Data Mining the Hair in Your Soup

Natural Language Processing in Economics Prediction Problems

Guillaume A. Pouliot and Mike Luca

Choice, Maxdiff, and Tweets, Monday, August 8 2014

Allocating Inspectors

The allocation problem

- The city of San Francisco sends sanitation inspectors to check if restaurants respect the hygiene regulations.
- They have a limited number of inspectors. Right now, these are allocated "at random".
- San Francisco would like a more efficient allocation of inspectors to restaurants.

Allocating Inspectors

The allocation problem

- The city of San Francisco sends sanitation inspectors to check if restaurants respect the hygiene regulations.
- They have a limited number of inspectors. Right now, these are allocated "at random".
- San Francisco would like a more efficient allocation of inspectors to restaurants.

Allocating Inspectors

The allocation problem

- The city of San Francisco sends sanitation inspectors to check if restaurants respect the hygiene regulations.
- They have a limited number of inspectors. Right now, these are allocated "at random".
- San Francisco would like a more efficient allocation of inspectors to restaurants.

Allocating Inspectors

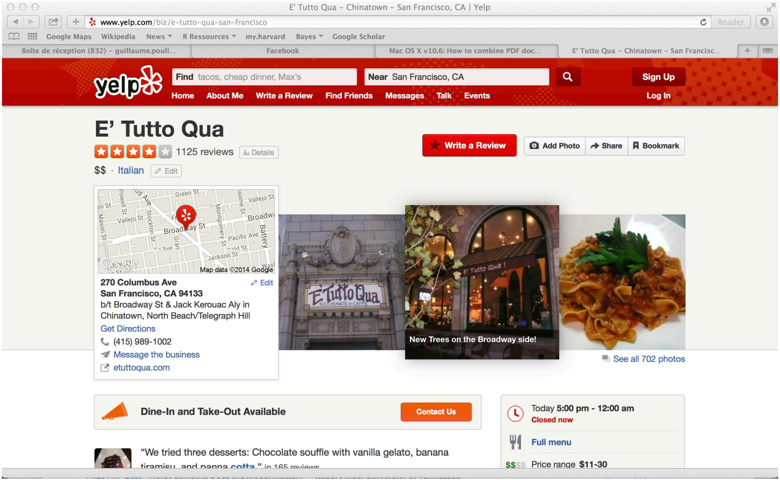
A suggestion to increase accuracy of prediction

- Prof. Luca's suggestion: using customer reviews to help find which restaurants are unsanitary. He thus put together a data set matching Yelp reviews (obtained from Yelp) to inspection scores (public).
- Our first question is then: **can we somehow use text data as a "covariate" to improve predictions?**

Allocating Inspectors

A suggestion to increase accuracy of prediction

- Prof. Luca's suggestion: using customer reviews to help find which restaurants are unsanitary. He thus put together a data set matching Yelp reviews (obtained from Yelp) to inspection scores (public).
- Our first question is then: **can we somehow use text data as a "covariate" to improve predictions?**


E' Tutto Qua - Chinatown - San Francisco, CA | Yelp
www.yelp.com/biz/e-tutto-qua-san-francisco
Reader
Google Maps
Wikipedia
News
R Ressources
my.harvard
Bayes
Google Scholar
Boîte de réception (832) – guillaume.pouli...
Facebook
Mac OS X v10.6: How to combine PDF doc...
E' Tutto Qua – Chinatown – San Francisc...
yelp
Find tacos, cheap dinner, Max's
Near San Francisco, CA
Sign Up
Home
About Me
Write a Review
Find Friends
Messages
Talk
Events
Log In
E' Tutto Qua
1125 reviews
Details
$$ · Italian
Edit
Write a Review
Add Photo
Share
Bookmark
Map data ©2014 Google
270 Columbus Ave
San Francisco, CA 94133
b/t Broadway St & Jack Kerouac Aly in
Chinatown, North Beach/Telegraph Hill
Get Directions
(415) 989-1002
Message the business
etuttoqua.com
E'Tutto Qua
New Trees on the Broadway side!
See all 702 photos
Dine-In and Take-Out Available
Contact Us
Today 5:00 pm - 12:00 am
Closed now
Full menu
Price range $11-30
"We tried three desserts: Chocolate souffle with vanilla gelato, banana

Beet and goat cheese salad

Was this review ...?

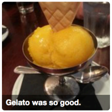
Useful 2 | Funny 2 | Cool 2

Misty B.
Fairfield, CA
3 friends
7 reviews

7/4/2014

1 check-in here

This is the best most authentic italian around. They are personal and fun. Everything we had was amazing. You can't go wrong here. We had a group of 8 on a holiday and they got us right in. Our waiter was great. Take a moment to experience Italy while I the city.



Gelato was so good.

The pizza!

Was this review ...?

Useful | Funny | Cool

Brandon R.
San Jose, CA
0 friends
21 reviews

7/3/2014

1 check-in here

The Data

- 4795 restaurants.
- 18 501 inspections.
- 1 107 477 reviews.

The Data

- 4795 restaurants.
- 18 501 inspections.
- 1 107 477 reviews.

The Data

- 4795 restaurants.
- 18 501 inspections.
- 1 107 477 reviews.

The Data

reviews: contains Yelp reviews; for each review we have the ID of the restaurant reviewed, the rating given by the reviewer, the date of the review, and the text of the review. Some example of reviews:

· *"Great tasting food. I hate over hyping a business but it seems that you cant go wrong with what ever you order here. I had the beef chow fun and vegetarian dumplings. Plus the salt and pepper tofu is crisp and delicious. The spring rolls are some of the best I've had ever."*

· *"This place was pretty good as for me and my girl came here to try it out. \n\nThe food was surprisingly cheap on the price wise and was pretty good.*

· *"Only draw back was a minor wait for seating but other than that the staff was quick and freindly."*

· *"Please, do not set your standards too high. Because by the time you reach the bathroom in this place, you want to get out."*

Latent Dirichlet Allocation

- A hierarchical model for text data in different documents.

Key Assumption

- "Bag-of-words" : The words (and the documents) are exchangeable.

- Through de Finetti's theorem (exchangeable sequence is conditionally i.i.d.), this makes the hierarchical modeling theoretically sensible.
- Blatantly false.

Latent Dirichlet Allocation

- A hierarchical model for text data in different documents.

Key Assumption

- "Bag-of-words" : The words (and the documents) are exchangeable.

- Through de Finetti's theorem (exchangeable sequence is conditionally i.i.d.), this makes the hierarchical modeling theoretically sensible.
- Blatantly false.

Latent Dirichlet Allocation

- A hierarchical model for text data in different documents.

Key Assumption

- "Bag-of-words" : The words (and the documents) are exchangeable.

- Through de Finetti's theorem (exchangeable sequence is conditionally i.i.d.), this makes the hierarchical modeling theoretically sensible.
- Blatantly false.

Latent Dirichlet Allocation

- A hierarchical model for text data in different documents.

Key Assumption

- "Bag-of-words" : The words (and the documents) are exchangeable.

- Through de Finetti's theorem (exchangeable sequence is conditionally i.i.d.), this makes the hierarchical modeling theoretically sensible.
- Blatantly false.

Latent Dirichlet Allocation

Notation

- Words w_i.
- Documents $\boldsymbol{w} = \{w_1, \ldots, w_N\}$.
- Corpus $\mathscr{D} = \{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_M\}$.

Latent Dirichlet Allocation
Notation

- Words w_i.
- Documents $\boldsymbol{w} = \{w_1, \ldots, w_N\}$.
- Corpus $\mathscr{D} = \{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_M\}$.

Latent Dirichlet Allocation

Notation

- Words w_i.
- Documents $\boldsymbol{w} = \{w_1, \ldots, w_N\}$.
- Corpus $\mathscr{D} = \{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_M\}$.

Latent Dirichlet Allocation

The model

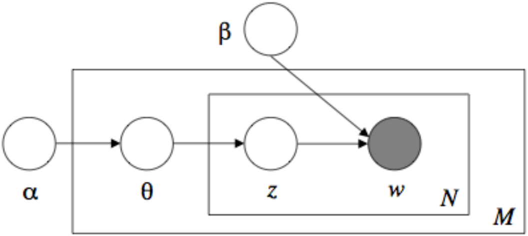
Generative Process

- For each document $\boldsymbol{w}$:
  $\theta \sim \mathrm{Dir}(\alpha)$
- For each of the N words w_n:
  (a) topic: $z_n \sim \mathrm{Mult}(\theta)$
  (b) word: $w_n \sim p(w_n | z_n, \beta)$,

z is assumed to be of dimension K(the number of topics), and $\beta = [\beta_{ij}] = p(w^j = 1 | z^i = 1)]$ is a $K \times V$ matrix, V is the size of the vocabulary.

Latent Dirichlet Allocation
The model

We can represent the model as a directed acyclic graph:



Latent Dirichlet Allocation

Example, K=2, N=4

Draw θ from a Dirichlet(α). We drew $\theta = (0.3, 0.7)$

$\theta =$	0.3	0.7	$V =$	fast	slow	good	carrot	tomato
$z =$			$\boldsymbol{w} =$					

Suppose we know **topic 1 is service**, and **topic 2 is ingredients**.

Latent Dirichlet Allocation

Example, K=2, N=4

Draw θ from a Dirichlet(α). We drew $\theta = (0.3, 0.7)$

$\theta =$	0.3	0.7	$V =$	fast	slow	good	carrot	tomato
	*							
$z =$			$\boldsymbol{w} =$					

Suppose we know **topic 1 is service**, and **topic 2 is ingredients**.

Latent Dirichlet Allocation

Example, K=2, N=4

Draw θ from a Dirichlet(α). We draw $\theta = (0.3, 0.7)$

$\theta =$	0.3	0.7	$V =$	fast	slow	good	carrot	tomato
	*			*				
$z =$			$\boldsymbol{w} =$					

Suppose we know **topic 1 is service**, and **topic 2 is ingredients**.

Latent Dirichlet Allocation

Example, K=2, N=4

Draw θ from a Dirichlet(α). We draw $\theta = (0.3, 0.7)$

$\theta =$	0.3	0.7
$z =$	*	
		*

$V =$	fast	slow	good	carrot	tomato
$\boldsymbol{w} =$	*				

Suppose we know **topic 1 is service**, and **topic 2 is ingredients**.

Latent Dirichlet Allocation

Example, K=2, N=4

Draw θ from a Dirichlet(α). We draw $\theta = (0.3, 0.7)$

$\theta =$	0.3	0.7	$V =$	fast	slow	good	carrot	tomato
	*			*				
$z =$		*	$\boldsymbol{w} =$					*

Suppose we know **topic 1 is service**, and **topic 2 is ingredients**.

Latent Dirichlet Allocation

Example, K=2, N=4

Draw θ from a Dirichlet(α). We draw $\theta = (0.3, 0.7)$

$\theta =$	0.3	0.7	$V =$	fast	slow	good	carrot	tomato
	*			*				
$z =$		*	$\boldsymbol{w} =$					*
		*						

Suppose we know **topic 1 is service**, and **topic 2 is ingredients**.

Latent Dirichlet Allocation

Example, K=2, N=4

Draw θ from a Dirichlet(α). We draw $\theta = (0.3, 0.7)$

$\theta =$	0.3	0.7	$V =$	fast	slow	good	carrot	tomato
	*			*				
$z =$		*	$\boldsymbol{w} =$					*
		*				*		

Suppose we know **topic 1 is service**, and **topic 2 is ingredients**.

Latent Dirichlet Allocation

Example, K=2, N=4

Draw θ from a Dirichlet(α). We draw $\theta = (0.3, 0.7)$

$\theta =$	0.3	0.7	$V =$	fast	slow	good	carrot	tomato
	*			*				
$z =$		*	$w =$					*
		*				*		
		*						

Suppose we know **topic 1 is service**, and **topic 2 is ingredients**.

Latent Dirichlet Allocation

Example, K=2, N=4

Draw θ from a Dirichlet(α). We draw $\theta = (0.3, 0.7)$

$\theta =$	0.3	0.7	$V =$	fast	slow	good	carrot	tomato
$z =$	*		$\mathbf{w} =$	*				
		*						*
		*				*		
		*				*		

Suppose we know **topic 1 is service**, and **topic 2 is ingredients**.

Latent Dirichlet Allocation

Example, K=2, N=4

- We observe the document

$$\boldsymbol{w} = \{\text{fast}, \ \text{good}, \ \text{good}, \ \text{tomato}\},$$

and we fit topics to it.

- Suppose we know that "**service**" gives high likelihood to "fast", "**ingredients**" gives high likelihood to "tomato", and both give more or less equal probability to "good". Hence you can imagine fitting θ from $\boldsymbol{w}$. More precisely, we can estimate the posterior $p(\theta, z | \boldsymbol{w}, \alpha, \beta)$.

Latent Dirichlet Allocation

Example, K=2, N=4

- We observe the document

$$\boldsymbol{w} = \{\text{fast, good, good, tomato}\},$$

and we fit topics to it.

- Suppose we know that "**service**" gives high likelihood to "fast", "**ingredients**" gives high likelihood to "tomato", and both give more or less equal probability to "good". Hence you can imagine fitting θ from $\boldsymbol{w}$. More precisely, we can estimate the posterior $p(\theta, z | \boldsymbol{w}, \alpha, \beta)$.

Some Topics Coefficients

Which Topics Are They?

Topic3	Topic4	...	Topic16	...	Topic26	Topic27
duck	vegetarian		bathroom		dim	sushi
lamb	vegan		damn		sum	roll
french	veggi		ass		tea	fish
bred	dog		shit		cheap	japanes
truffl	tofu		smell		bun	cheap
garcon	options		fuck		chines	udon
rude	decent		god		steam	salmon
chez	herbivor		dont		chinatown	tempura
sauce	sometim		dirti		dumpl	teriyaki
service	wrap		seriously		pumpkin	tuna

Most of the topics are restaurant categories.

Supervised LDA

Generative Process

- For each document $\boldsymbol{w}$:
  $\theta \sim \mathrm{Dir}(\alpha)$
- For each of the N words w_n:
  (a) topic: $z_n \sim \mathrm{Mult}(\theta)$
  (b) word: $w_n \sim p(w_n|z_n, \beta)$,
- Draw response for document:
  $y|z_{1:N}, \rho, \sigma^2 \sim N(\bar{z}^T \rho, \sigma^2)$

z is assumed to be of dimension K (the number of topics), and $\beta = [\beta_{ij}] = p(w^j = 1|z^i = 1)]$ is a $K \times V$ matrix, V is the size of the vocabulary.

Supervised LDA

Generative Process

- For each document $\boldsymbol{w}$:
  $\theta \sim \mathrm{Dir}(\alpha)$
- For each of the N words w_n:
  (a) topic: $z_n \sim \mathrm{Mult}(\theta)$
  (b) word: $w_n \sim p(w_n | z_n, \beta)$,
- Draw response for document:
  $y | z_{1:N}, \rho, \sigma^2 \sim N((X, \bar{z}^T)\rho, \sigma^2)$

z is assumed to be of dimension K (the number of topics), and $\beta = [\beta_{ij}] = p(w^j = 1 | z^i = 1)]$ is a $K \times V$ matrix, V is the size of the vocabulary.

Supervised LDA
Out-of-sample performence

We train the algorithms on the earlier 3/4 of the data set and predict the future.

	Unsup.	Sup. No cov.	Uncond. Sup.	Cond. Sup.
Classification	76%	72%	71%	73%
True Dirty	20%	37%	41%	47%
True Clean	93%	83%	81%	82%

An "always dirty" rule would have 23% classification score, and "always clean" a 67% score.

Lab Experiments

- See how people would try and trick the algorithm when they know there is one. What kind of fake reviews can we detect.
- Can we protect ourselves against fake reviews.

Summary

- Adding covariates obtained from text does help prediction.
- Controling for other covariates is difficult.
- Supervising with an algorithm more geared toward prediction could further improve results.

- Outlook
  - Obtaining a satisfying solution to the applied problem.
  - Embedding the statistical problem in an econ problem.
  - Getting efficient approximations.