Recipe Rating Prediction

Summary

Objective:
Predict the star rating of a recipe given its ingredients

Methods:
Compare linear regression and kernel methods and compare different ingredient embeddings to predict rating

Results:
Best model achieves average mean square error of 0.172

Dataset

● Online recipes from Food.com of any cuisine
● Each recipe has a variable-length list of ingredients and an average star rating out of five from users
● 5000 examples in training dataset, 1000 examples in validation dataset, 1000 examples in test dataset

Features

One-Hot Encoding
● Let each unique ingredient map to one-hot vector
● Recipe ingredient list as sum of one-hot vectors
● Total embedded vector length 3866

BERT Embeddings
● Embedding from BERT, a pre-trained NLP model
● Each ingredient list is represented as a “sentence”, then embedded as a length 768 vector
● Capture similarities between, for example, “olive oil”, “extra virgin olive oil”, and “cooking oil”

Results

<table>
<thead>
<tr>
<th>Models</th>
<th>Training Loss</th>
<th>Validation Loss</th>
<th>Test Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Regression, One-Hot</td>
<td>0.1958</td>
<td>0.1893</td>
<td>0.1719</td>
</tr>
<tr>
<td>Linear Regression, BERT</td>
<td>0.2114</td>
<td>0.1998</td>
<td>0.1840</td>
</tr>
<tr>
<td>Square Kernel, One-Hot</td>
<td>0.1181</td>
<td>0.2028</td>
<td>0.1866</td>
</tr>
<tr>
<td>Gaussian Kernel, One-Hot</td>
<td>0.266</td>
<td>5.843</td>
<td>6.962</td>
</tr>
<tr>
<td>Square Kernel, BERT</td>
<td>0.2338</td>
<td>0.2262</td>
<td>0.2125</td>
</tr>
<tr>
<td>Gaussian Kernel, BERT</td>
<td>0.0002</td>
<td>0.3886</td>
<td>0.3989</td>
</tr>
</tbody>
</table>

Discussion

● Conducted hyperparameter sweep over step size and number of iterations, using validation set to choose best values
● Linear regression with one-hot embedding performs best
  ○ L1 distance between predictions and true values is 0.436
  ○ Within one half of a star (out of five) on average
● Gaussian kernel generalizes poorly to unseen data
  ○ One-hot embedding is extremely sparse, making the “distance” between recipes not very informative
  ○ Improved performance on BERT embeddings supports this
  ○ Expect better performance on larger dataset

Future Work

● Apply work to larger dataset
  ○ Total of 226,000 recipes available, but dataset cannot fit in memory
● Compare to other text embeddings
  ○ BERT assumes sentence structure
  ○ Other text embeddings may better capture ingredient relationships
● Explore pathological performance of Gaussian kernel
● Compare to other kernels

Loss function:
Mean squared error

\[ J(\theta) = \frac{1}{2} \sum_{i=1}^{n} (h(\theta; x^{(i)}) - y^{(i)})^2 \]

Linear Regression:
Gradient descent to learn \( \theta \) for

\[ h(x) = \theta^T x \]

Kernel Methods:
Gradient descent to learn \( \beta \) for Square and Gaussian kernels

\[ h(x) = \sum_{i=1}^{n} \beta_i K(x^{(i)}, x) \]

\[ K(x, z) = (x^T z)^2 \quad K(x, z) = \exp \left( -\frac{||x - z||^2}{2\sigma^2} \right) \]