Learning Models of Users’ Informational Needs

Shawn Wolfe
CMPS 242
Aspects of Users’ Informational Need

• Typical models of a user’s need
  ▫ Includes simple model of content
  ▫ Little or nothing else

• What else might be a factor?
  ▫ Price
  ▫ Recency
  ▫ Readability
  ▫ Etc...

• How can we acquire more complex models? ML!
Prior work

- Previous study of Wolfe and Zhang
  - Domains
    - Airline Tickets
    - News Feed (information filtering)
  - Substantial improvement when all criteria used
- Follow-up study
  - Learn user model from past ratings (hold-one-out)
  - Best linear model: MSE 0.02689
  - Best nonlinear model: MSE 0.02390
- Can we do as well or better?
Data Set- YowNow

- From Yi Zhang’s dissertation 😊
  - ~ 9,000 instances
  - ~ 20 users
- Features (ratings)
  - Authority: 0 or 1
  - Novelty: 1, 2, 3, 4 or 5
  - Readability: 0 or 1
  - Relevance (to subject): 1, 2, 3, 4 or 5
  - Class value (user_like): 0, 0.25, 0.5, 0.75, 1
- Goal: learn class value from ratings
New Loss (Gain) Function

- **Common IR measure:** $F_\beta$
  - Harmonic mean of precision & recall
  - $F_\beta = \frac{(1 + \beta^2)P \cdot R}{\beta^2 P + R}$

- **Generalized to fractional relevance**
  - Prediction $\hat{Y}$, truth $Y$
  - $F_\beta = \frac{(1 + \beta^2) \cdot \|\min(\hat{Y}, Y)\|_1}{\beta^2 \cdot \|\hat{Y}\|_1 + \|Y\|_1}$

- **Issues**
  - Kinky!
  - Defined over entire set!
Regularization to Default Vector

• Use “reasonable” default $W_0$ to regularize
  ▫ Each criterion equally rated
  ▫ All min criteria $\rightarrow$ minimum target
  ▫ All max criteria $\rightarrow$ maximum target

• Linear Regression
  ▫ $W = (\lambda I + X^T X)^{-1}(\lambda W_0 + X^T Y)$

• Gradient Descent
  ▫ $W_t = W_{t-1} - (\eta/n)(\lambda \|W_{t-1} - W_0\|_1 + \text{Loss}(f(W_{t-1}X), Y)^T X)$
Nonlinear Representations

- **Partially nonlinear**
  - Pick two criteria, add 1 binary feature for each pair
  - **Example**
    - Authority (A) and Readability (R), each 0 or 1
    - New Features: F1 (A=0, R=0); F2 (A=0, R=1); F3 (A=1, R=0); F4 (A=1;R=1). Discard F1 and F4 as anchors.

- **Totally nonlinear**
  - Same process over all combinations
  - 2x5x2x5=100 new features (actually 96)
  - Discard original features
  - Also 5 binary class values
Square Loss Comparators

- **Machine Learning Demigod**
  - Makes optimal prediction based on criteria
  - Lower bound on square loss
- **Linear Regression on original representation**
  - Minimum loss on training set
  - Are nonlinear representations better?
- **Square loss from prior study**
  - Square loss from linear model: 0.02689
  - Square loss from best nonlinear model: 0.02390
Algorithms used

- Original Representation
  - Linear Regression (Squared Loss)
  - Gradient Descent (1-F_\beta Loss, Logistic Loss)
- Partially nonlinear
  - Linear Regression (Squared Loss)
- Fully nonlinear
  - Multiclass Logistic Regression (Accuracy)
  - Multiclass AdaBoost (Accuracy)
Procedure

- Repeat 10 times
  - Permute data
  - Reserve 20% data for testing, 80% training
  - Use 5-fold validation to search for optimal $\lambda$
  - Train model on entire training set
  - Use model to predict on testing set
- Report mean testing results
- Learn separate models for each user
Effect of Regularization on Linear Regression (Original Representation)
Effect of Regularization on Logistic Regression (Original Representation)
## Losses (Gain)

<table>
<thead>
<tr>
<th>Method</th>
<th>Squared</th>
<th>Logistic</th>
<th>1-F$_\beta$</th>
<th>(Accuracy)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DemiGod</td>
<td>0.02159</td>
<td>0.07016</td>
<td>0.07397</td>
<td>0.9186</td>
</tr>
<tr>
<td>Linear Reg.</td>
<td>0.02736</td>
<td>0.08681</td>
<td>0.09556</td>
<td>0.9087</td>
</tr>
<tr>
<td>GD Log. Reg.</td>
<td>0.02797</td>
<td>0.08849</td>
<td>0.09626</td>
<td>0.9036</td>
</tr>
<tr>
<td>GD 1-F$_\beta$</td>
<td>0.03731</td>
<td>0.1278</td>
<td>0.1106</td>
<td>0.8253</td>
</tr>
<tr>
<td>Combo 2+4</td>
<td><strong>0.02595</strong></td>
<td><strong>0.08322</strong></td>
<td><strong>0.09137</strong></td>
<td><strong>0.9073</strong></td>
</tr>
<tr>
<td>MC Log. Reg.</td>
<td></td>
<td></td>
<td></td>
<td>0.6219</td>
</tr>
<tr>
<td>MC AdaBoost</td>
<td></td>
<td></td>
<td></td>
<td>0.3941</td>
</tr>
<tr>
<td>Previous LR</td>
<td>0.02689</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prev. C 2+4</td>
<td>0.02390</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Conclusions

• Nonlinear representations
  ▫ Some nonlinearity can help
  ▫ Too much can hurt
• Gradient Descent
  ▫ Not finding global optimum
  ▫ Speed vs. analytical solution problematic
• Regularization- two edged sword
• $F_\beta$ as a gain (loss) function
  ▫ Appears to have too many local minima
  ▫ Work in progress