Big Data, Model Complexity, and Interpretability: Machine Learning and Finance

Sanmay Das

Washington University in St. Louis

MFM Summer School 2017
Statistics/Econometrics “vs.” Machine Learning

- Differences in communities, culture, practices
  - BUT: communities are learning a lot from each other and coming closer
  - Ideas come from both sides (e.g. bootstrap came from statistics to ML)

- (Supervised) ML has more of a focus on:
  - (Out-of-sample) prediction (vs. hypothesis testing or causal inference)
  - Algorithmic questions (esp. in high-dimensional problems)
  - Empirical (finite-sample) evaluation vs. asymptotics
Predictive Inference

- Clear use case: Solve a prediction problem and use the predictions for a meaningful task.
- E.g. “Prediction Policy Problems” (Kleinberg et al., 2015) example:

  - ≈ 500,000 Medicare beneficiaries receive hip or knee replacements every year
  - Costs are both monetary and quality-of-life (first 6 months are particularly tough for recipients, but outcomes improve by 12 months)
  - However, 1.4% of recipients die in the month after surgery and 4.2% in months 1-12
  - These 4.2% are highly predictable using ML methods. For this population, having the surgery was probably a bad decision in terms of QOL.
  - Don’t need to establish causality in order to improve outcomes in expectation (ethical issues can be a concern).
Goals

- Explain how machine learners view the world.
- Discuss what we’re good at, and what the state of the art is, with a focus on things that might be of value to MFM scholars.
- Talk about a couple of examples in a bit of detail.
Goals

- Explain how machine learners view the world.
- Discuss what we’re good at, and what the state of the art is, with a focus on things that might be of value to MFM scholars.
- Talk about a couple of examples in a bit of detail.
- Discuss causality: Machine learning has a lot to learn
- Talk about using ML to sell ad space on the web
Outline

- Basic framework: How (supervised) machine learners think
- State of practice in prediction problems
  - Algorithms: SVMs, ensemble methods, etc.
  - Optimization and design choices
- Prediction problems and applications in finance
- Text analysis and applications
Outline

- **Basic framework: How (supervised) machine learners think**
- State of practice in prediction problems
  - Algorithms: SVMs, ensemble methods, etc.
  - Optimization and design choices
- Prediction problems and applications in finance
- Text analysis and applications
The Supervised Learning Problem

- Unknown target function $f : \mathcal{X} \rightarrow \mathcal{Y}$
- Training data $\mathcal{D} : (x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)$ where $y_i = f(x_i)$ (possibly noisy).
- Want to learn $h$ “close to” $f$.
- Two central questions:
  - How do we learn $h$?
  - What can we say about how close $h$ is to $f$?

(Note: My development and notation here follows that of Abu-Mostafa, Magdon-Ismail, and Lin (2012)).
Generalization Error

I'D LIKE TO THANK ALL OF THE PEOPLE WHO HELPED DESIGN THE TECHNOLOGY TEST PARAMETERS.

THANKS TO YOUR INPUT, THE TEST HAD NOTHING IN COMMON WITH HOW THINGS WORK IN THE REAL WORLD.

SO I WASTED TWO WEEKS OF MY LIFE ON A TEST THAT IS NOT ONLY MEANINGLESS... ...BUT ALSO DANGEROUSLY MISLEADING.

THIS SLIDE SHOWS THE GAP BETWEEN THE TEST RESULTS AND REALITY.

WE'LL USE THE TEST RESULTS ANYWAY BECAUSE IT'S THE ONLY DATA WE HAVE.

FINE. I HOPE YOU ALL CHOKED TO DEATH ON YOUR LUNCHES.

WHY'S HE SO CRANKY? SOMETHING ABOUT DATA.

(©Scott Adams: http://dilbert.com/strips/comic/2010-11-07/)
Generalization Error

- Standard for closeness that we care about
  \[ E_{out}(h) = \Pr[h(x) \neq f(x)], \text{ where the probability is based on the sampling distribution on } \mathcal{X}. \]

- In practice, we estimate \( E_{out} \) by evaluating on a (held-out) test set.

- There are a ton of interesting problems when the sampling distribution for test data is not the same as that for \( \mathcal{D} \).
How Do We Learn $f$?

- Pick a hypothesis set $\mathcal{H} = \{h_1, h_2, \ldots, \}$
- Use a learning algorithm to select a hypothesis from $\mathcal{H}$ on the basis of $\mathcal{D}$.
- The choice of $\mathcal{H}$ and the learning algorithm are intertwined.
How Do We Learn $f$?

- Pick a hypothesis set $\mathcal{H} = \{h_1, h_2, \ldots, \}$
- Use a learning algorithm to select a hypothesis from $\mathcal{H}$ on the basis of $\mathcal{D}$.
- The choice of $\mathcal{H}$ and the learning algorithm are intertwined
- No free lunch in machine learning

![Diagram showing different hypotheses and their corresponding $f$ values]

\[ f = -1 \]
\[ f = +1 \]
\[ f = ? \]
Choosing $h$ from $\mathcal{H}$

- First thought: Minimize $E_{in}(g) = \frac{1}{n} \sum_{i=1}^{n} [h(x_i) \neq f(x_i)]$
- Many algorithms can be thought of within this broad framework.
  - Linear regression: Find a weight vector $w$ that minimizes
    $$E_{in}(w) = \frac{1}{n} \sum_{i=1}^{n} (w^T x_i - f(x_i))^2$$
  - Logistic regression: Find a linear function that minimizes
    $$E_{in}(w) = \frac{1}{n} \sum_{i=1}^{n} \ln(1 + e^{-y_i w^T x_i})$$
  - Decision trees: Find the tree that directly minimizes the above.
    Problem: Computationally intractable, so we use greedy heuristics
Minimizing $E_{out}$

But $E_{in}$ is not really our objective. Vapnik-Chervonenkis and PAC theory tell us (roughly) that $E_{out}(g) \leq E_{in}(g) + \Omega(\mathcal{H})$ where $\Omega(\mathcal{H})$ penalizes complexity of the hypothesis class. Gives us two objectives:

- Control hypothesis-class complexity
- Minimize $E_{in}$

(from (Abu-Mostafa, Magdon-Ismail, and Lin, 2012))
The Central Problems

- There are deep relationships between the stability and variance of a learning algorithm, hypothesis complexity, and generalization ability.
- Bigger data $\rightarrow$ more complex hypothesis spaces can generalize better.
- Different ML algorithms arise from different choices related to two questions:
  - What $\mathcal{H}$ to search
  - What and how to optimize in the search process
Outline

- Basic framework: How (supervised) machine learners think
- **State of practice in prediction problems**
  - **Algorithms:** SVMs, ensemble methods, etc.
  - Optimization and design choices
- Prediction problems and applications in finance
- Text analysis and applications
Regularized logistic regression: Minimize

\[ E_{\text{in}}(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} \ln(1 + e^{-y_i \mathbf{w}^T \mathbf{x}_i}) + \lambda \mathbf{w}^T \mathbf{w}. \]

- Often used with text data.
- In general performs better than Naive Bayes.

Support vector machines: Minimize

\[ E_{\text{in}}(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} (1 - y_i (\mathbf{w}^T \mathbf{x}_i))^+ + \lambda \mathbf{w}^T \mathbf{w}. \]

\[ (z)^+ \equiv \max(z, 0) \]

is the hinge loss.

- Same thing as maximizing the margin. The dual QP has a nice interpretation in terms of “support vectors” (points at or within the margin).
- When combined with the kernel trick, SVMs give a natural and powerful regularized non-linear classifier.
- Try quadratic and RBF kernels if needed.
Regularized logistic regression: Minimize
\[ E_{\text{in}}(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} \ln(1 + e^{-y_i \mathbf{w}^T \mathbf{x}_i}) + \lambda \mathbf{w}^T \mathbf{w}. \]
- Often used with text data.
- In general performs better than Naive Bayes.

Support vector machines: Minimize
\[ E_{\text{in}}(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} (1 - y_i (\mathbf{w}^T \mathbf{x}_i)_+) + \lambda \mathbf{w}^T \mathbf{w} \]
where \((z)_+ \equiv \max(z, 0)\) is the hinge loss.
- Same thing as maximizing the margin. The dual QP has a nice interpretation in terms of “support vectors” (points at or within the margin).
- When combined with the kernel trick, SVMs give a natural and powerful regularized non-linear classifier.
- Try quadratic and RBF kernels if needed.
Decision Trees

- Flexible and rich hypothesis space. Easy to express trees in terms of a combination of rules.
- Impossible to find the optimal tree, so we use greedy heuristics to construct the tree.
- High variance classifier
  - Because of the greedy nature, a small change can have huge cascading effect, leading to totally different trees.
- Frequently used as a building block for ensemble methods.
Random Forests

- Construct $k$ decision trees on bootstrapped replicates of the training set.
- Use random feature selection on the nodes (typically $\sqrt{\# \text{ features}}$) to decorrelate trees further.
- The ensemble votes on any new example.

Off-the-shelf state of the art performance. Out-of-bag error provides a great estimate of out-of-sample error for free. Implementations come along with a feature scoring method that gets used a lot (measuring decrease in accuracy when permuting a feature in all OOB samples).
Random Forests

- Construct $k$ decision trees on bootstrapped replicates of the training set.
- Use random feature selection on the nodes (typically $\sqrt{\# \text{ features}}$) to decorrelate trees further.
- The ensemble votes on any new example.
- Off-the-shelf state of the art performance.
- Out-of-bag error provides a great estimate of out-of-sample error for free.
- Implementations come along with a feature scoring method that gets used a lot (measuring decrease in accuracy when permuting a feature in all OOB samples)
Boosting

- Arose as the answer to a theoretical question: Given a black-box weak learner (slightly better than chance), can we “boost” it into a strong learner.
- Key idea: Each new weak learner is trained on a distribution of examples modified to give more weight to those the existing ensemble has most trouble with.
Boosting

- Arose as the answer to a theoretical question: Given a black-box weak learner (slightly better than chance), can we “boost” it into a strong learner.
- Key idea: Each new weak learner is trained on a distribution of examples modified to give more weight to those the existing ensemble has most trouble with.
- A variant called gradient boosting is also state-of-the-art.
- Typically use decision stumps or short decision trees as the weak learners.
- If you need a powerful ML algorithm, try random forests and/or gradient boosting first.
Deep Learning

- (Massively) multilayer neural networks, typically trained using GPUs with huge training datasets.
- Optimization, structure, and regularization are all arts that are slowly becoming more “off the shelf” but aren’t there yet.
- It appears they learn interesting intermediate representations.
- Have been hugely influential in computer vision and NLP.
- Why they work is still a major theoretical puzzle.
Outline

- Basic framework: How (supervised) machine learners think
- **State of practice in prediction problems**
  - Algorithms: SVMs, ensemble methods, etc.
  - **Optimization and design choices**
- Prediction problems and applications in finance
- Text analysis and applications
Model Selection

- How strong a regularizer? How many trees?
- Classic answer: Cross-validation and grid search over the space of parameters

Taking over: Bayesian Optimization

- Use a Gaussian Process prior over the value of the function being optimized (say error) and iteratively evaluate at different hyperparameter values (Shahriari et al., 2016; Snoek, Larochelle, and Adams, 2012; Osborne, Garnett, and Roberts, 2009).

- Spearmint (https://github.com/HIPS/Spearmint) is a well-known package for BO.
Model Selection

- How strong a regularizer? How many trees?
- Classic answer: Cross-validation and grid search over the space of parameters
- Taking over: Bayesian Optimization
  - Use a Gaussian Process prior over the value of the function being optimized (say error) and iteratively evaluate at different hyperparameter values (Shahriari et al., 2016; Snoek, Larochelle, and Adams, 2012; Osborne, Garnett, and Roberts, 2009).
  - Spearmint (https://github.com/HIPS/Spearmint) is a well-known package for BO.
Outline

- Basic framework: How (supervised) machine learners think
- State of practice in prediction problems
  - Algorithms: SVMs, ensemble methods, etc.
  - Optimization and design choices
- Prediction problems and applications in finance
- Text analysis and applications
Can predictive analytics provide business or regulatory value and insight?

- Khandani, Kim, and Lo (2010) show that banks can use ML on combined credit bureau and monthly transaction data to manage credit lines more profitably.

Many of these have a "horse race" element. One major issue is the Lucas Critique. Will get back to this...
Some Trends in Current Research

- Can predictive analytics provide business or regulatory value and insight?
  - Khandani, Kim, and Lo (2010) show that banks can use ML on combined credit bureau and monthly transaction data to manage credit lines more profitably.
  - Kong and Saar-Tsechansky (2014) discuss active learning in order to identify tax returns that would be particularly informative to audit.

Many of these have a “horse race” element. One major issue is the Lucas Critique. Will get back to this...
Some Trends in Current Research

- Can predictive analytics provide business or regulatory value and insight?
  - Khandani, Kim, and Lo (2010) show that banks can use ML on combined credit bureau and monthly transaction data to manage credit lines more profitably.
  - Kong and Saar-Tsechansky (2014) discuss active learning in order to identify tax returns that would be particularly informative to audit.
  - Giesecke, Sirignano, and Sadhwani (2016) use deep learning to estimate state transitions for mortgages.

- Many of these have a “horse race” element.
- One major issue is the Lucas Critique. Will get back to this...
Some Trends in Current Research

Can predictive analytics provide business or regulatory value and insight?

- Khandani, Kim, and Lo (2010) show that banks can use ML on combined credit bureau and monthly transaction data to manage credit lines more profitably.
- Kong and Saar-Tsechansky (2014) discuss active learning in order to identify tax returns that would be particularly informative to audit.
- Giesecke, Sirignano, and Sadhwani (2016) use deep learning to estimate state transitions for mortgages.
- Butaru et al. (2016) learn models for predicting CC delinquency across several large banks (and compare risk management practices across banks).
Some Trends in Current Research

- Can predictive analytics provide business or regulatory value and insight?
  - Khandani, Kim, and Lo (2010) show that banks can use ML on combined credit bureau and monthly transaction data to manage credit lines more profitably.
  - Kong and Saar-Tsechansky (2014) discuss active learning in order to identify tax returns that would be particularly informative to audit.
  - Giesecke, Sirignano, and Sadhwani (2016) use deep learning to estimate state transitions for mortgages.
  - Butaru et al. (2016) learn models for predicting CC delinquency across several large banks (and compare risk management practices across banks).

- Many of these have a “horse race” element.

- One major issue is the Lucas Critique. Will get back to this...
Predicting Credit Card Delinquency

- Rapidly growing consumer credit market.
- High charge-off rate and severe delinquency rate during the financial crisis highlight the need for robust and improved out-of-sample prediction models for risk management.

![Graph showing charge-off rate on credit card loans, all commercial banks from 1985 to 2015.](source)

With good predictions of delinquency, banks can actively manage credit to manage their exposure (Khandani, Kim, and Lo, 2010).
Project Overview

- Immediate goal: Support risk-based supervision
  - Compare performance of ML methods vs. “traditional” logistic regression.
  - Compare risk management across banks.

- Future goal: Measure systemic risk
  - Combine data across institutions.
  - Generate aggregate forecast models.
  - Use in stress testing?
Data Sources

- **Credit card data**
  - Detailed account-level characteristics (days past due, balance, utilization, FICO, etc.)
    - Cannot link the accounts across individuals
  - Collected by a major US regulator
  - Entire credit card portfolios of 6 large banks
  - Monthly data starting from January 2008

- **Attribute data**
  - Detailed borrower characteristics from a credit bureau (linked by account)
  - Quarterly starting in 2009

- **Macroeconomic variables**
  - Collected from various sources (linked by account ZIP)
  - Employment data, HPI, average wages, average hours, etc.

- In total, many TB of raw data
Sample Description

- Approx. 1% of each bank’s CC portfolio.
- Yields between 90K and 1M observations per period per bank.
- Portfolio size varies over time (some grow, some decline) and the sample size is representative of the true portfolio.
- Substantial heterogeneity in the time series and cross-sectional distribution of delinquency rates.
Modeling Dominates Credit Scores

Bank 2
Risk Ranking versus Credit Score
2 Quarter Forecast
Model: Logistic

Model Derived Risk Ranking (0-1)
Current Accounts
30 DPD Accounts
60 DPD Accounts
90+ DPD Accounts

Credit Score
Value-Added Analysis

- Compute a hypothetical cost-saving per customer, assuming the bank would cut lines of those predicted to default. Depends on:
  - Classification accuracy (the whole confusion matrix): **Quality of the model**
  - Run-up ratio (how much customers will increase their balance after time of prediction)
  - Profitability margin: Expected future cash-flows (opportunity cost of misclassification)

- Based on the framework of (Khandani, Kim, and Lo, 2010).
Value-Added Comparison

All Banks Logistic Models: Value Added (%)
2 Quarter Forecast

All Banks Random Forest Models: Value Added (%)
2 Quarter Forecast

Sanmay Das (WUSTL)
Most Important Attributes For Prediction

- Days past due
- Behavioral score
- Refreshed credit score
- Ratio of actual to minimum payments
- 1 month change in monthly utilization
- Payment equal to minimum in last 3 months
Further Insights

- Banks vary greatly in their risk management practices and effectiveness
  - 3 of the banks are very effective at cutting lines of accounts predicted to become delinquent
- Risk factors affecting delinquency vary across banks
- Macroeconomic variables affect different banks differently
  - Were more important factors in the model in 2012 and earlier than they are now.
Are Models For Predicting Failure Destined to Fail?

- So what if you can predict these things well in hindsight?
- The basic problem: Agent incentives and Goodhart’s law (Wikipedia’s formulation: “When a measure becomes a target, it ceases to be a good measure”).
- If the ML model (or any risk model (Danielsson, 2002)) is part of the regulatory system, it will break down, because the data generating process will change.

Rajan, Seru, and Vig (2015) study this empirically in the context of subprime mortgage loans originated from 1997–2006. Securitization changes lenders’ incentives and behavior. Over time, borrowers with similar “hard” information in the data become worse risks because they are worse along the “soft” information dimension (unobservable to investors).

“When incentive effects lead to a change in the underlying regime, the coefficients from a statistical model estimated on past data have no validity going forward, regardless of how sophisticated the model is or how well it fits the prior data.”

Sanmay Das (WUSTL)
Machine Learning and Finance
MFM Summer School 2017
Are Models For Predicting Failure Destined to Fail?

- So what if you can predict these things well in hindsight?
- The basic problem: Agent incentives and Goodhart’s law (Wikipedia’s formulation: “When a measure becomes a target, it ceases to be a good measure”).
- If the ML model (or any risk model (Danielsson, 2002)) is part of the regulatory system, it will break down, because the data generating process will change.
  - Securitization changes lenders’ incentives and behavior. Over time, borrowers with similar “hard” information in the data become worse risks because they are worse along the “soft” information dimension (unobservable to investors).
  - “[When] incentive effects lead to a change in the underlying regime, the coefficients from a statistical model estimated on past data have no validity going forward, regardless of how sophisticated the model is or how well it fits the prior data.”
Approach: Don’t Share the Model

New York Times, “Not communicating to banks exactly what they need to do to get their bankruptcy plans to pass muster is frustrating, confusing and part of the plan.”

How Regulators Mess With Bankers’ Minds, and Why That’s Good

Peter Eavis  @petereavis  APRIL 14, 2016

Bank regulators on Wednesday sent a message that big banks are still too big and too complex. They rejected special plans, called living wills, that the banks have to submit to show they can go through an orderly bankruptcy.

The thinking behind the regulators’ call for living wills is that if a large bank crash is orderly, there will be no need to save it and no need for taxpayer bailouts.

Pretty straightforward, right? Not for the banks. The regulators deliberately did not communicate the exact things the banks needed to do for their plans to pass muster. In this way, they kept them on their toes — and treating powerful banks this way may end up playing a surprisingly important role in keeping the financial regulation effective over time.
Issues

- The regulatory model will still affect the system even if it doesn’t explicitly change agent behavior
- Transparency can be important in many contexts
- Can the model be backed out from results?
- Not insurmountable in many applications
Approach: CS Work on Privacy and Fairness?

- Dwork, Hardt, et al. (2012) operationalize fairness
  - Given a (task-specific) similarity metric on individuals (can be designed by a regulator), the classifier must produce distributions on outcomes for these individuals that are bounded by a function of the distance between them.
  - Both they and Feldman et al. (2015) also consider statistical discrimination and disparate impact.

- Differential privacy (Dwork, McSherry, et al., 2006): Participation of a single person does not change the outcome significantly.

- While these ideas don’t immediately generalize to the “model in the loop” problem, could there be a solution with a similar flavor?
Approach: Robust Control?

- Explicitly think about closed-loop control with the prediction algorithm in the loop.
- Combine a structural or causal model of agent decisions with a predictive model that also includes other observables (Rajan, Seru, and Vig, 2015).
- Could inform policy at a broader level.
Adoption of ML

- These problems are really not specific to using ML techniques. Banks and regulators already use models to predict the future. ML typically does it better.
- Not meant to replace causal models where needed.
- In practice, have to re-train models regularly to deal with nonstationarities.
Adoption of ML

- These problems are really not specific to using ML techniques. Banks and regulators already use models to predict the future. ML typically does it better.
- Not meant to replace causal models where needed.
- In practice, have to re-train models regularly to deal with nonstationarities.
- The myth of interpretability
  - Are logistic regression coefficients really meaningful?
  - Single decision trees are equivalent to sets of rules, but decision tree learners are inherently unstable. How meaningful is the tree?
  - Often used synonymously with “manually sanity checkable”
  - There are ways to sanity check more complex models as well (random forest feature scores, checking which features drive changed in NN output, etc.)
Outline

- Basic framework: How (supervised) machine learners think
- State of practice in prediction problems
  - Algorithms: SVMs, ensemble methods, etc.
  - Optimization and design choices
- Prediction problems and applications in finance
- Text analysis and applications
Text Analysis

- Big idea: Ton of information in text that hasn’t been quantified and analyzed, but could be usable.
- Examples:
  - Gentzkow and Shapiro (2010) measure ideological slant by identifying trigrams used much more frequently by Democrats or Republicans in the congressional record, and use this to estimate the drivers of media slant.
Text Analysis

- Big idea: Ton of information in text that hasn’t been quantified and analyzed, but could be usable.
- Examples:
  - Gentzkow and Shapiro (2010) measure ideological slant by identifying trigrams used much more frequently by Democrats or Republicans in the congressional record, and use this to estimate the drivers of media slant.
Big idea: Ton of information in text that hasn’t been quantified and analyzed, but could be usable.

Examples:

- Gentzkow and Shapiro (2010) measure ideological slant by identifying trigrams used much more frequently by Democrats or Republicans in the congressional record, and use this to estimate the drivers of media slant.
- Manela and Moreira (2016) construct a measure of aggregate uncertainty “News Implied Volatility” (NVIX). ML (SVR) to map from bag-of-bigrams representations of WSJ articles to VIX in the observed period, and can then generalize to before VIX existed.
Methods

- Keyword counting. Where do we get the keyword lists?
- Linear models (SVMs, regularized logistic regression)
  - Mallet (http://mallet.cs.umass.edu/) is a great toolbox!
- Word embeddings (e.g. Word2vec) are hot (Mikolov et al., 2013).
- Deep models + word embeddings are promising, including for sentiment analysis (Santos and Gatti, 2014; Tang et al., 2014).
- Big question in many ML applications: where do we get the labels? (Now: Often crowdsourcing)
- Two big caveats: generalization across datasets and across time. Language changes fast!
A Cautionary Tale: Political Ideology Classification

Would it be feasible to train on one data from one context (e.g. speeches in congress) and learn the bias of writers in another (e.g. Wikipedia)?
Would it be feasible to train on one data from one context (e.g. speeches in congress) and learn the bias of writers in another (e.g. Wikipedia)?

We try it with data from (Yan, Lavoie, and Das, 2017):
- Salon and Townhall (media)
- Congressional Record (politicians)
- Conservapedia vs. RationalWiki (the crowd)

Algorithms:
- Logistic regression on n-grams (Bag-of-bigrams, TFIDF, feature hashing)
- Recursive autoencoder (RAE) (Socher et al., 2011).
## Generalizing Across Datasets

<table>
<thead>
<tr>
<th>Training</th>
<th>Test</th>
<th>Cong. Rec.</th>
<th>Salon &amp; Townhall</th>
<th>Wikis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cong. Rec.</td>
<td></td>
<td>0.79 (LR)</td>
<td>0.63 (LR)</td>
<td>0.52 (LR)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.78 (RAE)</td>
<td>0.56 (RAE)</td>
<td>0.46 (RAE)</td>
</tr>
<tr>
<td>Salon &amp; Townhall</td>
<td></td>
<td>0.55 (LR)</td>
<td>0.81 (LR)</td>
<td>0.54 (LR)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.54 (RAE)</td>
<td>0.93 (RAE)</td>
<td>0.62 (RAE)</td>
</tr>
<tr>
<td>Wikis</td>
<td></td>
<td>0.50 (LR)</td>
<td>0.49 (LR)</td>
<td>0.80 (LR)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.46 (RAE)</td>
<td>0.58 (RAE)</td>
<td>0.87 (RAE)</td>
</tr>
</tbody>
</table>
Generalizing Across Time

Salon–Townhall Predictions Over Time (Trained on 2006)

AUC vs Year


AUC: 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1

Sanmay Das (WUSTL)
Machine Learning and Finance
MFM Summer School 2017

Page 44 of 52
The Across-Dataset Problems Are Not Driven by Time

<table>
<thead>
<tr>
<th>Training</th>
<th>Test</th>
<th>Cong. Rec.</th>
<th>Salon &amp; Townhall</th>
<th>Wikis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cong. Rec.</td>
<td>0.82 (LR)</td>
<td>0.64 (LR)</td>
<td>0.51 (LR)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.81 (RAE)</td>
<td>0.59 (RAE)</td>
<td>0.47 (RAE)</td>
<td></td>
</tr>
<tr>
<td>Salon &amp; Townhall</td>
<td>0.56 (LR)</td>
<td>0.91 (LR)</td>
<td>0.50 (LR)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.54 (RAE)</td>
<td>0.90 (RAE)</td>
<td>0.55 (RAE)</td>
<td></td>
</tr>
<tr>
<td>Wikis</td>
<td>0.50 (LR)</td>
<td>0.54 (LR)</td>
<td>0.82 (LR)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.47 (RAE)</td>
<td>0.57 (RAE)</td>
<td>0.82 (RAE)</td>
<td></td>
</tr>
</tbody>
</table>
Takeaways

- Must be very careful in generalizing across types of text, especially for short texts.
- Temporal change of language can be a big issue.
- Big pro for finance: there may be natural labels to exploit.
- Perhaps combining structural and textual information will help?
  - But even semantically rich embeddings can be problematic in the political context (e.g. “private accounts” vs. “personal accounts” for social security).
In Conclusion

- Machine learning is very good with:
  - Out-of-sample prediction.
  - Big data.
  - Nonlinear models.
  - Designing new measures
  - “Not immediately quantitative” data (text, images)

- When these strengths speak well to a problem, it’s silly not to use the best techniques we have available!
References I


References II


References III


References IV


