Writer Adaptation using **Bottleneck Features** and **Discriminative Linear Regression** for Online Handwritten Chinese Character Recognition

Jun Du, Jin-Shui Hu, Bo Zhu, Si Wei, Li-Rong Dai
University of Science and Technology of China
iFlyteck Research

ICFHR 2014, Crete, Greece, September 1-4, 2014
Background

• Chinese handwriting recognition is popular
  – Especially on portable devices in mobile internet era

• User experience largely depends on the writing style
  – Mismatch even with more and more diversified training data

• Solution: writer adaptation
  – Can really improve the user experience for a specific writer
  – Supervised mode with automatically labeled data by users
Related Work for Writer Adaptation

• For handwriting recognition of western languages
  – Adaptable output layer of a time delay neural network (1993)
  – Adding a radial basis function to neural networks (1997)
  – MLLR and MAP for HMM based system (2001)
  – Biased regularization for SVM (2006)

• For Chinese handwriting recognition
  – STM: Style Transfer Mapping (2011)
Core Innovations

• Bottleneck features (BNF) for feature extraction
  – A highly nonlinear and discriminative transformation
  – Superior to linear transformation based on LDA

• Discriminative linear regression (DLR) for writer adaptation

• Incorporate BNF and DLR with prototype-based classifier
  – Significantly outperform STM
Multi-prototype based Classifier

• Classification with discriminant functions

\[
\begin{align*}
  r(x; \Lambda) &= \arg \max_i g_i(x; \lambda_i) \\
  g_i(x; \lambda_i) &= -\min_k \| x - m_{ik} \|^2
\end{align*}
\]

• Minimum Classification Error (MCE) criterion

\[
l(X; \Lambda) = \frac{1}{R} \sum_{r=1}^{R} \frac{1}{1 + \exp[-\alpha d(x_r; \Lambda) + \beta]}
\]

• Misclassification measure
  – Sample Separation Margin (SSM)

\[
d(x_r; \Lambda) = \frac{-g_p(x_r; \lambda_p) + g_q(x_r; \lambda_q)}{2 \| m_{p\hat{k}} - m_{q\hat{k}} \|}
\]
Bottleneck feature extractor

- Extracting from a bottleneck layer of DNN
  - DNN input: LDA transformed feature vector
  - DNN output: the posterior probability of character classes

- Hinton’s training recipe
  - Layer-by-layer RBM pre-training
  - Cross-entropy fine-tuning
Writer Adaptation via Linear Regression

- Feature transformation
  \[ x_r = \mathcal{F}(y_r; \Theta) = Ay_r + b \]

- Style transfer mapping
  \[
  \min_A \sum_{r=1}^{R'} f_r \| As_r - t_r \|_2^2 + \beta_1 \| A - I \|_2^2
  \]

- Discriminative linear regression (SSM-MCE)
  \[
  l(Y; \Lambda, \Theta) = \frac{1}{R'} \sum_{r=1}^{R'} \frac{1}{1 + \exp[-\alpha d(y_r; \Lambda, \Theta) + \beta]}
  \]
  \[
  d(y_r; \Lambda, \Theta) = \frac{-g_p(x_r; \lambda_p) + g_q(x_r; \lambda_q)}{2 \| m_{pk} - m_{qk} \|}
  \]
Experimental Setup

• Database
  – Training: 15167 character classes, totally 14846606 samples
  – Data from 105 real users written in several months
    • 5000-30000 character samples for each user
    • Random half for adaptation and testing

• Feature extraction
  – 392-dimensional raw feature: 8-directional features
  – LDA transformation: 392 -> 96

• DNN architecture for BNF: 96-1024-1024-1024-96-15167
No Adaptation: BNF vs. LDA

- BNF significantly outperforms LDA with LBG initialization
- The gap between BNF and LDA is smaller after SSM-MCE

Table 1. Performance (character error rate in %) comparison of systems using prototype-based classifiers with different features and different training criteria on the testing set of all 105 writers.

<table>
<thead>
<tr>
<th></th>
<th>#prototype</th>
<th>LBG</th>
<th>SSM-MCE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LDA</strong></td>
<td>1</td>
<td>33.97</td>
<td>22.16</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>30.63</td>
<td>20.20</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>27.08</td>
<td>19.14</td>
</tr>
<tr>
<td><strong>BNF</strong></td>
<td>1</td>
<td>26.06</td>
<td>19.66</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>23.56</td>
<td>19.12</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>22.01</td>
<td>18.79</td>
</tr>
</tbody>
</table>
Writer Adaptation using Different Approaches

• Both BNF and DLR bring significant improvements
• BNF and DLR are complementary (40% ERR over LDA+STM)
• More adaptation data is useful for DLR rather than STM

Table 2. Performance (character error rate in %) comparison of systems using different adaptation strategies averaged across each testing set of all 105 writers.

<table>
<thead>
<tr>
<th></th>
<th>LDA</th>
<th>BNF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>STM</td>
<td>DLR</td>
</tr>
<tr>
<td>Baseline</td>
<td>19.14</td>
<td>18.79</td>
</tr>
<tr>
<td>WA(1000)</td>
<td>13.49</td>
<td>11.96</td>
</tr>
<tr>
<td>WA(3000)</td>
<td>13.29</td>
<td>10.51</td>
</tr>
<tr>
<td>WA(5000)</td>
<td>13.24</td>
<td>10.11</td>
</tr>
</tbody>
</table>
Comparison for 25 selected writers

- In most cases, BNF+DLR achieves the best performance
Summary and Future Work

• BNF+DLR achieves promising results
  – Writer adaptation is easier in highly nonlinear feature space
  – Discriminatively trained linear regression is more powerful

• Future work
  – Unsupervised, semi-supervised adaptation
  – Extend the linear regression to nonlinear for writer adaptation
  – Writer adaptation on deep learning based classifiers