Theory of Evidence in Active Learning

Daniel Kałuża
Active Learning

Goal: Obtain the best possible model with limited labelling capabilities, assuming possibility of experts-model interaction.
Active Learning cycle

Source: A. Janusz, Ł. Grad, M. Grzegorowski, “Clash Royale Challenge: How to Select Training Decks for Win-rate Prediction”
Active Learning - approaches

Usually based on:

- Informativeness (E. g. Max Entropy, Prediction Margin)
- Representativeness (E. g. Clustering based, Distance based)
- Dissimilarity (E. g. Distance to the current batch)
Theory of Evidence - Basics

A different view on probability, distinguishing:

- subjective beliefs
  from
- objective chances

Focuses on sets of random events instead of single events.
Theory of Evidence - Rules

Let $\theta$ be a finite set of possible states. Then if function $\text{Bel}: 2^\theta \rightarrow [0, 1]$ satisfies conditions:

1. $\text{Bel}(\emptyset) = 0$
2. $\text{Bel}(\theta) = 1$
3. For every positive $n$ and every collection of subsets $A_1, A_2, \ldots, A_n$ of $\theta$:

$$\text{Bel}(A_1 \cup A_2 \cup \ldots \cup A_n) \geq \sum_{i=1}^{n} \text{Bel}(A_i) - \sum_{j=i+1}^{n} \text{Bel}(A_i \cap A_j)$$

$$+ \ldots + (-1)^{n+1} \text{Bel}(A_1 \cap A_2 \cap \ldots \cap A_n)$$

Then $\text{Bel}$ is called a belief function over $\theta$. 
Theory of Evidence - Example

Let $\theta = \{\theta_1, \theta_2\}$

$\theta_1$ - genuine

$\theta_2$ - counterfeit

$\text{Bel}(\theta_1) = a$

$\text{Bel}(\theta_2) = b$

$\text{Bel}(\emptyset) = 0$

$\text{Bel}(\theta) = 1$
Theory of Evidence - Uncertainty Intuition

Let's consider the following random events:
A - the dice number will be even
B - the dice number will be odd
C1 - the dice number will be 1
C3 - the dice number will be 3
C5 - the dice number will be 5

Bayesian uncertainty:
- \( P(A) = \frac{1}{2}, P(B) = \frac{1}{2} \)

What about:
- \( P(A) = \frac{1}{2}, P(C1) = \frac{1}{6}, P(C3) = \frac{1}{6}, P(C5) = \frac{1}{6} \)

In Theory of Evidence we can say:
\( \text{Bel}(\{X\}) = 0, \text{ for } X \text{ in } \{A, C1, C3, C5\} \)
\( \text{Bel}(\{A, C1, C3, C5\}) = 1 \)
Example of application - Neural Networks

Regular neural network classifier:

- softmax as an output
- output interpreted as probability distribution
- uncertainty measured on output, e.g. entropy
- optimized with cross-entropy and gradient based methods
Softmax - inflating the probabilities

\[ \sigma(x)_j = \frac{e^{x_j}}{\sum_k e^{x_k}} \]
Softmax - inflating the probabilities

Input pixels, $x$

Feedforward output, $y_i$

<table>
<thead>
<tr>
<th></th>
<th>cat</th>
<th>dog</th>
<th>horse</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>4</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Softmax output, $S(y_i)$

<table>
<thead>
<tr>
<th></th>
<th>cat</th>
<th>dog</th>
<th>horse</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.71</td>
<td>0.26</td>
<td>0.04</td>
<td></td>
</tr>
<tr>
<td>0.02</td>
<td>0.00</td>
<td>0.98</td>
<td></td>
</tr>
<tr>
<td>0.49</td>
<td>0.49</td>
<td>0.02</td>
<td></td>
</tr>
</tbody>
</table>

Shape: (3, 32, 32)  Shape: (3,)  Shape: (3,)

Source: https://ogunlao.github.io/images/softmax.png
Cross-entropy loss ~ Maximum Likelihood Estimation

MLE as a frequentist method, therefore it isn’t capable to describe the distribution variance!
MNIST example

Source: Sensoy et al. “Evidential Deep Learning to Quantify Classification Uncertainty”
Draft of idea - replace softmax with Dirichlet Distribution

Input → Forward propagation → $\alpha_1, \alpha_2, \ldots, \alpha_K$ → Transformation to beliefs & uncertainty → $b_1, b_2, \ldots, b_K, u$
Modeling DST with Subjective Logic & Dirichlet Distribution

\[ u + \sum_{k=1}^{K} b_k = 1, \quad u = \frac{K}{S}, \quad b_k = \frac{e_k}{S}, \quad S = \sum_{i=1}^{K} (e_i + 1) \]

- \( b_k \) - belief of mass corresponding to k-th singleton class
- \( u \) - uncertainty
- \( e_i \) - evidence for the i-th singleton class
- \( K \) - number of classes
Dirichlet Distribution

\[
D(p|\alpha) = \begin{cases} 
\frac{1}{B(\alpha)} \prod_{i=1}^{K} p_i^{\alpha_i - 1} & \text{for } p \in S_K, \\
0 & \text{otherwise,}
\end{cases}
\]

\[\alpha_k = e_k + 1 \quad b_k = \frac{e_k}{S}\]

- \(b_k\) - belief of mass corresponding to \(k\)-th singleton class
- \(u\) - uncertainty
- \(e_i\) - evidence for the \(i\)-th singleton class
- \(K\) - number of classes
- \(\alpha_k\) - parameter of Dirichlet distribution corresponding to \(k\)-th class
Loss & Training

\[
L_t(\Theta) = \sum_{j=1}^{K} \left( y_{ij} - \mathbb{E}[p_{ij}] \right)^2 + \text{Var}(p_{ij}) + \lambda_t \sum_{i=1}^{N} KL[D(p_i|\tilde{\alpha}_i) \parallel D(p_i|\langle 1, \ldots, 1 \rangle)],
\]

where \( t \) is an index of learning epoch

\[\lambda_t = \min(1.0, t/10) \in [0, 1]\]

- **KL** - Kullback-Leibler divergence

\[\tilde{\alpha}_i = y_i + (1 - y_i) \odot \alpha_i\]

- **K** - number of classes

- **\( \alpha_k \)** - parameter of Dirichlet distribution corresponding to k-th class
Results

Figure 1: Classification of the rotated digit 1 (at bottom) at different angles between 0 and 180 degrees. **Left:** The classification probability is calculated using the *softmax* function. **Right:** The classification probability and uncertainty are calculated using the proposed method.
Figure 5: Accuracy and entropy as a function of the adversarial perturbation $\epsilon$ on CIFAR5 dataset.
Results Active Learning CNN
Conclusions and thoughts

- interesting usage of Dirichlet distribution
- why authors are not using uncertainty for AL?
- can the same be done for other softmaxed methods? e.g. xgboost
- maybe there is a better way to incorporate DS theory to machine learning models?
Bibliography

Thank you for attention