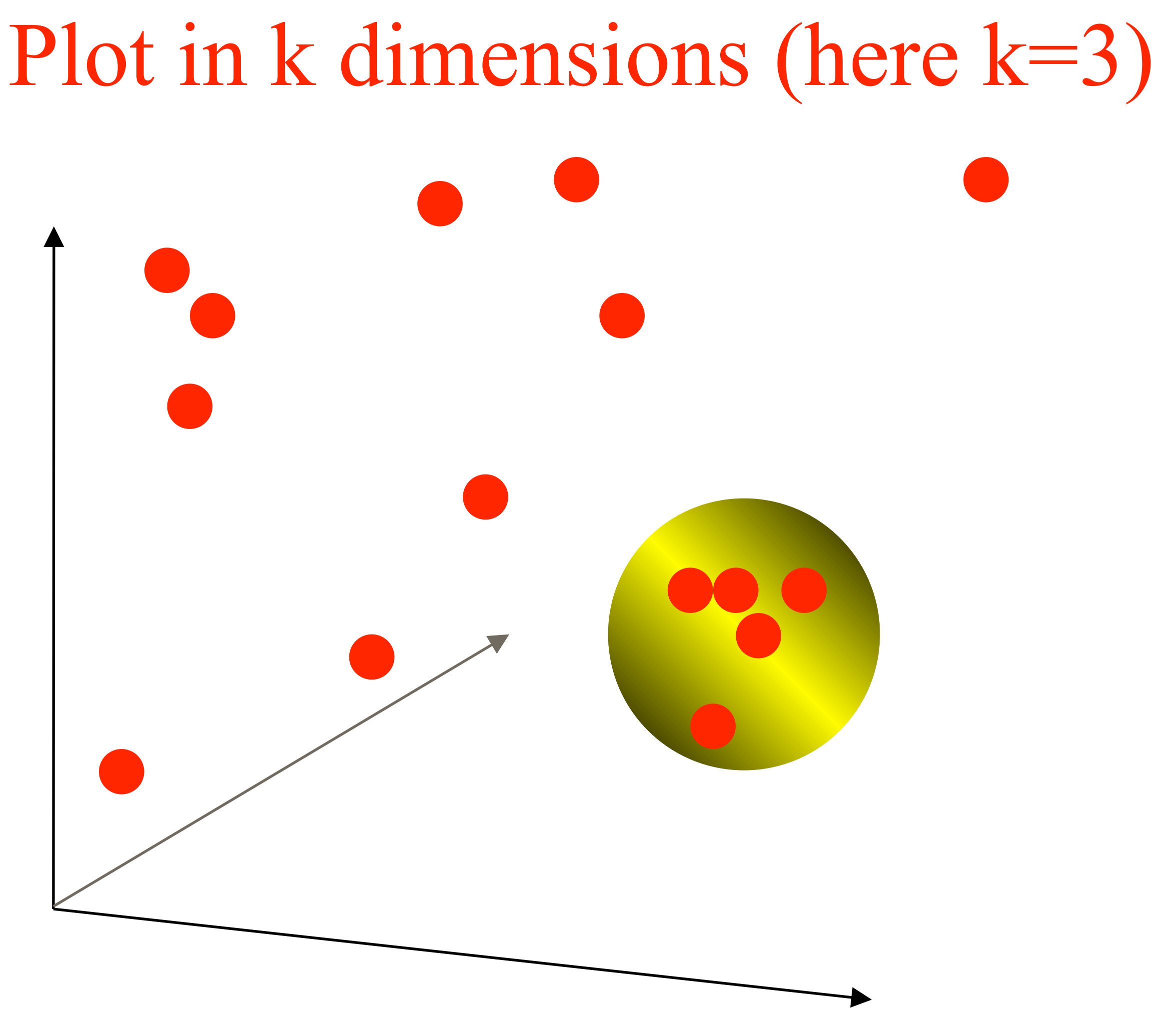
- Plot all word types in k-dimensional space
- Look for **clusters** of close-together types

Plot in k dimensions (here k=3)



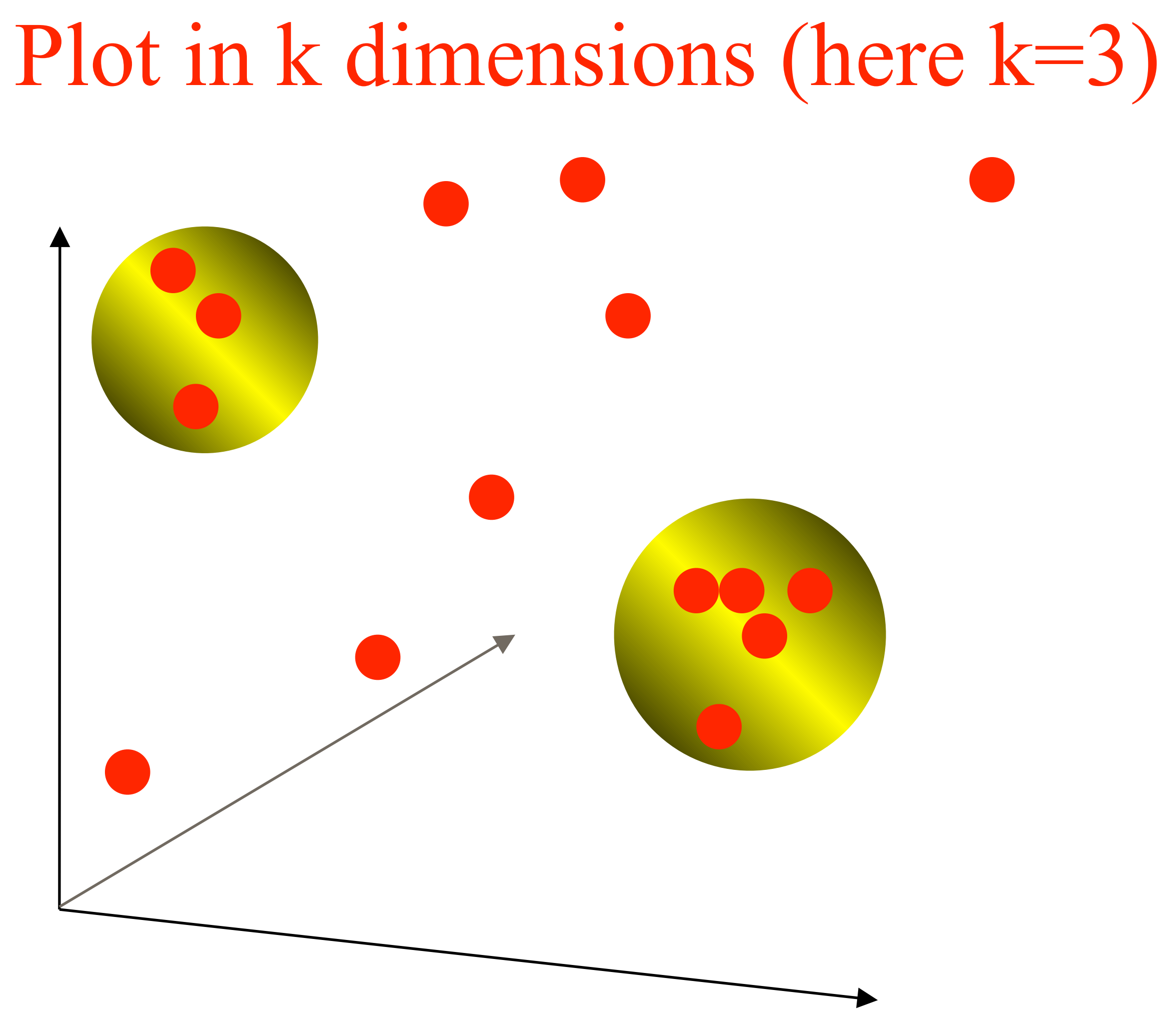
Learning Classes by Clustering

- Plot all word types in k-dimensional space
- Look for **clusters** of close-together types



Learning Classes by Clustering

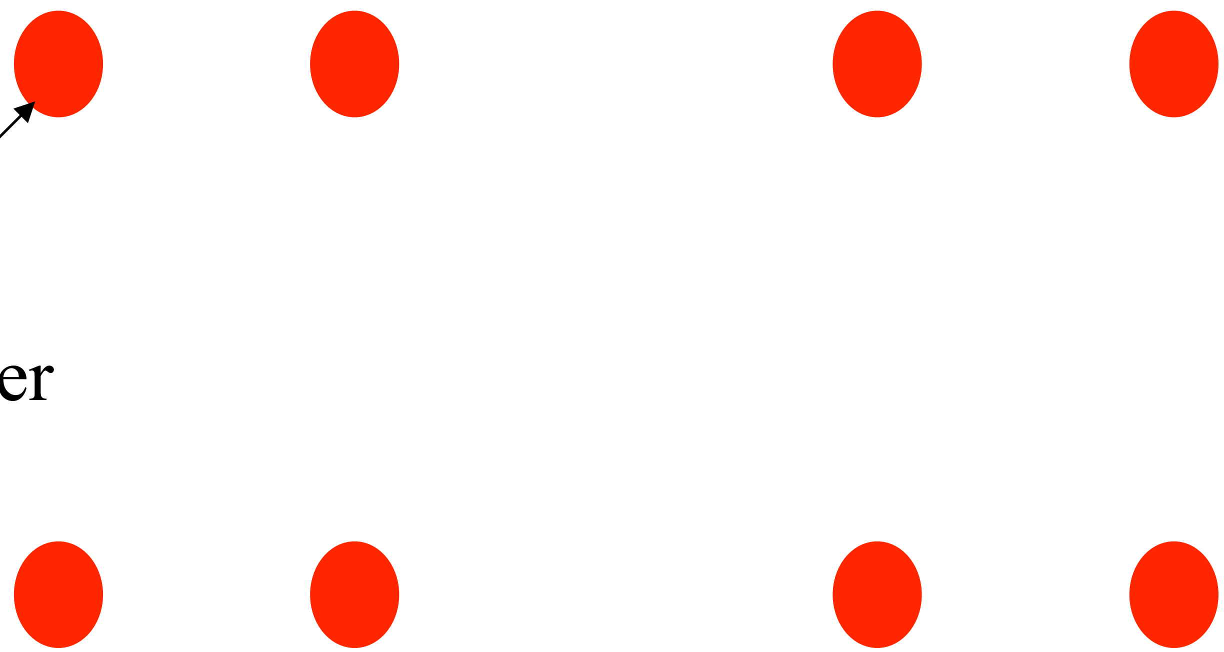
- Plot all word types in k-dimensional space
- Look for **clusters** of close-together types



Bottom-Up Clustering

- Start with one cluster per point
- Repeatedly merge 2 closest clusters
  - Single-link: dist(A,B) = min dist(a,b) for $a \in A$, $b \in B$
  - Complete-link: dist(A,B) = max dist(a,b) for $a \in A$, $b \in B$

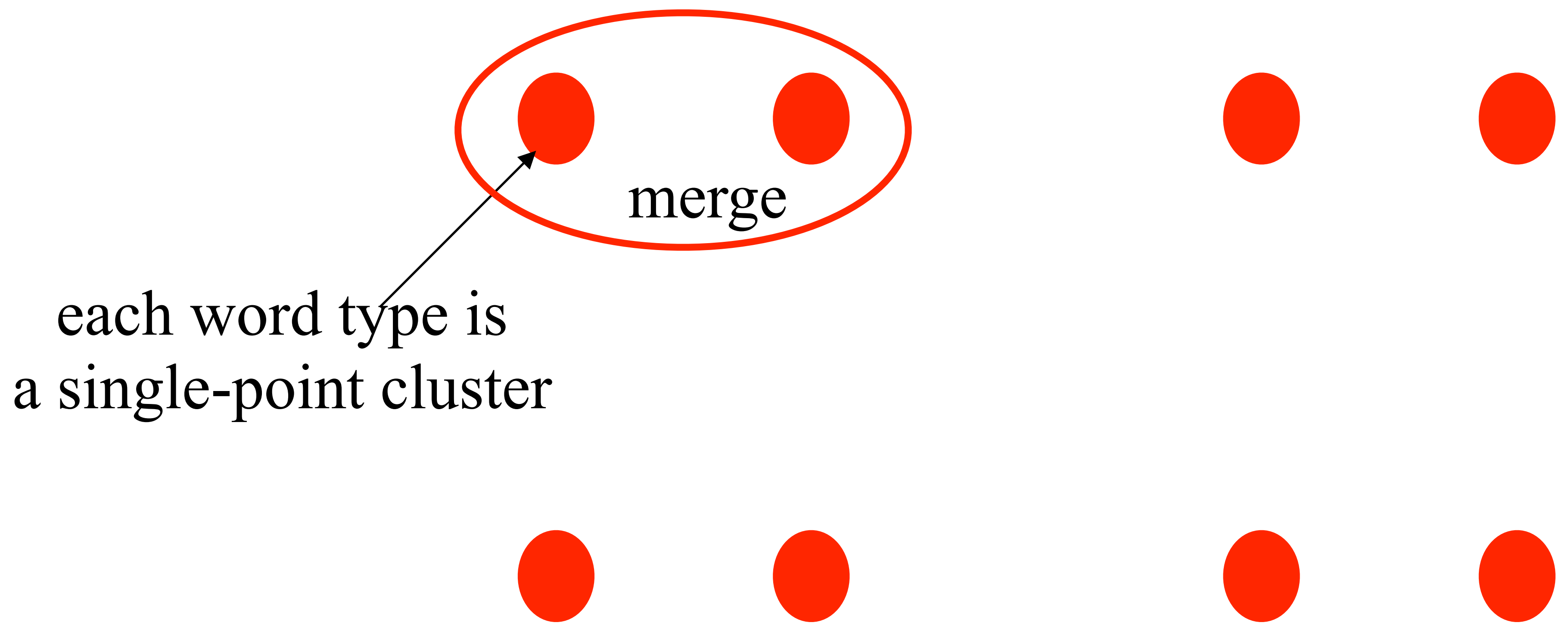
Bottom-Up Clustering – Single-Link



each word type is
a single-point cluster

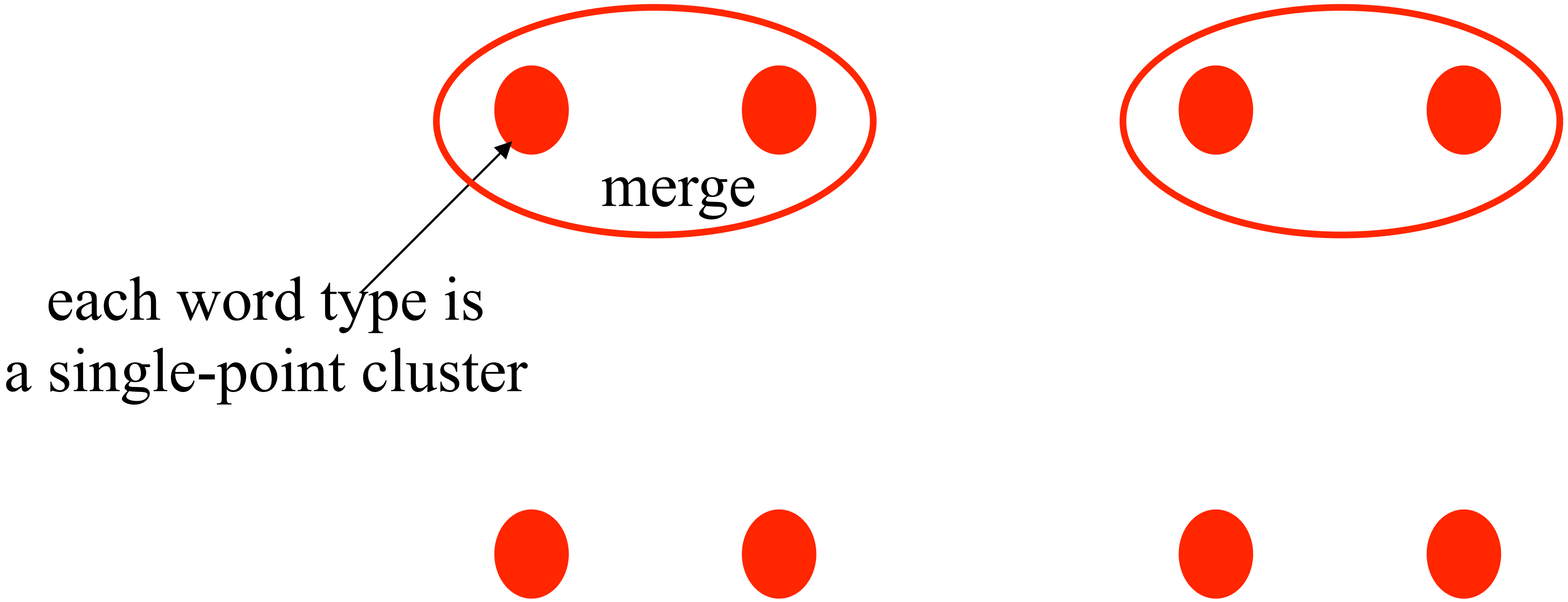
example from Manning & Schütze

Bottom-Up Clustering – Single-Link



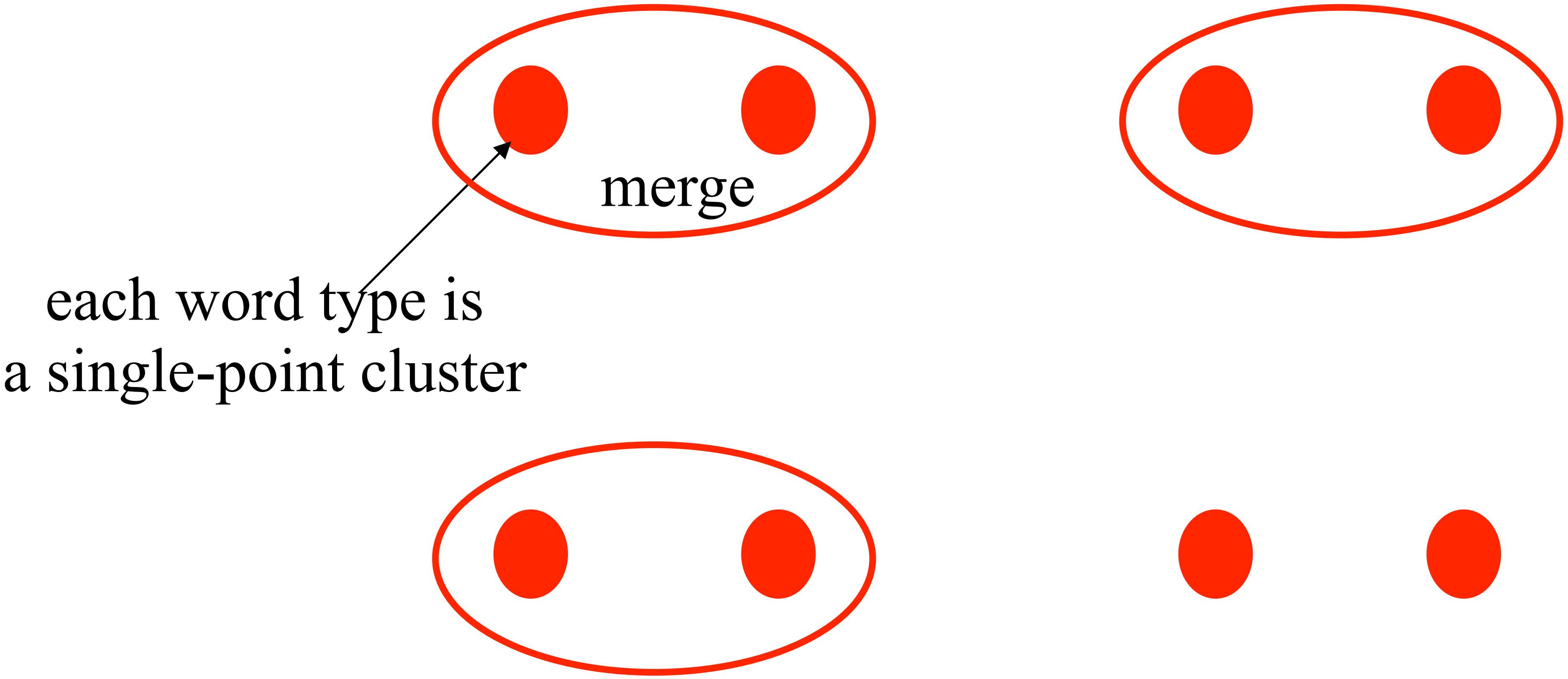
example from Manning & Schütze

Bottom-Up Clustering – Single-Link

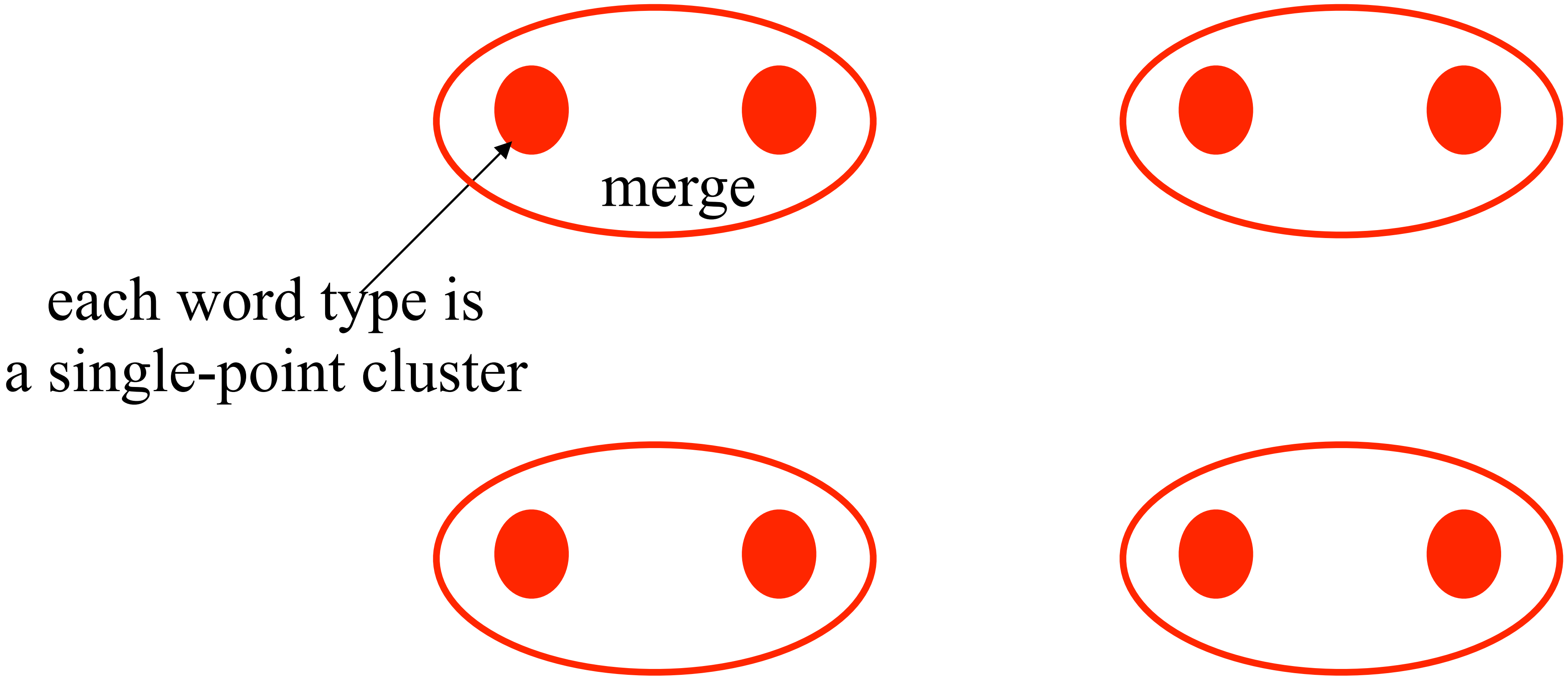


example from Manning & Schütze

Bottom-Up Clustering – Single-Link

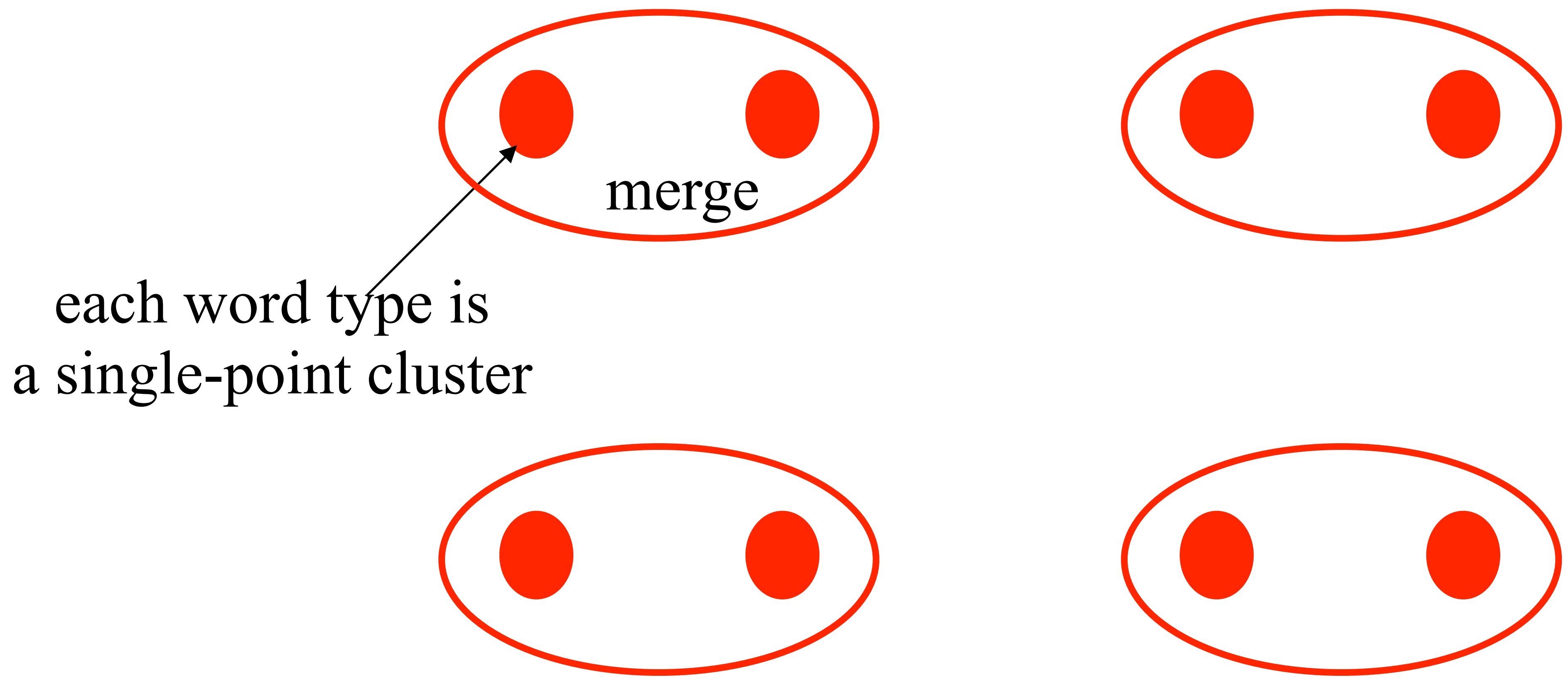


Bottom-Up Clustering – Single-Link



example from Manning & Schütze

Bottom-Up Clustering – Single-Link

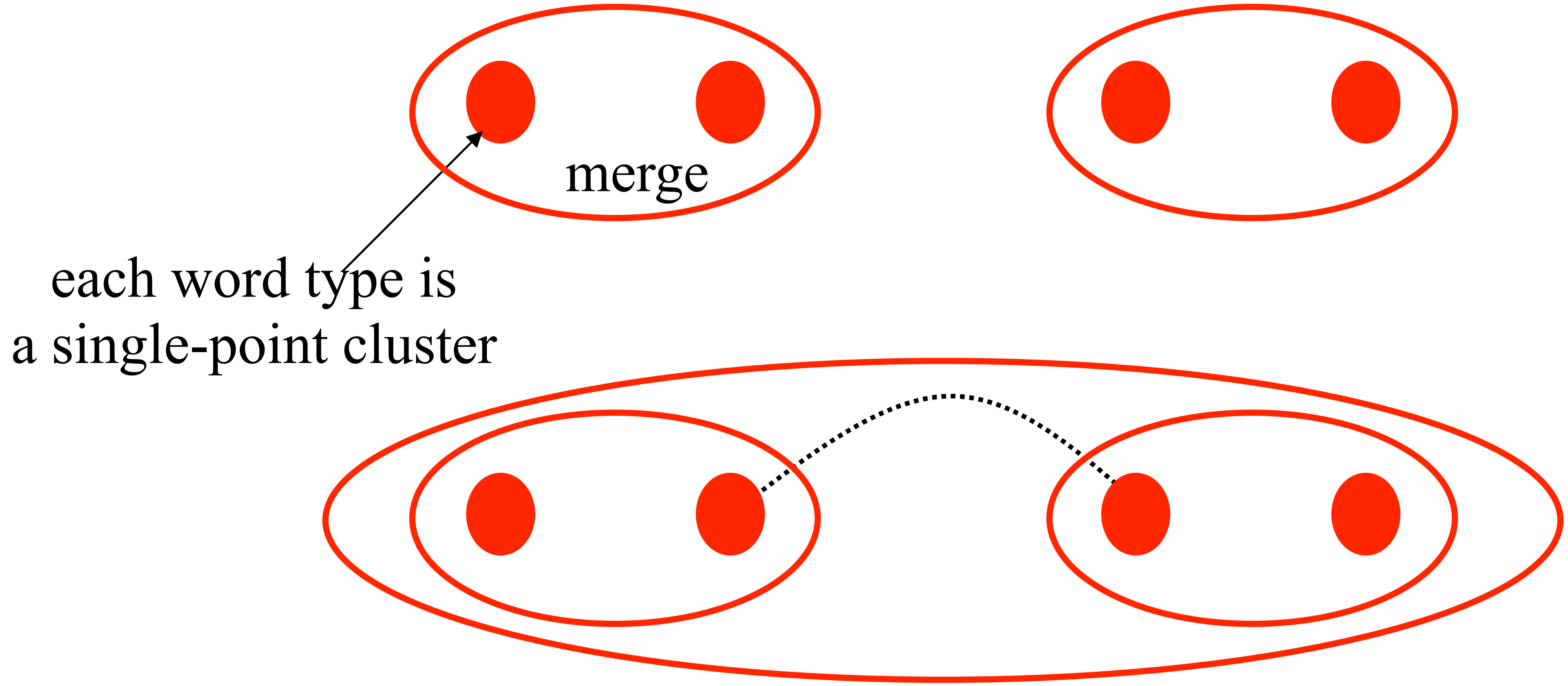


Again, merge closest pair of clusters:

Single-link: clusters are close if any of their points are

dist(A,B) = min dist(a,b) for a∈A, b∈B

Bottom-Up Clustering – Single-Link



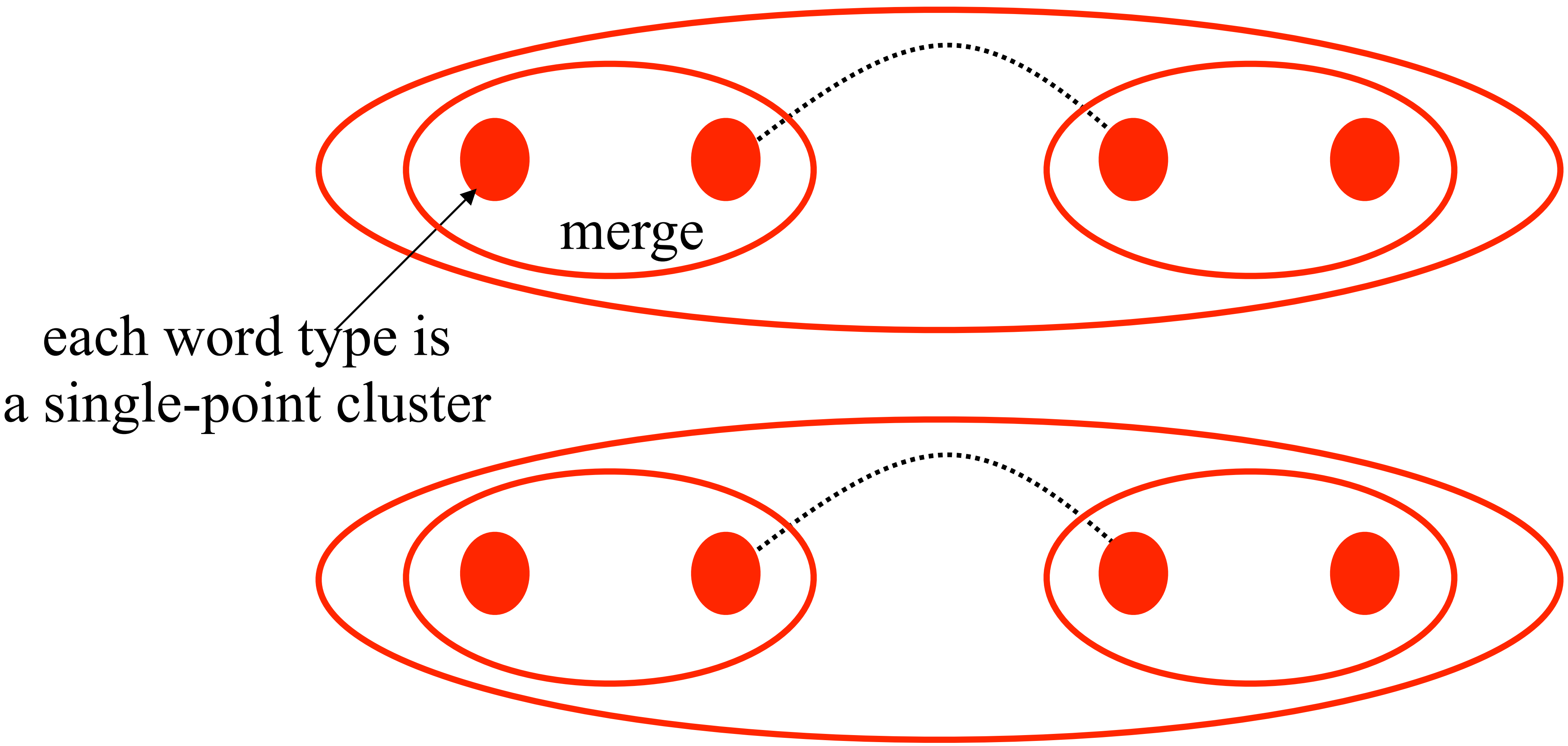
Again, merge closest pair of clusters:

Single-link: clusters are close if any of their points are

dist(A,B) = min dist(a,b) for a∈A, b∈B

example from Manning & Schütze

Bottom-Up Clustering – Single-Link



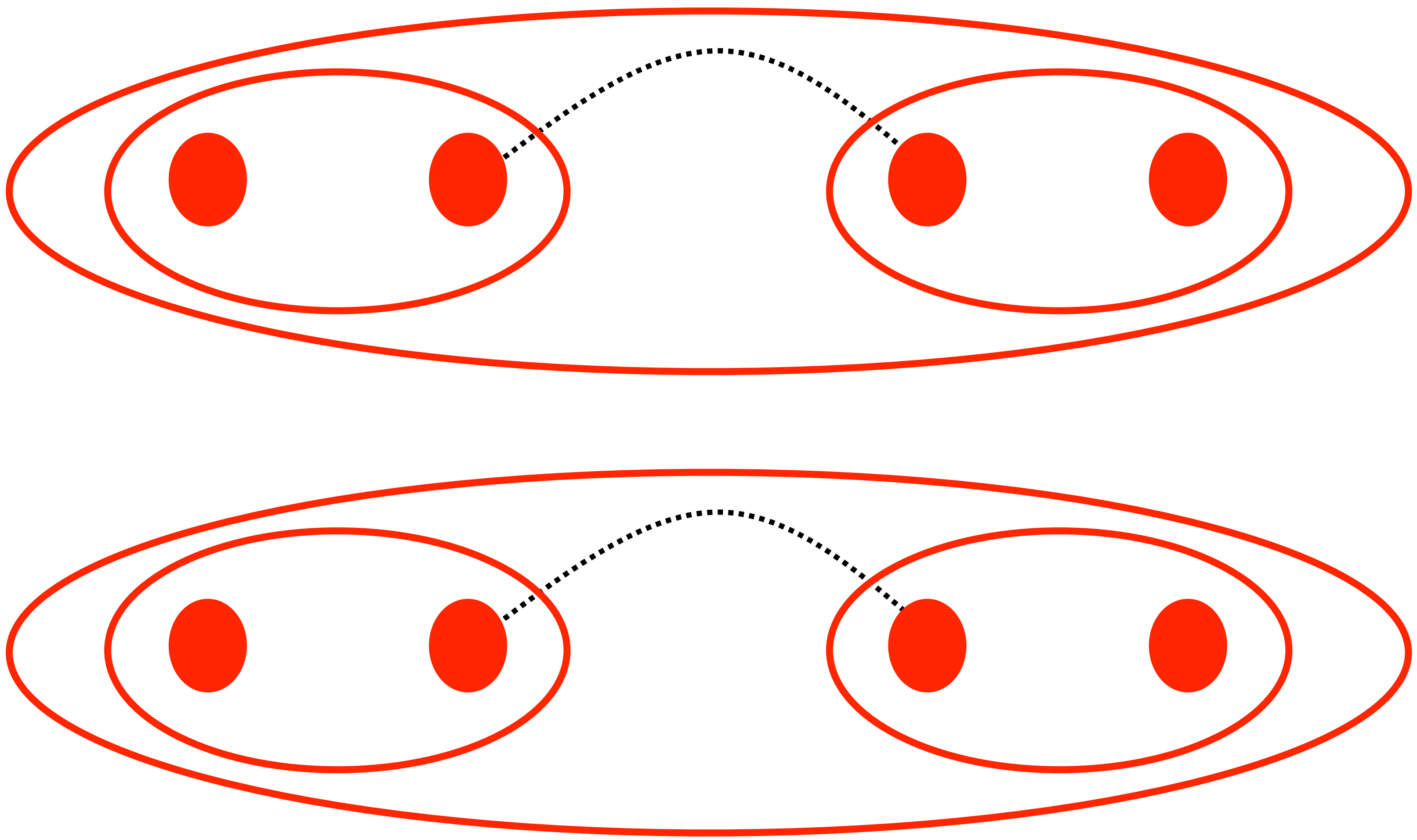
Again, merge closest pair of clusters:

Single-link: clusters are close if any of their points are

$\text{dist}(A,B) = \min \text{dist}(a,b)$ for $a \in A$, $b \in B$

example from Manning & Schütze

Bottom-Up Clustering – Single-Link



Again, merge closest pair of clusters:

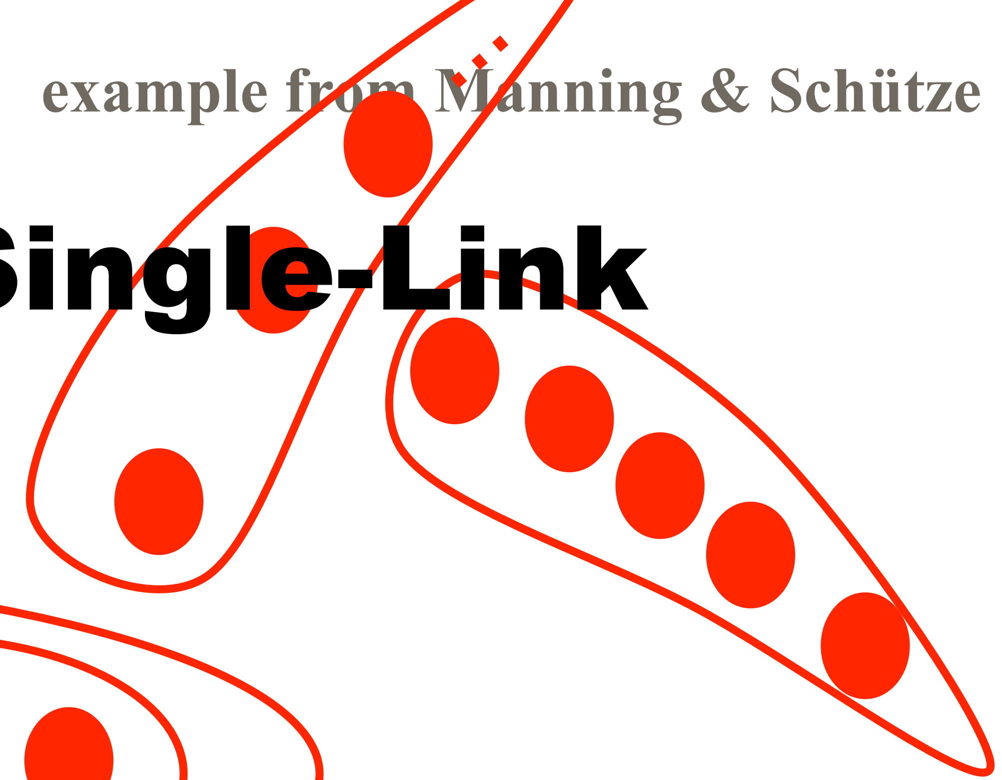
Single-link: clusters are close if any of their points are

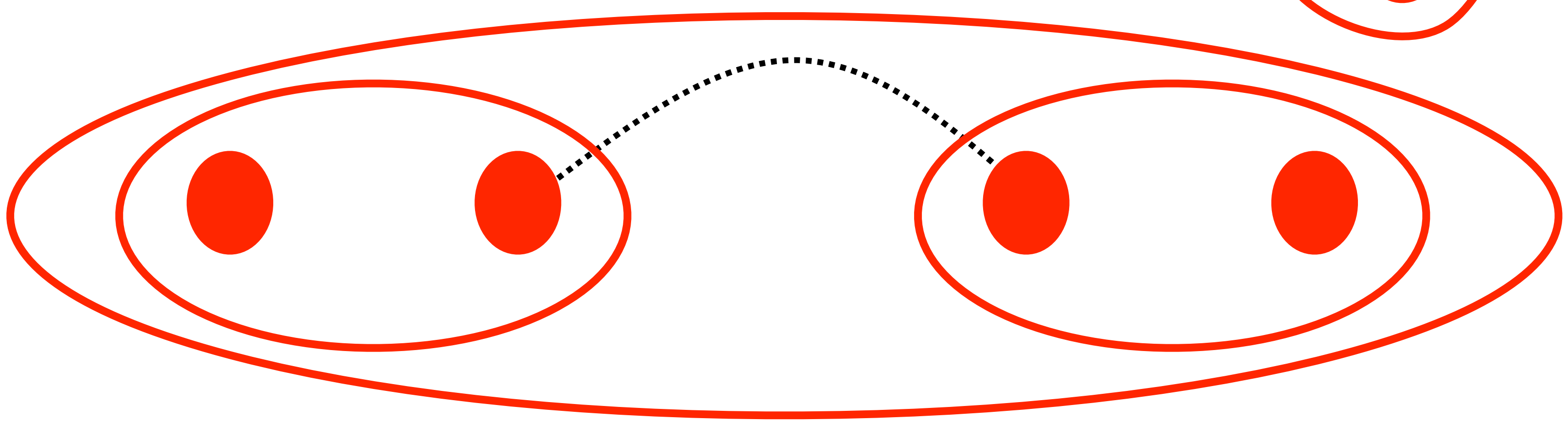
$\mathrm{dist}(A,B) = \min \mathrm{dist}(a,b)$ for $a \in A$, $b \in B$

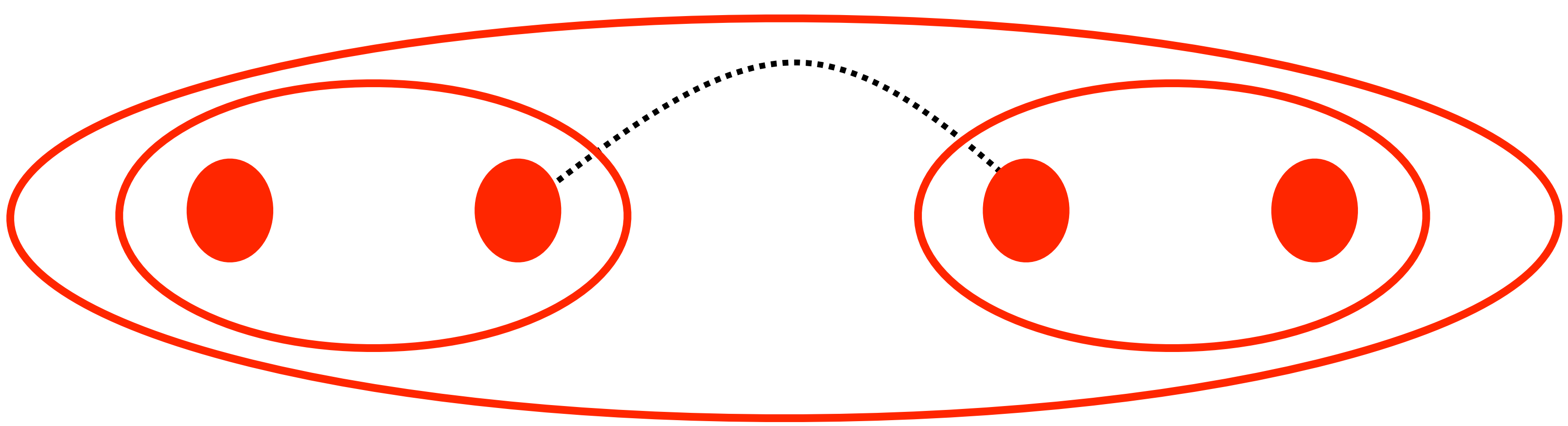
Fast, but tend to get long, stringy, meandering clusters

example from Manning & Schütze

Bottom-Up Clustering – Single-Link







Again, merge closest pair of clusters:

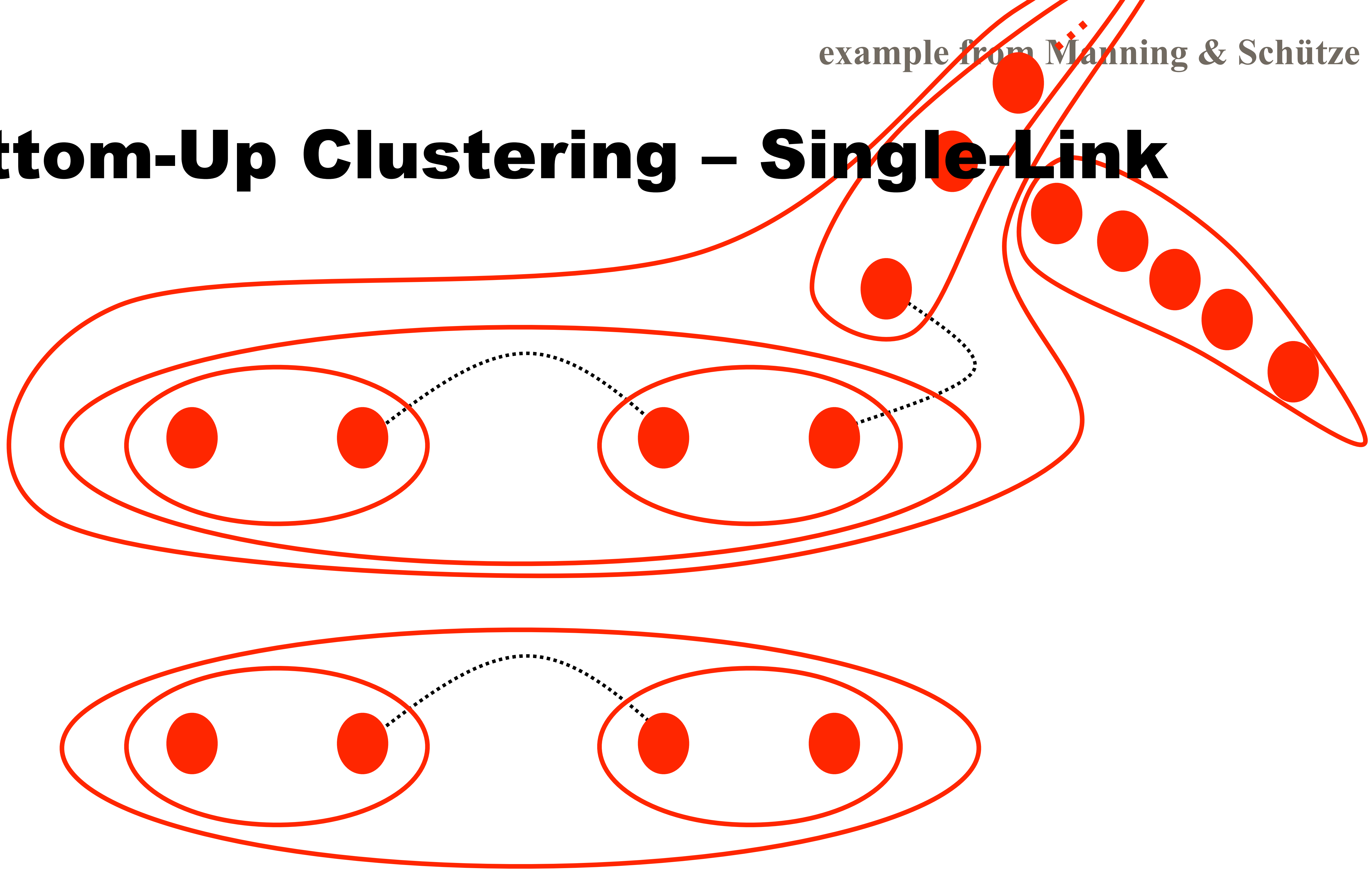
Single-link: clusters are close if any of their points are

dist(A,B) = min dist(a,b) for $a \in A$, $b \in B$

Fast, but tend to get long, stringy, meandering clusters

example from Manning & Schütze

Bottom-Up Clustering – Single-Link



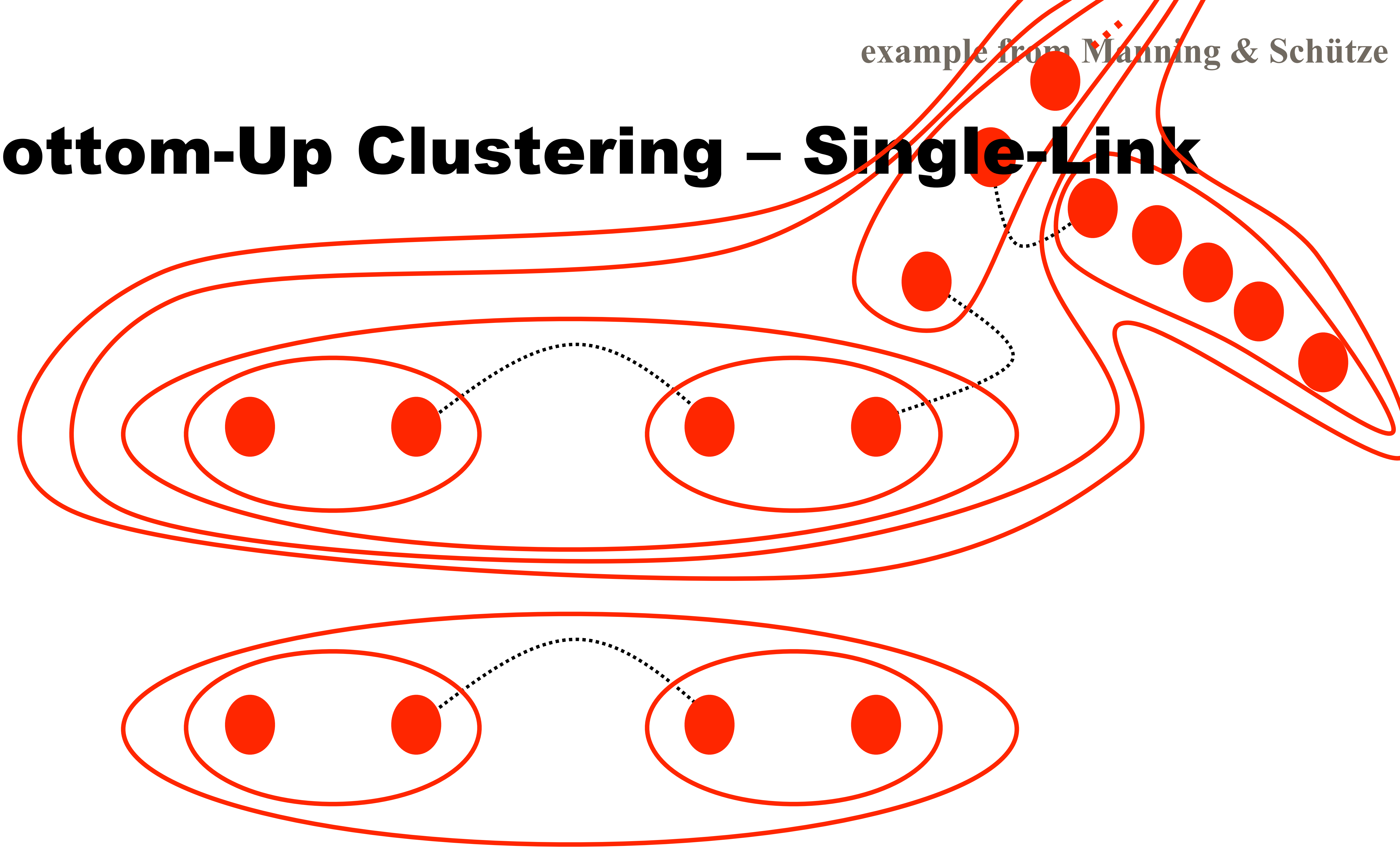
Again, merge closest pair of clusters:

Single-link: clusters are close if any of their points are

$\text{dist}(A,B) = \min \text{dist}(a,b)$ for $a \in A$, $b \in B$

Fast, but tend to get long, stringy, meandering clusters

Bottom-Up Clustering – Single-Link



Again, merge closest pair of clusters:

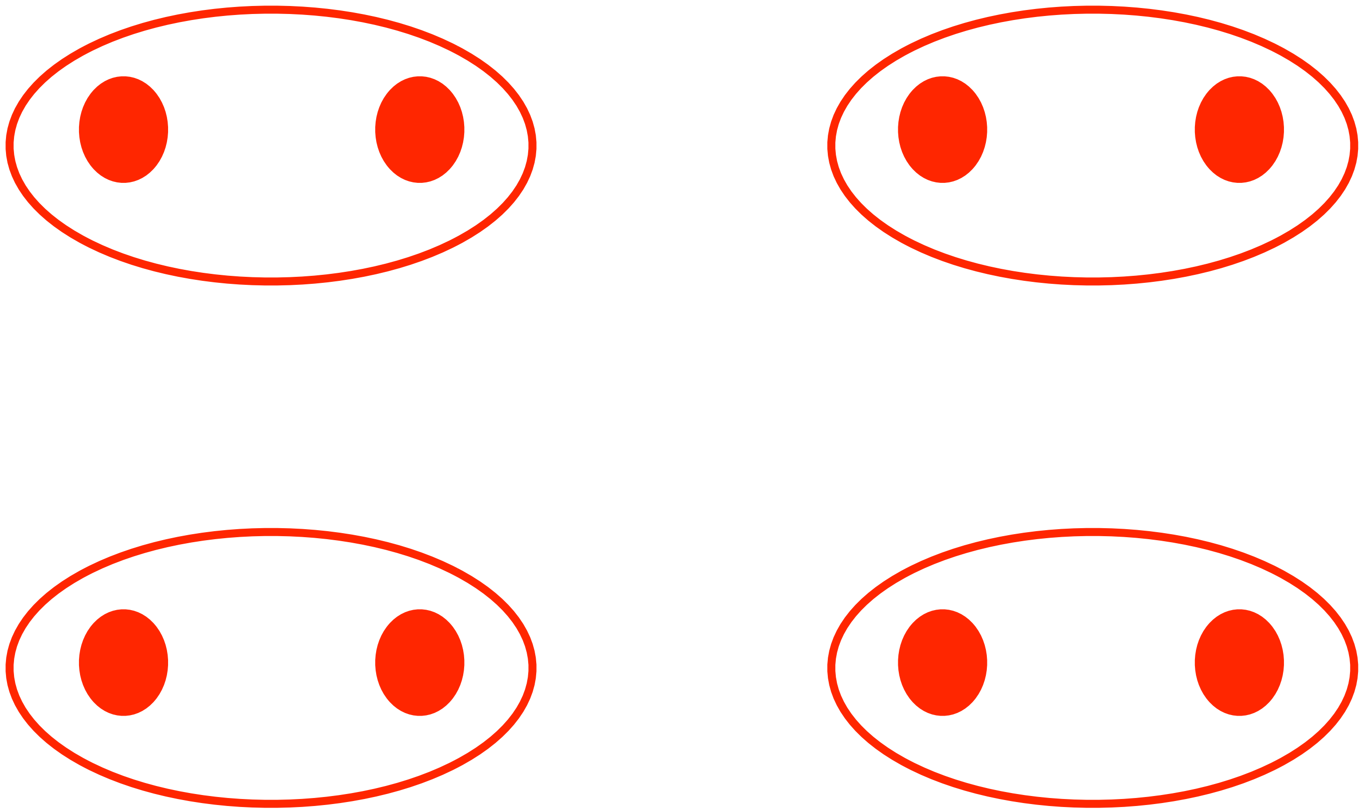
Single-link: clusters are close if any of their points are

dist(A,B) = min dist(a,b) for a∈A, b∈B

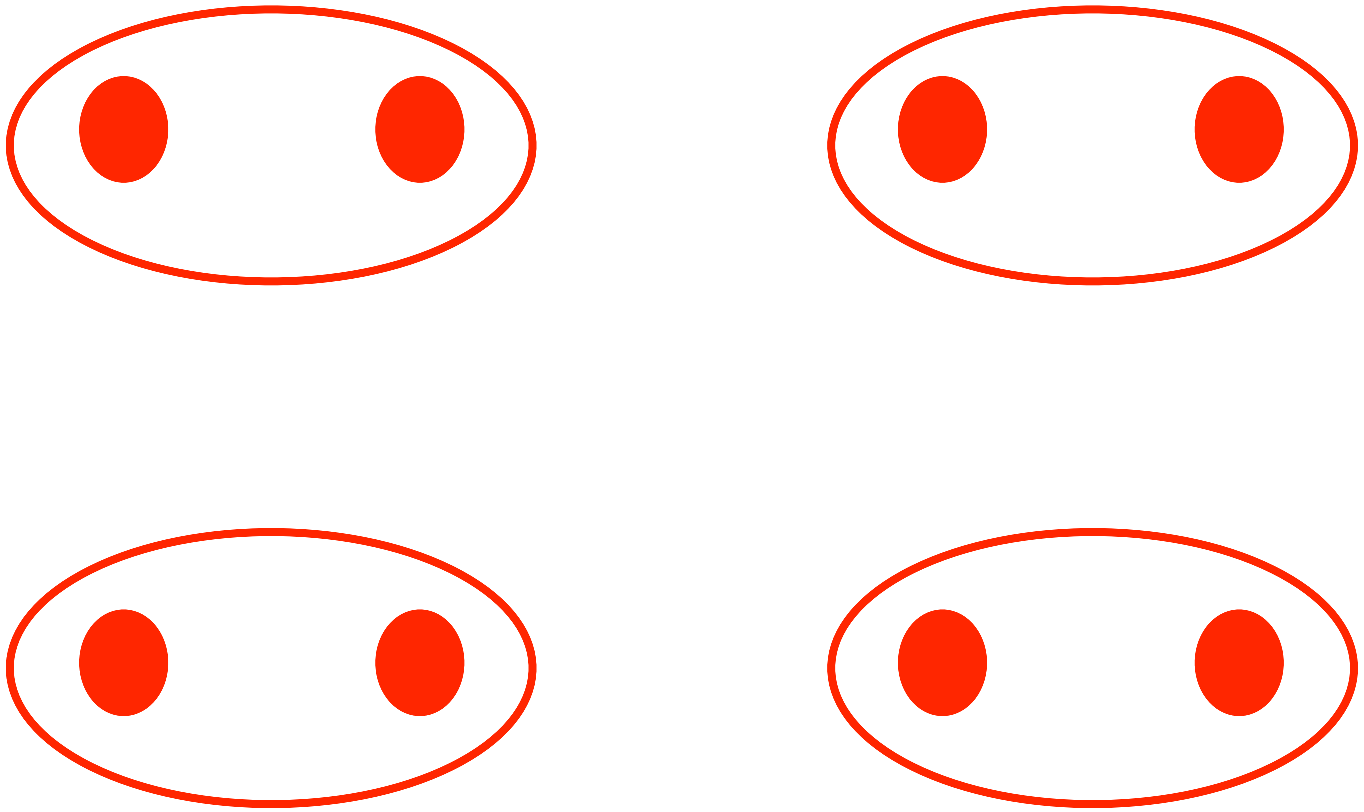
Fast, but tend to get long, stringy, meandering clusters

example from Manning & Schütze

Bottom-Up Clustering – Complete-Link



Bottom-Up Clustering – Complete-Link



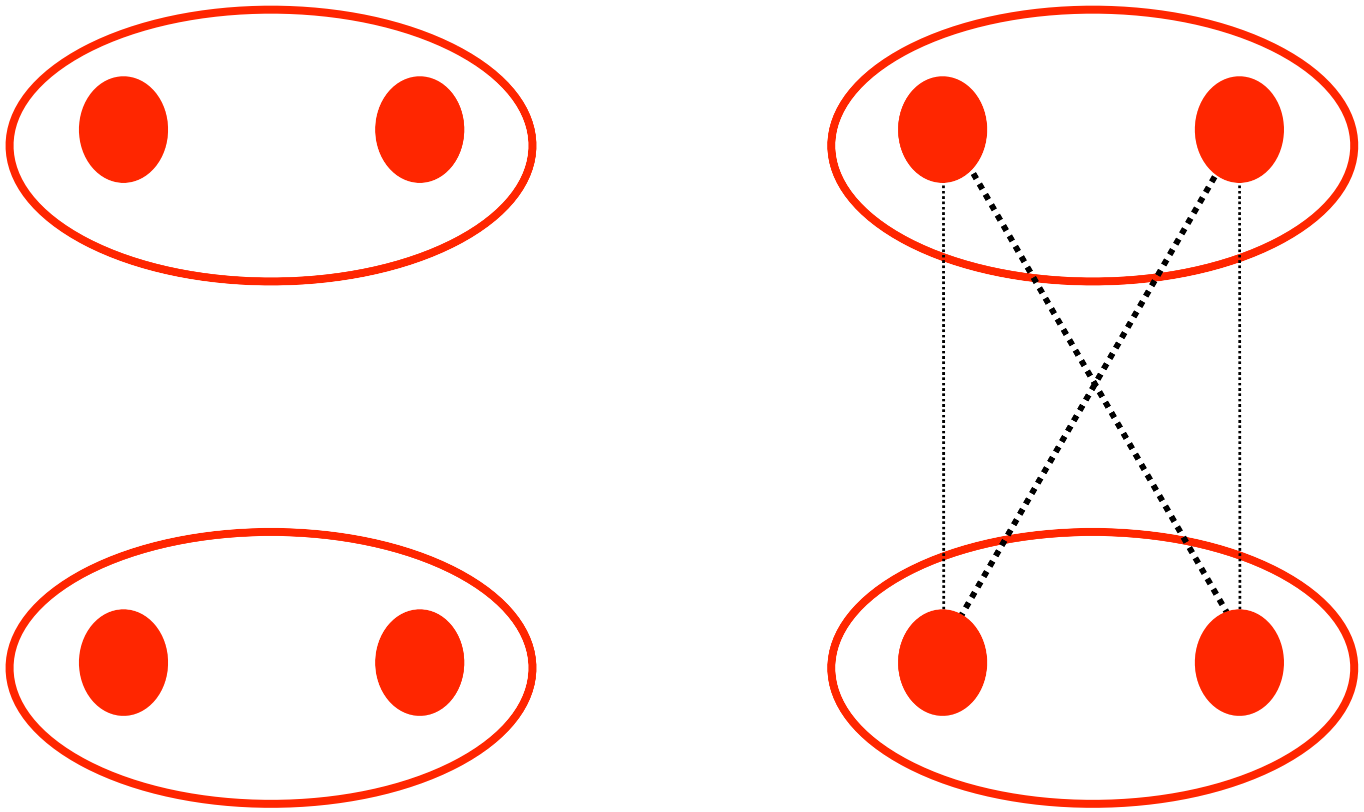
Again, merge closest pair of clusters:

Complete-link: clusters are close only if all of their points are

dist(A,B) = max dist(a,b) for $a \in A$, $b \in B$

example from Manning & Schütze

Bottom-Up Clustering – Complete-Link



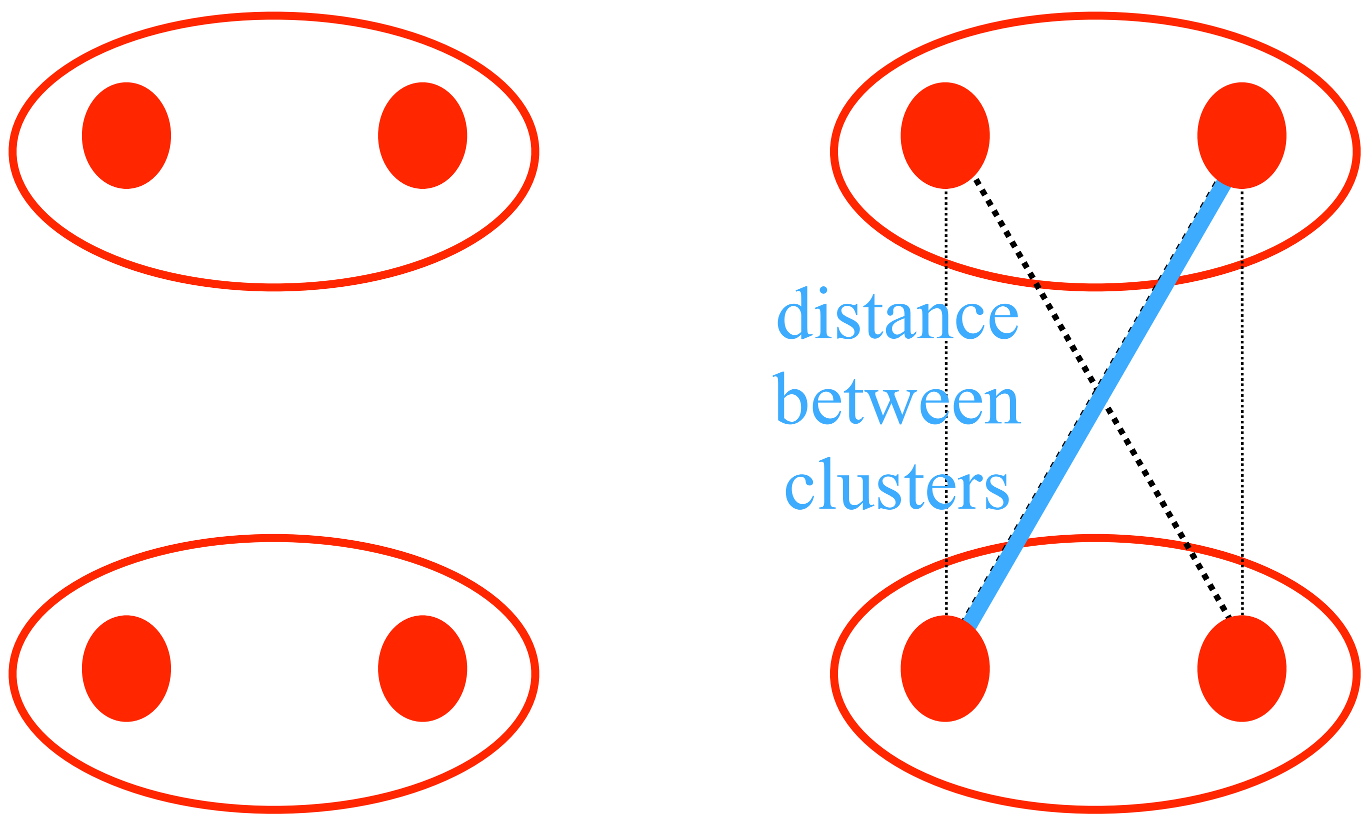
Again, merge closest pair of clusters:

Complete-link: clusters are close only if all of their points are

$\text{dist}(A,B) = \max \text{dist}(a,b)$ for $a \in A$, $b \in B$

example from Manning & Schütze

Bottom-Up Clustering – Complete-Link



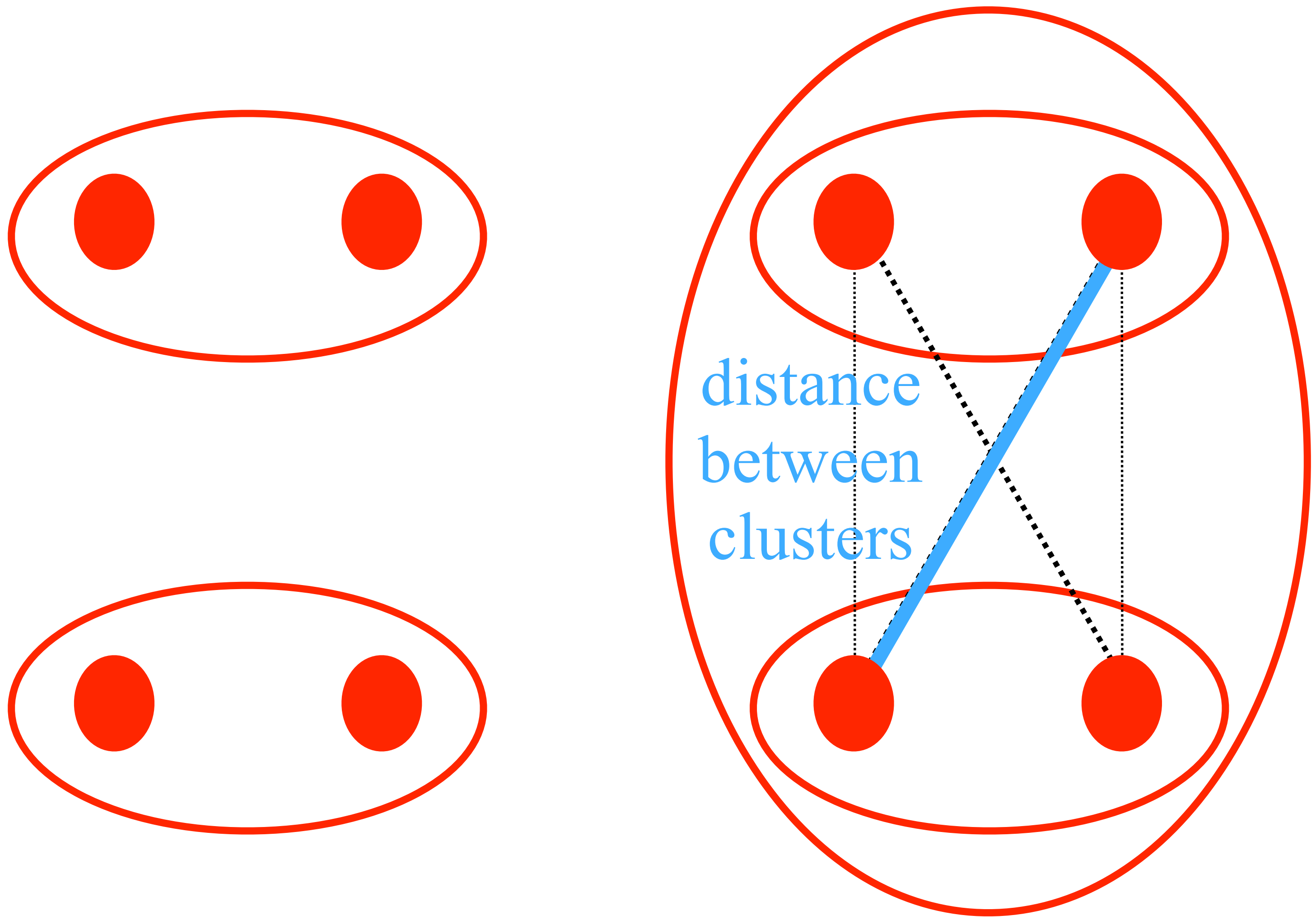
Again, merge closest pair of clusters:

Complete-link: clusters are close only if all of their points are

$\text{dist}(A,B) = \max \text{dist}(a,b)$ for $a \in A$, $b \in B$

example from Manning & Schütze

Bottom-Up Clustering – Complete-Link

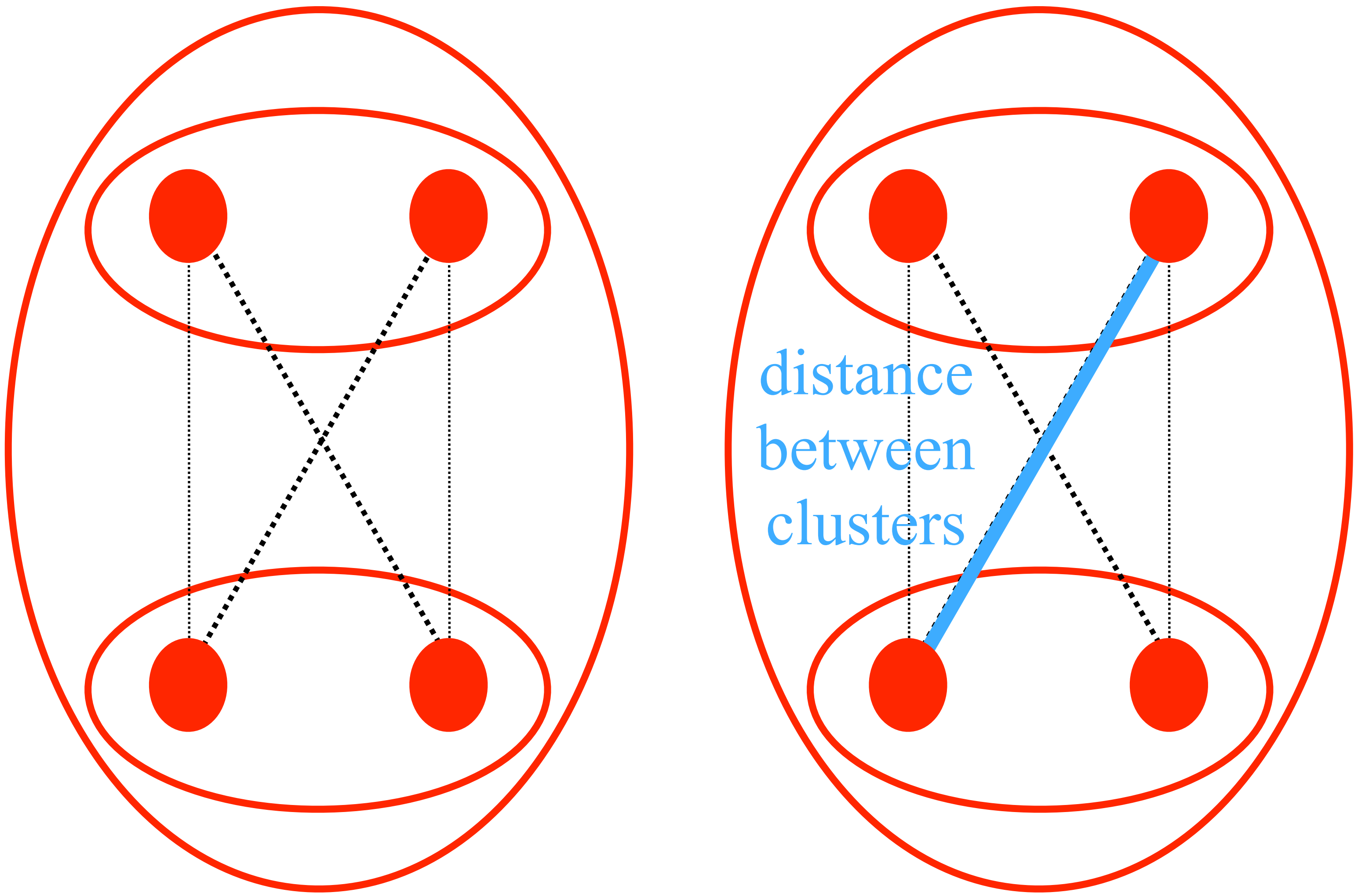


Again, merge closest pair of clusters:

Complete-link: clusters are close only if all of their points are

dist(A,B) = max dist(a,b) for $a \in A$, $b \in B$

Bottom-Up Clustering – Complete-Link

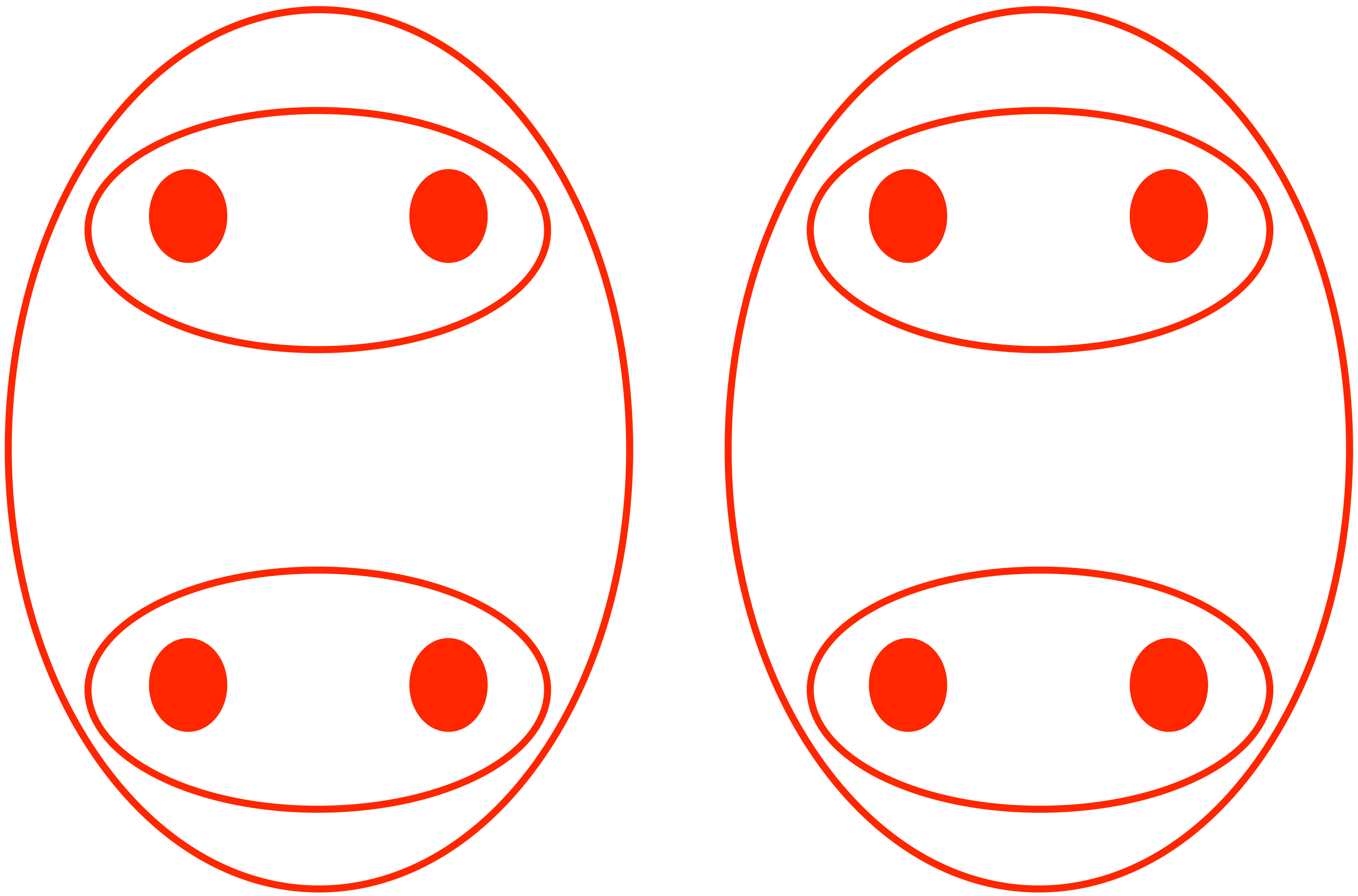


Again, merge closest pair of clusters:

Complete-link: clusters are close only if all of their points are

$\text{dist}(A,B) = \max \text{dist}(a,b)$ for $a \in A$, $b \in B$

Bottom-Up Clustering – Complete-Link



Again, merge closest pair of clusters:

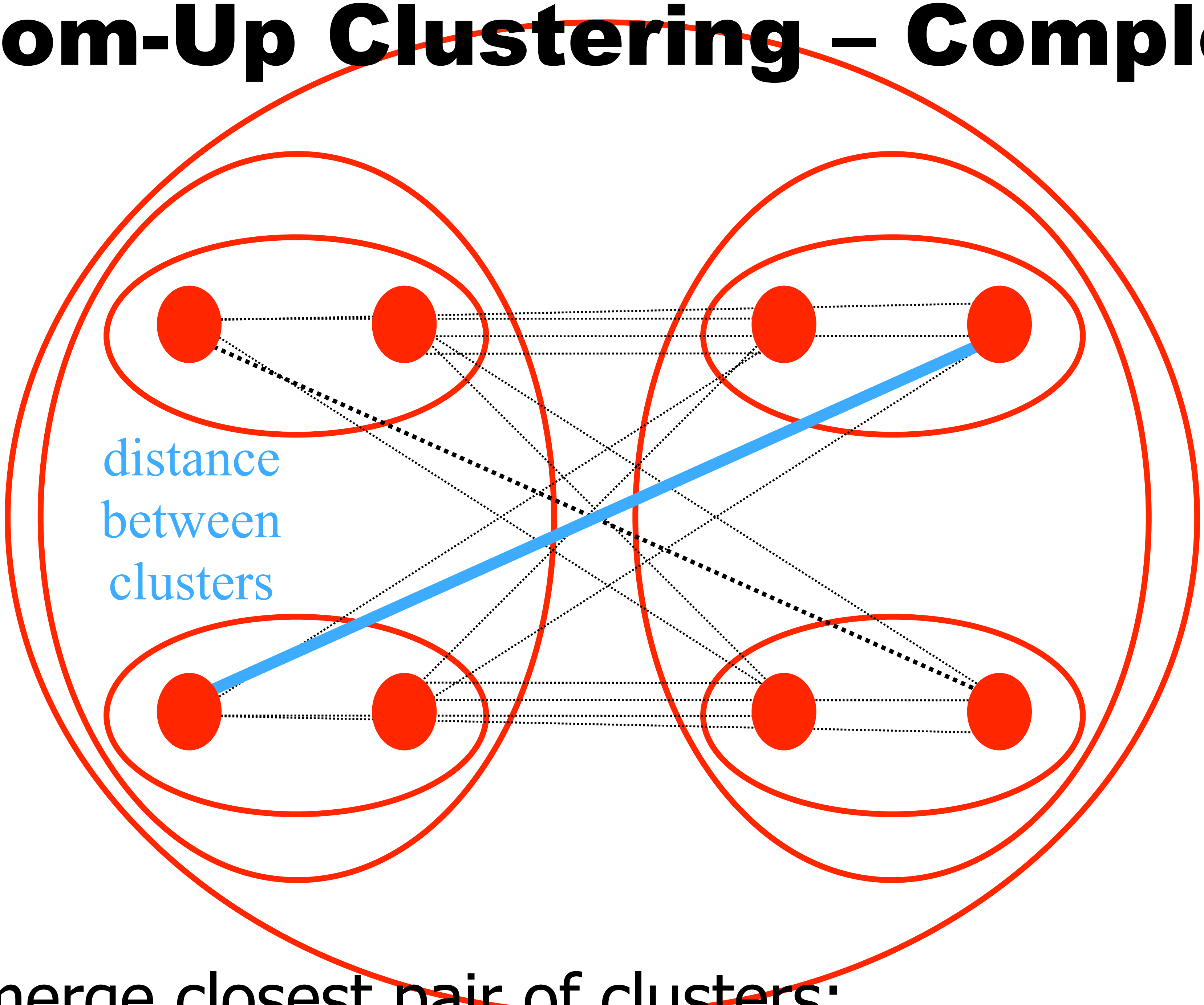
Complete-link: clusters are close only if all of their points are

$\text{dist}(A,B) = \max \text{dist}(a,b)$ for $a \in A$, $b \in B$

Slow to find closest pair – need quadratically many distances

example from Manning & Schütze

Bottom-Up Clustering – Complete-Link



Again, merge closest pair of clusters:

Complete-link: clusters are close only if all of their points are

$$\text{dist}(A,B) = \max \text{dist}(a,b) \text{ for } a \in A,\ b \in B$$

Slow to find closest pair – need quadratically many distances

Bottom-Up Clustering

- Start with one cluster per point
- Repeatedly merge 2 closest clusters
  - Single-link: dist(A,B) = min dist(a,b) for $a \in A$, $b \in B$
  - Complete-link: dist(A,B) = max dist(a,b) for $a \in A$, $b \in B$
    - too slow to update cluster distances after each merge; but ∃ alternatives!

Bottom-Up Clustering

- Start with one cluster per point
- Repeatedly merge 2 closest clusters
  - Single-link: dist(A,B) = min dist(a,b) for $a \in A$, $b \in B$
  - Complete-link: dist(A,B) = max dist(a,b) for $a \in A$, $b \in B$
    - too slow to update cluster distances after each merge; but $\exists$ alternatives!
  - Average-link: dist(A,B) = mean dist(a,b) for $a \in A$, $b \in B$
  - Centroid-link: dist(A,B) = dist(mean(A),mean(B))

Bottom-Up Clustering

- Start with one cluster per point
- Repeatedly merge 2 closest clusters
  - Single-link: dist(A,B) = min dist(a,b) for $a \in A$, $b \in B$
  - Complete-link: dist(A,B) = max dist(a,b) for $a \in A$, $b \in B$
    - too slow to update cluster distances after each merge; but ∃ alternatives!
  - Average-link: dist(A,B) = mean dist(a,b) for $a \in A$, $b \in B$
  - Centroid-link: dist(A,B) = dist(mean(A),mean(B))
- Stop when clusters are “big enough”
  - e.g., provide adequate support for backoff (on a development corpus)

Bottom-Up Clustering

- Start with one cluster per point
- Repeatedly merge 2 closest clusters
  - Single-link: dist(A,B) = min dist(a,b) for a∈A, b∈B
  - Complete-link: dist(A,B) = max dist(a,b) for a∈A, b∈B
    - too slow to update cluster distances after each merge; but ∃ alternatives!
  - Average-link: dist(A,B) = mean dist(a,b) for a∈A, b∈B
  - Centroid-link: dist(A,B) = dist(mean(A),mean(B))
- Stop when clusters are "big enough"
  - e.g., provide adequate support for backoff (on a development corpus)
- Some flexibility in defining dist(a,b)
  - Might not be Euclidean distance; e.g., use vector angle

EM Clustering (for k clusters)

EM Clustering (for k clusters)

- EM algorithm
  - Viterbi version – called "k-means clustering"
  - Full EM version – called "Gaussian mixtures"

EM Clustering (for k clusters)

- EM algorithm
  - Viterbi version – called "k-means clustering"
  - Full EM version – called "Gaussian mixtures"

EM Clustering (for k clusters)

- EM algorithm
  - Viterbi version – called “k-means clustering”
  - Full EM version – called “Gaussian mixtures”

- **Expectation step:** Use current parameters (and observations) to reconstruct hidden structure

EM Clustering (for k clusters)

- EM algorithm
  - Viterbi version – called “k-means clustering”
  - Full EM version – called “Gaussian mixtures”

- **Expectation step:** Use current parameters (and observations) to reconstruct hidden structure
- **Maximization step:** Use that hidden structure (and observations) to reestimate parameters

EM Clustering (for k clusters)

- EM algorithm
  - Viterbi version – called "k-means clustering"
  - Full EM version – called "Gaussian mixtures"

- **Expectation step:** Use current parameters (and observations) to reconstruct hidden structure
- **Maximization step:** Use that hidden structure (and observations) to reestimate parameters

EM Clustering (for k clusters)

- EM algorithm
  - Viterbi version – called "k-means clustering"
  - Full EM version – called "Gaussian mixtures"

- **Expectation step:** Use current parameters (and observations) to reconstruct hidden structure
- **Maximization step:** Use that hidden structure (and observations) to reestimate parameters

- Parameters: k points representing cluster centers

EM Clustering (for k clusters)

- EM algorithm
  - Viterbi version – called "k-means clustering"
  - Full EM version – called "Gaussian mixtures"

- **Expectation step:** Use current parameters (and observations) to reconstruct hidden structure
- **Maximization step:** Use that hidden structure (and observations) to reestimate parameters

- Parameters: k points representing cluster centers
- Hidden structure: for each data point (word type), which center generated it?

Brown Clustering

- Think back to Markov language models
  - E.g., first order $p(w_1, \ldots, w_n) \approx \prod_i p(w_i \mid w_{i-1})$
- Think back to hidden Markov models

$$p(w_1, \ldots, w_n, t_1, \ldots, t_n) \approx \prod_i p(w_i \mid t_i) p(t_i \mid t_{i-1})$$

- Class LM: deterministic word-tag mapping C

$$p(w_1, \ldots, w_n) \approx \prod_i p(w_i \mid C(w_i)) p(C(w_i) \mid C(w_{i-1}))$$

Brown Clustering

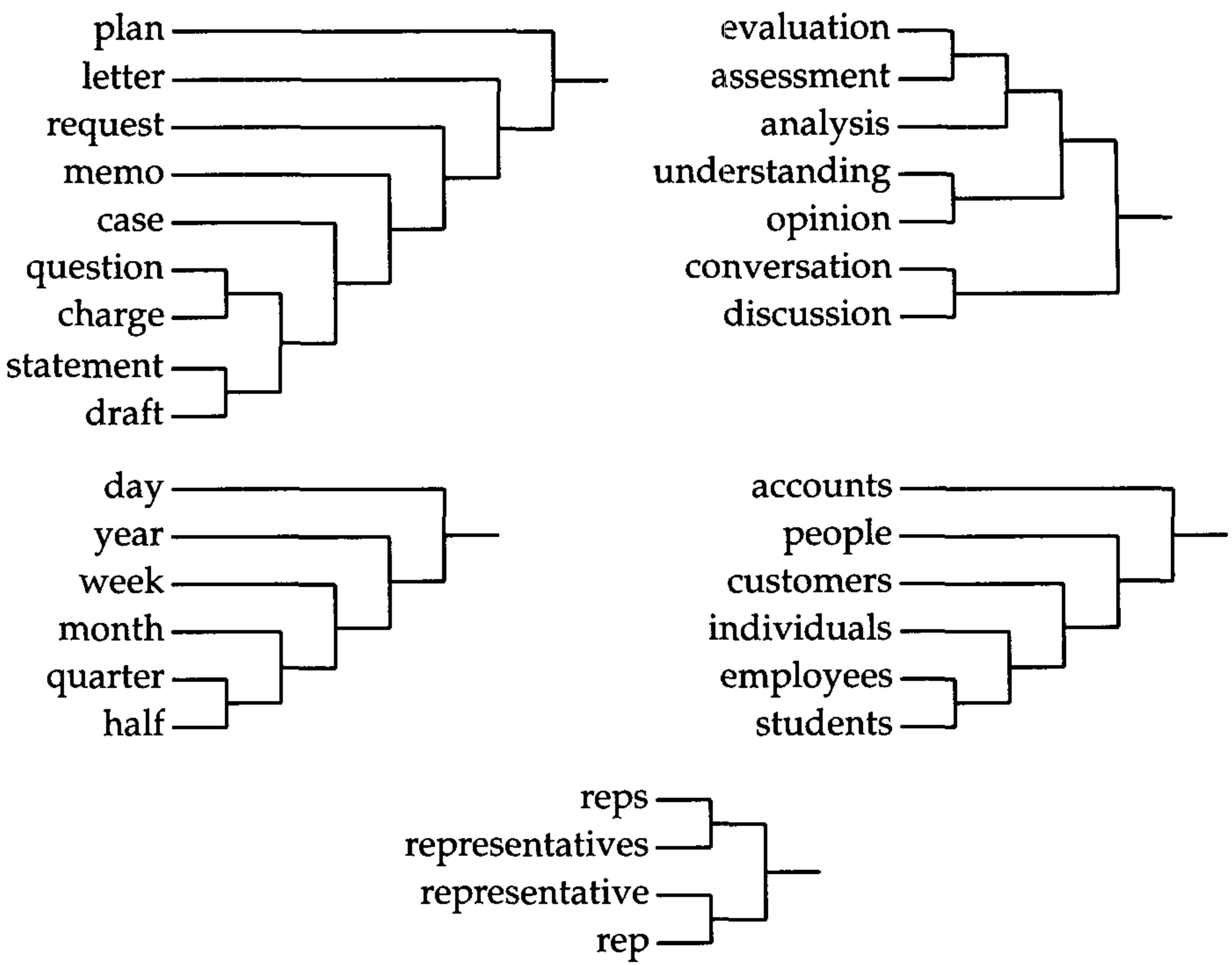
- Brown, Della Pietra, deSouza, Lai, Mercer (1992)
- Deterministic mapping admits Viterbi EM
- Greedy bottom-up cluster merging
- Cluster score

$$\begin{aligned} L(\pi) &= \sum_w \Pr(w) \log \Pr(w) + \sum_{c_1 c_2} \Pr(c_1 c_2) \log \frac{\Pr(c_2 \mid c_1)}{\Pr(c_2)} \\ &= -H(w) + I(c_1, c_2), \end{aligned}$$

- Merge to maximize

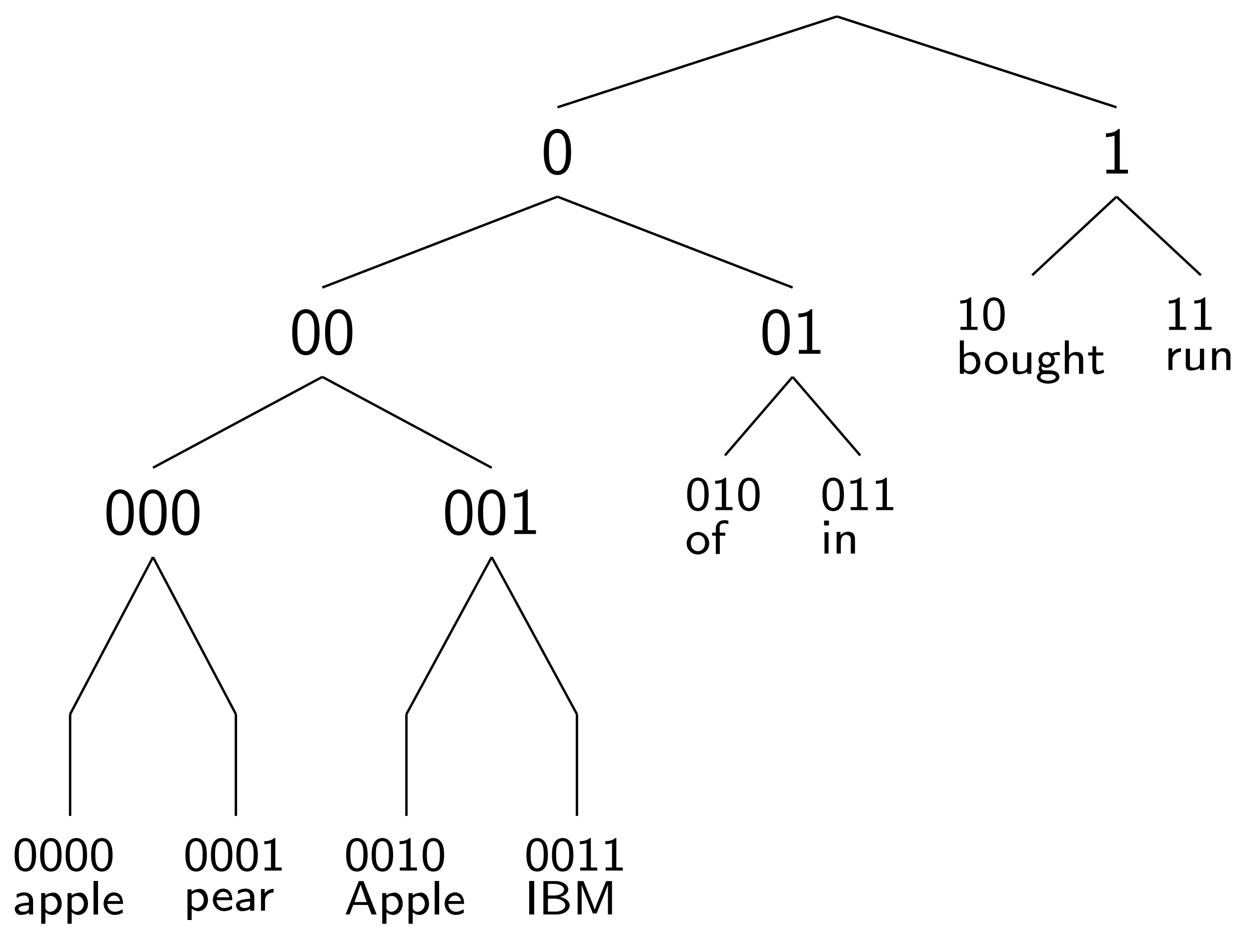
$$L(m,n) = \sum_{d \in \mathcal{C}'} I(m \cup n, d) - \sum_{d \in \mathcal{C}} (I(m,d) + I(n,d))$$

Brown Clustering



Brown Clustering

- Agglomerative clustering produces series of cluster assignments

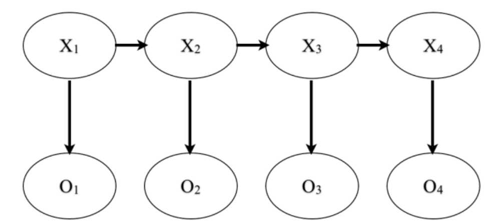


Brown Clustering

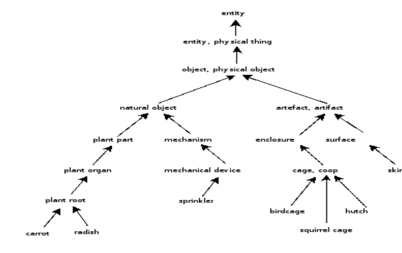
- Dependency parsing (Koo et al., 2008, Haffari et al., 2011, inter alia)
- PCFG parsing (Candito and Crabbé, 2009)
- Semantic dependency parsing (Zhao et al., 2009)
- Named-entity recognition (Turian et al., 2010, Miller et al., 2004)
- QA (Momtazi and Klakow, 2009)

Embedding

Deep Learning

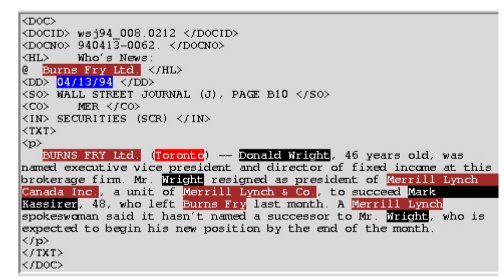


NER

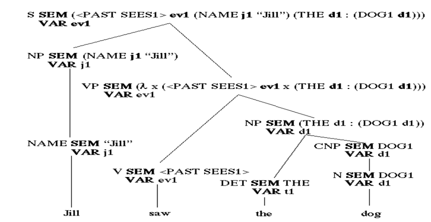


WordNet

Most current machine learning works well because of human-designed representations and input features

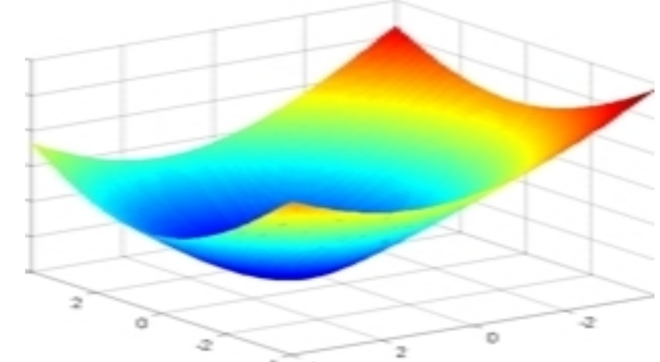


SRL



Parser

Machine learning becomes just optimizing weights to best make a final prediction



Representation learning attempts to automatically learn good features or representations

Deep learning algorithms attempt to learn multiple levels of representation of increasing complexity/abstraction

A Deep Architecture

Mainly, work has explored deep belief networks (DBNs), Markov Random Fields with multiple layers, and various types of multiple-layer neural networks

Output layer

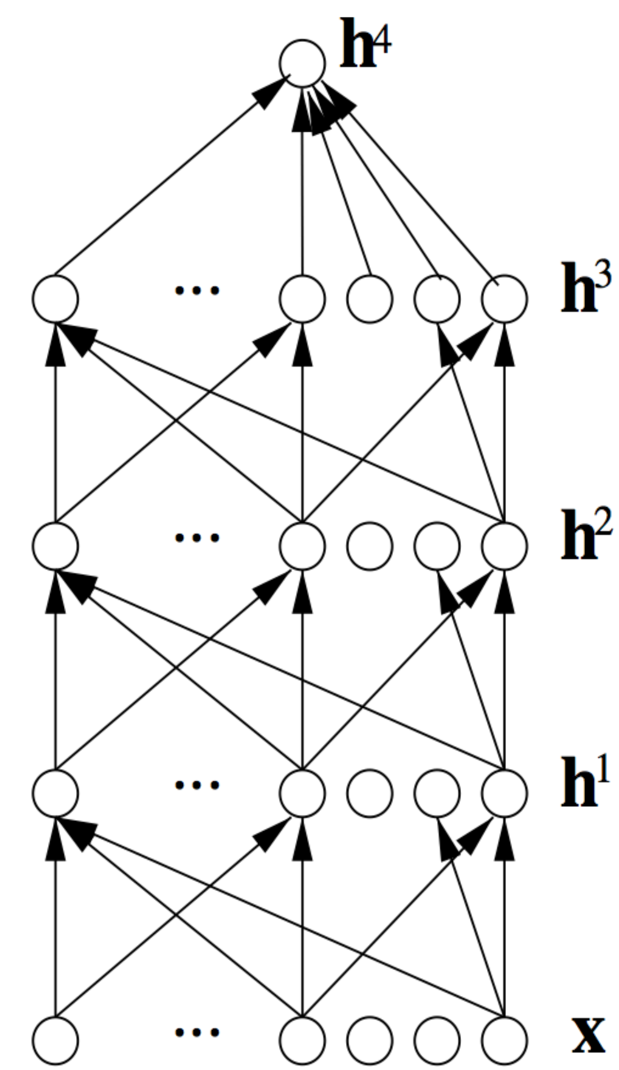
Here predicting a supervised target

Hidden layers

These learn more abstract representations as you head up

Input layer

Raw sensory inputs (roughly)



Demystification

Neural networks come with their own terminological baggage

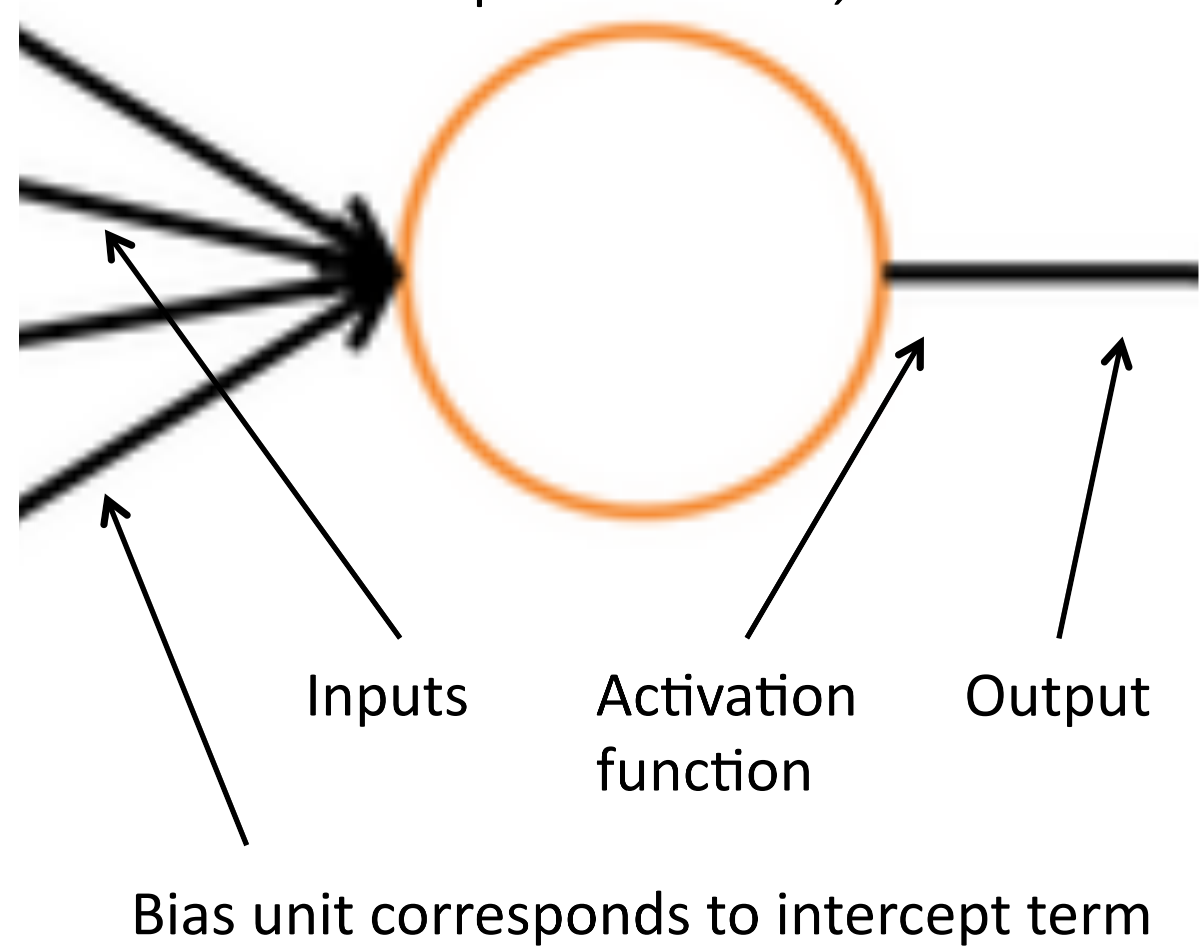
... just like SVMs

But if you understand how logistic regression or maxent models work

Then **you already understand** the operation of a basic neural network neuron!

A single neuron

A computational unit with *n* (3) inputs
and 1 output
and parameters *W, b*



Maxent to Neural Net

In NLP, a maxent classifier is normally written as:

$$P(c \mid d, \lambda) = \frac{\exp \sum_i \lambda_i f_i(c,d)}{\sum_{c' \in C} \exp \sum_i \lambda_i f_i(c',d)}$$

Supervised learning gives us a distribution for datum *d* over classes in *C*

Vector form: $$P(c \mid d, \lambda) = \frac{e^{\lambda^{\top} f(c,d)}}{\sum_{c'} e^{\lambda^{\top} f(c',d)}}$$

Such a classifier is used as-is in a neural network ("a softmax layer")

- Often as the top layer: $J = \text{softmax}(\lambda \cdot x)$

But for now we'll derive a two-class logistic model for one neuron

Maxent to Neural Net

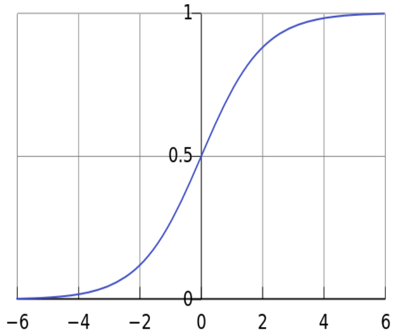
Vector form: $$P(c \mid d, \lambda) = \frac{e^{\lambda^\top f(c,d)}}{\sum_{c'} e^{\lambda^\top f(c',d)}}$$

Make two class:

$$P(c_1 \mid d, \lambda) = \frac{e^{\lambda^\top f(c_1,d)}}{e^{\lambda^\top f(c_1,d)} + e^{\lambda^\top f(c_2,d)}} = \frac{e^{\lambda^\top f(c_1,d)}}{e^{\lambda^\top f(c_1,d)} + e^{\lambda^\top f(c_2,d)}} \cdot \frac{e^{-\lambda^\top f(c_1,d)}}{e^{-\lambda^\top f(c_1,d)}}$$

$$= \frac{1}{1 + e^{\lambda^\top [f(c_2,d) - f(c_1,d)]}} = \frac{1}{1 + e^{-\lambda^\top x}}$$ for $x = f(c_1,d) - f(c_2,d)$

$$= f(\lambda^\top x)$$



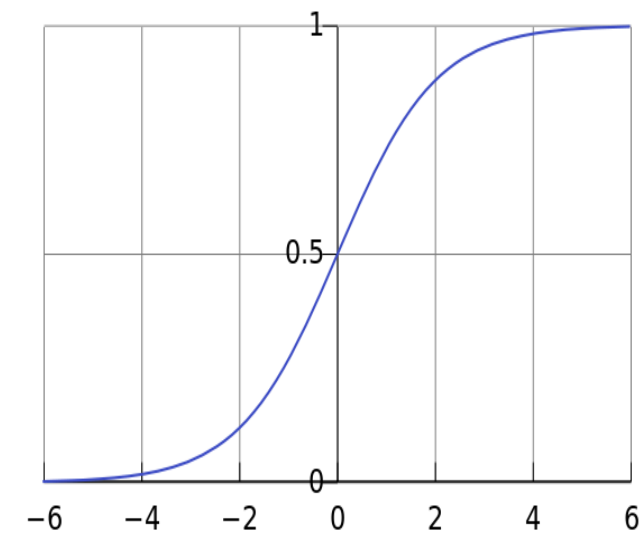
for $f(z) = 1/(1 + \exp(-z))$, the logistic function – a sigmoid non-linearity.

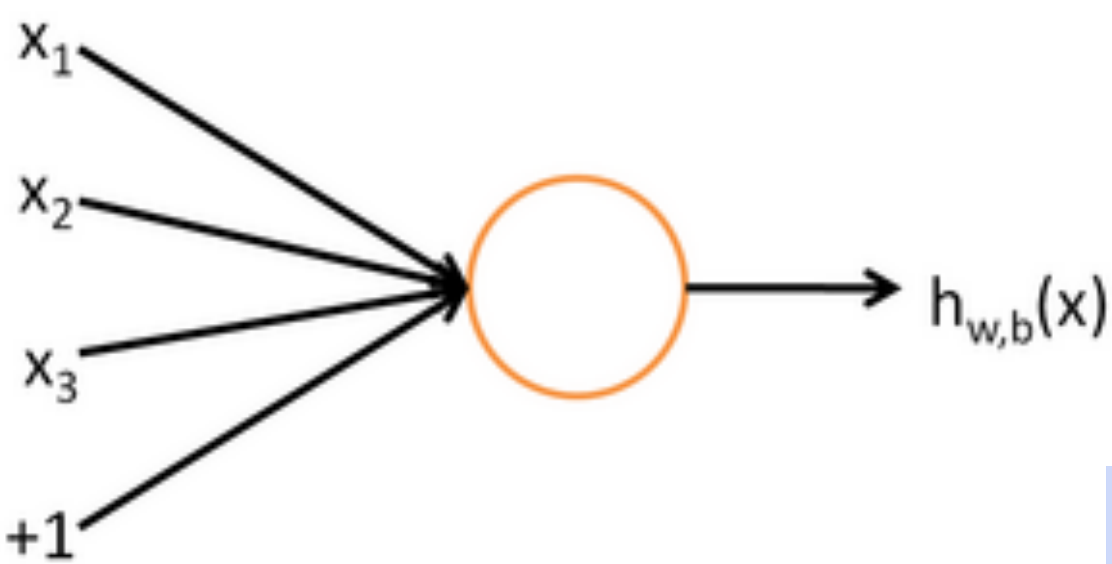
Now You've Got a Neuron

$$h_{w,b}(x) = f(w^{\mathsf{T}}x + b)$$

$$f(z) = \frac{1}{1+e^{-z}}$$

b: We can have an "always on" feature, which gives a class prior, or separate it out, as a bias term

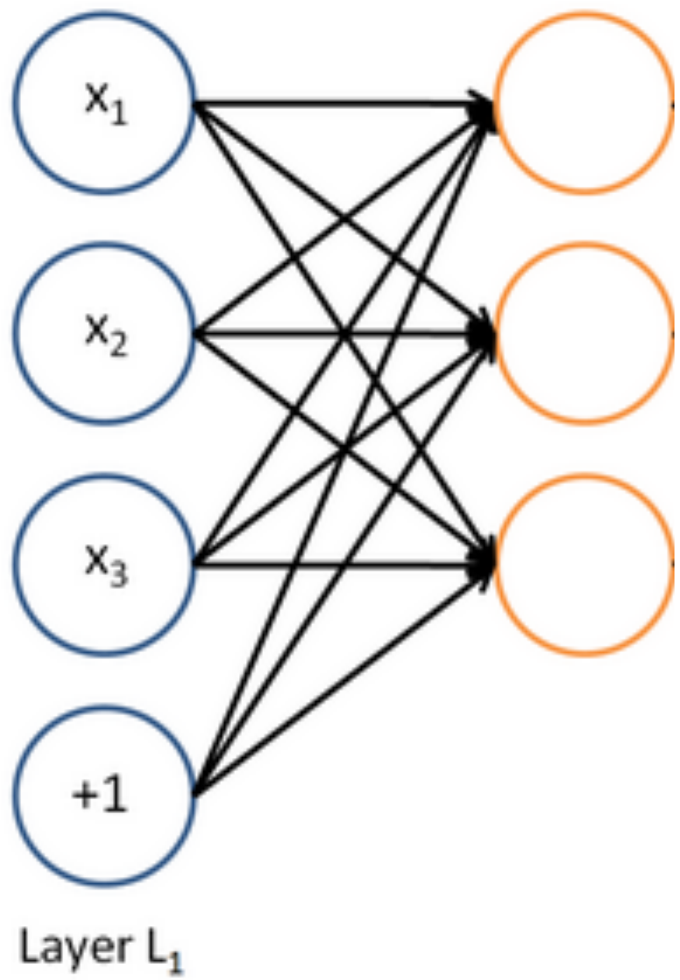




w, *b* are the parameters of this neuron i.e., this logistic regression model

Lots of Logistic Regressions!

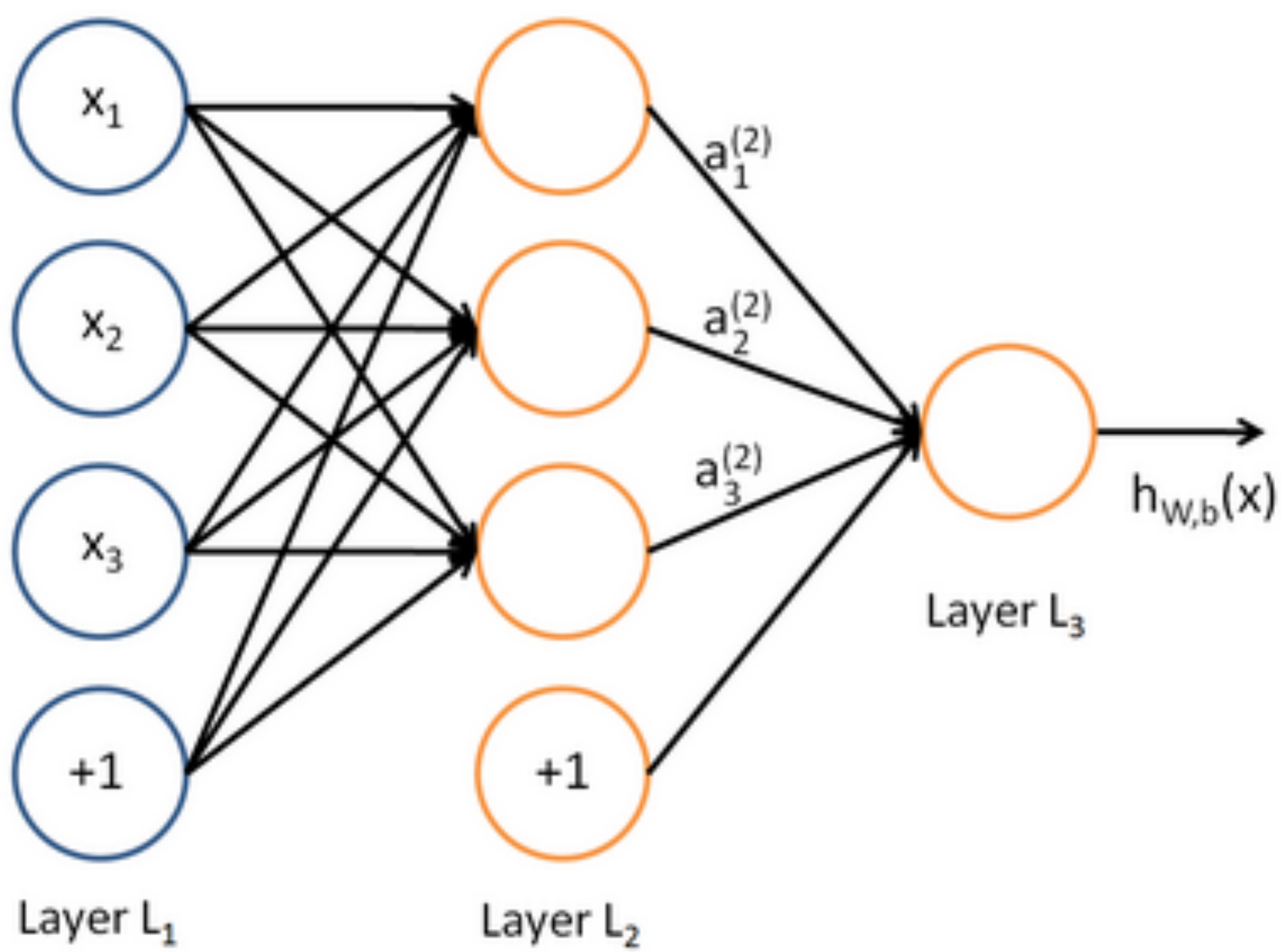
If we feed a vector of inputs through a bunch of logistic regression functions, then we get a vector of outputs …



But we don't have to decide ahead of time what variables these logistic regressions are trying to predict!

Lots of Logistic Regressions!

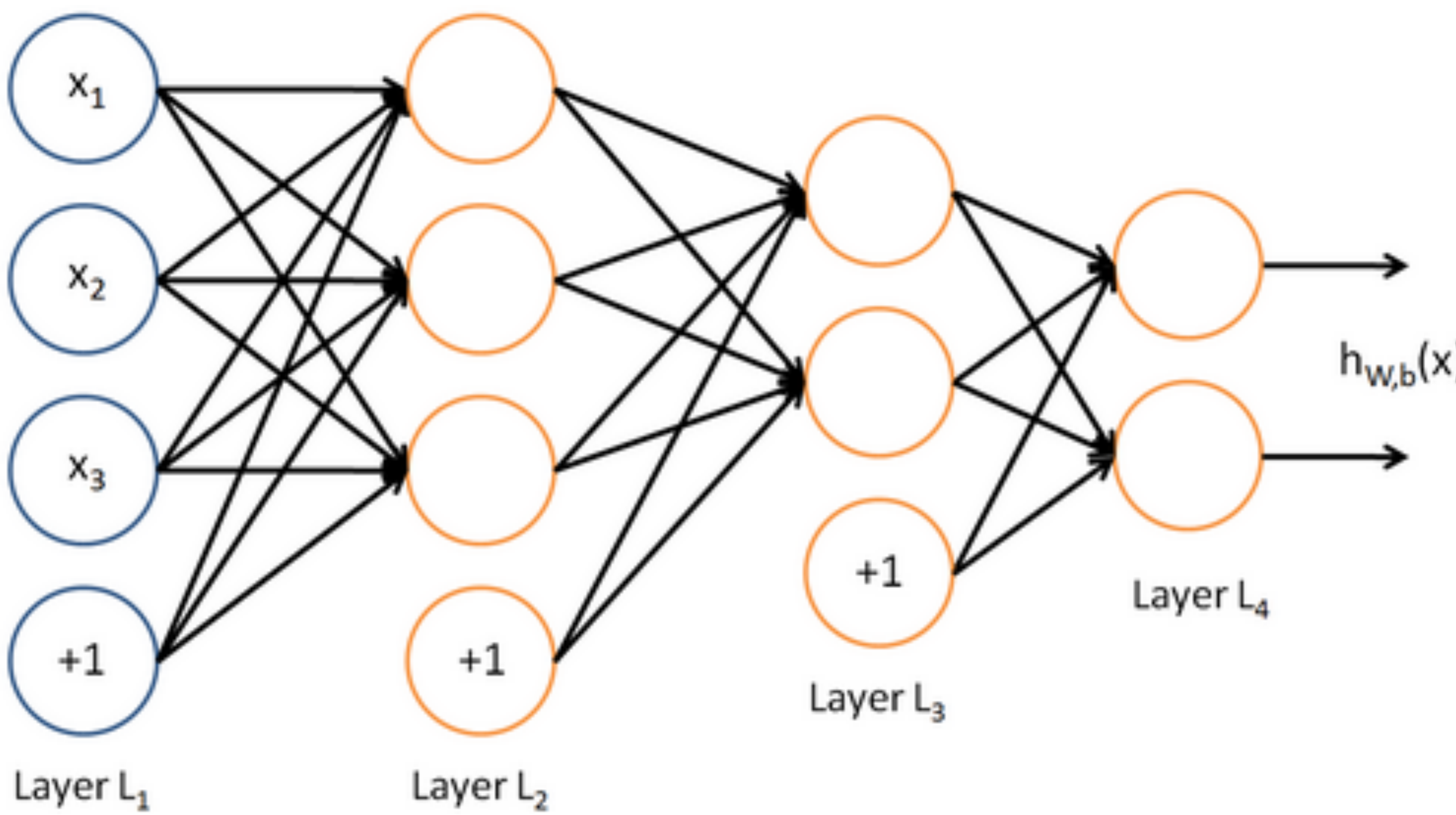
- … which we can feed into another logistic regression function



It is the training criterion that will direct what the intermediate hidden variables should be, so as to do a good job at predicting the targets for the next layer, etc.

Lots of Logistic Regressions!

Before we know it, we have a multilayer neural network….



Matrix Notation for a Layer

We have

$$a_1 = f(W_{11}x_1 + W_{12}x_2 + W_{13}x_3 + b_1)$$
$$a_2 = f(W_{21}x_1 + W_{22}x_2 + W_{23}x_3 + b_2)$$
$$\text{etc.}$$

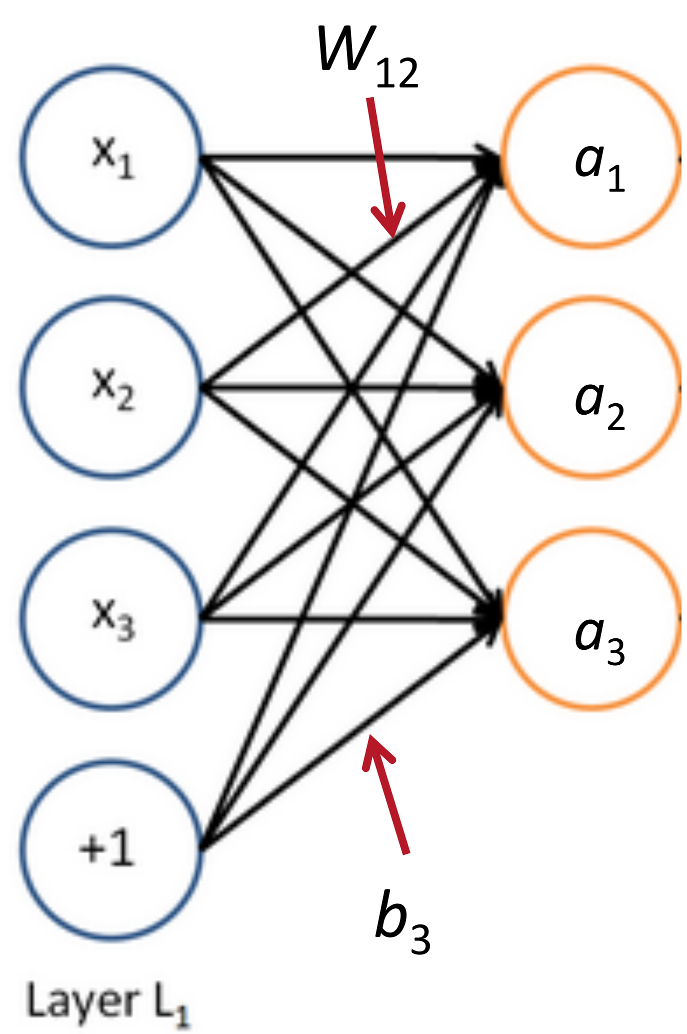
In matrix notation

$$z = Wx + b$$

$$a = f(z)$$

where *f* is applied element-wise:

$$f([z_1, z_2, z_3]) = [f(z_1), f(z_2), f(z_3)]$$



How to Estimate W?

- For a single supervised layer, we train just like a maxent model – we calculate and use error derivatives (gradients) to improve
  - Online learning: Stochastic gradient descent (SGD)
    - Or improved versions like AdaGrad (Duchi, Hazan, & Singer 2010)
  - Batch learning: Conjugate gradient or L-BFGS

- A multilayer net could be more complex because the internal ("hidden") logistic units make the function non-convex … just as for hidden CRFs [Quattoni et al. 2005, Gunawardana et al. 2005]
  - But we can use the same ideas and techniques
    - Just without guarantees …
  - We "backpropagate" error derivatives through the model

Translation Guide

You now understand the basics and the relation to other models

- Neuron = logistic regression or similar function
- Input layer = input training/test vector
- Bias unit = intercept term/always on feature
- Activation = response
- Activation function is a logistic (or similar “sigmoid” nonlinearity)
- Backpropagation = running stochastic gradient descent backward layer-by-layer in a multilayer network
- Weight decay = regularization / Bayesian prior

Standard Word Representation

The vast majority of rule-based **and** statistical NLP work regards words as atomic symbols: hotel, conference, walk

In vector space terms, this is a vector with one 1 and a lot of zeroes

[0 0 0 0 0 0 0 0 0 0 1 0 0 0 0]

Dimensionality: 20K (speech) – 50K (PTB) – 500K (big vocab) – 13M (Google 1T)

We call this a "one-hot" representation. Its problem:

motel [0 0 0 0 0 0 0 0 0 0 1 0 0 0 0] AND
hotel [0 0 0 0 0 0 0 1 0 0 0 0 0 0 0] = 0

Distributional Similarity

You can get a lot of value by representing a word by means of its neighbors

"You shall know a word by the company it keeps"

(J. R. Firth 1957: 11)

One of the most successful ideas of modern statistical NLP

government debt problems turning into banking crises as has happened in

saying that Europe needs unified banking regulation to replace the hodgepodge

↖ These words will represent *banking* ↗

You can vary whether you use local or large context to get a more syntactic or semantic clustering

Hard/Soft Clustering

Class based models learn word classes of similar words based on distributional information (~ class HMM)

- Brown clustering (Brown et al. 1992)
- Exchange clustering (Martin et al. 1998, Clark 2003)
- Desparsification and great example of unsupervised pre-training

Soft clustering models learn for each cluster/topic a distribution over words of how likely that word is in each cluster

- Latent Semantic Analysis (LSA/LSI), Random projections
- Latent Dirichlet Analysis (LDA), HMM clustering

Distributed Representation

Similar idea

Combine vector space semantics with the prediction of probabilistic models (Bengio et al. 2003, Collobert & Weston 2008, Turian et al. 2010)

In all of these approaches, including deep learning models, a word is represented as a dense vector

$$\textit{linguistics} = \begin{pmatrix} 0.286 \\ 0.792 \\ -0.177 \\ -0.107 \\ 0.109 \\ -0.542 \\ 0.349 \\ 0.271 \end{pmatrix}$$

Visualizing Embeddings

Vector Semantics

Mikolov, Yih & Zweig (2013)

These representations are *way* better at encoding dimensions of similarity than we realized!

- Analogies testing dimensions of similarity can be solved quite well just by doing vector subtraction in the embedding space

  Syntactically

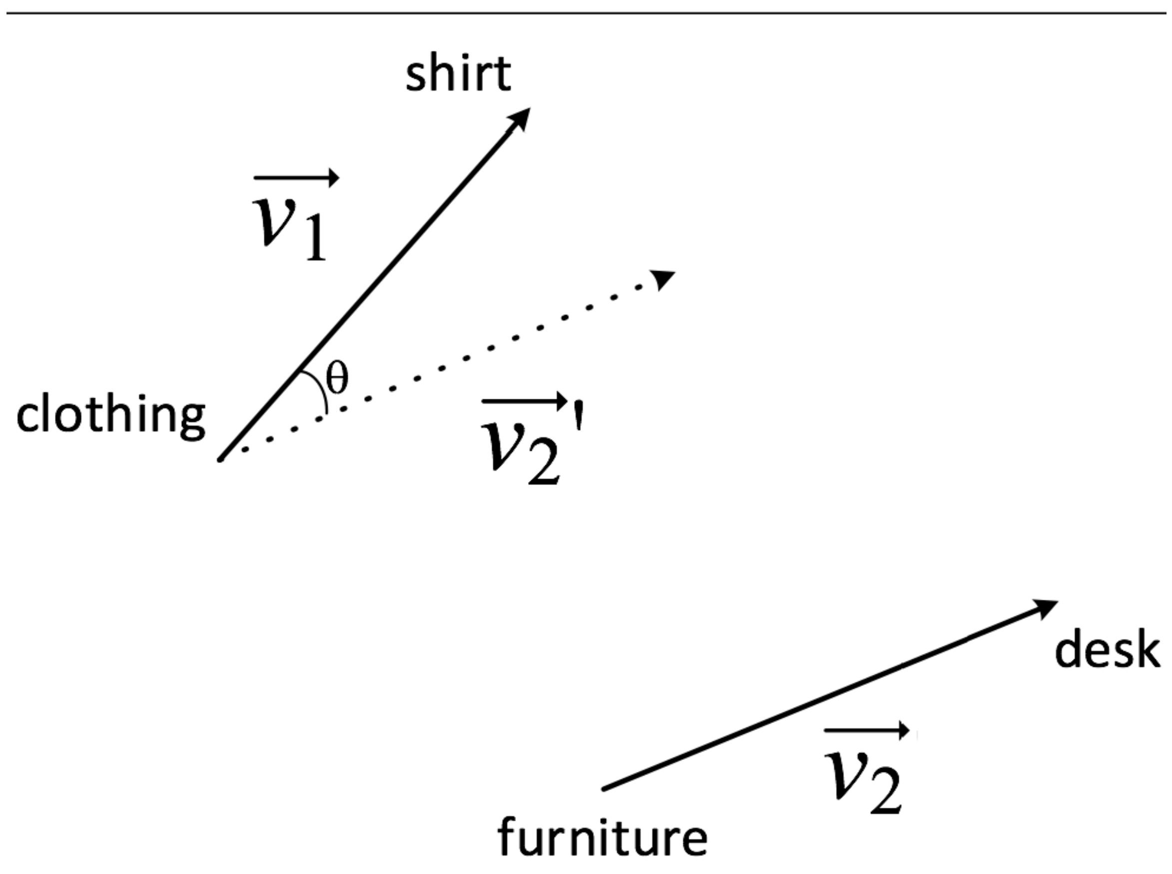
  - $x_{apple} - x_{apples} \approx x_{car} - x_{cars} \approx x_{family} - x_{families}$
  - Similarly for verb and adjective morphological forms

  Semantically (Semeval 2012 task 2)

  - $x_{shirt} - x_{clothing} \approx x_{chair} - x_{furniture}$

Vector Semantics

Mikolov, Yih & Zweig (2013)



Method	Syntax % correct
LSA 320 dim	16.5 [best]
RNN 80 dim	16.2
RNN 320 dim	28.5
RNN 1600 dim	39.6
Method	**Semantics Spearm ρ**
UTD-NB (Rink & H. 2012)	0.230 [Semeval win]
LSA 640	0.149
RNN 80	0.211
RNN 1600	0.275 [new SOTA]

Advantages of Embeddings

Compared to a method like LSA, neural word embeddings can become more meaningful through adding supervision from one or multiple tasks

“Discriminative fine-tuning”

For instance, sentiment is usually not captured in unsupervised word embeddings but can be in neural word vectors

We can build representations for large linguistic units

Reading

- Jurafsky & Martin, chapters 18–20
- NLTK book, chapter 10
- Yoav Goldberg, A Primer on Neural Network Models for Natural Language Processing, Tech report, October 2015 u.cs.biu.ac.il/~yogo/nnlp.pdf