DATA ANALYTICS

Unit 5: Advanced Techniques

Swati Pratap Jagdale
Department of Computer Science and Engineering

DATA ANALYTICS

Unit 5: Latent Semantic Analysis (LSA)

Swati Pratap Jagdale

Department of Computer Science and Engineering

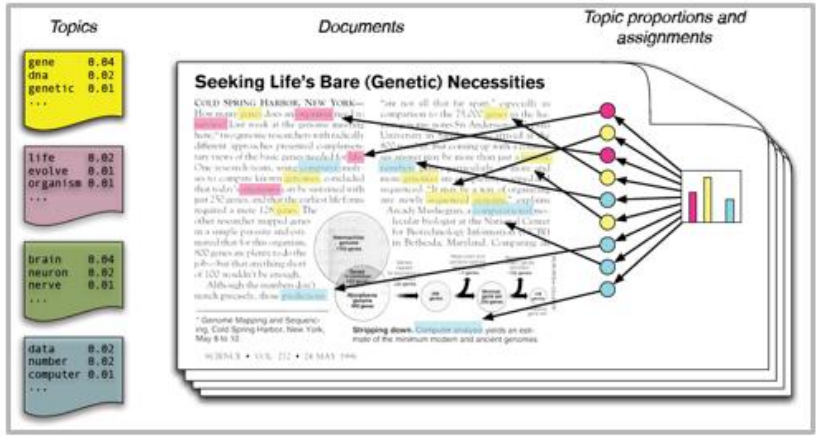
Latent Semantic Analysis (LSA)

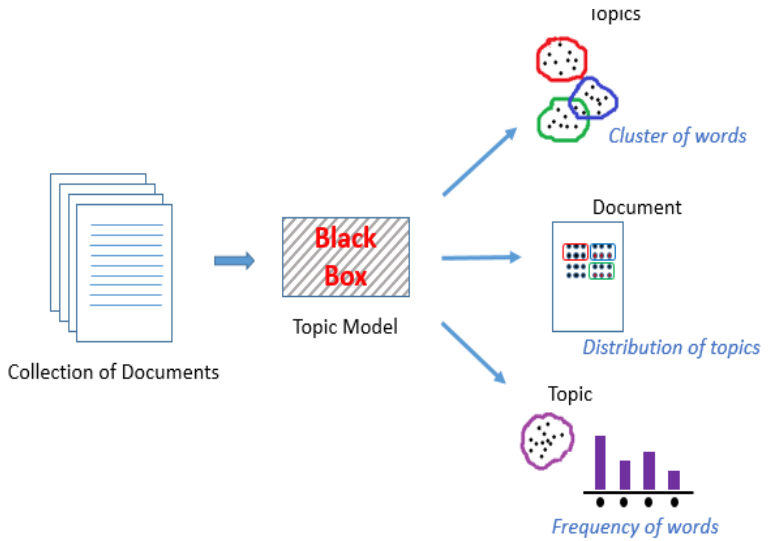
- Content-based recommendation systems: How do we extract keywords or 'topics' (latent patterns) in large documents (news articles, movie plots, book blurbs, relevant job descriptions, etc.) to create summaries or retrieve meaningful information?

- Latent Semantic Analysis, or LSA, is one of the foundation techniques in topic modeling.

- **What is a topic model?**
  An unsupervised technique to discover topics across various text documents.

  Every topic is defined by the proportion of different words it contains.





The problem and the vector space model

- Information Retrieval in the 1980s:
- Problem: Given a collection of documents: retrieve documents that are relevant to a given query match terms in documents to terms in query

- Solution approach: The vector space method
  - Create a term (rows) by document (columns) matrix, based on occurrence
  - Translate into vectors in a vector space; one vector for each document
  - Use cosine to measure distance between vectors (documents)
    small angle = large cosine = similar
    large angle = small cosine = dissimilar

- What can go wrong with this?

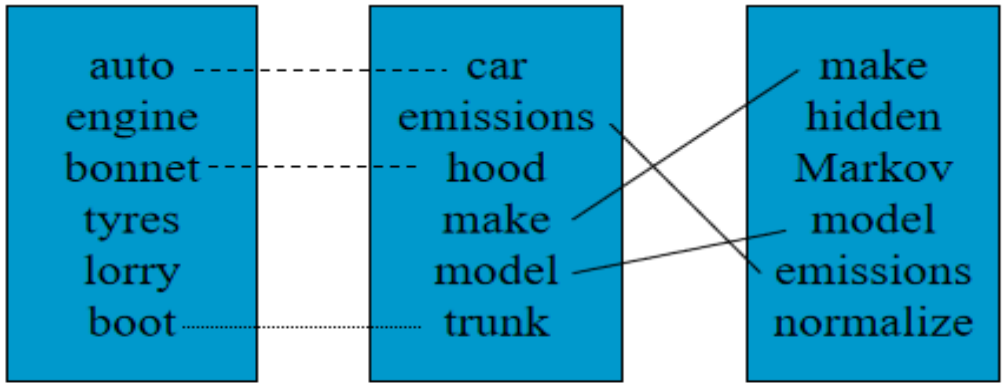
  - I liked his last **novel** quite a lot.
  - We would like to go for a **novel** marketing campaign.

In the first sentence, the word ‘novel’ refers to a book, and in the second sentence it means new or fresh.

Merely mapping words to documents will not suffice as a representation

Motivation for Latent Semantic Analysis (LSA)

- Vector Space Model (from Lillian Lee) has two problems; handling synonymy and polysemy



Synonymy

Will have small cosine

but are related

Not handling this leads to poor recall

Polysemy

Will have large cosine

but not truly related

Not handling this leads to poor precision

Latent Semantic Indexing was proposed to address these two problems to map 'concepts' better for effective Information Retrieval

Steps involved in Latent Semantic Analysis (LSA)

- Let's say we have m number of text documents with n number of total unique terms (words). We wish to extract k topics from all the text data in the documents. The number of topics, k, has to be specified by the user. Generate a document-term matrix of shape n x m
  An m x n term by document matrix (more generally term by context) tend to be sparse
- Convert matrix entries to weights, typically:
  - L(i,j) * G(i): local and global
  - a_ij -> log(freq(a_ij)) divided by entropy for row (-sum (p logp), over p: entries in the row) weight directly by estimated importance in passage
  - weight inversely by degree to which knowing that a word occurred, provides information about the passage it appeared in
- Rank-reduced Singular Value Decomposition (SVD) performed on matrix all but the k highest singular values are set to 0 produces k-dimensional approximation of the original matrix (in least-squares sense) this is the "semantic space"
- Compute similarities between entities in semantic space (usually with cosine)

Terms

Documents

	T1	T2	T3	...	Tn
D1	0.2	0.1	0.5	...	0.1
D2	0.1	0.3	0.4	...	0.3
D3	0.3	0.1	0.1	...	0.5
...	...	...	...	...	...
Dm	0.2	0.1	0.2	...	0.1

Singular Value Decomposition (SVD)

- SVD is basically a factorization of the matrix. Here, we reduce the number of rows (which means the number of words) while preserving the similarity structure among columns (which means paragraphs).
- Unique mathematical decomposition of a matrix into the product of three matrices: two with orthonormal columns and
  one with singular values on the diagonal
- A tool for dimension reduction
  - similarity measure based on co-occurrence
  - finds optimal projection into low-dimensional space

- Can be viewed as a method for rotating the axes in n-dimensional space, so that
  - the first axis runs along the direction of the largest variation among the documents
  - the second dimension runs along the direction with the second largest variation and so on
  - Generalized least-squares method

DATA ANALYTICS

Latent Semantic Analysis (LSA)

- **A Simple Example:** Technical Memo Titles

Topic: Human Computer Interaction (HCI)
c1: *Human* machine *interface* for ABC *computer* applications
c2: A *survey* of *user* opinion of *computer system response time*
c3: The *EPS user interface* management *system*
c4: *System* and *human system* engineering testing of *EPS*
c5: Relation of *user* perceived *response time* to error measurement

Topic: Graph theory (conceptually disjoint from HCI)
m1: The generation of random, binary, ordered *trees*
m2: The intersection *graph* of paths in *trees*
m3: *Graph minors* IV: Widths of *trees* and well-quasi-ordering
m4: *Graph minors*: A *survey*

Complete paper with detailed notes

Latent Semantic Analysis (LSA)

- Create a term-document matrix

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

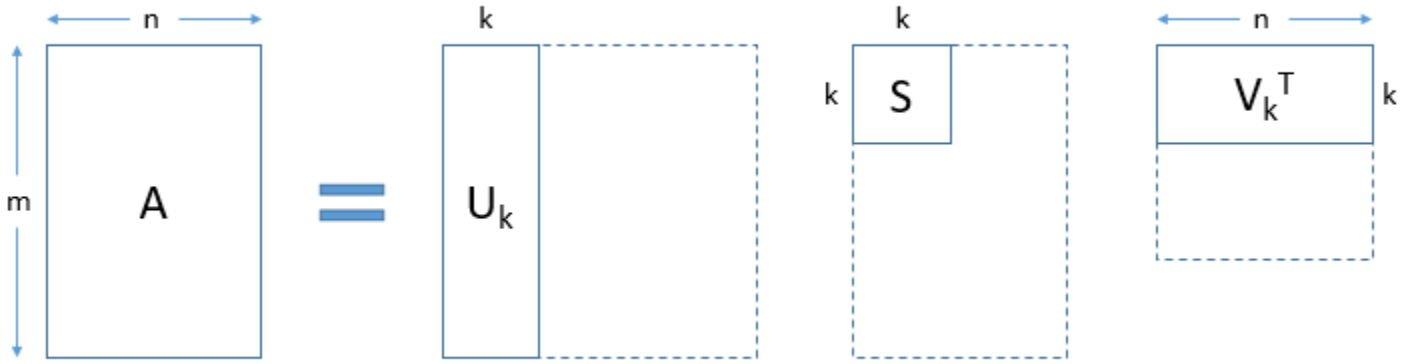
A word by context matrix, ***A***, formed from the titles of five articles about human-computer interaction and four about graph theory. Cell entries are the number of times that a word (rows) appeared in a title (columns) for words that appeared in at least two titles.

r (human,user) = -.38 r (human,minors) = -.29

Singular Value Decomposition

- The mxn term-document matrix is subject to singular value decomposition

$$A = USV^T$$



- Rank-reduced Singular Value Decomposition (SVD) performed on matrix, all but the k highest singular values are set to 0; this produces a k-dimensional approximation of the original matrix (in least-squares sense) this is the “semantic space”
- Compute similarities between entities in semantic space (usually with cosine)

Latent Semantic Analysis (LSA)

- Select k=2 rows of U

{U} =

0.22	-0.11	0.29	-0.41	-0.11	-0.34	0.52	-0.06	-0.41
0.20	-0.07	0.14	-0.55	0.28	0.50	-0.07	-0.01	-0.11
0.24	0.04	-0.16	-0.59	-0.11	-0.25	-0.30	0.06	0.49
0.40	0.06	-0.34	0.10	0.33	0.38	0.00	0.00	0.01
0.64	-0.17	0.36	0.33	-0.16	-0.21	-0.17	0.03	0.27
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.30	-0.14	0.33	0.19	0.11	0.27	0.03	-0.02	-0.17
0.21	0.27	-0.18	-0.03	-0.54	0.08	-0.47	-0.04	-0.58
0.01	0.49	0.23	0.03	0.59	-0.39	-0.29	0.25	-0.23
0.04	0.62	0.22	0.00	-0.07	0.11	0.16	-0.68	0.23
0.03	0.45	0.14	-0.01	-0.30	0.28	0.34	0.68	0.18

Latent Semantic Analysis (LSA)

- Select k=2 dimensions of Σ (the two largest Eigenvalues; denoted by the 2x2 submatrix S)

$\{\Sigma\} =$

$$\begin{matrix} 3.34 & & & & & & & & \\ & 2.54 & & & & & & & \\ & & 2.35 & & & & & & \\ & & & 1.64 & & & & & \\ & & & & 1.50 & & & & \\ & & & & & 1.31 & & & \\ & & & & & & 0.85 & & \\ & & & & & & & 0.56 & \\ & & & & & & & & 0.36 \end{matrix}$$

Latent Semantic Analysis (LSA)

- Select k=2 columns of V (or rows of V^T)

{V} =

0.20	0.61	0.46	0.54	0.28	0.00	0.01	0.02	0.08
-0.06	0.17	-0.13	-0.23	0.11	0.19	0.44	0.62	0.53
0.11	-0.50	0.21	0.57	-0.51	0.10	0.19	0.25	0.08
-0.95	-0.03	0.04	0.27	0.15	0.02	0.02	0.01	-0.03
0.05	-0.21	0.38	-0.21	0.33	0.39	0.35	0.15	-0.60
-0.08	-0.26	0.72	-0.37	0.03	-0.30	-0.21	0.00	0.36
0.18	-0.43	-0.24	0.26	0.67	-0.34	-0.15	0.25	0.04
-0.01	0.05	0.01	-0.02	-0.06	0.45	-0.76	0.45	-0.07
-0.06	0.24	0.02	-0.08	-0.26	-0.62	0.02	0.52	-0.45

Latent Semantic Analysis (LSA)

- Original term-document matrix

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

r (human.user) = -.38 r (human.minors) = -.29

After LSA

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	0.16	0.40	0.38	0.47	0.18	-0.05	-0.12	-0.16	-0.09
interface	0.14	0.37	0.33	0.40	0.16	-0.03	-0.07	-0.10	-0.04
computer	0.15	0.51	0.36	0.41	0.24	0.02	0.06	0.09	0.12
user	0.26	0.84	0.61	0.70	0.39	0.03	0.08	0.12	0.19
system	0.45	1.23	1.05	1.27	0.56	-0.07	-0.15	-0.21	-0.05
response	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
time	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
EPS	0.22	0.55	0.51	0.63	0.24	-0.07	-0.14	-0.20	-0.11
survey	0.10	0.53	0.23	0.21	0.27	0.14	0.31	0.44	0.42
trees	-0.06	0.23	-0.14	-0.27	0.14	0.24	0.55	0.77	0.66
graph	-0.06	0.34	-0.15	-0.30	0.20	0.31	0.69	0.98	0.85
minors	-0.04	0.25	-0.10	-0.21	0.15	0.22	0.50	0.71	0.62

r (human,user) = .94 r (human,minors) = -.83

Effect of SVD on the correlation matrix

LSA Titles example:
Correlations between titles in raw data

	c1	*c2*	*c3*	*c4*	*c5*	*m1*	*m2*	*m3*
c2	-0.19							
c3	0.00	0.00						
c4	0.00	0.00	0.47					
c5	-0.33	0.58	0.00	-0.31				
m1	-0.17	-0.30	-0.21	-0.16	-0.17			
m2	-0.26	-0.45	-0.32	-0.24	-0.26	0.67		
m3	-0.33	-0.58	-0.41	-0.31	-0.33	0.52	0.77	
m4	-0.33	-0.19	-0.41	-0.31	-0.33	-0.17	0.26	0.56

0.02
-0.30 0.44

Correlations in first-two dimension space post LSA

c2	0.91							
c3	1.00	0.91						
c4	1.00	0.88	1.00					
c5	0.85	0.99	0.85	0.81				
m1	-0.85	-0.56	-0.85	-0.88	-0.45			
m2	-0.85	-0.56	-0.85	-0.88	-0.44	1.00		
m3	-0.85	-0.56	-0.85	-0.88	-0.44	1.00	1.00	
m4	-0.81	-0.50	-0.81	-0.84	-0.37	1.00	1.00	1.00

0.92
-0.72 1.00

Some FAQ's on LSA

- **Why, and under what circumstances would reducing the dimensionality of representation be beneficial?**
  - When the original data are generated from a source of the same dimensionality and general structure as the reconstruction.
  - Suppose, for example, that speakers or writers generate paragraphs by choosing words from a k-dimensional space in such a way that words in the same paragraph tend to be selected from nearby locations. If listeners or readers try to infer the similarity of meaning from these data, they will do better if they reconstruct the full set of relations in the same number of dimensions as the source. Among other things, given the right analysis, this will allow the system to infer that two words from nearby locations in semantic space have similar meanings even though they are never used in the same passage, or that they have quite different meanings even though they often occur in the same utterances.

- **How is k, the number of dimensions to be retained in LSA, selected?**
  - Empirically
  - Some external criterion of validity is sought, such as the performance on a synonym test or prediction of the missing words in passages if some portion are deleted in forming the initial matrix.

Pros and Cons of LSA

Pros:

- LSA is fast and easy to implement.
- It gives decent results, much better than a plain vector space model.

Cons:

- Since it is a linear model, it might not do well on datasets with non-linear dependencies.
- LSA assumes a Gaussian distribution of the terms in the documents, which may not be true for all problems.
- LSA involves SVD, which is computationally intensive and hard to update as new data comes up.

Additional References

https://www.analyticsvidhya.com/blog/2018/10/stepwise-guide-topic-modeling-latent-semantic-analysis/

https://www.analyticsvidhya.com/blog/2016/08/beginners-guide-to-topic-modeling-in-python/

Paper on LSA with a detailed explanation of the example:
http://lsa.colorado.edu/papers/dp1.LSAintro.pdf

DATA ANALYTICS

Unit 5: Sparse data processing, sparse PCA

Swati Pratap Jagdale

Department of Computer Science and Engineering

Sparse Data Processing

- Sparse data is when there are more 0's than there are entries in the data matrix (or flatfile or record, etc.)

User	Movie	Rating
A	Parasite	5
A	Joker	4
A	Avengers: Endgame	4
B	Parasite	2
B	Spotlight	4
B	The Great Beauty	3
C	Avengers: Endgame	5
D	There will be blood	4
E	Avengers: Endgame	4

Dense matrix

Users					Movies						Target
A	B	C	D	E	Parasite	Joker	Avengers	Spotlight	The Great Beauty	There will be blood	Rating
1	0	0	0	0	1	0	0	0	0	0	5
1	0	0	0	0	0	1	0	0	0	0	4
1	0	0	0	0	0	0	1	0	0	0	4
0	1	0	0	0	1	0	0	0	1	0	2
0	1	0	0	0	0	0	0	1	0	0	4
0	1	0	0	0	0	0	0	0	1	0	3
0	0	1	0	0	0	0	1	0	0	0	5
0	0	0	1	0	0	0	0	0	0	1	4
0	0	0	0	1	0	0	1	0	0	0	4

Sparse matrix

- What is the problem?
  - Space complexity: very large term-document matrices or matrices showing links between websites or users need to be stored in memory for processing
  - Time complexity: it takes needlessly long to perform operations on sparse matrices (given most of the data is 0 and need not be processed!)

https://machinelearningmastery.com/sparse-matrices-for-machine-learning/
https://towardsdatascience.com/working-with-sparse-data-sets-in-pandas-and-sklearn-d26c1cfbe067

Sparse Data Processing

Some workarounds

- Ignore zero values; only nonzero values can be stored and processed
- Use a different representation:

  - Dictionary of Keys. A dictionary is used where a row and column index is mapped to a value.

  - List of Lists. Each row of the matrix is stored as a list, with each sublist containing the column index and the value.

  - Coordinate List. A list of tuples is stored with each tuple containing the row index, column index, and the value.

  - Compressed Sparse Row (CSR). The sparse matrix is represented using three one-dimensional arrays for the non-zero values, the extents of the rows, and the column indexes.

  - Compressed Sparse Column. The same as the Compressed Sparse Row method except the column indices are compressed and read first before the row indices.

- Use dimensionality reduction techniques such as sparse PCA

https://machinelearningmastery.com/sparse-matrices-for-machine-learning/
https://towardsdatascience.com/working-with-sparse-data-sets-in-pandas-and-sklearn-d26c1cfbe067

Sparse PCA

- **Sparse principal component analysis (sparse PCA)** is a specialised technique used in statistical analysis and, in particular, in the analysis of multivariate data sets.
- It extends the classic method of principal component analysis (PCA) for the reduction of dimensionality of data by introducing sparsity structures to the input variables.
- A particular disadvantage of ordinary PCA is that the principal components are usually linear combinations of all input variables.
- Sparse PCA overcomes this disadvantage by finding linear combinations that contain just a few input variables.
- Contemporary datasets often have the number of input variables comparable with or even much larger than the number of samples.

Sparse PCA

Mathematical Formulation

- Given a data matrix X n rows (each row is an independent sample) with p columns (attributes)
- One assumes each column of X has mean zero, otherwise one can subtract column-wise mean from each element of X.
- Let $\Sigma = \frac{1}{n-1} X^\top X$ be the empirical covariance matrix of X, which has

dimensions p x p. Given integer k, 1<=k<=p, the sparse PCA problem can be formulated as maximizing the variance along a direction represented by vector v, while constraining its cardinality.

$$\begin{aligned} \max \quad & v^T \Sigma v \\ \text{subject to} \quad & \|v\|_2 = 1 \\ & \|v\|_0 \leq k. \end{aligned}$$

- The constraints specify that *v* is a unit vector and $||v||_0$ represents the L_0 norm of v, defined as the number of its non-zero components (<=k).
- k is much smaller than p; the result is the k-sparse largest eigenvalue.
- If one takes k=p, the problem reduces to the ordinary PCA, and the optimal value becomes the largest eigenvalue of covariance matrix Σ.

Sparse PCA

- After finding the optimal solution v, one deflates Σ to obtain a new matrix.

$$\Sigma_1 = \Sigma - (v^T \Sigma v) v v^T,$$

- Iterate this process to obtain further principal components.
- However, unlike PCA, sparse PCA cannot guarantee that different principal components are orthogonal. In order to achieve orthogonality, additional constraints must be enforced.

- The following equivalent definition is in matrix form.
- Let, V be a p×p symmetric matrix, one can rewrite the sparse PCA problem as:

$$\begin{aligned} \max \quad & Tr(\Sigma V) \\ \text{subject to} \quad & Tr(V) = 1 \\ & \|V\|_0 \leq k^2 \\ & Rank(V) = 1, V \succeq 0. \end{aligned}$$

- Tr is the matrix trace, and $||V||_0$ represents the non-zero elements in matrix V. The last line specifies that V has matrix rank one and is positive semidefinite. The last line means that one has **V= vv^T**

$$\begin{aligned} \max \quad & Tr(\Sigma V) \\ \text{subject to} \quad & Tr(V) = 1 \\ & \mathbf{1}^T |V| \mathbf{1} \leq k \\ & V \succeq 0. \end{aligned}$$

Applications of Sparse PCA

Financial Data Analysis

- Suppose ordinary PCA is applied to a dataset where each input variable represents a different asset, it may generate principal components that are weighted combination of all the assets
- In contrast, sparse PCA would produce principal components that are weighted combination of only a few input assets, so one can easily interpret its meaning.
- Furthermore, if one uses a trading strategy based on these principal components, fewer assets imply less transaction costs.

Biology

- Consider a dataset where each input variable corresponds to a specific gene. Sparse PCA can produce a principal component that involves only a few genes, so researchers can focus on these specific genes for further analysis.

High-dimensional Hypothesis Testing

- Contemporary datasets often have the number of input variables (p) comparable with or even much larger than the number of samples (n)
- It has been shown that if p/n does not converge to zero, the classical PCA is not consistent. But sparse PCA can retain consistency even if p>>n

References

https://www.analyticsvidhya.com/blog/2014/01/logistic-regression-rare-event/

https://en.wikipedia.org/wiki/Sparse_PCA#Financial_Data_Analysis

DATA ANALYTICS

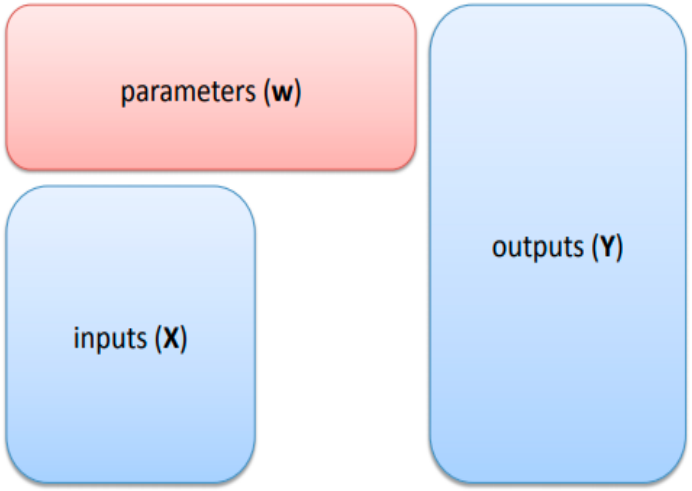
Unit 5: Concept of hidden variables

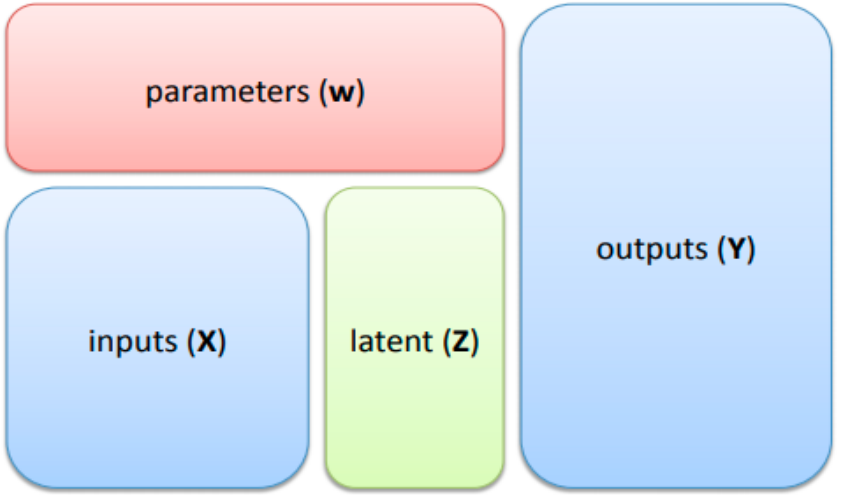
Swati Pratap Jagdale

Department of Computer Science and Engineering

Hidden Variables

- Random variables in supervised learning



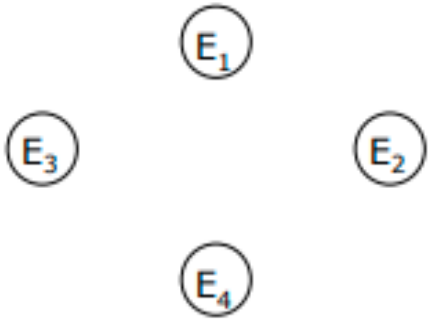


- 'Hidden' (or latent) variable is one that we never 'see'
- Not even in training
- Sometimes we believe they are real
  and sometimes they approximate reality (as it happens in Physics)

- 'Learning' or 'decoding' are both understood as inference problems

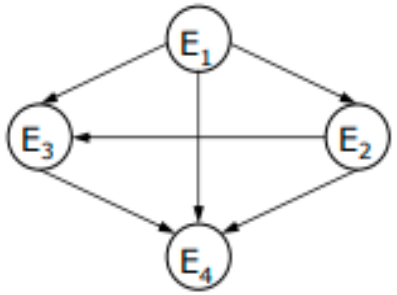
Learning With Hidden Variables

- Consider a situation in which you can observe a whole bunch of different evidence variables, E_1 through E_n. Maybe they're all the different symptoms that a patient might have. Or maybe they represent different movies and whether someone likes them.



Learning With Hidden Variables

- If those variables are all conditionally dependent on one another, then we'd need a highly connected graph that's capable of representing the entire joint distribution between the variables.
- Because the last node has n-1 parents, it will take on the order of 2^n parameters to specify the conditional probability tables in this network.



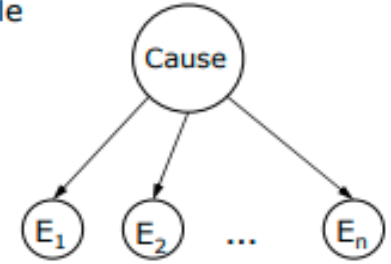
Without the cause, all the evidence is dependent on each other

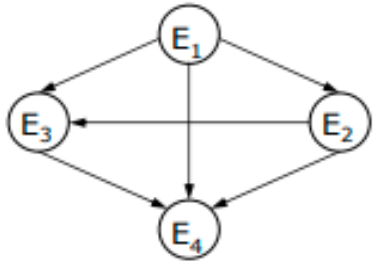
$O(2^n)$ parameters

Learning With Hidden Variables

- But, in some cases, we can get a considerably simpler model by introducing an additional “cause” node.
- It might represent the underlying disease state that was causing the patients’ symptoms or some division of people into those who like westerns and those who like comedies.

Cause is unobservable



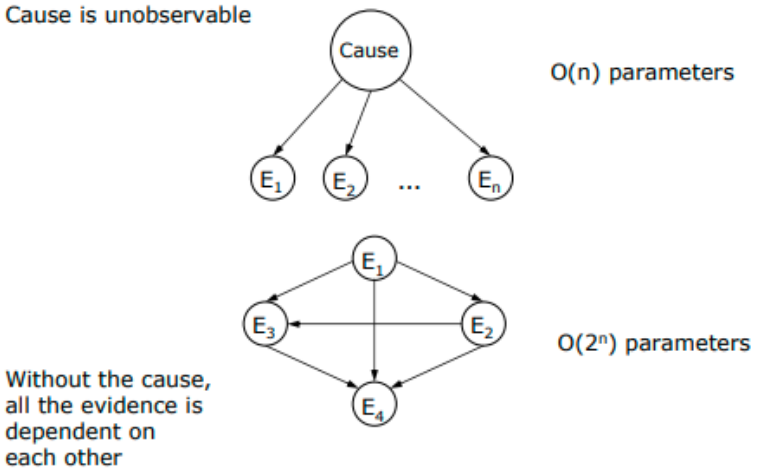


$O(2^n)$ parameters

Without the cause, all the evidence is dependent on each other

Learning With Hidden Variables

- In the simpler model, the evidence variables are conditionally independent given the causes. That means that it would only require on the order of n parameters to describe all the CPTs in the network, because at each node, we just need a table of size 2 (if the cause is binary; or k if the cause can take on k values), and one (or k-1) parameter to specify the probability of the cause.



- So, what if you think there's a hidden cause? How can you learn a network with unobservable variables?

Simpson's Paradox

- Edward Hugh Simpson, a statistician and former cryptanalyst at Bletchley Park, described the statistical phenomenon - Simpson's paradox

- The art of data science is seeing beyond the data — using and developing methods and tools to get an idea of what that hidden reality looks like.

- Simpson's paradox showcases the importance of skepticism and interpreting data with respect to the real world, and also the dangers of oversimplifying a more complex truth by trying to see the whole story from a single data-viewpoint.

Simpson's Paradox

- ***Simpson's Paradox:***

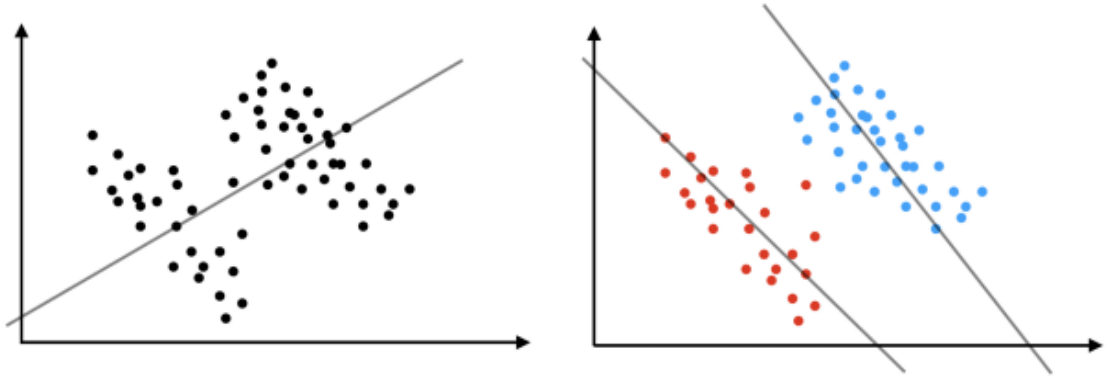
A trend or result that is present when data is put into groups that reverses or disappears when the data is combined.

Example: UC Berkley's suspected gender-bias.
At the beginning of the academic year in 1973, UC Berkeley's graduate school had admitted roughly 44% of their male applicants and 35% of their female applicants.
Was there a discrimination against female applicants?

When the data was studied department-wise, it was observed that:

- there was a statistically significant gender bias **in favor of women** for 4 out of the 6 departments, and no significant gender bias in the remaining 2
- It is discovered that **women tended to apply to departments that admitted a smaller percentage of applicants overall**, and that this hidden variable affected the marginal values for the percentage of accepted applicants in such a way as to reverse the trend that existed in the data as a whole

Simpson's Paradox



A visual example: the overall trend reverses when data is grouped by some colour-represented category.

Simpson's Paradox

A simple example in business:

- Suppose the soft drinks industry is trying to choose between two new flavors they have produced. We could sample public opinion on the two flavors

Flavour	Sample Size	# Liked Flavour
Sinful Strawberry	1000	800
Passionate Peach	1000	750

- 80% of people enjoyed 'Sinful Strawberry' whereas only 75% of people enjoyed 'Passionate Peach'. So 'Sinful Strawberry' is more likely to be the preferred flavor.

Simpson's Paradox

- Some other information while conducting the survey, such as the sex of the person sampling the drink. What happens if we split our data up by sex?

- 84.4% of men and 40% of women liked 'Sinful Strawberry' whereas 85.7% of men and 50% of women liked 'Passionate Peach'

Flavour	# Men	# Liked Flavour (Men)	# Women	# Liked Flavour (Women)
Sinful Strawberry	900	760	100	40
Passionate Peach	700	600	300	150

Simpson's Paradox

- *According to our sample data, generally people prefer 'Sinful Strawberry', but both men and women separately prefer 'Passionate Peach'.*
- *This is an example of Simpson's Paradox!*

Flavour	# Men	# Liked Flavour (Men)	# Women	# Liked Flavour (Women)
Sinful Strawberry	900	760	100	40
Passionate Peach	700	600	300	150

Simpson's Paradox

Lurking variables (Hidden Variables)

- Simpson's paradox arises when there are hidden variables that split data into multiple separate distributions.
- Such a hidden variable is aptly referred to as a **lurking variable**, and they can often be difficult to identify.

Consider the lurking variable (sex) and a little bit of probability theory:-

- P(Liked Strawberry) = P(Liked Strawberry | Man)P(Man) + P(Liked Strawberry | Woman)P(Woman)

- 800/1000 = (760/900)×(900/1000) + (40/100)×(100/1000)

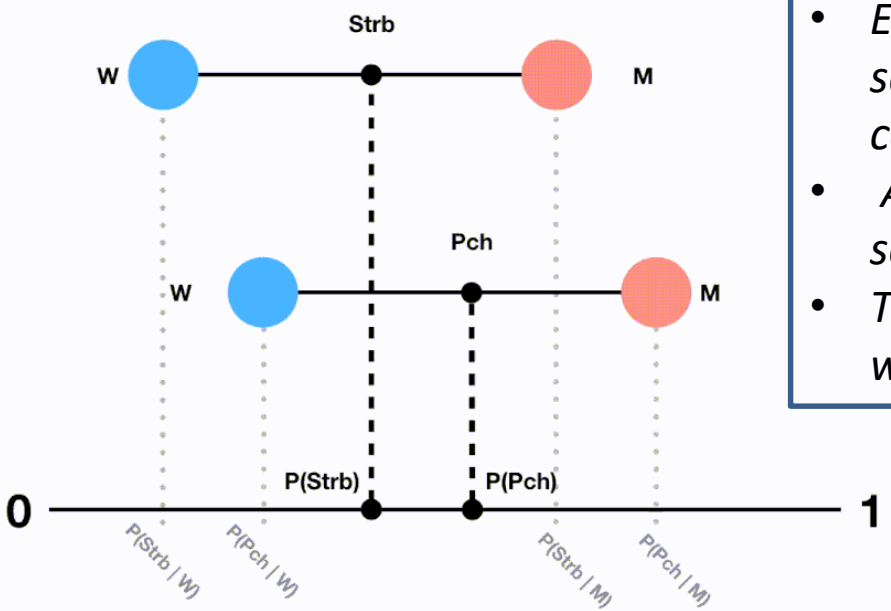
- P(Liked Peach) = P(Liked Peach | Man)P(Man) + P(Liked Peach | Woman)P(Woman)

- 750/1000 = (600/700)×(700/1000) + (150/300)×(300/1000)

Simpson's Paradox

- **Lurking variables (Hidden Variables)**

- We can think of the marginal probabilities of sex (P(Man) and P(Woman)) as weights that, in the case of 'Sinful Strawberry', cause the total probability to be significantly shifted towards the male opinion.



- *Each coloured circle represents either the men or women that sampled each flavour, the position of the centre of each circle corresponds to that group's probability of liking the flavour.*
- *As the circles grow (i.e. sample proportions change) we can see how the marginal probability of liking the flavour changes.*
- *The marginal distributions shift and switch as samples become weighted with respect to the lurking variable (sex).*

Approaches to Infer Hidden Variables

From other variables in the model i.e., from observations or evidence

- Viterbi algorithm for decoding the transition of hidden states
- “Learning” of Hidden Markov Models (or expectation maximization (EM))

These are popular approaches to infer the transition probabilities between hidden states or the effect of hidden variables on a system

Think about this:
Given a model, how does one postulate the presence of a hidden (or latent) variable?

(This is where an understanding of the problem domain comes in…)

References

http://www.cs.cmu.edu/~nasmith/psnlp/lecture5.pdf

https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-825-techniques-in-artificial-intelligence-sma-5504-fall-2002/lecture-notes/Lecture18FinalPart1.pdf

https://towardsdatascience.com/simpsons-paradox-and-interpreting-data-6a0443516765

THANK YOU

Swati Pratap Jagdale
Department of Computer Science
swatigambhire@pes.edu