History Dependent Domain Adaptation


*Presenting; Rensselaer Polytechnic Institute
**Google Pittsburgh
Overview

● Standard methods are too myopic
  ○ Typical focus is on instantaneous performance
  ○ Many systems retrain while acting as a service

● Outline
  ○ Introduce problem
  ○ Our solutions
  ○ Experimental results
An extreme example

● Periodically re-train a classifier
● Mix of old and new examples at each step
● Small additional cost for repeating errors
  ○ Humans manually correct errors
● Two systems with high accuracy at any step
  ○ Distinct errors: approach 100% human effort
  ○ Consistent errors: very little human effort
Extreme example (continued)

Distinct errors

Consistent errors

Old Errors

New Errors

Fewer Incremental Errors

High cumulative error

Low cumulative error
General problem

- Loss depends on previous classifications
  - Low cost to repeat errors
- Incomplete feedback
  - Human corrections take time
  - Reviewing every classification can be too expensive
- Can we learn while minimizing new errors?
  - May not know which classifications are errors
Averaging

\[ h_{t+1}(x) = \langle \sum_{i=1}^{t+1} \beta_i w_i, x \rangle \]

- Exponentially weighted
  \[ w \leftarrow \alpha w_{t+1} + (1 - \alpha) w \]

- Average weights or model outputs
  - Equivalent in the linear case
  - Simple baseline
Warm-start

- Use a small step size, fewer steps
  - Reduce divergence from previous hypotheses
- More generally, online learning
Weight nearness constraint

\[ \| w_{t+1} - w_t \| \leq \delta \]

- Full optimization, with a hard constraint
  - Equivalent to regularizing around previous hypothesis
Prediction regularization

- Encourage model similarity on training data
- Squared
  \[ \sum_{x \in D_{t+1}} (\langle w_{t+1}, x \rangle - \langle w_t, x \rangle)^2 \]
- Hinge
  \[ \sum_{x} \max\{0, 1 - h_t(x)w_{t+1}^T x\} \]
  - Equivalent to adding more (weighted) examples
  \[ (x, h_t(x))_{x \in X} \]
  - Weight depends on coefficient of regularization term
Evaluating methods

● Area under the ROC curve (AUC)
  ○ Instantaneous performance
  ○ Avoid decreasing too much

● Cumulative Unique False Positives (CUFP)
  ○ Overall performance
  ○ Number of examples misclassified at least once

● Train on previous data, test on new data
Data

- **Adversarial advertisements (Sculley 2011)**
  - Adversarial (positive) or non-adversarial (negative)
  - Sparse, high-dimensional

- **Malicious URL Identification (Ma 2009)**
  - Malicious (positive) or non-malicious (negative)
  - Qualitatively similar, public
Adversarial advertisement results

40% CUFP reduction with a 0.4% AUC increase
Malicious URL results

50% CUFP reduction with a 0.05% AUC decrease
Summary & Future work

● History dependent adaptation problem
  ○ Loss depends on previous classifications
  ○ Many real-world applications

● Evaluated several solutions
  ○ Up to a 50% reduction in CUFP!
  ○ Maintain high AUC

● Lots still to do
  ○ Why do certain methods outperform others?
  ○ Can we do better?
  ○ Make use of unlabeled data
Questions?

Thank you!

Allen Lavoie: lavoia@cs.rpi.edu
Matthew Otey: otey@google.com
Nathan Ratliff: ratliffn@google.com
D. Sculley: dsculley@google.com