LT2212 Statistical methods in NLP, Winter 2019

Lecture 3: Word vectors; perceptrons and SVMs; an application

Asad Sayeed
with content from Jon Dehdari

University of Gothenburg

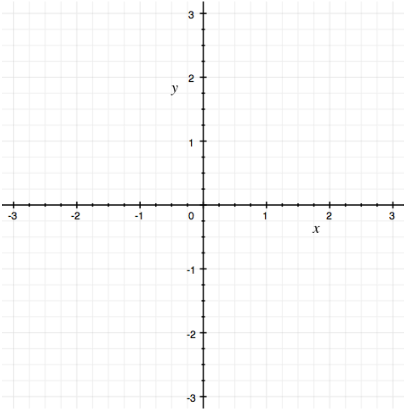
Today's agenda:

1. Vector-space classification
2. Word vectors
3. Beyond LSA
4. An application in distributional semantics

Assignment 2 questions?

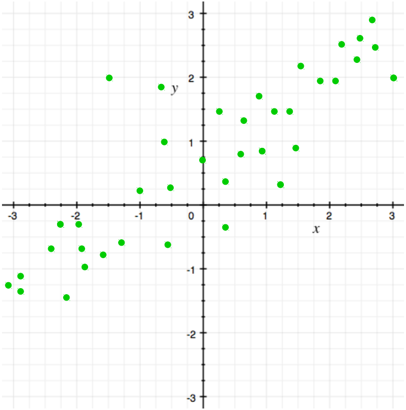
Part 1: Vector-space classification

Regression



Instead of drawing the line based on the equation. . .

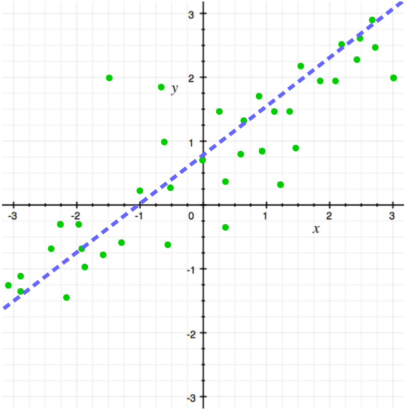
Regression



Instead of drawing the line based on the equation. . .

- What if we already know the points . . . and they're not on a line?

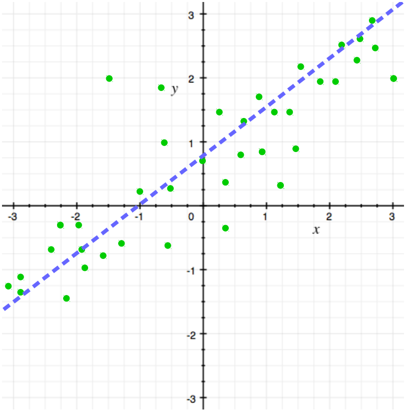
Regression



Instead of drawing the line based on the equation...

- What if we already know the points ... and they're not on a line?
- We can still make a generalization about them by fitting a line: this is **linear regression**.

Regression



Instead of drawing the line based on the equation. . .

- What if we already know the points . . . and they're not on a line?
- We can still make a generalization about them by fitting a line: this is **linear regression**.
- Each choice of m and b in ($y = mx + b$) is a hypothesis about the "real" source of the data.

Regression

Back to the notion of the hyperplane:

N-dimensional linear equation

$$y = w_0 x_0 + w_1 x_1 \ldots w_{N-2} x_{N-2} + b$$

Regression

Back to the notion of the hyperplane:

N-dimensional linear equation

$$y = w_0 x_0 + w_1 x_1 \ldots w_{N-2} x_{N-2} + b$$

- In linear regression, $x_0 \ldots x_{N-2}$ are the **predictors**.

Regression

Back to the notion of the hyperplane:

N-dimensional linear equation

$$y = w_0x_0 + w_1x_1 \dots w_{N-2}x_{N-2} + b$$

- In linear regression, $x_0 \dots x_{N-2}$ are the **predictors**.
- y is the **response** to the predictors.

Regression

Back to the notion of the hyperplane:

N-dimensional linear equation

$$y = w_0x_0 + w_1x_1 \ldots w_{N-2}x_{N-2} + b$$

- In linear regression, $x_0 \ldots x_{N-2}$ are the **predictors**.
- y is the **response** to the predictors.
- The weights $w_0 \ldots w_{N-2}$ are the **coefficients** that represent the strength of each factor.

Regression

Back to the notion of the hyperplane:

N-dimensional linear equation

$$y = w_0 x_0 + w_1 x_1 \ldots w_{N-2} x_{N-2} + b$$

- In linear regression, $x_0 \ldots x_{N-2}$ are the **predictors**.
- y is the **response** to the predictors.
- The weights $w_0 \ldots w_{N-2}$ are the **coefficients** that represent the strength of each factor.
- The intercept b represents the response if no factor was present.

Regression

Back to the notion of the hyperplane:

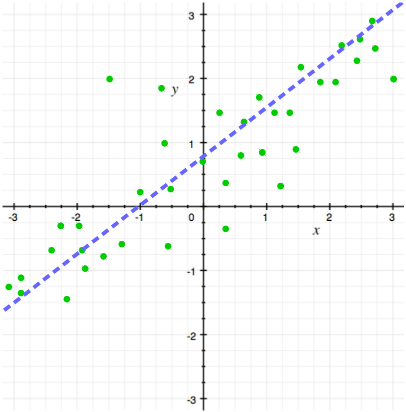
N-dimensional linear equation

$$y = w_0x_0 + w_1x_1 \dots w_{N-2}x_{N-2} + b$$

- In linear regression, $x_0 \dots x_{N-2}$ are the **predictors**.
- y is the **response** to the predictors.
- The weights $w_0 \dots w_{N-2}$ are the **coefficients** that represent the strength of each factor.
- The intercept b represents the response if no factor was present.

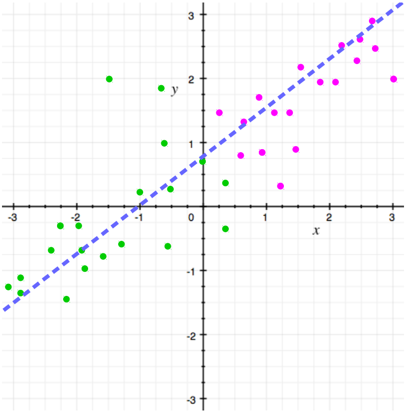
"Task" of linear regression: find best-fitting hyperplane via **w** and b, according to techniques we won't talk about here.

Classification



But sometimes we want a different hypothesis.

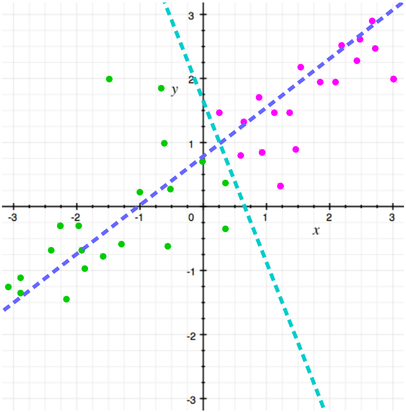
Classification



But sometimes we want a different hypothesis.

- Instead, we're not looking for the response, but the **class**.

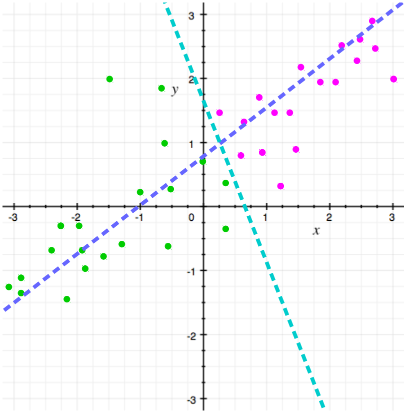
Classification



But sometimes we want a different hypothesis.

- Instead, we're not looking for the response, but the **class**.
- Which means we're looking for an entirely different hyperplane.

Classification

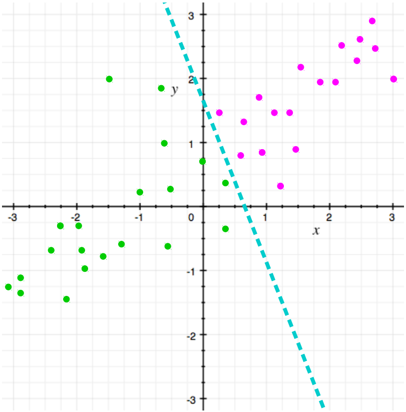


But sometimes we want a different hypothesis.

- In fact, we're trying to find **w** and b such that

$$f(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{w} \cdot \mathbf{x} + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

Classification



But sometimes we want a different hypothesis.

- In fact, we're trying to find $\mathbf{w}$ and b such that

$$f(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{w} \cdot \mathbf{x} + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

- And then we can decide the hyperplane $y = \mathbf{w} \cdot \mathbf{x} + b$.

Continuous and categorical features

In classification, predictors are **features**.
How do we map to points in the space?

Continuous and categorical features

In classification, predictors are **features**.

How do we map to points in the space?

- For continuous features (e.g. "height"):
    - Typical practice: normalize to values between -1 and 1.

Continuous and categorical features

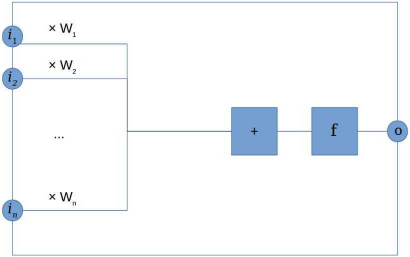
In classification, predictors are **features**.

How do we map to points in the space?

- For continuous features (e.g. "height"):
  - Typical practice: normalize to values between -1 and 1.
- For categorical features (e.g. "car brand name"):
  - Convert each category into a separate binary feature.
  - E.g. if "car brand name" has values "volvo", "subaru", "ford", you will turn it into three separate features: "brand-volvo", "brand-subaru" and "brand-ford" with 0 or 1 value.

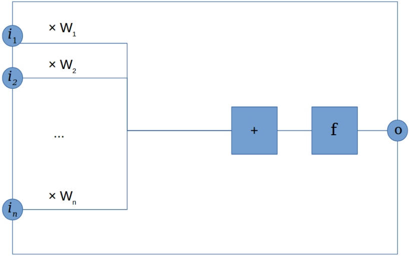
The perceptron: a very simple neural network

(from Wikipedia)



The perceptron: a very simple neural network

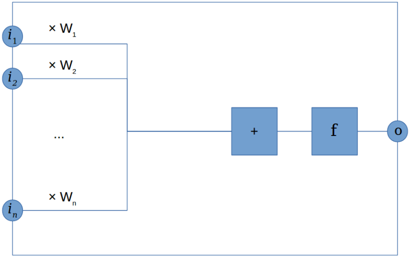
(from Wikipedia)



- Each instance vector **x**'s values are fed as inputs *i* to the network.

The perceptron: a very simple neural network

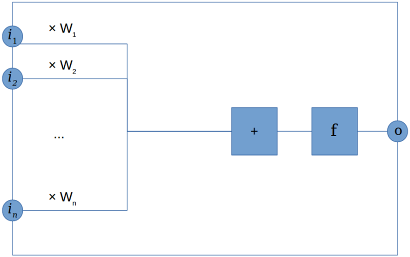
(from Wikipedia)



- Each instance vector **x**'s values are fed as inputs *i* to the network.
- Feature function *f* is applied (remember: 1 or 0 output).

The perceptron: a very simple neural network

(from Wikipedia)



- Each instance vector **x**'s values are fed as inputs *i* to the network.
- Feature function *f* is applied (remember: 1 or 0 output).
- Weights adjusted based on output correctness.

Perceptron algorithm (roughly)

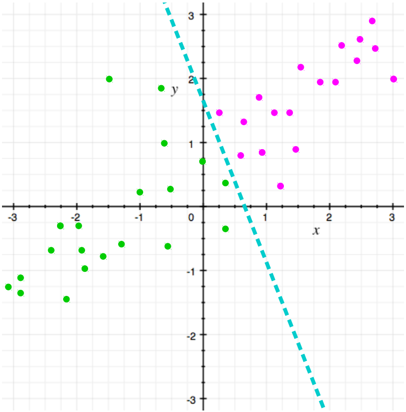
Initialize weights $\mathbf{w}$ and bias (usually to (close to) 0).

Given n feature vectors $\mathbf{x}$ and corresponding "ground truth" values d, for vector $\mathbf{x}_i$:

- Calculate $f(\mathbf{x})$ as 1 or 0 using $\mathbf{w} \cdot \mathbf{x}_i + b$.
- Update weights as $\mathbf{w} \leftarrow \mathbf{w} + (d_i - f(\mathbf{x_i}))\mathbf{x_i}$.
- Move to next $\mathbf{x}$ feature vector, cycling through vectors until convergence.

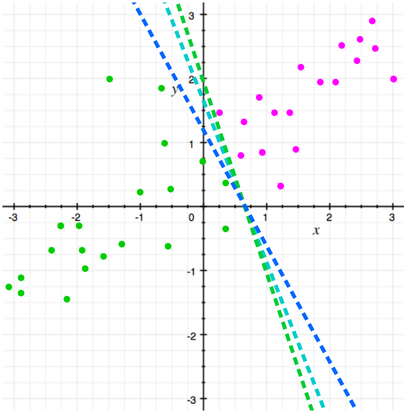
(There is a theoretical upper bound on how many iterations are required to converge.)

Perceptron limitations



What line do you choose?

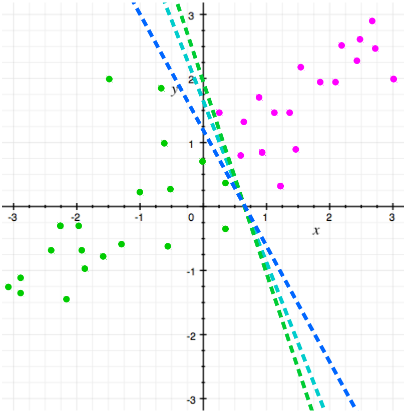
Perceptron limitations



What line do you choose?

- Many are possible, perceptron does not fit one reliably.

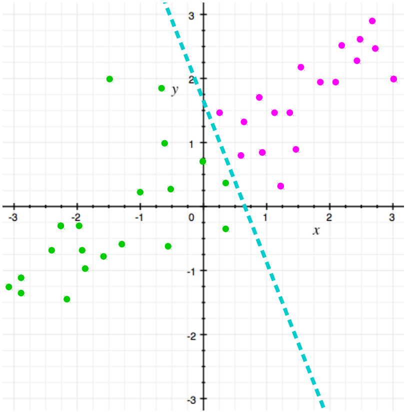
Perceptron limitations



What line do you choose?

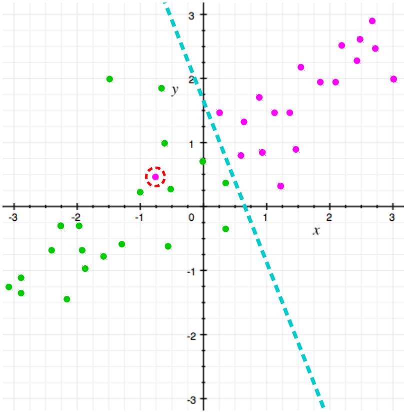
- Many are possible, perceptron does not fit one reliably.
- Why is this a problem?

Perceptron limitations



Our picture so far has been very convenient.

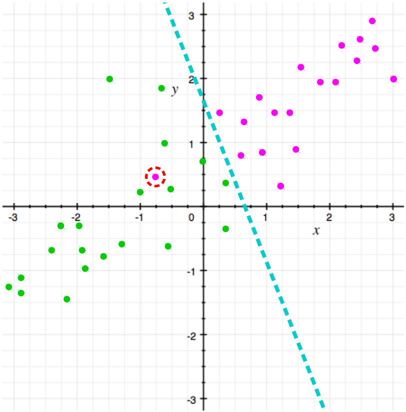
Perceptron limitations



Our picture so far has been very convenient.

- What if a point of one class was surrounded by points of the other class?

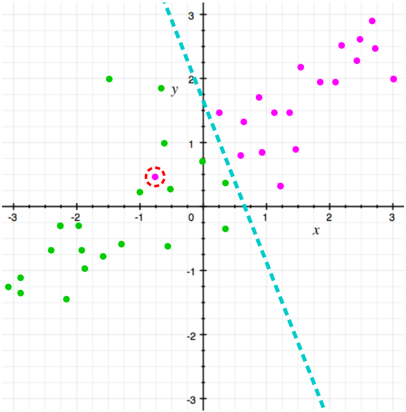
Perceptron limitations



Our picture so far has been very convenient.

- What if a point of one class was surrounded by points of the other class?
- Perceptrons don't converge if space is not **linearly separable**.

Perceptron limitations



Our picture so far has been very convenient.

- What if a point of one class was surrounded by points of the other class?
- Perceptrons don't converge if space is not **linearly separable**.
- Setting a "tolerance" doesn't help much – need more complex variant.

Support vector machines

Support vector machines (SVMs) find hyperplanes in the same sense as perceptrons. But they're more versatile.

Support vector machines

Support vector machines (SVMs) find hyperplanes in the same sense as perceptrons. But they're more versatile.

- Linear SVM – not only find a separating hyperplane, but also the "optimal" hyperplane, given some tolerance.

Support vector machines

Support vector machines (SVMs) find hyperplanes in the same sense as perceptrons. But they're more versatile.

- Linear SVM – not only find a separating hyperplane, but also the "optimal" hyperplane, given some tolerance.
- Nonlinear SVM – Same as linear SVM, but find hyperplanes that work for linearly inseparable data.

Support vector machines

Support vector machines (SVMs) find hyperplanes in the same sense as perceptrons. But they're more versatile.

- Linear SVM – not only find a separating hyperplane, but also the "optimal" hyperplane, given some tolerance.
- Nonlinear SVM – Same as linear SVM, but find hyperplanes that work for linearly inseparable data.
- The algorithms are more advanced than a perceptron: quadratic programming, gradient descent – we'll get into gradient descent probably in a later class.

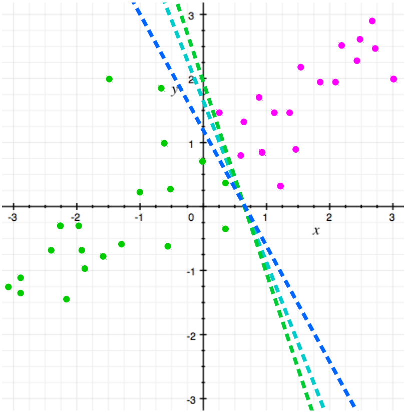
Support vector machines

Support vector machines (SVMs) find hyperplanes in the same sense as perceptrons. But they're more versatile.

- Linear SVM – not only find a separating hyperplane, but also the "optimal" hyperplane, given some tolerance.
- Nonlinear SVM – Same as linear SVM, but find hyperplanes that work for linearly inseparable data.
- The algorithms are more advanced than a perceptron: quadratic programming, gradient descent – we'll get into gradient descent probably in a later class.

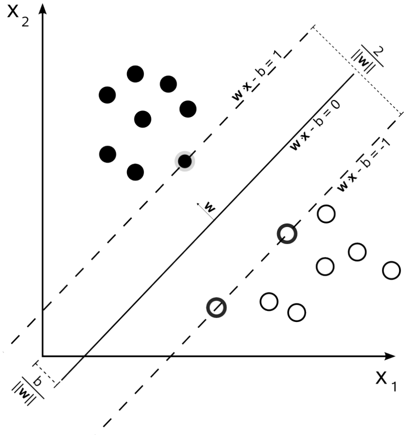
But what is a support vector?

Hard-margin linear SVM



Perceptrons have too much freedom.

Hard-margin linear SVM

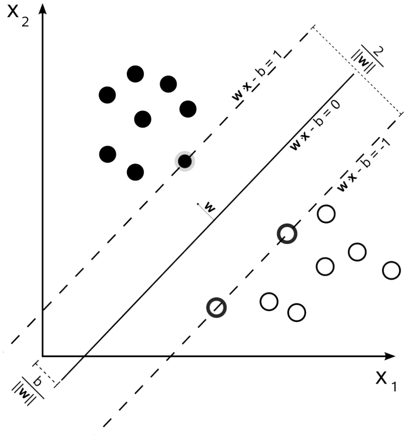


(from Wikipedia)

Perceptrons have too much freedom.

- If data is linearly separable, choose two parallel hyperplanes corresponding to $\mathbf{w} \cdot \mathbf{x} + b = 1$ and $\mathbf{w} \cdot \mathbf{x} + b = -1$.

Hard-margin linear SVM

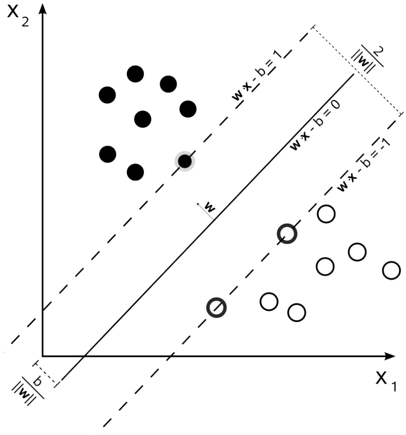


(from Wikipedia)

Perceptrons have too much freedom.

- If data is linearly separable, choose two parallel hyperplanes corresponding to $\mathbf{w} \cdot \mathbf{x} + b = 1$ and $\mathbf{w} \cdot \mathbf{x} + b = -1$.
- Maximize distance of planes by minimizing magnitude of $\mathbf{w}$.

Hard-margin linear SVM

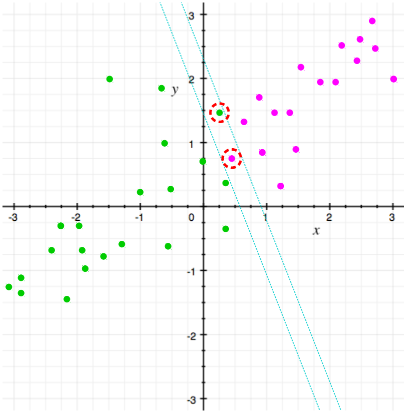


(from Wikipedia)

Perceptrons have too much freedom.

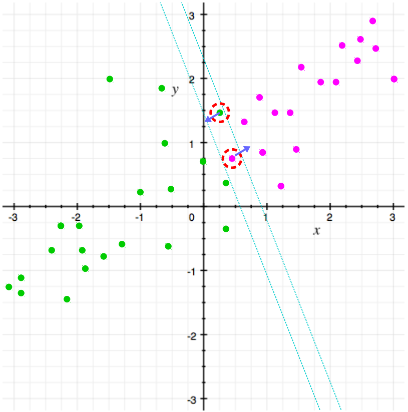
- If data is linearly separable, choose two parallel hyperplanes corresponding to $\mathbf{w} \cdot \mathbf{x} + b = 1$ and $\mathbf{w} \cdot \mathbf{x} + b = -1$.
- Maximize distance of planes by minimizing magnitude of **w**.
- Vectors closest to plane are the support vectors.

Soft-margin linear SVM



“Minor” linear-separability problem: instances inside margins.

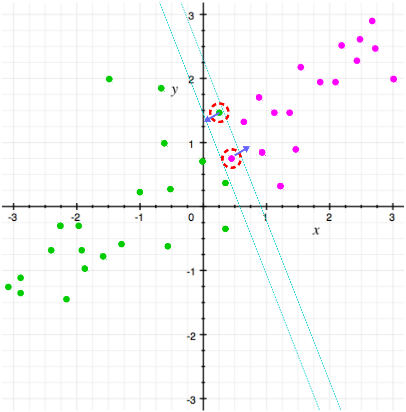
Soft-margin linear SVM



"Minor" linear-separability problem: instances inside margins.

- Solution: virtually "push" them back with an adjustment.

Soft-margin linear SVM

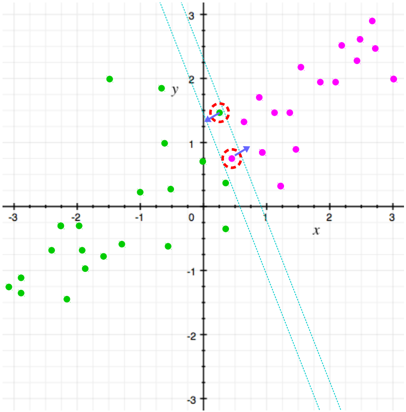


"Minor" linear-separability problem: instances inside margins.

- Solution: virtually "push" them back with an adjustment.
- "Hinge loss" function:

$$\max(0, 1 - y_i(\mathbf{w} \cdot \mathbf{x}_i + b))$$

Soft-margin linear SVM



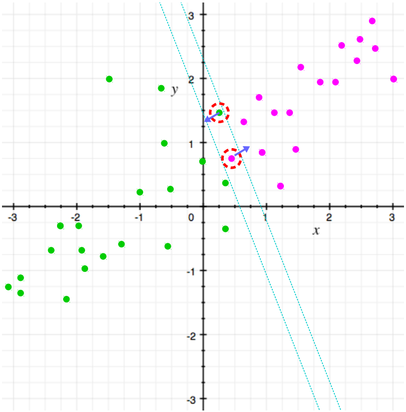
"Minor" linear-separability problem: instances inside margins.

- Solution: virtually "push" them back with an adjustment.
- "Hinge loss" function:

  $$\max(0, 1 - y_i(\mathbf{w} \cdot \mathbf{x}_i + b))$$

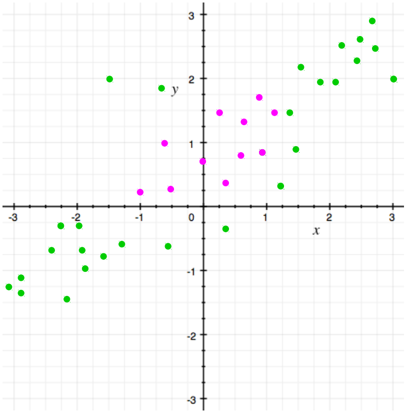
- Add to goals of learner: minimize hinge loss across all instances, with small tolerance for expanding margin.

Nonlinearity



Soft-margin is OK for small overlaps. . .

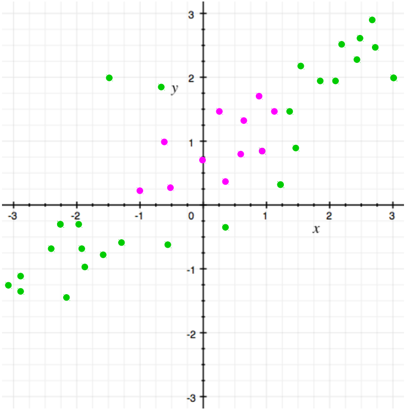
Nonlinearity



Soft-margin is OK for small overlaps. . .

- . . . but sometimes no separability adjustment helps.

Nonlinearity

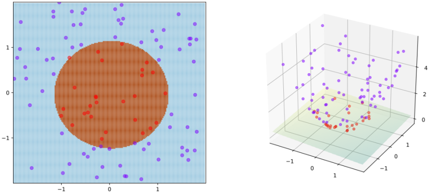


Soft-margin is OK for small overlaps...

- ...but sometimes no separability adjustment helps.
- If you don't like the space you have, go to another space!
    - Apply a function that either maps all points nonlinearly or into a higher dimension, or both.

Nonlinearity

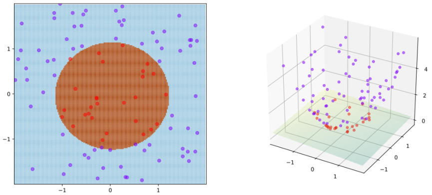
(from Wikipedia)



- Full (possibly expensive) space transformation can be avoided via the **kernel trick**.

Nonlinearity

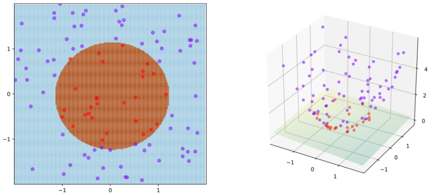
(from Wikipedia)



- Full (possibly expensive) space transformation can be avoided via the **kernel trick**.
- Suppose $\phi(\mathbf{x})$ transforms x into the new space. Kernel function $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$

Nonlinearity

(from Wikipedia)



- Full (possibly expensive) space transformation can be avoided via the **kernel trick**.
- Suppose $\phi(\mathbf{x})$ transforms x into the new space. Kernel function $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$
- Now we can can compute dot products for optimization without having the explicit space.

Kernel functions

Some very basic ones. (They can in theory be quite "bespoke" to your problem.)

- Polynomial kernel:
$$k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j)^d$$
- Radial basis function:
$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2)$$

Often similar to the nonlinearities used in "real" neural networks.

Part 2: Word vectors

Remember Borges?

An Argentinian philosopher and fiction writer. One of his stories mentions 'a certain Chinese Encyclopedia', the *Celestial Emporium of Benevolent knowledge.* It contains a classifcation of animals.

- those that belong to the emperor
- embalmed ones
- those that are trained
- suckling pigs
- mermaids
- fabulous ones
- stray dogs

Remember Borges?

…actually, it goes on.

- those that are included in the present classification
- those that tremble as if they are mad
- innumerable ones
- those drawn with a very fine camelhair brush
- others
- those that have just broken a flower vase
- those that from a long way off look like flies

What words are

So far we've talked about words in order. But words have a relationship to each other.

- We use dictionaries in real life for a reason.
- We need to make fine-grained distinctions, draw connections, and so on.
- Humans make judgements about similarities.
    - You know that "motorcycle" can be used in most, but not all contexts that "car" can be used.
    - English-German bilinguals know that "pride" and "Stolz" are quite similar.

Define “chair”

From dictionary.com (just the noun version):

Define “chair”

From dictionary.com (just the noun version):

- A seat, especially for one person, usually having four legs for support and a rest for the back and often having rests for the arms.
- Something that serves as a chair or supports like a chair: “two men clasped hands to make a chair for their injured companion”.
- A position of authority, as of a judge, professor, etc.
- The person occupying a seat of office, especially the chairperson of a meeting: “the speaker addressed the chair”
- (in an orchestra) the position of a player, assigned by rank; desk: “first clarinet chair”.
- “the chair”, Informal. electric chair.

Words in terms of other words

That doesn't seem very helpful, but it gives us a place to start.
Define "chair" in terms of features:

- +one-person, +four-legs, +support, +backrest, +armrest
- +authority
- +occupies-chair
- +orchestra
- +execution

Words in terms of other words

OK, that gives us the definition of a chair in terms of (rather specific) features.

Define the noun "cockpit". Let's go to dictionary.com again. I get as features:

- +enclosed, +airplane, +controls, +panel, +seats
- +instrumentation, +automobile
- +pit, +cockfights
- +conflict

Very little overlaps.

So can we compare them?

Encode features as 1 or 0

	chair	cockpit
one-person	1	0
backrest	1	0?
four-legs	1	0
support	1	0?
armrest	1	0?
authority	1	0?
enclosed	0	1
airplane	0	1
seats	0?	1
...		

Similarity

- What we've just defined is a vector space.

Similarity

- What we've just defined is a vector space.
- Dimension = feature. So far it's a low-dimensional space.

Similarity

- What we've just defined is a vector space.
- Dimension = feature. So far it's a low-dimensional space.
- How can we measure the similarity between them? Common answer: cosine similarity.

Similarity

- What we've just defined is a vector space.
- Dimension = feature. So far it's a low-dimensional space.
- How can we measure the similarity between them? Common answer: cosine similarity.

Cosine similarity

$\mathrm{sim}(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|}$

Similarity

- What we've just defined is a vector space.
- Dimension = feature. So far it's a low-dimensional space.
- How can we measure the similarity between them? Common answer: cosine similarity.

Cosine similarity

$$\mathrm{sim}(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|}$$

- So what would the similarity of "chair" and "cockpit" be in our space? Probably zero!

Words in terms of other words

We need a new data source. Collect it from a real corpus. Let's try Google.

Words in terms of other words

We need a new data source. Collect it from a real corpus. Let's try Google.

	chair	cockpit
one-person		
backrest		
four-legs		
support		
armrest		
authority		
enclosed		
airplane		
seats		
...		

Words in terms of other words

We need a new data source. Collect it from a real corpus. Let's try Google.

	chair	cockpit
one-person		
backrest		
four-legs		
support		
armrest		
authority		
enclosed		
airplane		
seats		
...		

Now it's not so bad: we can get a non-zero similarity. Yay?

Words in terms of other words

- In fact, rather than using dictionary definitions of explicit features, cut out the middle man.

Words in terms of other words

- In fact, rather than using dictionary definitions of explicit features, cut out the middle man.
- "Learn" a vector for each word by counting corpus context. Ways of learning:
  - Simple co-occurrence counts based on a window.
    - The vocabulary basically becomes the feature space.

Words in terms of other words

- In fact, rather than using dictionary definitions of explicit features, cut out the middle man.
- "Learn" a vector for each word by counting corpus context. Ways of learning:
  - Simple co-occurrence counts based on a window.
    - The vocabulary basically becomes the feature space.
  - More complex counts, such as POS tags, bits of parse trees.

Words in terms of other words

- In fact, rather than using dictionary definitions of explicit features, cut out the middle man.
- "Learn" a vector for each word by counting corpus context. Ways of learning:
  - Simple co-occurrence counts based on a window.
    - The vocabulary basically becomes the feature space.
  - More complex counts, such as POS tags, bits of parse trees.
- Sometimes raw counts aren't what you need: smoothing, reweighting.

Words in terms of other words

These are “count” vectors. What are the problems with doing it this way?

Words in terms of other words

These are “count” vectors. What are the problems with doing it this way?

- Sparsity: many words just never appear with other words.

Words in terms of other words

These are "count" vectors. What are the problems with doing it this way?

- Sparsity: many words just never appear with other words.
- Dimensionality: especially if you use fancy features (syntax, etc), you get million dimensional spaces.

Words in terms of other words

These are "count" vectors. What are the problems with doing it this way?

- Sparsity: many words just never appear with other words.
- Dimensionality: especially if you use fancy features (syntax, etc), you get million dimensional spaces.

What we need? Dimensionality reduction, or some other way to start from a compressed space.

Words in terms of other words

These are “count” vectors. What are the problems with doing it this way?

- Sparsity: many words just never appear with other words.
- Dimensionality: especially if you use fancy features (syntax, etc), you get million dimensional spaces.

What we need? Dimensionality reduction, or some other way to start from a compressed space.

- Sharing dimensions helps generalization.

Words in terms of other words

These are “count” vectors. What are the problems with doing it this way?

- Sparsity: many words just never appear with other words.
- Dimensionality: especially if you use fancy features (syntax, etc), you get million dimensional spaces.

What we need? Dimensionality reduction, or some other way to start from a compressed space.

- Sharing dimensions helps generalization.
- Nevertheless, there’s value in count vectors (for things that require explicit linguistic knowledge)

Words in terms of other words

These are "count" vectors. What are the problems with doing it this way?

- Sparsity: many words just never appear with other words.
- Dimensionality: especially if you use fancy features (syntax, etc), you get million dimensional spaces.

What we need? Dimensionality reduction, or some other way to start from a compressed space.

- Sharing dimensions helps generalization.
- Nevertheless, there's value in count vectors (for things that require explicit linguistic knowledge)

So now... "predict" vectors...

Words as Integers

- Our previous representations of words (and word classes) have been fairly flat
- For example, the word ‘*monkey*’ can be represented as an integer, such as ‘7’

Words as Integers

- Our previous representations of words (and word classes) have been fairly flat
- For example, the word '*monkey*' can be represented as an integer, such as '7'
- **One-hot encoding** represents that as:

0	0	0	0	0	0	1	0	0	0	...	0

Words as Integers

- Our previous representations of words (and word classes) have been fairly flat
- For example, the word '*monkey*' can be represented as an integer, such as '7'
- **One-hot encoding** represents that as:

0	0	0	0	0	0	1	0	0	0	...	0

- and the word class (eg. 2) containing '*monkey*':

0	1	0	0

Words as Integers

- Our previous representations of words (and word classes) have been fairly flat
- For example, the word '*monkey*' can be represented as an integer, such as '7'
- **One-hot encoding** represents that as:

0	0	0	0	0	0	1	0	0	0	...	0

- and the word class (eg. 2) containing '*monkey*':

0	1	0	0

- Both of these are sparse vectors of booleans, with just one entry having a 'true' value

Words as Integers

- Our previous representations of words (and word classes) have been fairly flat
- For example, the word '*monkey*' can be represented as an integer, such as '7'
- **One-hot encoding** represents that as:

0	0	0	0	0	0	1	0	0	0	...	0

- and the word class (eg. 2) containing '*monkey*':

0	1	0	0

- Both of these are sparse vectors of booleans, with just one entry having a 'true' value
- Either way, we're working with integers (..., -2, -1, 0, 1, 2, ...)

Words as $\mathbb{R}$eal Numbers

- We can do more with real numbers (eg. -1.5, 0.23, 55.01)

Words as $\mathbb{R}$eal Numbers

- We can do more with real numbers (eg. -1.5, 0.23, 55.01)
- We can represent the word '*monkey*' as a dense vector of real numbers:

0.38	-1.27	-0.55	1.44

Words as $\mathbb{R}$eal Numbers

- We can do more with real numbers (eg. -1.5, 0.23, 55.01)
- We can represent the word '*monkey*' as a dense vector of real numbers:

0.38	-1.27	-0.55	1.44

- We can have the plural form, '*monkeys*' be close in that vector space:

0.31	-1.27	**-0.61**	1.44

Words as $\mathbb{R}$eal Numbers

- We can do more with real numbers (eg. -1.5, 0.23, 55.01)
- We can represent the word '*monkey*' as a dense vector of real numbers:

0.38	-1.27	-0.55	1.44

- We can have the plural form, '*monkeys*' be close in that vector space:

0.31	-1.27	**-0.61**	1.44

- We can also have a related word, like '*ape*' be close in that vector space, *but in different dimensions*:

0.38	**-1.33**	-0.55	**1.49**

Applications of Word Vectors

- **Word distances**. For example, closest words to '*Sweden*':

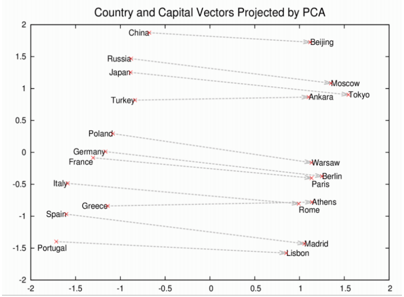
Word	Cosine Distance
Norway	0.75
Denmark	0.72
Finland	0.62
Switzerland	0.59
...	

Applications of Word Vectors

- **Word distances**. For example, closest words to '*Sweden*':

Word	Cosine Distance
Norway	0.75
Denmark	0.72
Finland	0.62
Switzerland	0.59
...	

- **Analogy**. E.g., *Japan* is to *Tokyo* as *Germany* is to *Berlin*

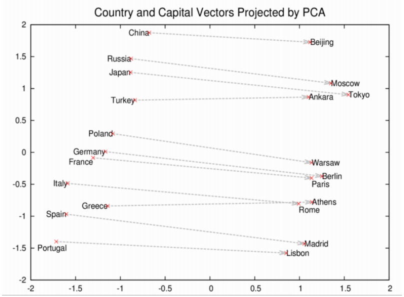


Applications of Word Vectors

- **Word distances**. For example, closest words to '*Sweden*':

Word	Cosine Distance
Norway	0.75
Denmark	0.72
Finland	0.62
Switzerland	0.59
...	

- **Analogy**. E.g., *Japan* is to *Tokyo* as *Germany* is to *Berlin*



Japan – Tokyo ≈ Germany – Berlin

Applications of Word Vectors

- **Sentence Completion** (actually just restricted language modeling):
- "All red-headed men who are above the age of [800 | seven | twenty-one | 1,200 | 60,000] years , are eligible."
- "That is his [generous | mother's | successful | favorite | main] fault , but on the whole he's a good worker."

Applications of Word Vectors

- **Sentence Completion** (actually just restricted language modeling):
- "All red-headed men who are above the age of [800 | seven | twenty-one | 1,200 | 60,000] years , are eligible."
- "That is his [generous | mother's | successful | favorite | main] fault , but on the whole he's a good worker."
- Mikolov et al (2013b) selected the test word that best predicted the context

Kinds of vector spacess

Vector spaces can be divided into two overall categories:

Kinds of vector spacess

Vector spaces can be divided into two overall categories:

- Count-based
  - Corpus counts (however chosen) are either taken "literally" or adjusted by an information statistic: pointwise mutual information, local mutual information, tf/idf, etc.

Kinds of vector spacess

Vector spaces can be divided into two overall categories:

- Count-based
  - Corpus counts (however chosen) are either taken “literally” or adjusted by an information statistic: pointwise mutual information, local mutual information, tf/idf, etc.
- Prediction-based
  - Counts are readjusted by applying machine learning techniques to “compress” the data (a form of dimensionality reduction. . .)
  - Word contexts no longer necessarily human-comprehensible.

Those were fairly fashionable NLP uses of vector spaces, but. . .

. . . classification can also be seen as a vector space problem.

Take the contexts seriously – as "documents".

. . . classification can also be seen as a vector space problem.

Take the contexts seriously – as “documents”.

- Another word for the collection of vectors: a “term-document matrix.”

...classification can also be seen as a vector space problem.

Take the contexts seriously – as “documents”.

- Another word for the collection of vectors: a “term-document matrix.”
  - “Document” can mean even just a few words context, for example.

. . . classification can also be seen as a vector space problem.

Take the contexts seriously – as "documents".

- Another word for the collection of vectors: a "term-document matrix."
    - "Document" can mean even just a few words context, for example.
- Instances are contexts/documents, no longer words.

. . . classification can also be seen as a vector space problem.

Take the contexts seriously – as "documents".

- Another word for the collection of vectors: a "term-document matrix."
  - "Document" can mean even just a few words context, for example.
- Instances are contexts/documents, no longer words.
- Words (and anything else) are features of the document.

. . . classification can also be seen as a vector space problem.

Take the contexts seriously – as “documents”.

- Another word for the collection of vectors: a “term-document matrix.”
  - “Document” can mean even just a few words context, for example.
- Instances are contexts/documents, no longer words.
- Words (and anything else) are features of the document.
- Classification problem: finding a hyperplane that divides up the space.

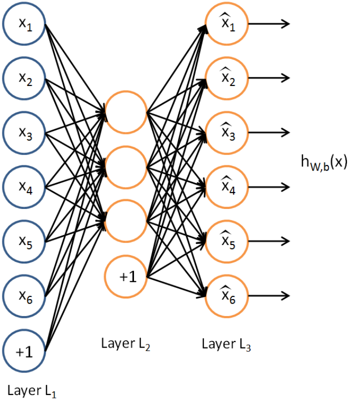
Now we start getting ahead of ourselves, so bear with me. . .

. . . but back to dimensionality reduction!

Part 3: Beyond LSA

Autoencoder

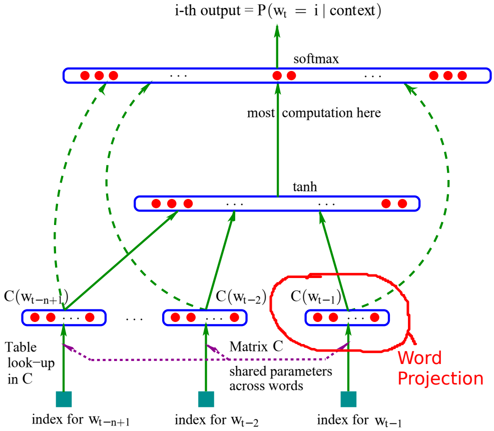
From Stanford deep learning tutorial:



Learn compressed representation of the input by learning the identity function via a neural network.

Projection Layer in Neural Language Models

- **Neural Language Modeling** – this was actually one of the earliest uses of word vectors. We'll talk more about these next semester



word2vec

- Tomáš Mikolov and colleagues found that you don't need the full neural-net language model to get useful word vectors

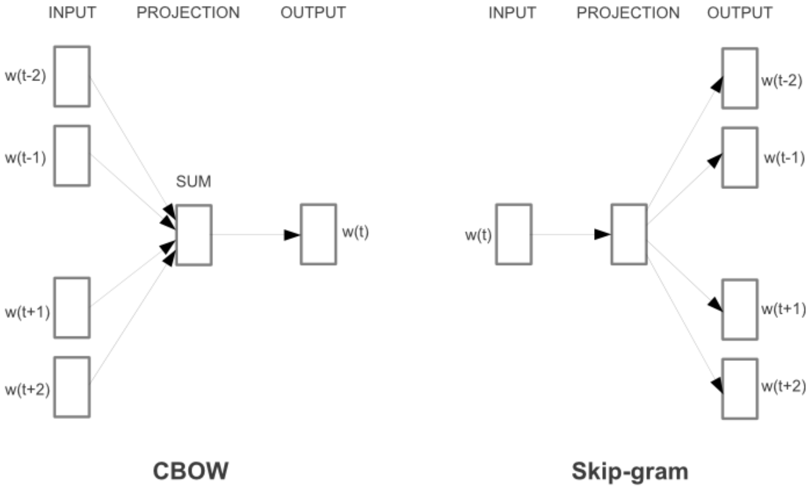
word2vec

- Tomáš Mikolov and colleagues found that you don't need the full neural-net language model to get useful word vectors
- In fact, you don't need a neural network at all. He removed the hidden layer, giving a traditional log-linear model

word2vec

- Tomáš Mikolov and colleagues found that you don't need the full neural-net language model to get useful word vectors
- In fact, you don't need a neural network at all. He removed the hidden layer, giving a traditional log-linear model
- He developed a simplified form of training called negative sampling (derived from earlier NCE). It's a little like a binary MaxEnt classifier

word2vec: CBOW & Skip-gram



Hyperparameters

(What is a parameter? Usually, the model weights. Example hyperparameter: how many parameters. . .)

Hyperparameters

(What is a parameter? Usually, the model weights. Example hyperparameter: how many parameters...)

- Window size: how much surrounding context to use
- Normalization: softmax (traditional) vs. hierarchical softmax vs. negative sampling
- Vector dimensions: 100–500 common
- Number of negative samples: 3–10 common
- Number of training epochs, initial learning rate, negative sample distribution ($\alpha = 0.75$), model, ...

Part 4: an application in distributional semantics

World knowledge

Is a systematic study of world knowledge possible?

World knowledge

Is a systematic study of world knowledge possible?

- Yes: let's look at work on generalized event knowledge [McRae and Matsuki, 2009]:

World knowledge

Is a systematic study of world knowledge possible?

- Yes: let's look at work on generalized event knowledge [McRae and Matsuki, 2009]:
    - Prototypical knowledge of events and their participants

World knowledge

Is a systematic study of world knowledge possible?

- Yes: let's look at work on generalized event knowledge [McRae and Matsuki, 2009]:
  - Prototypical knowledge of events and their participants
    - Acquired from first- and second-hand experience, i.e., from language too, available in our memory

World knowledge

Is a systematic study of world knowledge possible?

- Yes: let's look at work on generalized event knowledge [McRae and Matsuki, 2009]:
  - Prototypical knowledge of events and their participants
    - Acquired from first- and second-hand experience, i.e., from language too, available in our memory
  - Activated by words in isolation, which cue concepts from typical scenarios (e.g. going to doctor, eating in restaurant).

World knowledge

Is a systematic study of world knowledge possible?

- Yes: let's look at work on generalized event knowledge [McRae and Matsuki, 2009]:
  - Words rapidly combine to generate expectations about upcoming input.
    - e.g., Donna used the hose to wash her filthy . . . **car**/hair

World knowledge

Is a systematic study of world knowledge possible?

- Yes: let's look at work on generalized event knowledge [McRae and Matsuki, 2009]:
  - Words rapidly combine to generate expectations about upcoming input.
    - e.g., Donna used the hose to wash her filthy . . . **car**/hair
  - **Thematic fit**: the typicality of a filler for a given semantic argument slot.
    - . . . e.g., "car" should have a higher thematic fit than "hair" in the above example.

World knowledge

Is a systematic study of world knowledge possible?

- Yes: let's look at work on generalized event knowledge [McRae and Matsuki, 2009]:
  - Words rapidly combine to generate expectations about upcoming input.
    - e.g., Donna used the hose to wash her filthy . . . **car**/hair
  - **Thematic fit**: the typicality of a filler for a given semantic argument slot.
    - . . . e.g., "car" should have a higher thematic fit than "hair" in the above example.

Possible to make predictions and verify hypotheses regarding world knowledge and its role in linguistic processing.

General knowledge cued by the verb

Let's make it more concrete:

General knowledge cued by the verb

Let's make it more concrete:

- People rapidly integrate various types of semantic and syntactic knowledge

General knowledge cued by the verb

Let's make it more concrete:

- People rapidly integrate various types of semantic and syntactic knowledge
- Verb meaning and situation structure: relations among the entities that commonly participate in the event

General knowledge cued by the verb

Let's make it more concrete:

- People rapidly integrate various types of semantic and syntactic knowledge
- Verb meaning and situation structure: relations among the entities that commonly participate in the event
- A thematic role is a concept formed through everyday experiences (people learning who and what play specific roles in specific situations)

General knowledge cued by the verb

Let's make it more concrete:

- People rapidly integrate various types of semantic and syntactic knowledge
- Verb meaning and situation structure: relations among the entities that commonly participate in the event
- A thematic role is a concept formed through everyday experiences (people learning who and what play specific roles in specific situations)
  - We learn about *accusing* and its agent role from our experiences with people who accuse others and linguistic descriptions of them

General knowledge cued by the verb

Let's make it more concrete:

- People rapidly integrate various types of semantic and syntactic knowledge
- Verb meaning and situation structure: relations among the entities that commonly participate in the event
- A thematic role is a concept formed through everyday experiences (people learning who and what play specific roles in specific situations)
  - We learn about *accusing* and its agent role from our experiences with people who accuse others and linguistic descriptions of them
- Does reading or hearing a verb result in the immediate computation of information regarding typical agents, patients, instruments and locations?

. . . and that heavily with reference to events

What does an event consist of?

. . . and that heavily with reference to events

What does an event consist of?

- A predicate – whatever is happening (or in some cases the description of a state)

. . . and that heavily with reference to events

What does an event consist of?

- A predicate – whatever is happening (or in some cases the description of a state)
- Participants – the objects/entities/abstract constructs that make the predicate specific.

...and that heavily with reference to events

What does an event consist of?

- A predicate – whatever is happening (or in some cases the description of a state)
- Participants – the objects/entities/abstract constructs that make the predicate specific.

Participants are usually defined by "thematic" or semantic roles.

. . . and that heavily with reference to events

What does an event consist of?

- A predicate – whatever is happening (or in some cases the description of a state)
- Participants – the objects/entities/abstract constructs that make the predicate specific.

Participants are usually defined by “thematic” or semantic roles.

- Traditionally: agent, patient, goal, etc.
- Some roles are “required” by particular events (often agents and patients for transitive verbs), most are “adjuncts” (locations, instruments, etc.)

Thematic fit

A challenge in building computational models of events.

Thematic fit

A challenge in building computational models of events.

The thematic fit problem

Given a verb/event-type v, an entity x, how well does v fit x in role r?

Thematic fit

A challenge in building computational models of events.

The thematic fit problem

Given a verb/event-type v, an entity x, how well does v fit x in role r?

We ask typically ask humans to give us this data.

Thematic fit

A challenge in building computational models of events.

The thematic fit problem

Given a verb/event-type v, an entity x, how well does v fit x in role r?

We ask typically ask humans to give us this data.

- Need to get ratings. Possible questions:
  - "How common is it for a cake to bake something?" (agent)
  - "An oven is something you can use for baking." (instrument)

  Rate from 1-7.

Thematic fit

A challenge in building computational models of events.

The thematic fit problem

Given a verb/event-type v, an entity x, how well does v fit x in role r?

We ask typically ask humans to give us this data.

- Need to get ratings. Possible questions:
  - “How common is it for a cake to bake something?” (agent)
  - “An oven is something you can use for baking.” (instrument)

  Rate from 1-7.

(How you ask actually makes things complicated…)

Agent/patient (subj/obj) ratings

Verb	Noun	Semantic role	Score
advise	doctor	subj	6.8
advise	doctor	obj	4.0
confuse	baby	subj	3.7
confuse	baby	obj	6.0
eat	lunch	subj	1.1
eat	lunch	obj	6.9
kill	lion	subj	2.7
kill	lion	obj	4.9

Data source for thematic fit norms

Here are some widely available thematic fit rating sources.

Data source for thematic fit norms

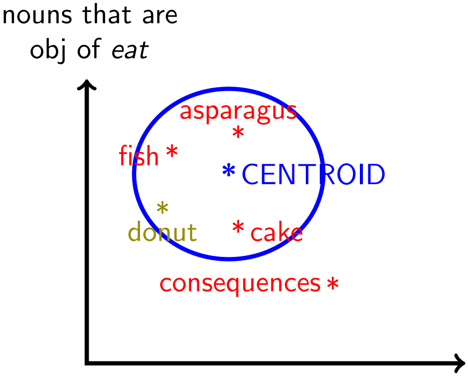
Here are some widely available thematic fit rating sources.

- Padó agent/patient ratings
  - Balanced rating set of 18 verbs with 12 nouns each extracted from WSJ.
- McRae agent/patent ratings: 1444 ratings, unbalanced
- Ferretti et al.: instruments (248) and locations (274).
- Greenberg et al.: patients balanced for number of senses (from WordNet).

Now assume for a moment that we have a vector space.

How to evaluate thematic fit with a vector space

Query: how good is “donut” as an object of “eat”?

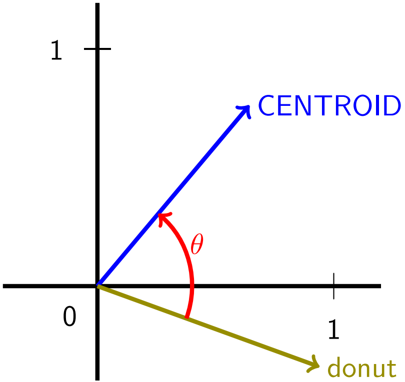


(Special thanks to A. Zarcone.)

Find an average vector (centroid) based on 20 nouns that are typical “eat”-objects.

Thematic fit measurement

Query: how good is “donut” as an object of “eat”?



Then take the cosine of “donut” with the centroid.

So assuming a cosine similarity approach. . .

. . . what do we need to build a vector space of this kind?

So assuming a cosine similarity approach. . .

. . . what do we need to build a vector space of this kind?

- At minimum, a space that allows us to assess most frequent fillers of given role.

So assuming a cosine similarity approach. . .

. . . what do we need to build a vector space of this kind?

- At minimum, a space that allows us to assess most frequent fillers of given role.
- Bonus: space that gives us semantically-relevant features to compare.

So assuming a cosine similarity approach. . .

. . . what do we need to build a vector space of this kind?

- At minimum, a space that allows us to assess most frequent fillers of given role.
- Bonus: space that gives us semantically-relevant features to compare.

First let's try a count space . . .

Distributional Memory

Baroni and Lenci [2010]: Distributional Memory (DM) approach:

Distributional Memory

Baroni and Lenci [2010]: Distributional Memory (DM) approach:

1. Parse entire corpus (using MaltParser).

Distributional Memory

Baroni and Lenci [2010]: Distributional Memory (DM) approach:

1. Parse entire corpus (using MaltParser).
2. Read into data structure (order-3 tensor) as counts of $<$*word0, link, word1*$>$ dimensions.
   - Where "link" is a feature derived from a dependency between "word0" and "word1".

Distributional Memory

Baroni and Lenci [2010]: Distributional Memory (DM) approach:

1. Parse entire corpus (using MaltParser).
2. Read into data structure (order-3 tensor) as counts of $<$*word0, link, word1*$>$ dimensions.
   - Where "link" is a feature derived from a dependency between "word0" and "word1".
3. Reweight counts with Local Mutual Information (LMI).

Local mutual information

$$O \log \frac{O}{E}$$

where O is observed counts of triples in corpus and E is counts expected under independence of words and links.

Distributional Memory

Baroni and Lenci [2010]: Distributional Memory (DM) approach:

1. Parse entire corpus (using MaltParser).
2. Read into data structure (order-3 tensor) as counts of <*word0, link, word1*> dimensions.
   - Where "link" is a feature derived from a dependency between "word0" and "word1".
3. Reweight counts with Local Mutual Information (LMI).

Local mutual information

$$O \log \frac{O}{E}$$

where O is observed counts of triples in corpus and E is counts expected under independence of words and links.

This process results in a tensor space of tens of millions of dims.

What are the feature spaces like?

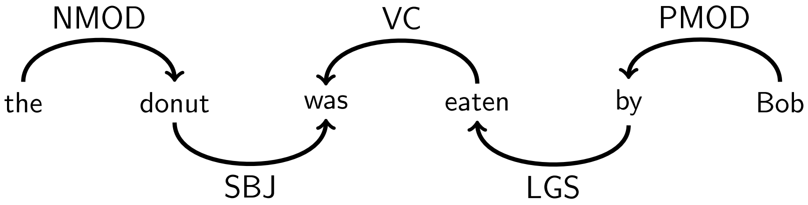
Baroni and Lenci come up with three different tensors:

- DepDM – Raw dependencies from MaltParser, adjusted in process similar to ours.
- LexDM – Lexicalized links based on DepDM, expanded by handcrafted rules.
- TypeDM (publicly available) – Counts reflect *number of types* of links in LexDM, rather than raw counts.

Corpora: UKWAC, WackyPedia, BNC.

TypeDM feature space

Baroni and Lenci's TypeDM model: "semantic" features hand-crafted from syntactic dependencies.



Donut

A small excerpt of a Baroni and Lenci DM

	$\langle$*verb,bomb*$\rangle$	$\langle$*subj,kill*$\rangle$	$\langle$*verb,gun*$\rangle$	$\langle$*subj,shoot*$\rangle$	$\langle$*verb,book*$\rangle$	$\langle$*subj,read*$\rangle$
marine	40.0	82.1	85.3	44.8	3.2	3.3
teacher	5.2	7.0	9.3	4.7	48.4	53.6

How well does all of this work?

Evaluation via Spearman's ρ.
(Rank-based correlation – is this a good idea?)

How well does all of this work?

Evaluation via Spearman's ρ.
(Rank-based correlation – is this a good idea?)

- Average human agreement $= 68$ on Padó data.

How well does all of this work?

Evaluation via Spearman's ρ.
(Rank-based correlation – is this a good idea?)

- Average human agreement = 68 on Padó data.
- TypeDM on Padó agent/patient: Spearman's ρ correlation: 53

How well does all of this work?

Evaluation via Spearman's ρ.
(Rank-based correlation – is this a good idea?)

- Average human agreement $= 68$ on Padó data.
- TypeDM on Padó agent/patient: Spearman's ρ correlation: 53

Other roles do significantly worse. (e.g. Ferretti locations get 23)...

Accounting for quality: visualization

To put a long story short...

I and some colleagues did some years of work on investigating these spaces.

- Hard to account for why some things work and some don't.

Accounting for quality: visualization

To put a long story short. . .

I and some colleagues did some years of work on investigating these spaces.

- Hard to account for why some things work and some don't.
- Value for visualization: investigate linguistic relationships *ad hoc* for error analysis, etc.

Accounting for quality: visualization

To put a long story short...
I and some colleagues did some years of work on investigating these spaces.

- Hard to account for why some things work and some don't.
- Value for visualization: investigate linguistic relationships *ad hoc* for error analysis, etc.
- Need to project the space down to two or three dimensions to visualize.

Accounting for quality: visualization

To put a long story short...
I and some colleagues did some years of work on investigating these spaces.

- Hard to account for why some things work and some don't.
- Value for visualization: investigate linguistic relationships *ad hoc* for error analysis, etc.
- Need to project the space down to two or three dimensions to visualize.

Hence, "Roleo": `http://roleo.coli.uni-saarland.de`