1. Multi-label Classification

Assign a subset of candidate labels to an object (image, document, video).

2. Existing Approaches

- Binary Relevance: predict each binary label independently
- Power-Set: treat each subset as a class + multi-class
- CRF: specify label dependencies with graphical models
- PCC: predict next label based on previous labels

3. Proposed Model: Conditional Bernoulli Mixtures

Approximate the conditional joint by a Conditional Bernoulli Mixture (CBM) with fully factorized mixture components. \( y = \text{binary label vector of length } L \).

\[
\text{CBM: } p(y|x) = \sum_{z} p(z=k|x; \alpha) \prod_{b=1}^{L} b(y_b|z_k; \beta_k)
\]

- \( p(z=k|x; \alpha) \): probability of belonging to component \( k \) instantiated with a multi-class classifier; e.g., multinomial LR
- \( b(y_b|z_k; \beta_k) \): probability of getting label \( y \) in component \( k \) instantiated with a binary classifier; e.g., binary LR

\( \alpha \) automatically capture label dependencies; \( \beta \) a flexible reduction method: multi-label subsume Binary Relevance and Power-Set as special cases

4. Capturing Label Dependencies: Illustration

Top 4 most influential CBM components for the example image

5. Simple Training with EM

Given training dataset \( \{(x_n, y_n)\} \), use EM to minimize an upper bound of negative log likelihood:

\[
\sum_{n=1}^{N} \text{KL}(f(z_n)||p(z_n|x; \alpha)) + \sum_{k=1}^{K} \sum_{l=1}^{L} \sum_{n=1}^{N} \gamma_{k,l} \text{KL}(\text{Ber}(y_{n,l}|y_{n,k}; \beta_k))
\]

\( f(z_n) \) is the posterior membership distribution \( p(z_n|x; \alpha) \).

\( \text{Ber}(y_{n,l}|y_{n,k}; \beta_k) \) is the Bernoulli distribution with head probability \( y_{n,k} \).

\( \gamma_{k,l} \) is the posterior probability of \( z_{n,l} = k \) for \( y_{n,k} \).

**E step:** Re-estimate posterior membership probabilities:

\[
\gamma_{k,l} = \frac{\pi(z_{n,l} = k|x; \alpha)}{\sum_{k=1}^{K} \pi(z_{n,l} = k|x; \alpha)} \prod_{b=1}^{L} b(y_{n,b}|y_{n,k}; \beta_k)
\]

**M step:** Update model parameters. Standard multi-class and binary classifier learning:

\[
\alpha_{\text{new}} = \arg \min_{\alpha} \sum_{n=1}^{N} \text{KL}(f(z_n)||p(z_n|x; \alpha)) \quad \text{multi-class classification}
\]

\[
\beta_{\text{new}}^{y_{n,l}} = \arg \min_{\beta} \sum_{n=1}^{N} \text{KL}(\text{Ber}(y_{n,l}|y_{n,k}; \beta_k)) \quad \text{binary classification}
\]

Two concrete instantiations:
- with logistic regression (LR) learners: EM + gradient descent/LBFGS
- with gradient boosted trees (GB) learners: EM + gradient boosting

6. Fast Prediction by Dynamic Programming

Two common difficulties in prediction:

\( \hat{y}(y) \) how to find argmax \( p(y|x) \) without enumerating \( 2^L \) possibilities of \( y \)?

\( \hat{y}(y) \) how to predict unseen subsets \( y \)?

Find the exact argmax \( p(y|x) \) efficiently by dynamic programming:
- to get a high overall probability, at least one component probability must be high
- in each component, list label subsets in a decreasing probability order with DP
- iterate round-robin across components and prune remaining suboptimal subsets

7. Results

We use the most stringent evaluation measure: subset accuracy = \( \frac{1}{N} \sum_{n=1}^{N} 1[y_n = y] \).

A predicted subset is considered correct only when it matches the true subset exactly.

Test subset accuracy of different methods on five datasets. All numbers are in percentages.

<table>
<thead>
<tr>
<th>Method Learner</th>
<th>SCENE image</th>
<th>RCV1 text</th>
<th>TMCC2007 text</th>
<th>MEDIALL image</th>
<th>NUS-WIDE image</th>
</tr>
</thead>
<tbody>
<tr>
<td>BinRel LR</td>
<td>51.5</td>
<td>40.4</td>
<td>25.3</td>
<td>9.6</td>
<td>24.7</td>
</tr>
<tr>
<td>PowSet LR</td>
<td>68.1</td>
<td>50.2</td>
<td>28.2</td>
<td>9.0</td>
<td>26.6</td>
</tr>
<tr>
<td>CC LR</td>
<td>62.9</td>
<td>48.2</td>
<td>26.2</td>
<td>10.9</td>
<td>26.0</td>
</tr>
<tr>
<td>PCC LR</td>
<td>64.8</td>
<td>48.3</td>
<td>26.8</td>
<td>10.9</td>
<td>26.3</td>
</tr>
<tr>
<td>ECC-label LR</td>
<td>60.6</td>
<td>46.5</td>
<td>26.0</td>
<td>11.3</td>
<td>26.0</td>
</tr>
<tr>
<td>ECC-subset LR</td>
<td>63.1</td>
<td>49.2</td>
<td>25.9</td>
<td>11.5</td>
<td>26.0</td>
</tr>
<tr>
<td>CDN LR</td>
<td>59.9</td>
<td>12.6</td>
<td>16.8</td>
<td>5.4</td>
<td>17.1</td>
</tr>
<tr>
<td>pairCRF linear</td>
<td>68.8</td>
<td>46.4</td>
<td>28.1</td>
<td>10.3</td>
<td>26.4</td>
</tr>
<tr>
<td>CBM GB</td>
<td>69.7</td>
<td>49.9</td>
<td>28.7</td>
<td>13.5</td>
<td>27.3</td>
</tr>
<tr>
<td>BinRel GB</td>
<td>59.3</td>
<td>30.1</td>
<td>25.4</td>
<td>11.2</td>
<td>24.4</td>
</tr>
<tr>
<td>PowSet GB</td>
<td>70.5</td>
<td>38.2</td>
<td>23.1</td>
<td>10.1</td>
<td>23.6</td>
</tr>
<tr>
<td>CBM GB</td>
<td>70.5</td>
<td>43.0</td>
<td>27.5</td>
<td>14.1</td>
<td>26.5</td>
</tr>
</tbody>
</table>

8. Analysis

Test subset accuracy on TMC dataset with varying number of components \( K \) for CBM+LR

- \( K = 1 \), CBM only estimates marginals and performs similarly to Binary Relevance
- \( K > 1 \), CBM becomes a better joint estimator and achieves better subset accuracy
- \( K = 30 \), performance asymptotes

Our code is available at https://github.com/cheng-li/pyramid