Quantum Machine Learning for Election Modeling

September 28, 2017
Max Henderson, Ph.D.
Survey finds Hillary Clinton has ‘more than 99% chance’ of winning election over Donald Trump

The Princeton Election Consortium found Ms Clinton has a projected 312 electoral votes across the country and only 270 are needed to win

Rachael Revesz New York | @RachaelRevesz | Saturday 5 November 2016 16:44 GMT | 106 comments

Where did the models go wrong?

A model that has correctly predicted the winner of every U.S. presidential race since Ronald Reagan in 1980 is forecasting a big victory for Hillary Clinton.

Clinton is expected to get 332 electoral votes, while Trump is predicted to get just 206, according to the Moody’s Analytics model, which is based on three economic and three political factors.
State-by-state correlations

- Major issue: failure to model correlations\(^1-3\) between states
- Most models assumed independence between results of each state
- An accurate correlation matrix can capture higher-level, richer structure in the data and account for systemic errors in polls

First, there are errors of analysis. As an example, if you had a model of last year's election that concluded that Clinton had a 95 or 99 percent chance of winning, you committed an analytical error. \(^4\) Models that expressed that much confidence in her chances had a host of technical flaws, such as ignoring the correlations in outcomes between states. \(^5\)

### Similar states usually have similar outcomes

<table>
<thead>
<tr>
<th></th>
<th>Ala.</th>
<th>Calif.</th>
<th>Fl.</th>
<th>Minn.</th>
<th>N.C.</th>
<th>N.M.</th>
<th>R.I.</th>
<th>Wis.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alabama</td>
<td>1.0</td>
<td>.61</td>
<td>.53</td>
<td>.72</td>
<td>.54</td>
<td>.41</td>
<td>.55</td>
<td></td>
</tr>
<tr>
<td>California</td>
<td>.60</td>
<td>1.0</td>
<td>.73</td>
<td>.67</td>
<td>.69</td>
<td>.80</td>
<td>.61</td>
<td>.68</td>
</tr>
<tr>
<td>Florida</td>
<td>.61</td>
<td>.73</td>
<td>1.0</td>
<td>.67</td>
<td>.75</td>
<td>.70</td>
<td>.63</td>
<td>.76</td>
</tr>
<tr>
<td>Minnesota</td>
<td>.53</td>
<td>.67</td>
<td>.67</td>
<td>1.0</td>
<td>.68</td>
<td>.58</td>
<td>.64</td>
<td>.84</td>
</tr>
<tr>
<td>N. Carolina</td>
<td>.72</td>
<td>.69</td>
<td>.75</td>
<td>.68</td>
<td>1.0</td>
<td>.60</td>
<td>.53</td>
<td>.67</td>
</tr>
<tr>
<td>New Mexico</td>
<td>.54</td>
<td>.80</td>
<td>.70</td>
<td>.58</td>
<td>.60</td>
<td>1.0</td>
<td>.54</td>
<td>.64</td>
</tr>
<tr>
<td>Rhode Island</td>
<td>.41</td>
<td>.61</td>
<td>.84</td>
<td>.64</td>
<td>.53</td>
<td>.54</td>
<td>1.0</td>
<td>.69</td>
</tr>
<tr>
<td>Wisconsin</td>
<td>.55</td>
<td>.68</td>
<td>.76</td>
<td>.84</td>
<td>.67</td>
<td>.64</td>
<td>.69</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Difficulty of sampling from correlated graphs

- Even with perfect data on correlations between states, using the correlation matrix is difficult due to the computational cost of sampling from fully-connected graphs.
- Sampling from fully-connected graphs is analogous to sampling from a properly trained Boltzmann machine.
  - Training coefficients of Boltzmann machines requires performing calculations on all possible states of the model.
  - As this is intractable on large problem sizes, heuristics or other models are typically implemented instead.
Forecasting elections on a quantum computer

- Quantum computing (QC) research has shown potential speedups in training deep neural networks\(^1-3\)

- Core idea: By using QC-trained models to simulate election results we can achieve:
  - More efficient sampling / training
  - Intrinsic, tuneable state correlations
  - Inclusion of additional error models

Step 1: Mapping an election to a Boltzmann machine

Available data is limited

• What we would like:
  • Detailed breakdowns of demographics
  • Meticulously curated biases and correlations
  • All of the data that 538 has spent years and thousands of dollars curating

• What we have:
  • Publicly available results of previous US elections
  • State probabilities, as told by polls
  • Publicly accessible data from 538
Calculating the missing second order moments

- In lieu of better curated data concerning second order moments, we calculated our own terms from previous US election results
- Our methodology should not “break” first order moments

Assumptions in this model:
- In each previous election, if two states had the same election result, that increased their correlation
- Elections that were more recent have a higher weight
Step 2: Mapping a Boltzmann machine to the QC

The update equations for training the model:

\[
\Delta w_{ij} = -\frac{1}{\eta} \left( \langle s_i s_j \rangle_D - \langle s_i s_j \rangle_M \right)
\]

\[
\Delta \theta_i = -\frac{1}{\eta} \left( \langle s_i \rangle_D - \langle s_i \rangle_M \right)
\]

Potential quantum advantage
Graph embedding – Qubit chains

Example of embedding a problem (left) into a fixed graph structure (right)
Effect of embedding: Short qubit chains

- To validate the approach, we randomly chose first and second order terms for a hypothetical 5-state nation
- Using the smallest embedding chains, this network was unable to properly train
  - “Hopfield” like results; optimal solutions rather than probabilistic results
  - Leads to huge changes in weights/biases, causing network instability

Diagonal = \( \langle s_i \rangle_M \)
Off diagonal = \( \langle s_i s_j \rangle_M \)
Effect of embedding: Long qubit chains

- For larger problem sizes, the embedding will necessarily have longer qubit chains
- To simulate this for our small network, we artificially increased the qubit chains
- With this approach, arbitrary first and second order moments were learned by the networks

\[
\text{Diagonal} = \langle s_i \rangle_M \\
\text{Off diagonal} = \langle s_i s_j \rangle_M
\]
Primary experiment

• Goal: Using historical data and the QC-training methodology presented here, reproduce election forecasts over time

• Some caveats:
  • Multiple models needed for modeling national error; 25 were used here
  • Limited time windows of D-Wave access, so results were generated every two weeks instead of daily
  • Limited hardware size made us omit 1 state and province (sorry Maryland and DC... you always vote D anyway)
  • For simplification, Maine and Nebraska were considered winner-take-all
Results – Training errors

Examples testing extremes of correlations: negative, random, & positive

Red lines = $\langle S_i \rangle_D$
Blue lines = $\langle S_i \rangle_M$
Large errors emerge when polls are updated and large changes occur.
State errors

• 538 individual state errors used a t-distribution with 10 degrees of freedom (df)
• Probabilistic sampling from QC naturally led to state errors with similar distribution and parameters
Summary

• The QC-trained networks were able to learn structure in polling data to make election forecasts in line with the models of 538.

• Trump was given a higher likelihood of victory, even though the first order moments remained unchanged.
  • Ideally in the future, we could rerun this method using correlations known with more detail in-house from 538.

• Each iteration of the training model quickly produced 25,000 simulations (one for each national error model), which eclipses the 20,000 simulations 538 performs each time they rerun their models.