About Me

- UW PhD Student in AI
- CS886 Alumnus
- 7 years doing ML research
Outline

1. My CS886 Experience
2. Short Break?
3. Other Studies and Questions
The Point

- See an example of a "good" project
- Understand my thought process at the time
- Improve your projects, and avoid pitfalls
The Topic
Finding a Great Topic

Pick a topic that:

- you know about
Finding a Great Topic

Pick a topic that:
- you know about
- you’re excited about
Finding a Great Topic

Pick a topic that:

- you know about
- you’re excited about
- is self contained and testable
Voting

- A way to take collective actions
- Incorporates everyone’s opinion
- (Potentially) produces a fair outcome
### Applications of Voting

*Guest Lecture*

John A. Doucette (U. Waterloo)

October 15, 2014

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Barrett</td>
<td>Scot Barrett, Senior Lecturer, TCD; Department of Economics, Arts Building, University of Dublin, Trinity College Dublin, DUBLIN 2.</td>
</tr>
<tr>
<td>Coleman</td>
<td>Marc Coleman, Broadcaster, Journalist, Economist; 8 St. John's Close, Mounttown Road, DUN LAOGHAIRE.</td>
</tr>
<tr>
<td>Cox</td>
<td>Meave Cox, TCD Graduate, Barrister-at-Law Degree Student; Scranghagh School House, Airlow, CO. WICKLOW.</td>
</tr>
<tr>
<td>Donnelly</td>
<td>Francis Vincent Donnelly, Chief Financial Officer, Company Director; 9 Carrigill Lower, Ballyshannon, Kilcullen, CO. KILDARE.</td>
</tr>
<tr>
<td>Dubsky</td>
<td>Karin Dubsky, Environmental Scientist; White Wells, Balymoney, Gorey, CO. WEXFORD.</td>
</tr>
<tr>
<td>Dudgeon</td>
<td>Jeffrey Edward Anthony Dudgeon, Author, Human Rights Campaigner, Former Civil Servant; 56 Mount Prospect Park, Belfast, BT9 7BG, CO. ANTRIM.</td>
</tr>
<tr>
<td>Frost</td>
<td>Dermot Frost, I.T. Specialist; 119 Clarranard Road, Donnycarney, DUBLIN 5.</td>
</tr>
<tr>
<td>Guérin</td>
<td>Maurice Guérin, Medical Doctor; Mimosa Lodge, 46 Fortfield Road, Terenure, DUBLIN 9W.</td>
</tr>
</tbody>
</table>
# Ranked Ballot Voting

**Electorate of Brindabella**

**Number five boxes from 1 to 5 in the order of your choice**

You may then show as many further preferences as you wish by writing numbers from 6 onwards in other boxes.

<table>
<thead>
<tr>
<th>A CANBERRA LIBERALS</th>
<th>B THE ACT GREENS</th>
<th>C ACT LABOR</th>
<th>D BULLET TRAIN FOR CANBERRA</th>
<th>E AUSTRALIAN MOTORIST PARTY</th>
<th>UNGROUPED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Andrew WALL</td>
<td>Johnathan DAVIS</td>
<td>Joy BURCH</td>
<td>Mark ERWOOD</td>
<td>Kieran JONES-ELLIS</td>
<td>Michael LINFIELD</td>
</tr>
<tr>
<td>Zed SESELJA</td>
<td>Ben MURPHY</td>
<td>Rebecca CODY</td>
<td></td>
<td></td>
<td>INDEPENDENT</td>
</tr>
<tr>
<td>Val JEFFERY</td>
<td>Amanda BRESNAN</td>
<td>Karl MAFTOUM</td>
<td></td>
<td></td>
<td>Calvin PEARCE</td>
</tr>
<tr>
<td>Nicole LAWDER</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>INDEPENDENT</td>
</tr>
<tr>
<td>Brendan SMYTH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mark GIBBONS</td>
</tr>
</tbody>
</table>

Remember, number at least five boxes from 1 to 5 in the order of your choice.
**Advantages**

- Easier to be fair
- Easy to find
  nth-place
- Lots of systems
Why Use Ranked Ballot Voting?

John A. Doucette (U. Waterloo)
What I was thinking:

- Voting is exciting!
- New to me, but some experience
- Need a self-contained topic
Completing a ranked ballot is hard:

- Expensive to learn about candidates
- Too many candidates
- Too many interdependent values
Could we use learning to help?

- Candidates often *ordered*
- Voters often predictable
- Voters think alike
Can ML Help?

John A. Doucette (U. Waterloo)
Can ML Help?

John A. Doucette (U. Waterloo)
Guest Lecture
October 15, 2014
Previously Shown

- “Safe” when done carefully
- “Possible” to learn preferences
- Lots of bias toward popular candidates.
Outstanding Questions

Questions:
- Which algorithm works best?
- Which features are predictive?
- Are we fair to all candidates?
The Final Topic

- Find the most accurate combination of features and algorithms
- Find the least unfair combination of features and algorithms
- Compare across many datasets to eliminate bias
Meta: Narrowing Your Topic

**Narrowing Your Topic**

- Find an expert and ask for ideas
- Find papers that solve problems without ML
- Make sure no one’s solved your problem already!
Finding Voting Data

1. Read papers in the area, found “standard” data
2. Browsed government databases, found some data, but low quality
3. Asked around, heard about a new repository
Finding Data

- Pitch to everyone
- If a paper uses the dataset you want, email the authors!
- Don’t be afraid to spend some time finding the best data you can
Aside: Data Authenticity

Issues with data repositories:

- Research optimizes against the repo
- Repo data is not always realistic
- Active debate in many areas, be aware
Preprocessing

My Data

- Clean, uniform format.
- Not all usable (poor ground truth)
- Format is non-standard, and poorly documented
### Preprocessing

### Cleaning

- Removed unusable data
- Is what’s left representative?
Ablation Studies

1. Measure missingness in raw data
2. Clean
3. Randomly ablate cleaned data using measured distribution
4. Repeat 3 many times
Aside: Preprocessing

### Problems

- Do voters with complete ballots have different views?
- Ablation study doesn’t account for this
- **Check pertinent aggregates in raw and ablated sets**
Feature Extraction

1. Absolute Positions
2. Relative Positions
3. Indicators
4. Triplet relations

\[ a \succ b \succ c \]

1. \( \text{Pos}(a) = 1 \)
2. \( \text{Pos}(a) - \text{Pos}(b) = -1 \)
3. \( I(a \succ b) = 1 \)
4. \( I(a \succ b \land a \succ c) = 1 \)
Picking Good Features

- Information *you* use to answer the question
- Information that your algorithm can work with
- Pitch to others, ask how what they’d use
Feature Selection

- Small scale testing to find potentially useful techniques
- Settled on IG and PCA
- Parametrization matters
Tools

- Bash/Perl/Python scripts to turn raw data into matrix of features (fast prototyping, easy to verify correctness)
- Heavier tools (mostly R) to run feature selection
- Not always easy to use
Development Choices

- Fast development is hard to maintain, and bugs are hard to find
- More is learned from rolling your own, but get it right!
- No one thinks “I wish I’d spent less time designing my experiment.”
Classifiers for Voting

- Multinomial Dist. Learner (multi-class)
- Naive Bayes (baseline)
- SVM (higher order relations)
Finding the Right ML Algorithms

- May not find the “right” algorithm
- Should prefer available/working alg to “right” alg
- Workshop and tutorial proceedings to find esoteric algs
Use or Build?

- Build to learn more about how the algorithm works
- Use for faster speeds and more features
- Build if custom data formats are important
Know your Tools!

- **Parameters matter**
- Implementations are wrong or broken
- Implementations are “temperamental”
- Algorithms are “temperamental”

Pedregosa et al. 2011
My Problems

- Multinomial implementation broke with redundant features
- NB implementation didn’t smooth by default
- SVM needed lots and lots of expensive parameter tuning
Multi-class Classification

- Not all algorithms support $> 2$ classes
- Lots don’t handle \textit{constrained} MCC
- More than doubled implementation time

Budney 2013
Experiment Design

The Basics
- 10 sets
- 3 Feature Sets
- 3 Classifiers
- 100 Samples
- Over 9000 runs!

Additional Concerns
- Bootstrapping with training/test, not CV
- **Identical** data to allow paired tests
- Control of FDR
**Performance Measures**

**BC Error**

\[
\sum_{c \in C} \left| \text{count}(c) - G(c) \right| \quad \frac{1}{\sum_{c \in C} G(c)}
\]

**Bias**

\[
R^2 = \text{cor}(\text{count}(c) - G(c), G(c))^2
\]
Designing Performance Comparisons

- What is the system for?
- What’s the simplest technique?
- What’s the best technique?

![BC Error for Random Imputation (n=500)](image)
CPU Limited Experiments

- Write or use multi-threaded code
- Select implementation languages with this in mind
- Get some extra CPU power
- Reduce experiment size

```c
void trainModel(ProblemInstance pi, ulong length, immutable Candidate[ulong] alts){
    T[] results = new T[alts.length];
    foreach(ulong current_class : parallel(iota(1,alts.length+1))){
        Vector y_i;
        foreach(ulong j : pi.y.keys)
            y_i[j] = pi.y[j]==current_class?1.0:-1.0;
        results[current_class-1] = new T();
        results[current_class-1].train(pi.x, assumeUnique(y_i));
    }
    foreach(ulong i, T model; results){
        this.models[length][alts[i-1].dup] = model;
    }
}```
Processing Results

- R is great (e.g. pairwise.t.test)
- Tables often better than graphs
Results

- SVM + IG is the best combination of accuracy and fairness
- Multinomial & NB generally do not perform well
Writing Tips

- Start now: you have everything but the results
- Read some other papers for style/layout
- Get as much feedback as possible
Tips for Reviewing Papers

- Don’t prioritize spelling & grammar
- Think carefully about the paper’s methodology
- Does the paper do what it claims?
Plagiarism

- Assume about 20% plagiarism rate (yes, even here)
- Breaks in text, changes in voice, stylistic glitches
- Not necessarily malicious

Biliæ-Zulle et al. 2005
Conclusions

- Project wrapped up nicely, despite many hiccups
- Concrete question answered, definitively
- More questions generated, but the constructed testbed made them easy to answer too
**Future Work**

Lots of New Questions

- IG parameters never optimized
- Does this actually select good outcomes?
- Would a tailored method work better?
Meta: Where am I now?

Since then:
- Instant thesis topic
- Much improved mining system
- Custom ML models and better metrics

$$|\hat{\theta}_i - \theta_i| < \frac{t \sqrt{\hat{\theta}_i (1 - \hat{\theta}_i)}}{\sqrt{n_i} + \frac{|C|}{\sqrt{n_i}}} + \frac{\theta_i |C| + 1}{n_i + |C|}$$

$$n_i = n_{i-1} (\theta_i - \epsilon)$$
10 minute break / come talk to me.
We can:

- Hangout / do Q & A
- Talk about working with temporal data at edX
- Talk about evolutionary approaches, multi-objective learning, and embedded feature selection
- Talk about Compression classifiers, the stock market, and ML on unstructured data
EdX:
- Massive Online Classroom service
- Thousands of students per class
- Behavior tracked to the millisecond
- Demographics, behavior and grades are linked
The Data

- Held at MIT on airgapped machines
- Strict usage protocol
- Haphazard processing
- Stratified sampling
The Question

Research Questions

- Who is learning?
- How much do they learn?
- What makes them learn better?
Ethics Training

Training

- 12 hour US training
- 6 hour Canadian training
- IRB approval as additional researcher
Processing and Confounds

- Demographics explain many effects when present
- Many non-numeric fields
- Many students DNF

<table>
<thead>
<tr>
<th>Cohort</th>
<th>% of S.MReV</th>
<th>Mean</th>
<th>Standard deviation</th>
<th>Standard error</th>
<th>Relative improvement (SE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PhD</td>
<td>8</td>
<td>0.67</td>
<td>0.93</td>
<td>0.10</td>
<td>0.16 (0.06)</td>
</tr>
<tr>
<td>Masters</td>
<td>19</td>
<td>0.26</td>
<td>0.91</td>
<td>0.06</td>
<td>-0.06 (0.05)</td>
</tr>
<tr>
<td>College</td>
<td>29</td>
<td>-0.08</td>
<td>0.99</td>
<td>0.06</td>
<td>-0.11 (0.04)</td>
</tr>
<tr>
<td>High school</td>
<td>11</td>
<td>-0.20</td>
<td>0.93</td>
<td>0.07</td>
<td>-0.11 (0.06)</td>
</tr>
<tr>
<td>Less than HS</td>
<td>6</td>
<td>-0.05</td>
<td>0.84</td>
<td>0.10</td>
<td>-0.21 (0.10)</td>
</tr>
<tr>
<td>No response</td>
<td>23</td>
<td>0.02</td>
<td>1.04</td>
<td>0.07</td>
<td>0.01 (0.07)</td>
</tr>
<tr>
<td>Physics teachers</td>
<td>17</td>
<td>0.39</td>
<td>0.97</td>
<td>0.07</td>
<td>0.00 (0.05)</td>
</tr>
<tr>
<td>S.011 students</td>
<td>3</td>
<td>-1.05</td>
<td>0.50</td>
<td>0.08</td>
<td>-</td>
</tr>
</tbody>
</table>

Note. Degree listed is highest degree attained. For example, “High School” refers to students who have obtained a high school diploma and may be enrolled in college.

Colvin et al. 2014
First Attempts

Classification

- Predict over/under performance
- Aggregate time spent studying + demographics
- Only generally predictive features are pretest and slope

Colvin et al. 2014
Transition Model

Idea

- Not what you do, but how you do it
- Order of interaction could be predictive
- Learn as a Markov Chain

Joxemai and Falcorian 2014
Data Issues

- Time window or event counts?
- Window size and automatic-transition events
- Normalized or raw counts?
Analysis Issues

- Support levels for transitions
- Effect sizes
- Accounting for FDR
More Careful Analysis

Useful Results

- Build “typical” models for over-/under-performers
- Pooled t-tests for parameter differences
- Controlled with support & FDR

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
<th>Relative Risk (sample size)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[problem_check]</td>
<td>[problem_show]</td>
<td>1.682 (75)</td>
</tr>
<tr>
<td>[save_problem_check]</td>
<td>[goto_position]</td>
<td>1.300 (180)</td>
</tr>
<tr>
<td>[checkpoint_problem]</td>
<td>[goto_position]</td>
<td>0.979 (682)</td>
</tr>
<tr>
<td>[problem_get]</td>
<td>[course_material]</td>
<td>0.978 (679)</td>
</tr>
<tr>
<td>[course_material]</td>
<td>[problem_get]</td>
<td>0.966 (682)</td>
</tr>
<tr>
<td>[course_material]</td>
<td>[goto_position]</td>
<td>0.961 (682)</td>
</tr>
<tr>
<td>[checkpoint_problem]</td>
<td>[course_material]</td>
<td>0.961 (681)</td>
</tr>
<tr>
<td>[goto_position]</td>
<td>[problem_check]</td>
<td>0.949 (665)</td>
</tr>
<tr>
<td>[problem_get]</td>
<td>[about_course]</td>
<td>0.937 (603)</td>
</tr>
<tr>
<td>[course_material]</td>
<td>[posttest_problem_get]</td>
<td>0.932 (554)</td>
</tr>
<tr>
<td>[pretest_save_problem_check]</td>
<td>[pretest_save_problem_check]</td>
<td>0.913 (89)</td>
</tr>
</tbody>
</table>
**Example Results**

- Frequently looking at your grades ↓
- Doing many homework problems in a row ↑
- Frequent homework interruptions ↓
Embedded Feature Selection and Evolutionary Approaches
Traffic Analysis

- Very fast models
- Minimal, easily computed features
- Potentially frequent retraining
Genetic Programming

- ‘Evolve’ a program mapping inputs to outputs
- Pros: Programs can be complex and arbitrary mappings.
- Cons: Weak guarantees; ‘artful’ parameter selection; very large search spaces
Symbiotic Bid Based GP

- One modern alg
- Evolves very simple programs
- Simple programs as instruction set for complex programs
- Can layer indefinitely.
Data

- Reuters + FS Challenge Data
  - Old, but ‘real’ data
  - Many features, very sparse
  - Can we get a tiny and powerful model?

Table 1: Data set properties.

<table>
<thead>
<tr>
<th>Data set</th>
<th>Exemplar Count (train (test))</th>
<th>Feature Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Handwritten character recognition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multifeature</td>
<td>1,510 (490)</td>
<td>649</td>
</tr>
<tr>
<td>Gisette</td>
<td>6,000 (1,000)</td>
<td>5,000</td>
</tr>
<tr>
<td>Document Classification: Bag-of-words</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NIPS</td>
<td>7,000 (3,500)</td>
<td>12,419</td>
</tr>
<tr>
<td>Enron</td>
<td>7,000 (3,500)</td>
<td>28,102</td>
</tr>
<tr>
<td>NY Times</td>
<td>7,000 (3,500)</td>
<td>102,660</td>
</tr>
</tbody>
</table>
Comparisons

‘Out-of-the-Box’ Approaches

- Logres + Regularization
- F + SVM + Regularization
- Decision Tree + Confidence Factor

John A. Doucette (U. Waterloo)
Results

SBB vs. Simple models

- SBB solutions are reliably simpler and/or better
- Often we can find great solutions with fewer than 10 features.
Results

SBB vs. F+SVM

- Similar Results
- But what about processing times?
Results

SBB vs. F+SVM

- SBB: Program size (# ops)
- SVM: # ops to classify new point
- Exponential advantage for SBB.
Conclusions

- Great results, great models
- Technique now used in ML for security
- Careful though: GP hard to use, not the best tool for every domain.
My Get Rich Quick Scheme

Idea

- Annual financial statements contain information.
- Formats are non-uniform, hard to process
- Can we predict future stock performance using unprocessed reports?
Source Data

- All Form 10k’s for all DJIA companies, 1992-2011
- Manually collected (never again)
- Manually labeled (never again)
Compression as Classification

- Idea: Similar data compresses well together
- Build compression codes for all data from each class
- Classify by best compression ratio
- **No preprocessing required!**

Cormack and Horspool 1987
Difficulties

- Expensive training time, especially with later reports
- Many, many, many, problems with the data
- Original DMC code needed modifications
Classification of Stocks

- Beats the market in expectation (4% higher returns)
- Considerable variance
Results

Prediction of Bubbles?

- Do we really need data from the last n years?
- Possibly allows crash prediction?
Conclusions

- Neat idea, surprising that it works at all
- Many possible confounding issues, needs more work
- It’s finance, so it’s hard to say what would happen if you used this in practice.