Design of Text Mining Experiments

Matt Taddy, University of Chicago Booth School of Business

faculty.chicagobooth.edu/matt.taddy/research

Active Learning: a flavor of design of experiments

'Optimal': consider the regression model when choosing data.

- classically developed for additive regression models.

Adaptive/Sequential: look where the model is least certain.

- get the best precision for a given testing budget

Simple idea, but practical application can be tough. For example, we need to be very careful with model sensitivity.

DOE question: how useful are these methods for some of our contemporary super complicated modelling schemes?

An approach that has worked well in relative low-D:

- Add points iteratively (greedy search).
- while using Monte Carlo to average over model/design-criterion uncertainty (Bayesian).

Surprisingly robust: the basic technique has been used and abused under different models and experimental settings.
Search optimization, field experiments, model calibration

Experiment Design Lesson: Be a Greedy Bayesian

Taddy, Lee, Gray, Griffen 2009 Technometrics
Taddy, Gramacy, Polson 2011 JASA
Gramacy, Lee, + ... 2008-12

Switching Gears: Analysis of Sentiment in Text

Text comes connected to interesting “author” variables

- Positive or negative opinion/feeling
- What you buy, what you watch, your reviews
- political beliefs, market/economic beliefs

Here, sentiment is *very* loosely defined:
Observables linked to variables motivating language choice

Regression Problem: model the relationship between text and sentiment inorder to predict ’missing sentiment’ from new text.

Modelling and Measuring Sentiment in Text

Text is super high dimensional,
and it gets higher dimensional as you observe more speech.

Most successful approaches *tokenize* text into words/phrases, and represent each document via term counts ('bag of words').

> *All the world's a stage, and all the men and women merely players*
> ⇒ `[all.world, stage, all, men.and.women, mere, play]`

The statistician's data units are vocabulary-length ('p') term count $\mathbf{x}$ and frequency $\mathbf{f} = \mathbf{x}/m$ vectors.

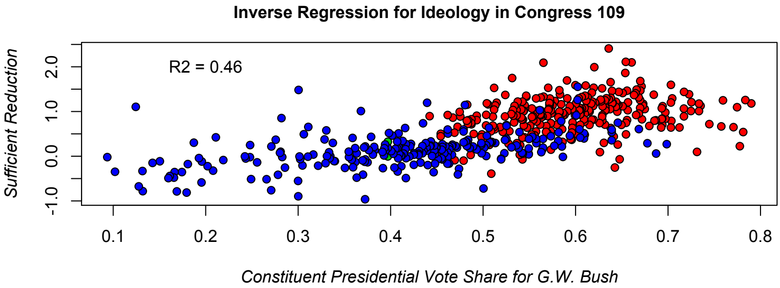
Everything is multinomial...

Multinomial Inverse Regression

Given a logistic inverse regression for sentiment y,

$$\mathbf{x}_i \sim \text{MN}(\mathbf{q}(y_i), m_i) \quad \text{with} \quad \log\left(\frac{q_{ij}}{q_{i0}}\right) = \eta_{ij} = \alpha_j + \varphi_j y_i$$

then $\mathbf{f}'\varphi$ is a *sufficient* dimension reduction: $y \perp\!\!\!\perp \mathbf{f} \mid \mathbf{f}'\varphi$.



A sort of partial least squares for count data. *(Taddy 2011)*

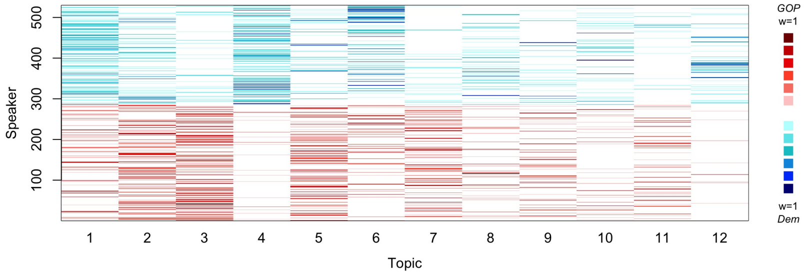
Estimation via a joint penalty-coefficient MAP algorithm.

Multinomial Topic Models

$$\mathbf{x}_i \sim \mathrm{MN}(\omega_{i1}\boldsymbol{\theta}_1 + \ldots + \omega_{iK}\boldsymbol{\theta}_K, m_i), \quad \sum_k \omega_{ik} = 1.$$

Each latent 'topic' $\boldsymbol{\theta}_k$ is a probability vector over all p terms, and $\boldsymbol{\omega}_i$ provides a low dimensional document representation.

Document-topic weights for speeches in the 109th congress



A sort of principle components analysis for multinomial data.

Pritchard, Stephens, Donnelly 2000; Blei, Ng, Jordan 2003; Taddy 2012

Joint Topic-Weight MAP estimation

Standard Approach: Introduce topic-memberships $\mathbf{z}_i$ and estimate $\boldsymbol{\Theta}$ from $\mathrm{p}(\Theta|\mathbf{X})$ via computational (MCMC) or analytic (VEM) approximate integration over $\mathbf{Z}$.

Encouraged by MNIR: how bad would a joint MAP do instead?

We use EM, without $\mathbf{Z}$, and Quadratic Programming for $\boldsymbol{\Omega}|\boldsymbol{\Theta}$

Builds on Alexander: full conditional QP, + Hoffman: EM with $\mathbf{Z}$.

Re-parametrize: solve for $\boldsymbol{\Omega}$ and $\boldsymbol{\Theta}$ transformed into natural exponential family (NEF) parameterization.

$$\text{e.g.,} \quad \boldsymbol{\varphi} \text{ where } \omega_k = \frac{\exp[\varphi_{k-1}]}{\sum_{h=0}^{K-1} \exp[\varphi_h]}, \quad \varphi_0 = 0$$

EM updates $\hat{\mathbf{\Theta}} \to \mathbf{\Theta} \mid \mathbf{\Omega}$

Topic k LHD approx is $\mathrm{MN}(\hat{\mathbf{x}}_k; \boldsymbol{\theta}_k, \hat{t}_k)$, with

$$\hat{x}_{kj} = \sum_{i=1}^{n} x_{ij} \frac{\hat{\theta}_{kj}\omega_{ik}}{\sum_{h=1}^{K} \hat{\theta}_{hj}\omega_{ih}}, \quad \hat{t}_k = \sum_{j=1}^{p} \hat{x}_{kj}.$$

Given we're maximizing in NEF space, updates are $\theta_{kj} = (\hat{x}_{kj} + \alpha_{kj})/[\hat{t}_k + \sum_{j=1}^{p} \alpha_{kj}]$.

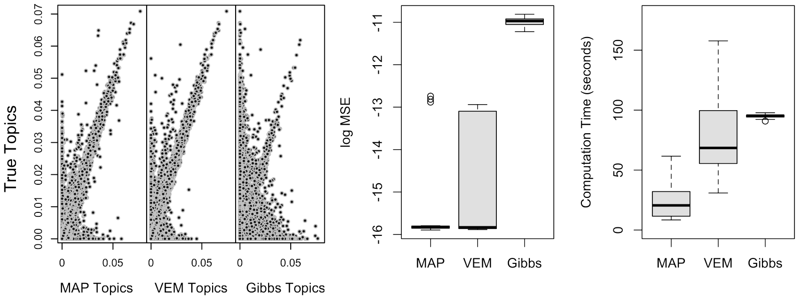
Quadratic Programming for $\mathbf{\Omega} \mid \mathbf{\Theta}$

Conveniently, $\boldsymbol{\omega}_i$ are independent given $\mathbf{\Theta}$. In NEF space, just maximize each individual

$$l(\boldsymbol{\omega}) = \sum_{j=1}^{p} x_j \log\left(\boldsymbol{\omega}\boldsymbol{\theta}_{\cdot j}\right) + \sum_{k=1}^{K} \frac{\log(\omega_k)}{K}.$$

This speeds-up EM by orders of magnitude.

Topic Fit with Simulated Data



Topic estimation with $K = 10$, $\sum_j x_j = 200$, $n = 500$
(VB via `topicmodels` and Gibbs via `lda` packages).
The more efficient MAP procedure does not suffer in accuracy.

Choosing K via Bayes Factors

Maximize marginal likelihood, approximated via Laplace as

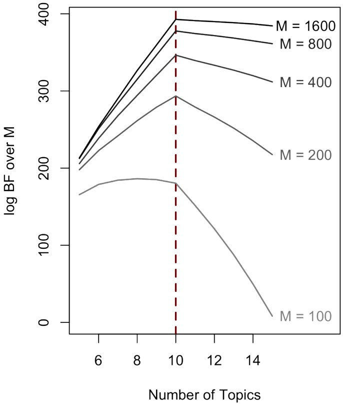
$$\mathrm{p}(\mathbf{X}|K) \approx \mathrm{p}\left(\mathbf{X}, \hat{\mathbf{\Theta}}, \hat{\mathbf{\Omega}}\right) | - \mathbf{H}|^{-\frac{1}{2}} (2\pi)^{\frac{d}{2}} K!$$

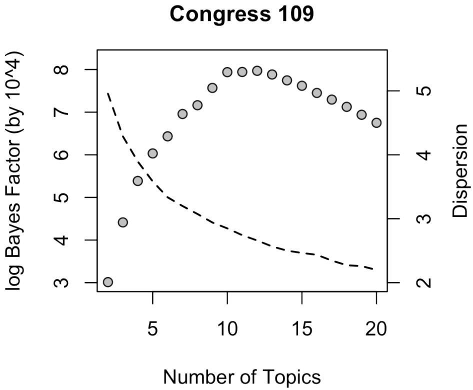
Easy to calculate *except* $|\mathbf{H}|$, posterior Hessian determinant.

Fortunately, $\mathbf{H}$ can be organized to be sparse except for blocks $\frac{\partial^2 L}{\partial \varphi_{ik} \partial \varphi_{ih}}$ and $\frac{\partial^2 L}{\partial \theta_{kj} \partial \theta_{hj}}$, and we can use a block-diagonal determinant approximation for $|\mathbf{H}|$ (precision increases with n).

This is trivial to calculate given MAP parameter estimates.

Model Selection: Choosing K





This shows selection for simulated and real data. The block diagonal Hessian approx, and Laplace approximation, appear to be doing a decent job. This will be useful in DOE...

Tracking social media brand engagement

Classify tweets as ‘pos’, ‘neg’, or ‘neutral’ on a given subject.

Categorizing Tweets about the Chicago Bears

We are investigating posts concerning the Chicago Bears NFL football team, including its players, fans, and brand.

Your task is to read the text and determine if it is positive, negative, neutral or junk as defined below:

Positive feelings regarding the Chicago Bears (e.g., this represents excitement, support, respect, or optimism)

Negative feelings regarding the Chicago Bears (e.g., this represents anger, disgust, boredom, or doubt)

Neutral contains any reference to the Chicago Bears, but it is neither positive nor negative

Junk not related to the Chicago Bears (e.g., it references another type of bear, or is spam)

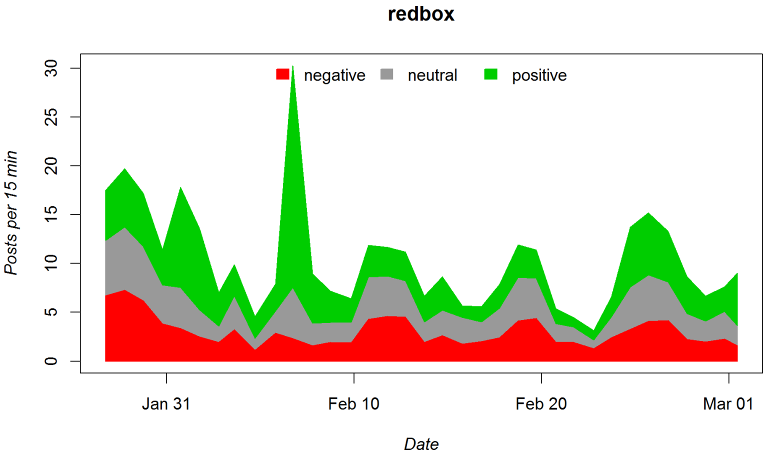
Post: *bears are eagles kryptonite; lose to them every year*

Categorize: ○ **Positive** ○ **Negative** ○ **Neutral** ○ **Junk**

Use MNIR for dimension reduction, then fit a low-D classifier.

We actually have two IR factors: general sentiment trained on 3 mil tweets, plus brand specific sentiment.

Example: Redbox dvd rental



Model updating: There are tons of tweets available, but matching them to sentiment is 'expensive' (around 10¢ each).

$\Rightarrow$ subselect an experiment design from available tweets.

Optimal Design for Text Experiments

Goal: choose $[\mathbf{x}_1, \ldots, \mathbf{x}_M]$ to minimize variance of $\mathbf{f}'\varphi$.

Problems with optimal design for the MNIR model

- Multivariate 'response' and IR trickery means that standard univariate learning metrics do not apply.

  It's not clear how to build a search criterion
- Vocabulary is growing, which is good, but which can also increase variance. Plus, we want to learn when p=1/2.
- Uncertainty about φ is expensive to quantify (e.g., the information matrix for φ is dense and high-dimensional) and very sensitive to current fit.

Instead, leverage what we have lots of: text!

We can fit a big topic model without knowing y.

Leap-of-faith: *sentiment is linear in latent topic-factor space*

$\Rightarrow$ linear model techniques for selecting $\mathbf{W}_M = [\boldsymbol{\omega}_1, \ldots, \omega_M]'$.

Topic D-Optimal Designs: maximize $|\mathbf{W}'\mathbf{W}|$.

i.e., minimize determinant of least squares covariance.

Be Greedy!

$$D_M = |\mathbf{W}_M'\mathbf{W}_M| \Rightarrow D_{M+1} = D_M\left[1 + \omega_{M+1}'(\mathbf{W}_M'\mathbf{W}_M)^{-1}\omega_{M+1}\right]$$

So we just select ω_{M+1} to max $\omega'(\mathbf{W}_M'\mathbf{W}_M)^{-1}\omega$.

This is easy in reduced dimension (K).

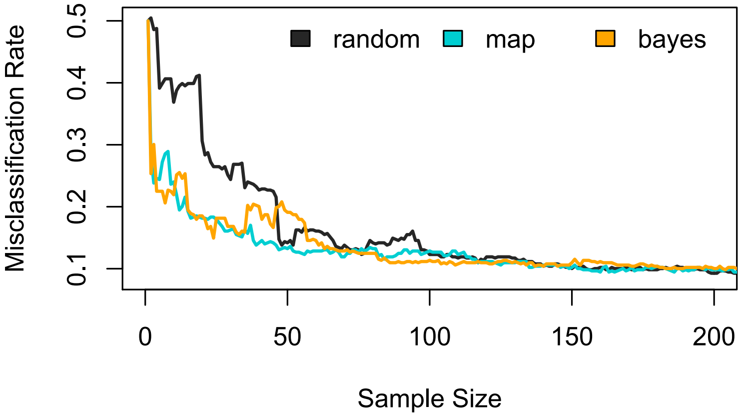
Be Bayesian!

ω's are MAP estimated: there is uncertainty. However...

- They are roughly independent of each other given $\boldsymbol{\Theta}$.
- Reparam $\omega_k = \dfrac{\exp[\varphi_{k-1}]}{\sum_{h=0}^{K-1}\exp[\varphi_h]}$ and things look Gaussian.

We can use the same Laplace approx as in marginal likelihood calculation and simulate $\boldsymbol{\omega}_i$'s to max average for D_{M+1}.

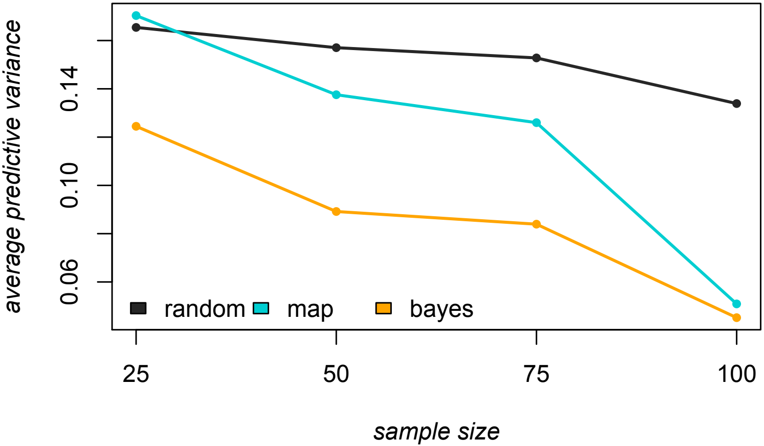
109th Congress: Designed Sentiment Sampling



In this example, we have a ground truth to compare against

Both greedy approaches give big initial gains. The Bayesian version is more stable (it never pops up like the MAP).

Redbox: a little predictive variation experiment



Metric is $\mathbb{E}[\mathrm{var}(\varphi'\mathbf{F})]$, the variance of our d.r. projection.

All of this is a bit hasty so far...

- Is our variance approximation capturing what we need?
- What are the effects of growing vocabulary?

Would it be better to just count significant tokens?

Thanks for listening!