Classification: From Expert to Ensemble to Crowd

Lorenza Saitta
saitta@mfn.unipmn.it
Classification

• Classification (only indirectly learning)
• Data independent setting
• Framework where alternative classification strategies can be compared

• PROBLEMS
  – Error estimation
  – Class assignment
  – Optimization of the classification process
Parameters of the Classification Process

• Number of classifiers
  - Single
  - Ensemble (Bagging-like)
  - Crowdsourcing

• Number of classes
  - Two classes (binary)
  - More than two classes (multi-class)

• Aggregation function
  - Majority voting
  - Weighted Majority voting
  - Plurality voting
**Problem Definition**

- Set $\Phi = \{\varphi_j | 1 \leq \varphi \leq R\}$ of classifiers
- Finite set $\mathcal{X}$ of examples to classify: $|\mathcal{X}| = N$
- Examples may be duplicated – only $n$ are different

Empirical frequency $f_k$ of example $x_k$

$$f_k = \frac{n_k}{N} \quad \sum_{k=1}^{n} n_k = N \quad \sum_{k=1}^{n} f_k = 1$$

Decision Matrix

<table>
<thead>
<tr>
<th></th>
<th>$\varphi_1$</th>
<th>$\varphi_j$</th>
<th>$\varphi_k$</th>
<th>$p_k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>$p_1$</td>
</tr>
<tr>
<td>$x_k$</td>
<td>...</td>
<td>$p_j(x_k)$</td>
<td>...</td>
<td>$p_k$</td>
</tr>
<tr>
<td>$x_n$</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>$p_n$</td>
</tr>
<tr>
<td>$r_j$</td>
<td>$r_1$</td>
<td>$r_j$</td>
<td>...</td>
<td>$r$</td>
</tr>
</tbody>
</table>

$p_j(x_k) \in \{0,1\}$

Probability that $\varphi_j$ correctly classifies $x_k$

$r_j = \text{Probability that a given } \varphi_j \text{ correctly classifies an example } x_k \text{ chosen according to } f$

$p_k = \text{Probability that a given example } x_k \text{ is correctly classified by a randomly chosen classifier } \varphi_j$
Example of Decision Matrix

<table>
<thead>
<tr>
<th></th>
<th>$\varphi_1$</th>
<th>$\varphi_2$</th>
<th>$\varphi_3$</th>
<th>$\varphi_4$</th>
<th>$\varphi_5$</th>
<th>$\varphi_6$</th>
<th>$\varphi_7$</th>
<th>$\varphi_8$</th>
<th>$\varphi_9$</th>
<th>$p_k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.67</td>
</tr>
<tr>
<td>$x_2$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.67</td>
</tr>
<tr>
<td>$x_3$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.56</td>
</tr>
<tr>
<td>$x_4$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.44</td>
</tr>
<tr>
<td>$x_5$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.44</td>
</tr>
<tr>
<td>$x_6$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.44</td>
</tr>
<tr>
<td>$x_7$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.33</td>
</tr>
<tr>
<td>$r_j$</td>
<td>0.71</td>
<td>0.71</td>
<td>0.57</td>
<td>0.57</td>
<td>0.57</td>
<td>0.43</td>
<td>0.43</td>
<td>0.29</td>
<td>0.29</td>
<td>r = 0.51</td>
</tr>
</tbody>
</table>

Values of $p_k$ decrease top down

Values of $r_j$ decrease left to right
Conflicting Points of View

Classifier-centred strategy
Each classifier in $\Phi$ is only interested in how many examples in $X$ it correctly classifies, independently from:
- the identity of the misclassified examples
- the accuracies of other classifiers

Example-centred strategy
Each example in $X$ is only interested in being itself correctly classified, independently from:
- the identity of the used classifiers
- whether or not any other example is correctly classified
“Expert” Classification

• **Classifier-centred** strategy

• Monte Carlo matrix has:
  - n rows, each corresponding to a set of \( n \) instances \( x_{k,i} \) of example \( x_k \)
  - just one column, corresponding to a single classifier \( \varphi_1 \)

• Probability that instance \( x_{k,i} \), randomly extracted from \( X \), is correctly classified:

\[
\pi^{(s)}(x_{k,i}) = p_j(x_k) \in \{0, 1\}
\]

\[
E_f[\pi^{(s)}(x_{k,i})] = \sum_{k=1}^{n} f_k p_1(x_{k,i}) = r_1
\]

\[
\text{Var}_f[\pi^{(s)}(x_{k,i})] = r_1 (1 - r_1)
\]

• Accuracy of the classifier (constant):

\[
\rho^{(s)} = \sum_{k=1}^{n} f_k \pi^{(s)}(x_k) = r_1
\]

• Assigned class: \( \varphi_1(x_{k,i}) \)
(Bagging-like) Ensemble Classification (1)

- **Classifier-centred strategy**
- **Monte Carlo matrix** has:
  - n rows, each corresponding to a set of $n_k$ instances $x_{k,i}$ of example $x_k$
  - T columns, corresponding to R classifiers $\varphi_1, \ldots, \varphi_T$
- **Probability** that instance $x_{k,i}$, randomly extracted from $X$, is correctly classified:

  $S = \sum_{k=1}^n n_k$ and $V = \sum_{k=u+1}^v n_k$

  $\pi_T^{(e)}(x_{k,i}) = I_{p_k > 1/2} + \frac{1}{2} I_{p_k = 1/2}$

  $\pi_T^{(e)} = \mathbb{E}_T [\pi_T^{(e)}(x_{k,i})] = \sum_{k=1}^n f_k I_{p_k > 1/2} + \frac{1}{2} \sum_{k=1}^n f_k I_{p_k = 1/2} = \frac{S}{N} + \frac{V}{2N}$

  $\text{Var}_T [\pi_T^{(e)}(x_{k,i})] = \pi_T^{(e)}(1 - \pi_T^{(e)}) - \frac{V}{4N}$

  $\rho_T^{(e)} = \sum_{k=1}^n f_k \pi_T^{(e)}(x_{k,i}) = \sum_{k=1}^n f_k I_{p_k > 1/2} + \frac{1}{2} \sum_{k=1}^n f_k I_{p_k = 1/2} = \frac{S}{N} + \frac{V}{2N} = \pi_T^{(e)}$

  $\text{Max}(0, 2r - 1) \leq \rho_T^{(e)} < \text{Min}(1, 2r)$
(Bagging-like) Ensemble Classification (2)

If we consider a randomly extracted instance \( x_{k,i} \) from \( \mathcal{X} \) according to \( f \), we can compute the probability that it is correctly classified, independently from the other ones:

\[
\pi_T^{(e)} < 1 - \sum_{s=0}^{\frac{T}{2}} \sum_{J_s \subseteq J_T} G(J_s) \\
G(J_s) = \prod_{i \in J_s} r_i \prod_{i \in J_T - J_s} (1 - r_i)
\]

\( J_T = \{1, 2, \ldots, T\} \quad J_s = \{j_1, j_2, \ldots, j_s\} \)

If \( r_j = r \quad (1 \leq i \leq T) \):

\[
\pi_T^{(e)} = 1 - \sum_{s=0}^{\frac{T}{2}} \binom{T}{s} r^s (1 - r)^{T-s}
\]
Monte Carlo Classification

- **Example-centred strategy**

  We extract randomly, with replacement, from Φ a subset of $T << R$ classifiers, and use them to classify $x_{k,i}$

  If $T \to \infty$, the whole set $\Phi$ is used:

  $\mathcal{P}^{(MC)}_{\infty}(x_{k,i}) = I_{p_k > 1/2} + \frac{1}{2} I_{p_k = 1/2}$

  $\rho^{(MC)}_{\infty} = \frac{S}{N} + \frac{V}{2N} = \rho^{(e)}_T$

  If $T$ is finite:

  $\mathcal{P}^{(MC)}_T(x_{k,i}) = \sum_{t=1+\left\lfloor \frac{T}{2} \right\rfloor}^{T} \binom{T}{t} p_k^t (1-p_k)^{T-t}$

  Each example is classified independently. Not all examples have the same difficulty.
Mean and Variance of the Accuracy

Probability that a given subset of T classifier is chosen:

\[
\zeta(\Phi_T^{(k,i)}) \triangleq \Pr(t_1, t_2, ..., t_R) = \left( t_1 \begin{array}{cccc} T \end{array} t_2 \cdots t_R \right) \prod_{j=1}^{R} \left( \frac{1}{R} \right)^{t_j} \quad \text{with} \quad \sum_{j=1}^{R} t_j = T
\]

\[
\mathbb{E}_\zeta \left[ \rho_T^{(MC)} \right] = \frac{1}{N} \mathbb{E}_\zeta \left[ c(T) \right] = \frac{1}{N} \sum_{k=1}^{n} \mathbb{E}_\zeta \left[ c_k(T) \right] = \sum_{k=1}^{n} f_k \pi(T, p_k)
\]

\[
\text{Var}_\zeta \left[ \rho_T^{(MC)} \right] = \frac{1}{N^2} \sum_{k=1}^{n} \text{Var}_\zeta \left[ c_k(T) \right] = \frac{1}{N} \sum_{k=1}^{n} f_k \pi(T, p_k) [1 - \pi(T, p_k)]
\]
Probability Distribution of the Accuracy

Exact probability distribution -> classifications of examples are independent

Very good approximation:

\[ \theta_T^{(MC)}(\xi) \approx \frac{1}{\sqrt{2\pi \text{ Var}[\rho_T^{(MC)}]}} e^{-\frac{(\xi - E[\rho_T^{(MC)}])^2}{2 \text{ Var}[\rho_T^{(MC)}]}} \]
Crowdsourcing Classification

- Example-centred strategy
- Number R of workers (classifiers) and number N of tasks (examples) very large
- Each example is classified by T workers and the answers are aggregated
- Reliability of workers is unknown (hammer, spammer)

Crowdsourcing classification can be performed EXACTLY as Monte Carlo suggests
Crowdsourcing and Machine Learning (ML)

- Single classification is impossible
- Ensemble is impractical, especially Adaboost-like (we do not know the true labels of examples)
- ML is not concerned with the way in which classification is performed
- Monte Carlo theory allows redundancies and independencies to be exploited at their best
- Multi-class problem: impossible to reduce to pairwise classification or one-vs-all

Monte Carlo classification handles in a uniform way both binary and multiclass problems
Label Assignment

Most difficult problem
- Accuracy of workers is unknown
- Ground truth is unavailable

An extra criterion is needed
- Aggregation
- Hypothesis about the distribution of the workers among hammers and spammers
- Correct classification satisfies Maximum Likelihood principle for some parameter

Two approaches
- One-step aggregation
- Iterative process (EM alternating computation of probability of label and reliability of workers)
Summary

• Monte Carlo framework
  – Binary/multiclass
  – Spammer resistant
  – Accuracies of workers not required to be all greater than \( \frac{1}{2} \)

• Experiments
  – (Amazon Mechanical Turk)
  – Hung et al (2013) -> Framework for synthetic data
Conclusions

• Trade-off between Quantity and Quality
• Concentrated vs diffuse knowledge
• Generating knowledge has a cost
• Human -> Machine -> Human

• Crowdsourcing is well suited to problems where the natural abilities of humans can be easily exploited

• Relationships with Machine Learning?