CSE 519: Data Science
Steven Skiena
Stony Brook University

Lecture 22: Topics in Machine Learning

The World of Many Weak Features

Often we have many relatively weak features to apply to a classification problem.

In text classification problems, we often have the frequency of each word in documents of positive and negative classes: e.g. the frequency of ``sale” in spam and real email.

Bayesian Classifiers

To classify a vector $X = (x_1, \ldots x_n)$ into one of *m* classes, we can use Bayes Theorem:

$$p(C_i|X) = \frac{p(C_i)p(X|C_i)}{p(X)}$$

This reduces decisions about the class given the input to the input given the class.

Identifying the Most Probable Class

Argmax is the class with the highest probability:

$$C(X) = \max_{i=1}^{m} \frac{p(C_i)p(X|C_i)}{p(X)} = \max_{i=1}^{m} p(C_i)p(X|C_i)$$

P(C_i) is the prior probability of class *i*.

P(X) is the probability of seeing input *X* over all classes. This is dicey, but can be ignored for classification because it is constant.

Tabulation Yields Marginal Probabilities

Day	Outlook	Temp	Humidity	Beach?
1	Sunny	High	High	Yes
2	Sunny	High	Normal	Yes
3	Sunny	Low	Normal	No
4	Sunny	Mild	High	Yes
5	Rain	Mild	Normal	No
6	Rain	High	High	No
7	Rain	Low	Normal	No
8	Cloudy	High	High	No
9	Cloudy	High	Normal	Yes
10	Cloudy	Mild	Normal	No

P(X|Class)	Probability in Class	
Outlook	Beach	No Beach
Sunny	3/4	1/6
Rain	0/4	3/6
Cloudy	1/4	2/6
Temperature	Beach	No Beach
High	3/4	2/6
Mild	1/4	2/6
Low	0/4	2/6
Humidity	Beach	No Beach
High	2/4	2/6
Normal	2/4	4/6
P(Beach Day)	4/10	6/10

Independence and Naive Bayes

But what is $P(X|C)$, where X is a complex feature vector?

If (a,b) are independent, then $P(ab)=P(a)\ P(b)$

This calculation is much simpler than factoring in correlations and interactions of multiple factors, but:

What's the probability of having two size 9 feet?

Complete Naive Bayes Formulation

We seek the argmax of:

$$C(X) = \max_{i=1}^{m} p(C_i)p(X|C_i) = \max_{i=1}^{m} p(C_i) \prod_{j=1}^{n} p(x_j|C_i)$$

Multiplying many probabilities is bad, so:

$$C(X) = \max_{i=1}^{m}(\log(p(C_i)) + \sum_{j=1}^{n} \log(p(x_j|C_i)))$$

Is a Sunny-Mild-High a Beach Day?

$$
\begin{aligned}
&P(\text{Beach}|(\text{Sunny,Mild,High})) \\
&\qquad = (P(\text{Sunny}|\text{Beach}) \times P(\text{Mild}|\text{Beach}) \times P(\text{High}|\text{Beach}) \times P(\text{Beach}) \\
&\qquad = (3/4) \times (1/4) \times (2/4) \times (4/10) = 0.0375
\end{aligned}
$$

$$
\begin{aligned}
&P(\text{No Beach}|(\text{Sunny,Mild,High})) \\
&\qquad = (P(\text{Sunny}|\text{No}) \times P(\text{Mild}|\text{No}) \times P(\text{High}|\text{No})) \times P(\text{No}) \\
&\qquad = (1/6) \times (2/6) \times (2/6) \times (6/10) = 0.0111
\end{aligned}
$$

Dealing with Zero Counts

You may never have seen it before, but what is the probability my next word is defenestrate?

Observed counts do not accurately capture the frequency of rare events, for which there is typically a long tail.

Laplace asked: “What is the probability the sun will rise tomorrow?”

+1 Discounting

Discounting is a statistical technique to adjust counts for yet-as-unseen events.

The simplest technique is add one discounting, where we add one to the frequency all outcomes, including unseen.

Thus after seeing 5 reds and 3 greens, P(new-color)=1/((5+1)+(3+1)+(0+1)) = 1/11

Feature Engineering

Domain-dependent data cleaning is important:

- Z-scores and normalization
- Creating bell-shaped distributions.
- Imputing missing values
- Dimension reduction, like SVD
- Explicit incorporation of non-linear combinations like products and ratios.

Commissions on Art Auctions

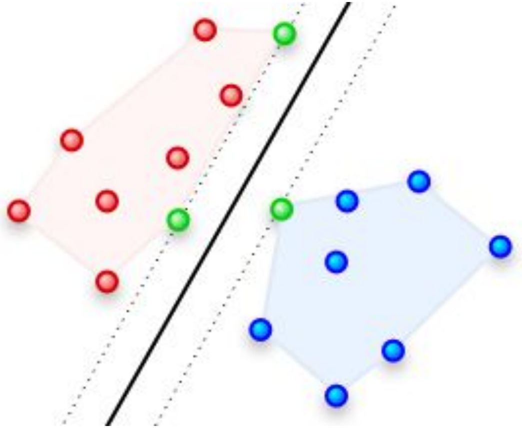
When you buy a painting at an auction, you pay the house a specified percentage as a fee.

How is this best represented as a feature?

- The commission percentage (e.g. *10%*)
- The actual commission paid *(0.1*1M=$100k)*
- Change the target variable from hammer price to total amount paid: *($33M to $36.3M)*

Support Vector Machines

SVMs are an important way to build non-linear classifiers.

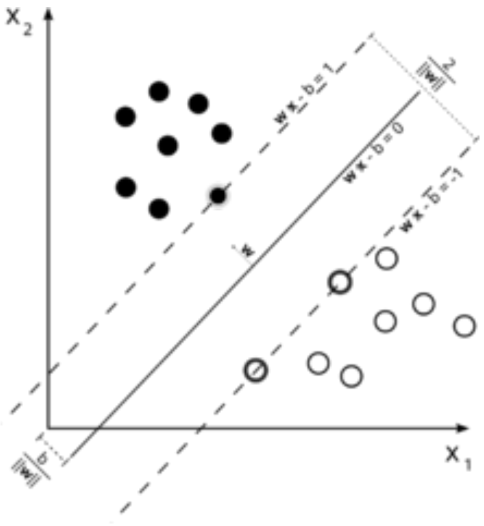


They work by seeking maximum margin linear separators between the two classes.

Optimization Problem

Optimize the coefficient size $\|\mathbf{w}\|$ subject to the constraints $y_i(\mathbf{w}\cdot\mathbf{x_i} - b) \geq 1.$ for all $i = 1, \ldots, n$

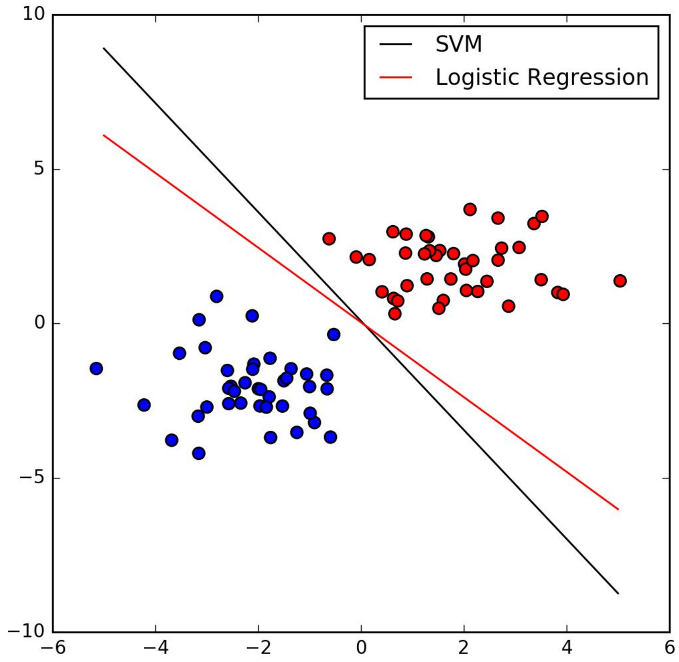
Only a few points touch the boundary of the separating channel.
Near-vertical lines are closer than horizontal lines even b +/- 1 are 2 apart, hence minimizing on ||w||.



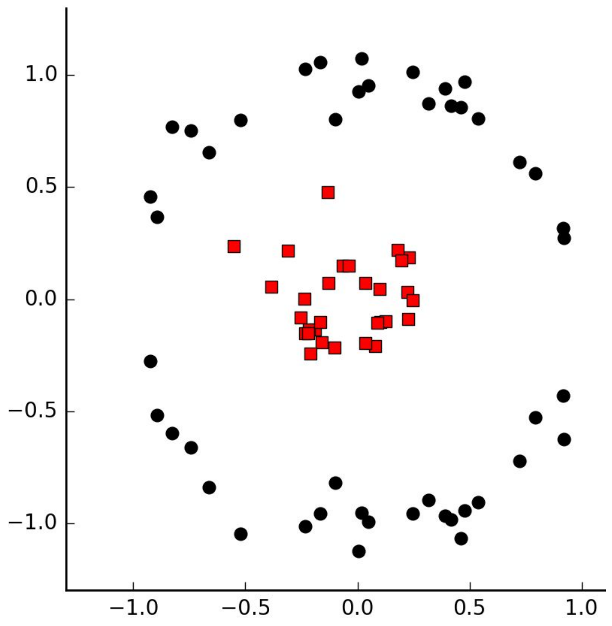
SVMs vs. Logistic Regression

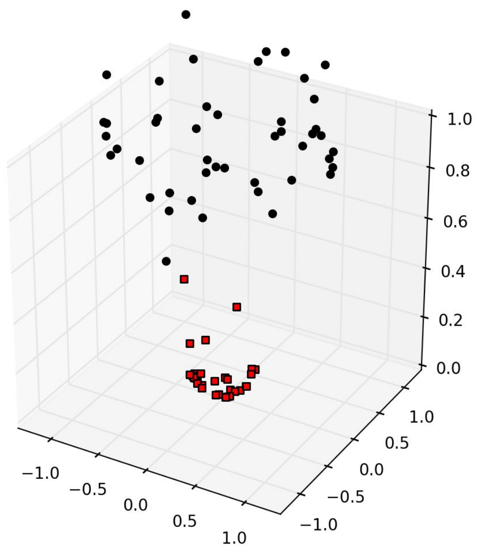
Both methods find separating planes, but different ones.

LR values all points, but SVM only the points at the boundary.



Projecting to Higher Dimensions





Adding enough dimensions makes everything linearly separable.

Here (x,y) ->(x,y,x^2+y^2) does the job.

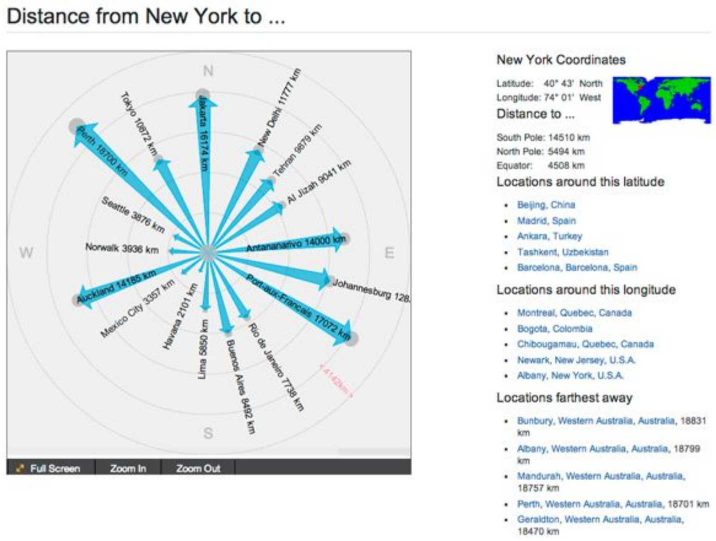
Efficient solvers like LibSVM are available for this.

Projecting to Higher Dimensions

The non-linearity depends upon how space is projected to higher dimensions.

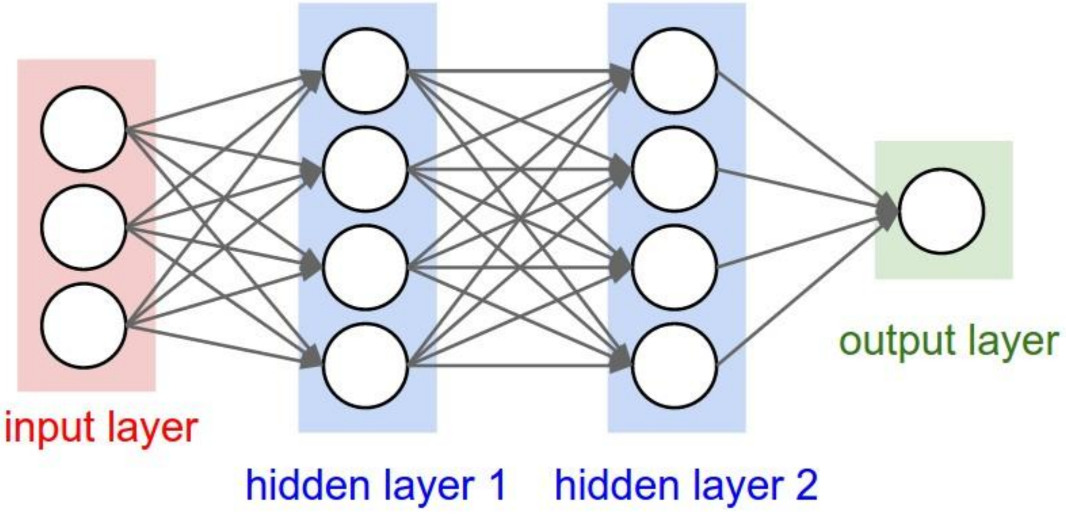
The distance from all *n* input points to the target creates an n-dimensional feature vector.

Kernal functions give the power to use such features efficiently, without building the n*n matrix.



Neural Networks / Deep Learning

The hottest area of machine learning today involves large, deep neural network architectures.



Basic Principles of Deep Learning

- That the weight of each edge is a distinct parameter means large networks exploits large training sets (*like nearest neighbors..*)
- The depth of the networks means they can build up hierarchical representations of features: e.g. pixels, edges, regions, objects
- Toolkits like TensorFlow make it easy to build DL models if you have the data.

Node Computations

Each node in the network typically computes a nonlinear function Phi(v) of a weighted input sum:

$$v_i = \beta + \sum_i w_i x_i$$

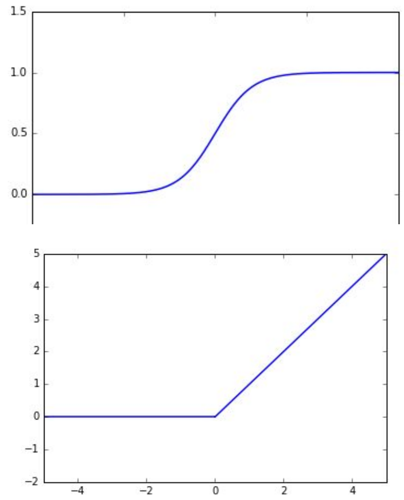
The beta term is the bias, the activation in the absence of input.

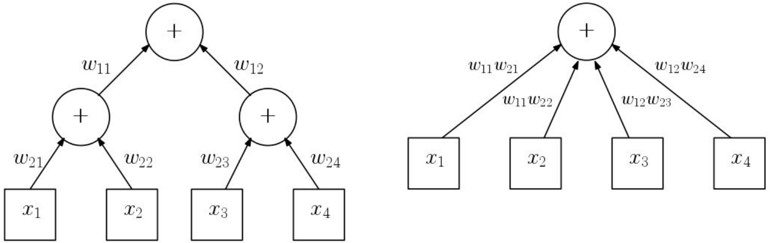
Many dot products implies matrix multiplication!

Non-Linearity

The logit and RELU functions make good candidates for Phi.

Linear function like addition cannot exploit depth, because hidden layers add no power.





Backpropagation

NNs are trained by a stochastic gradient descent-like algorithm, with changes for each training example pushed down to lower levels.

Non-linear functions result in a non-convex optimization function, but this generally produces good results.

Word Embeddings

One NN application I have found particularly useful is word2vec, constructing 100 dimensional word representations from text corpora.

The goal is to try to predict missing words by context: We would **** to improve

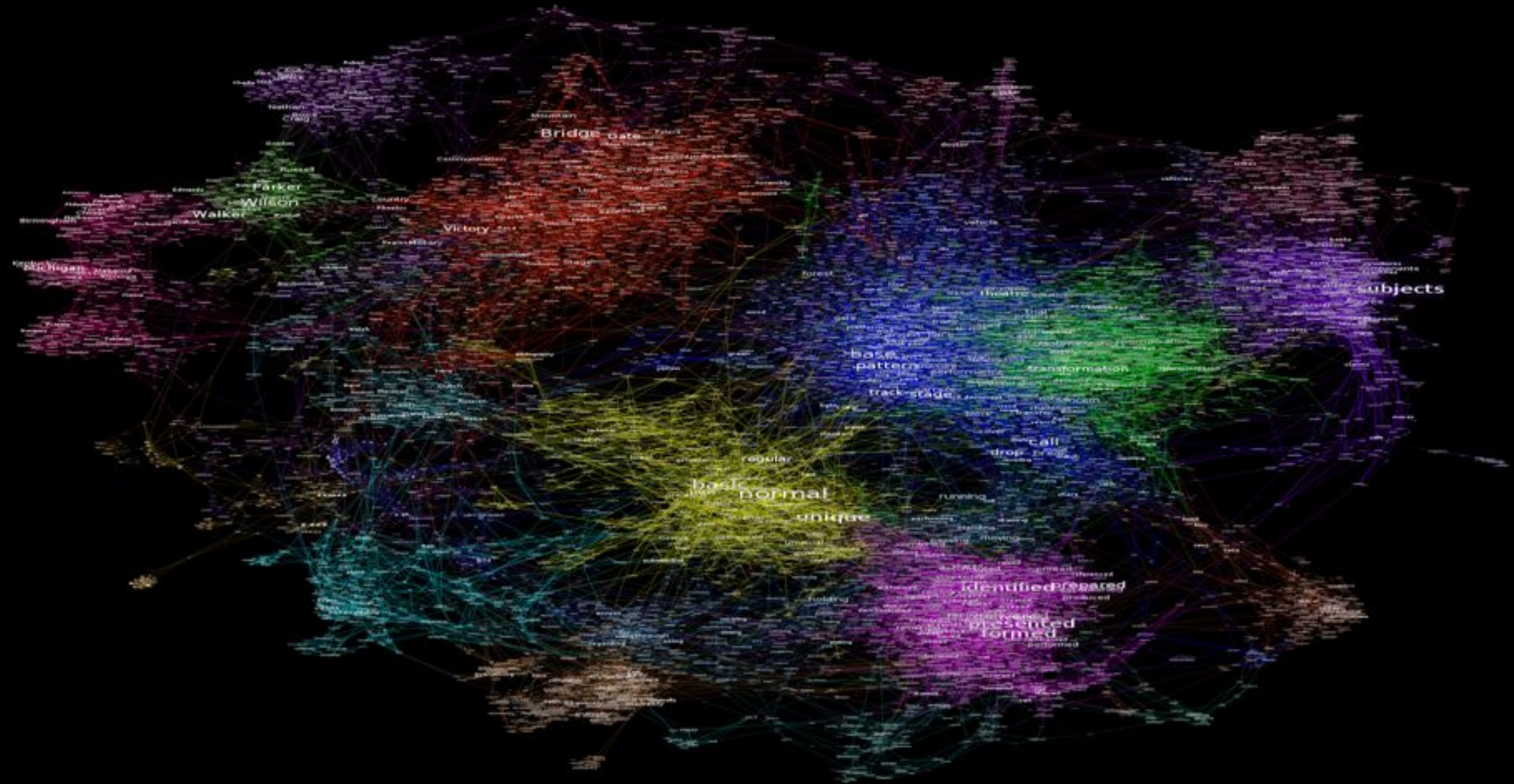
Thus large volumes of training data can be construction from text without supervision.

Nearest Neighbors in Embeddings

	Word	Translation
French	**rouge**	**red**
	juane	yellow
	rose	pink
	blanc	white
	orange	orange
	bleu	blue
Arabic	**اركش**	**thanks**
	اركشو	and thanks
	يتايحت	greetings
	اركش	thanks + diacritic
	اركشو	and thanks + diacritic
	ابحرم	hello
Russian	**Путин**	**Putin**
	Янукович	Yanukovych
	Троцкий	Trotsky
	Гитлер	Hitler
	Сталин	Stalin
	Медведев	Medvedev

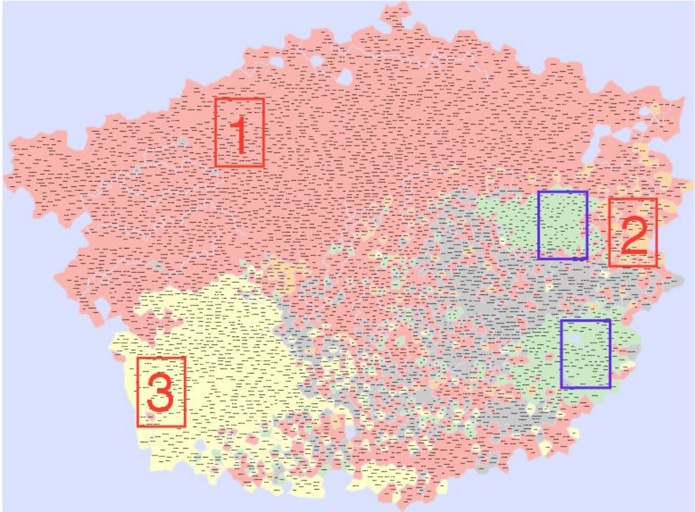
	Word	Translation
Spanish	**dentista**	**dentist**
	peluquero	barber
	ginecólog	gynecologist
	camionero	truck driver
	oftalmólogo	ophthalmologist
	telegrafista	telegraphist
Arabic	**نادلو**	**two boys**
	نانبا	two sons
	نيدلو	two boys
	نلافط	two children
	نينبا	two sons
	ناتنبا	two daughters
	Transliteration	
Chinese	**dongzhi**	**Winter Solstice**
	chunfen	Vernal Equinox
	xiazhi	Summer solstice
	qiufen	Autumnal Equinox
	ziye	Midnight
	chuxi	New Year's Eve

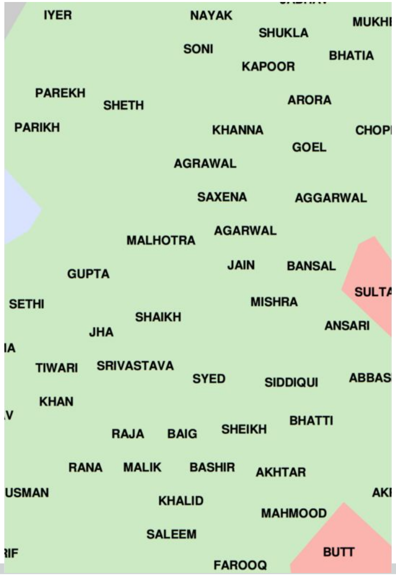
	Word	Word
English	**Mumbai**	**Bombay**
	Chennai	Madras
	Bangalore	Shanghai
	Kolkata	Calultta
	Cairo	Bangkok
	Hyderabad	Hyderabad
German	**Eisenbahnbetrieb**	**rail operations**
	Fahrbetrieb	driving
	Reisezugverkehr	passenger trains
	Fährverkehr	ferries
	Handelsverkehr	Trade
	Schülerverkehr	students Transport
Italian	**papa**	**Pope**
	Papa	Pope
	pontefice	pontiff
	basileus	basileus
	canridnale	cardinal
	frate	friar


Bridge
Gate
Parker
Wilson
Walker
Michigan
Victory
subjects
theatre
base
pattern
transformation
call
drop
regular
normal
unique
identified
prepared
presented
formed

Name Embeddings

Word2vec on email contact lists encode gender and ethnicity because of homophily:





Graph Embeddings (DeepWalk)

Networks based on similarity or links define very sparse feature vectors.

Random walks on networks (sequences of vertices) look like sentences (sequences of words).

Thus we can use word2vec to train network representations!

Nearest Neighbors in Wikipedia

The links between pages defines the network.

Ludwig van Beethoven
- Franz Schubert (0.489)
- Johannes Brahms (0.532)
- Wolfgang Mozart (0.567)
- Robert Schumann (0.576)
- Gustav Mahler (0.635)

Barack Obama
- George W. Bush (0.474)
- Hillary Clinton (0.657)
- Bill Clinton (0.658)
- Joe Biden (0.750)
- Al Gore (0.791)

Scarlett Johansson
- Kirsten Dunst (0.784)
- Natalie Portman (0.786)
- Gwyneth Paltrow (0.796)
- Brad Pitt (0.858)
- Cameron Diaz (0.891)

Mick Jagger
- John Lennon (0.687)
- Keith Richards (0.687)
- Paul McCartney (0.796)
- Ronnie Wood (0.822)
- Eric Clapton (0.833)

Albert Einstein
- Richard Feynman (1.049)
- Max Planck (1.073)
- Freeman Dyson (1.107)
- Stephen Hawking (1.153)
- Robert Oppenheimer (1.156)

Steven Skiena
- Larry Page (1.597)
- Sergey Brin (1.598)
- Danny Hillis (1.644)
- Andrei Broder (1.652)
- Mark Weiser (1.653)