Multilabel Classification and Deep Learning

Zachary Chase Lipton

Critical Review of RNNs:
http://arxiv.org/abs/1506.00019

Learning to Diagnose:
http://arxiv.org/abs/1511.03677

Conditional Generative RNNs:
http://arxiv.org/abs/1511.03683
Outline

• Introduction to Multilabel Learning

• Evaluation

• Efficient Learning & Sparse Models

• Deep Learning for Multilabel Classification

• Classifying Multilabel Time Series with RNNs
Supervised Learning

- General problem, desire a labeling function
  \[ f : \mathcal{X} \to \mathcal{Y} \]

- ERM principle - choose the model \( \hat{f} \) in hypothesis class \( \mathcal{H} \) that minimizes loss on the training sample \( S \in \{ \mathcal{X} \times \mathcal{Y} \}^n \)

- Most research assumes simplest case
  \( \mathcal{X} = \mathbb{R}^d, \mathcal{Y} = \{0, 1\} \)

- Real world much messier
Binary Classification

\[ y \in \{0, 1\} \]
Multiclass Classification

\[ y \in \{ c_1, c_2, \ldots, c_L \} \]
Multilabel Classification

$y \subseteq \{c_1, c_2, \ldots, c_L\}$
Why Multilabel?

• **Superset of both BC and MC:**
  BC when $|L| = 1$, MC when $y \in L$

• **Natural for many real problems:**
  Clinical diagnosis
  Predicting purchases
  Auto-tagging news articles
  Activity recognition
  Object detection

• **Easy to formulate:**
  Take L tasks and slap them together
Naive Baseline

• **Binary relevance:**
  Separately train $|L|$ classifiers $f_i : \mathcal{X} \rightarrow \{0, 1\}$

• **Pros:**
  Simple to execute, easy to understand
  strong baseline

• **Cons:**
  Computational cost: $|L| \times$
  Leaves some information on the table (correlation betw. labels)
Challenges

• **Efficiency**
  Develop classifiers that do not scale in time or space complexity with the number of labels

• **Performance**
  Make use of the extra labels to achieve better accuracy, generalization

• **Evaluation**
  How do we evaluate a multilabel classifier’s performance across 10s, 100s, 1000s, or even 1M labels?
Outline

• Introduction to Multilabel Learning

• Evaluation

• Efficient Learning & Sparse Models

• Deep Learning for Multilabel Classification

• Classifying Multilabel Time Series with RNNs
Why not accuracy?

- **Often extreme class imbalance**
  When blind classifier gets 99.99%, can be optimal to be uninformative

- **Varying base rates across labels**
  E.g.: MeSH dataset: Human applies to 71% of articles, platypus in <.0001%
F1 Score

• Easy to calculate from confusion matrix

<table>
<thead>
<tr>
<th>Predicted +</th>
<th>Actual +</th>
<th>Actual -</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>tp</td>
<td>fp</td>
</tr>
<tr>
<td>Predicted -</td>
<td>fn</td>
<td>tn</td>
</tr>
</tbody>
</table>

• Harmonic mean of precision $\frac{tp}{tp + fp}$ and recall $\frac{tp}{tp + fn}$

$$F1 = \frac{2 \cdot tp}{2 \cdot tp + fp + fn}$$
F1 given fixed base rate
Compared to Accuracy
Expected F1 for Uninformative Classifier
# Multilabel Variations

**Micro F1 calculated over all entries**

<table>
<thead>
<tr>
<th></th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>TN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example 1</td>
<td>TP</td>
<td>FP</td>
<td>FN</td>
<td>TN</td>
</tr>
<tr>
<td>Example 2</td>
<td>FP</td>
<td>FP</td>
<td>FN</td>
<td>TP</td>
</tr>
<tr>
<td>Example 3</td>
<td>FN</td>
<td>TP</td>
<td>FN</td>
<td>FP</td>
</tr>
<tr>
<td>...</td>
<td>TN</td>
<td>TP</td>
<td>TP</td>
<td>TN</td>
</tr>
</tbody>
</table>
Macro F1

- **Macro**: F1 calculated separately for each label and averaged

<table>
<thead>
<tr>
<th></th>
<th>Label 1</th>
<th>Label 2</th>
<th>Label 3</th>
<th>Label 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example 1</td>
<td>TP</td>
<td>FP</td>
<td>FN</td>
<td>TN</td>
</tr>
<tr>
<td>Example 2</td>
<td>FP</td>
<td>FP</td>
<td>FN</td>
<td>TP</td>
</tr>
<tr>
<td>Example 3</td>
<td>FN</td>
<td>TP</td>
<td>FN</td>
<td>FP</td>
</tr>
<tr>
<td>...</td>
<td>TN</td>
<td>TP</td>
<td>TP</td>
<td>TN</td>
</tr>
</tbody>
</table>
Characterizing the Optimal Threshold

• Threshold can be expressed in terms of the conditional probabilities of scores given labels

\[
\frac{b \cdot p(s|t = 1)}{(1 - b) \cdot p(s|t = 0)} \geq J
\]

• When scores are calibrated probabilities, optimal threshold is precisely half the F1 it achieves.

\[
s \geq \frac{tp}{2tp + fn + fp} = \frac{F}{2}
\]
Problems with F1

- Sensitive to thresholding strategy
- Hard to tell who has the best algorithms and who is smart about thresholding
- Micro-F1 biased towards common labels
- Macro-F1 biased against them
Some alternatives

• Any threshold indicates a cost sensitivity: When you know the cost, specify it and use weighted accuracy

• AUC exhibits same dynamic range for every label (blind classifier gets 0, perfect is 1)

• Macro-averaged AUC scores may give a better sense of performance across all labels

**high AUC for rare labels can be misleading. can achieve AUC of .99 produce useless results for IR**
Outline

• Introduction to Multilabel Learning
• Evaluation
  • Efficient Learning & Sparse Models
• Deep Learning for Multilabel Classification
• Classifying Multilabel Time Series with RNNs
The problem

• With many labels, binary relevance models can be huge and slow

• 10k labels + 1M features = 80GB of parameters

• We want compact models
  Fast to train and evaluate, cheap to store
Linear Regression

• The bulk of computation is label agnostic (compute inverse \((X^TX)^{-1}\))

\[
\theta = (X^TX)^{-1}X^Tb
\]

\[
\theta = (X^TX)^{-1}X^TB
\]

• Can do this especially fast when we reduce dimensionality of \(X\) via SVD.

• Problem: Unsupervised dim reduction -> lose signal of rare features -> mess up rare labels
Sparsity

- For auto-tagging tasks, features are often high-dimensional sparse bag-of-words or n-grams
- Datasets for web-scale information retrieval tasks are large in the number of examples, thus SGD is the default optimization procedure
- Absent regularization, the gradient is sparse and training is fast
- Regularization destroys the sparsity of the gradient
- Number of features and labels are large, dense stochastic updates are computationally infeasible
Regularization

• Goals: achieve model sparsity, prevent overfitting

• $\ell_1$ regularization is induces sparse models

• $\ell_2$ regularization is thought to achieve more accurate models in practice

• Elastic net, balances the two

$$F(w) = L(w) + \lambda_1 \cdot |w|_1 + \frac{1}{2} \lambda_2 \cdot |w|_2^2$$
Balancing Regularization with Efficiency

• To regularize while maintaining efficiency, can use a lazy updating scheme, first described by Carpenter (2008)

• For each feature, remember the last time it was nonzero

• When a feature is nonzero at some step t+k, perform a closed form update

• We derive lazy updates for elastic net regularization on both standard SGD and FoBoS (Duchi & Singer)
Lazy Updates for Elastic Net

**Theorem 1** To bring the weight $w_j$ current from time $\psi_j$ to time $k$ using SGD, the constant time update is

$$w_j^{(k)} = \text{sgn}(w_j^{(\psi_j)}) \left[ |w_j^{(\psi_j)}| \frac{P(k - 1)}{P(\psi_j - 1)} - P(k - 1) \cdot (B(k - 1) - B(\psi_j - 1)) \right]_+$$

where $P(t) = (1 - \eta(t)\lambda_2) \cdot P(t - 1)$ with base case $P(-1) = 1$ and $B(t) = \sum_{\tau=0}^{t} \eta^{(\tau)}/P(\tau - 1)$ with base case $B(-1) = 0$.

**Theorem 2** A constant-time lazy update for FoBoS with elastic net regularization and decreasing learning rate to bring a weight current at time $k$ from time $\psi_j$ is

$$w_j^{(k)} = \text{sgn}(w_j^{(\psi_j)}) \left[ |w_j^{(\psi_j)}| \frac{\Phi(k - 1)}{\Phi(\psi_j - 1)} - \Phi(k - 1) \cdot \lambda_1 (\beta(k - 1) - \beta(\psi_j - 1)) \right]_+$$

where $\Phi(t) = \Phi(t - 1) \cdot \frac{1}{1 + \eta(t)\lambda_2}$ with base case $\Phi(-1) = 1$ and $\beta(t) = \beta(t - 1) + \frac{\eta(t)}{\Phi(t-1)}$ with base case $\beta(-1) = 0$. 
Empirical Validation

- On two largest datasets in Mulan repository of multilabel datasets, we can train to convergence on a laptop in just minutes

- $rcv1$: 490x speedup, $bookmarks$: 20x speedup
Outline

• Introduction to Multilabel Learning
• Evaluation
• Efficient Learning & Sparse Models
• Deep Learning for Multilabel Classification
• Classifying Multilabel Time Series with RNNs
Performance

• Efficiency is nice, but we’d also like performance

• Neural networks can learn \textit{shared representations} across labels.

• Both regularizes each label’s model and exploits correlations between labels

• In extreme multilabel, may use significantly less parameters than logistic regression
Neural Network
Training w Backpropagation

- Goal: calculate the derivative of loss function with respect to each parameter (weight) in the model

- Update the weights by gradient following:

\[ w \leftarrow w - \eta \nabla_w L_i \]
Forward Pass
Backward Pass

The diagram illustrates the Backward Pass in a neural network, where the gradient of the loss function with respect to the output is calculated. The network has an input layer, a hidden layer, and an output layer. The gradient of the loss function is propagated backward through the network to update the weights. The symbols $\delta$ represent the error signal at each layer.
Multilabel MLP
Outline

• Introduction to Multilabel Learning
• Evaluation
• Efficient Learning & Sparse Models
• Deep Learning for Multilabel Classification

• Classifying Multilabel Time Series with RNNs
To Model Sequential Data: Recurrent Neural Networks
Recurrent Net (Unfolded)

\[ h^{(t)} = \sigma(W_{hx}x^{(t)} + W_{hh}h^{(t-1)} + b_h) \]

\[ \hat{y}^{(t)} = \text{softmax}(W_{yh}h^{(t)} + b_y) \]
LSTM Memory Cell
(Hochreiter & Schmidhuber, 1997)
LSTM Forward Pass

\[ g^{(t)} = \phi(W_{gx}x^{(t)} + W_{ih}h^{(t-1)} + b_g) \]
\[ i^{(t)} = \sigma(W_{ix}x^{(t)} + W_{ih}h^{(t-1)} + b_i) \]
\[ f^{(t)} = \sigma(W_{fx}x^{(t)} + W_{fh}h^{(t-1)} + b_f) \]
\[ o^{(t)} = \sigma(W_{ox}x^{(t)} + W_{oh}h^{(t-1)} + b_o) \]
\[ s^{(t)} = g^{(t)} \odot i^{(i)} + s^{(t-1)} \odot f^{(t)} \]
\[ h^{(t)} = s^{(t)} \odot o^{(t)} \]
LSTM (full network)
Unstructured Input

\[ x_i = \]
Modeling Problems

- **Examples:** 10,401 episodes
- **Features:** 13 time series (sensor data, lab tests)
- **Complications:** Irregular sampling, missing values, varying-length sequences
How to models sequences?

- Markov models
- Conditional Random Fields
- **Problem: Cannot model long range dependencies**
Simple Formulation

$\mathbf{x}_1 \rightarrow \mathbf{x}_2 \rightarrow \mathbf{x}_3 \rightarrow \mathbf{x}_4 \rightarrow \mathbf{x}_5 \rightarrow \mathbf{x}_6 \rightarrow \text{Targets}$
Target Replication
Auxiliary Targets
# Results

## Classification performance for 128 ICU phenotypes

<table>
<thead>
<tr>
<th>Model</th>
<th>Micro AUC</th>
<th>Macro AUC</th>
<th>Micro F1</th>
<th>Macro F1</th>
<th>Prec. at 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base Rate</td>
<td>0.7128</td>
<td>0.5</td>
<td>0.1346</td>
<td>0.0343</td>
<td>0.0788</td>
</tr>
<tr>
<td>Logistic Regression, First 6 + Last 6</td>
<td>0.8122</td>
<td>0.7404</td>
<td>0.2324</td>
<td>0.1081</td>
<td>0.1016</td>
</tr>
<tr>
<td>Logistic Regression, Expert features</td>
<td>0.8285</td>
<td>0.7644</td>
<td>0.2502</td>
<td>0.1373</td>
<td>0.1087</td>
</tr>
<tr>
<td>MLP, First 6 + Last 6</td>
<td>0.8375</td>
<td>0.7770</td>
<td>0.2698</td>
<td>0.1286</td>
<td>0.1096</td>
</tr>
<tr>
<td>MLP, Expert features</td>
<td><strong>0.8551</strong></td>
<td><strong>0.8030</strong></td>
<td><strong>0.2930</strong></td>
<td><strong>0.1475</strong></td>
<td><strong>0.1170</strong></td>
</tr>
</tbody>
</table>

### LSTM Models with two 64-cell hidden layers

<table>
<thead>
<tr>
<th>Model</th>
<th>Micro AUC</th>
<th>Macro AUC</th>
<th>Micro F1</th>
<th>Macro F1</th>
<th>Prec. at 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM</td>
<td>0.8241</td>
<td>0.7573</td>
<td>0.2450</td>
<td>0.1170</td>
<td>0.1047</td>
</tr>
<tr>
<td>LSTM, AuxOut (Diagnoses)</td>
<td>0.8351</td>
<td>0.7746</td>
<td>0.2627</td>
<td>0.1309</td>
<td>0.1110</td>
</tr>
<tr>
<td>LSTM-AO (Categories)</td>
<td>0.8382</td>
<td>0.7748</td>
<td>0.2651</td>
<td>0.1351</td>
<td>0.1099</td>
</tr>
<tr>
<td>LSTM-TR</td>
<td>0.8429</td>
<td>0.7870</td>
<td>0.2702</td>
<td>0.1348</td>
<td>0.1115</td>
</tr>
<tr>
<td>LSTM-TR-AO (Diagnoses)</td>
<td>0.8391</td>
<td>0.7866</td>
<td>0.2599</td>
<td>0.1317</td>
<td>0.1085</td>
</tr>
<tr>
<td>LSTM-TR-AO (Categories)</td>
<td>0.8439</td>
<td>0.7860</td>
<td>0.2774</td>
<td>0.1330</td>
<td>0.1138</td>
</tr>
</tbody>
</table>

### LSTM Models with Dropout (probability 0.5) and two 128-cell hidden layers

<table>
<thead>
<tr>
<th>Model</th>
<th>Micro AUC</th>
<th>Macro AUC</th>
<th>Micro F1</th>
<th>Macro F1</th>
<th>Prec. at 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM-DO</td>
<td>0.8377</td>
<td>0.7741</td>
<td>0.2748</td>
<td>0.1371</td>
<td>0.1110</td>
</tr>
<tr>
<td>LSTM-DO-AO (Diagnoses)</td>
<td>0.8365</td>
<td>0.7785</td>
<td>0.2581</td>
<td>0.1366</td>
<td>0.1104</td>
</tr>
<tr>
<td>LSTM-DO-AO (Categories)</td>
<td>0.8399</td>
<td>0.7783</td>
<td>0.2804</td>
<td>0.1361</td>
<td>0.1123</td>
</tr>
<tr>
<td>LSTM-DO-TR</td>
<td><strong>0.8560</strong></td>
<td><strong>0.8075</strong></td>
<td><strong>0.2938</strong></td>
<td><strong>0.1485</strong></td>
<td><strong>0.1172</strong></td>
</tr>
<tr>
<td>LSTM-DO-TR-AO (Diagnoses)</td>
<td>0.8470</td>
<td>0.7929</td>
<td>0.2735</td>
<td>0.1488</td>
<td>0.1149</td>
</tr>
<tr>
<td>LSTM-DO-TR-AO (Categories)</td>
<td>0.8543</td>
<td>0.8015</td>
<td>0.2887</td>
<td>0.1446</td>
<td>0.1116</td>
</tr>
<tr>
<td>LSTM-DO-TR (Linear Gain)</td>
<td>0.8480</td>
<td>0.7986</td>
<td>0.2896</td>
<td><strong>0.1530</strong></td>
<td>0.1160</td>
</tr>
</tbody>
</table>

### Ensembles of Best MLP and Best LSTM

<table>
<thead>
<tr>
<th>Model</th>
<th>Micro AUC</th>
<th>Macro AUC</th>
<th>Micro F1</th>
<th>Macro F1</th>
<th>Prec. at 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean of LSTM-DO-TR &amp; MLP</td>
<td>0.8611</td>
<td>0.8143</td>
<td>0.2981</td>
<td>0.1553</td>
<td>0.1201</td>
</tr>
<tr>
<td>Max of LSTM-DO-TR &amp; MLP</td>
<td><strong>0.8643</strong></td>
<td><strong>0.8194</strong></td>
<td><strong>0.3035</strong></td>
<td><strong>0.1571</strong></td>
<td><strong>0.1218</strong></td>
</tr>
</tbody>
</table>
Outline

• Introduction to Multilabel Learning
• Evaluation
• Efficient Learning & Sparse Models
• Deep Learning for Multilabel Classification
• Jointly Learning to Generate and Classify Beer Reviews
RNN Language Model
Past Supervised Approaches relied upon Encoder-Decoder Model
Bridging Long Time Intervals with Concatenated Inputs
A.5 FRUIT/VEGETABLE BEER
<STR>On tap at the brewpub. A nice dark red color with a nice head that left a lot of lace on the glass. Aroma is of raspberries and chocolate. Not much depth to speak of despite consisting of raspberries. The bourbon is pretty subtle as well. I really don’t know that I find a flavor this beer tastes like. I would prefer a little more carbonization to come through. It’s pretty drinkable, but I wouldn’t mind if this beer was available. <EOS>
Character-based Classification

![Graph showing the likelihood of different types of beer](image-url)
“Love the Strong Hoppy Flavor”
Thanks!

Contact:
zlipton@cs.ucsd.edu
zacklipton.com

Critical Review of RNNs:
http://arxiv.org/abs/1506.00019

Learning to Diagnose:
http://arxiv.org/abs/1511.03677

Conditional Generative RNNS:
http://arxiv.org/abs/1511.03683