Problem and Motivation
Multi-label classification: predict multiple outputs, e.g., identify multiple objects in an image. Often desirable to model count structure. E.g., in order to identify which objects there are, it is helpful to know how many there are.

- Simple idea: multiple independent logistic regression (LR) classifiers.
- Problem: LR uses the same parameters to both identify and count objects.
- Even though there are always 1 to 4 objects in each image, logistic regression may predict 0 objects, 5 objects, 6 objects, etc., this limits its modeling ability.

The Binary n-Choose-k Model (BnCk)
Setting: learn to predict multiple binary outputs.

<table>
<thead>
<tr>
<th>Features</th>
<th>Parameters</th>
<th>Model Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
<td>$W$</td>
<td>$\theta = Wx$</td>
<td>$y \in {1, 2, \ldots, R}$</td>
</tr>
</tbody>
</table>

- Define a subset of the variables by $c \subseteq \{1, 2, \ldots, D\}$ and complement $\bar{c} = \{1, 2, \ldots, D\} \setminus c$.
- Draw $k$ from a prior distribution $p(k)$ over counts $k$.
- Draw $k$ variables to take on label 1, where the probability of choosing subset $c$ is given by

$$p(y_c = 1, y_{\bar{c}} = 0 | k) = \frac{\exp(\sum_{r \in c} \theta_r)}{\exp(\sum_{r \in \bar{c}} \theta_r)}$$

- Training objective is convex, only tuning parameter is L2 penalty strength.
- At test-time, the quality of a ranking is evaluated using a gain function, e.g., NDCG, Precision@K. Finding the optimal ranking over BnCk is trivial for these measures.

**Theorem 1.** Under an ordinal n-choose-k model, the optimal decision theoretic predictions for monotonic ranking gains, such as NDCG and Precision@K, are made by sorting the $\theta$ scores.

Connection to Logistic Regression
- Multiple output logistic regression can be viewed as a binary n-choose-k model.
- Let $Z_k(\theta) = \exp(\sum_{r \in c} \theta_r)$, $Z(\theta) = \sum_{k} Z_k(\theta)$, and assume $\sum_{y_k = k} = k$.

$$p(y, k; \theta) = p(y | k; \theta)p(y | k; \theta) = \frac{Z_k(\theta)}{Z(\theta)}(\begin{pmatrix} \exp(\sum_{r \in c} \theta_r) \\ \exp(\sum_{r \in \bar{c}} \theta_r) \end{pmatrix}) = \prod_{r \in c} \frac{\exp(\theta_r)}{1 + \exp(\theta_r)}$$

- Logistic regression implicitly models counts using the “prior” $p(k; \theta) = \frac{Z_k(\theta)}{Z(\theta)}$.
- Induces independence between output variables.
- The prior distribution is called a Poisson-Binomial distribution (Chen et al., 1994).
- Distribution over the number of successes in independent Bernoulli trials with different probabilities.

Efficient Likelihood Computation
- The PnCk count-conditional likelihood involves summing over all subsets of size $k$.
- Can be viewed as a Markov Random Field with unary potentials and a global cardinality potential.
- Can efficiently compute the count conditional likelihood in $O(D \log^2 D)$.

Belfiore, 1995, Tarlow et al., 2012

Experiments
Modeling the number of objects in an image.
- We train a BnCk model with an input-dependent count prior $p(k|x)$.
- Separates which digits appear in an image from how many.
- LR test set log-likelihood: -2.84, BnCk test set log-likelihood: -1.95.

Ranking with weak labels on the LETOR 3.0 datasets.
- Each input is a query with multiple documents and associated relevance scores.
- Output is a ranking of the documents within each query.

Top-k Classification
- The inputs are an image and a single ground-truth label, the outputs are the top $k$ predictions of the model.
- We train to maximize the expected accuracy under a top-$k$ evaluation criterion.
- Overfitting can be an issue, but training and testing with top-k is promising.

<table>
<thead>
<tr>
<th>Training objective</th>
<th>Evaluation Criterion</th>
</tr>
</thead>
<tbody>
<tr>
<td>LR</td>
<td>0.069 / 0.750 / 0.612</td>
</tr>
<tr>
<td>NDK</td>
<td>0.574 / 0.736 / 0.679</td>
</tr>
<tr>
<td>Top 1</td>
<td>0.854 / 0.834 / 0.834</td>
</tr>
<tr>
<td>Top 5</td>
<td>0.875 / 0.852 / 0.860</td>
</tr>
<tr>
<td>Top 10</td>
<td>0.892 / 0.871 / 0.891</td>
</tr>
<tr>
<td>NDCG Truncation Level (K)</td>
<td>0.523 / 0.767 / 0.813</td>
</tr>
</tbody>
</table>

Top 1 is equivalent to softmax regression