Computing the median string of a distribution

Jorge Calvo-Zaragoza  Colin de la Higuera  Jose Oncina

Pattern Recognition and Artificial Intelligence Group

December 3, 2014
Bayesian Decision Theory

- Given input $x$, we want to guess its correct label from a set of categories $C = (c_1, \ldots, c_{|C|})$
- Bayesian Decision Theory: take the action that minimizes the empirical risk

$$R(c_i|x) = \sum_{j=1}^{|C|} Pr(c_j|x) \lambda(c_j|c_i)$$
Loss Function

The 0/1 loss function is widely used for classification

\[ R(c_i|x) = \sum_{j=1}^{|C|} Pr(c_j|x) \lambda(c_j|c_i) = \sum_{j\neq i} Pr(c_j|x) = 1 - Pr(c_i|x), \]

which leads to a MAP criterion

\[ \arg\min_{c_i} R(c_i|x) = \arg\max_{c_i} Pr(c_i|x) \]
Median String criterion

- In fields like ASR, MT or OCR the categories are strings ($\Sigma^*$).
- The loss function is the Edit distance ($d_{edit}$)

$$R(w|x) = \sum_{v \in \Sigma^*} Pr(v|x)d_{edit}(w, v),$$

- The string that minimizes the risk is the Median string

Inconsistency

- MAP criterion is typically used for these tasks
Distribution over strings

- In such tasks, the string set of categories is defined by probabilistic models \( \text{HMM, PFA, } \ldots \) 
- Given input \( x \), these models are able to give a posterior probability \( Pr(x|w), \forall w \in \Sigma^* \) 
- We are interested in computing the median string of a distribution generated by a \( \text{PFA} \)
Let us denote a probabilistic finite automaton $\mathcal{A}$ with $n$ states by:

- $S \in \mathbb{Q}^{1 \times n}$ represents the probabilities of starting at each state.
- $M = \{M_a \in \mathbb{Q}^{n \times n} | a \in \Sigma\}$ represents the transition probabilities.
- $F \in \mathbb{Q}^{n \times 1}$ represents the probabilities of ending in each state.

Given a string $w = w_1 \cdots w_k$ we compute $Pr_{\mathcal{A}}(w)$ as:

$$Pr_{\mathcal{A}}(w) = S \prod_{i=1}^{\left|w\right|} M_{w_i} F$$
Basic problems

Name: Computing the edit distance to a PFA (EDP)
Instance: A PFA $\mathcal{A}$ over an alphabet $\Sigma$. A string $w$.
Question: Compute $d_{edit}(w, \mathcal{A}) = \sum_{v \in \Sigma^*} Pr_A(v)d_{edit}(w, v)$

Name: Median String from a PFA search (MSP)
Instance: A PFA $\mathcal{A}$ over an alphabet $\Sigma$.
Question: Find a string $w$ in $\Sigma^*$ such that $\forall x \in \Sigma^*$, $d_{edit}(w, \mathcal{A}) \leq d_{edit}(x, \mathcal{A})$
MSP is NP-Hard

- Finding the median of a set of strings is NP-Hard (Casacuberta and de la Higuera, 2000)
- MSP is also NP-Hard
- Otherwise, we can solve the first problem by building a PFA giving an equal probability to, and only to, those strings of the set
Distance string-automaton (EDP)

Problem formulation

- Let $X$ denote the random variable \textit{string of $A$}
- Given a string $w \in \Sigma^*$, we are interested in $E[d_{edit}(w, X)]$
- Is such value finite?

Proof

Considering that $d_{edit}(u, v) \leq |u| + |v|,$

$$E[d_{edit}(w, X)] \leq E[|w| + |X|] = |w| + E[|X|]$$

Then, since

$$E[|X|] = SM_{\Sigma}(I - M_{\Sigma})^{-2}F < \infty,$$

$E[d_e(w, X)]$ is finite
First attempts

- Some first attempts to solve EDP:
  - Dynamic Programming scheme
  - Edit Automaton
EDP with DP

- Attempt with acyclic $P_{FA}$
- Given a string $w = w_1 \ldots w_{|w|}$, let us define

\[
T[i, j, d] = \sum_{x \in \Sigma^*: d_{edit}(x, w_1 \ldots w_i) = d} Pr_{qj}(x)
\]

- Distance between $w$ and $A$ can be defined as:

\[
\sum_{d \in \mathbb{N}} d \sum_{q_j \in Q} T[|w|, j, d] \cdot F_{qj}
\]
EDP with DP

Initialization

- $\forall q_j \in Q, T[0, j, 0] = l_{q_j}$
- $\forall q_j \in Q, T[0, j, d] = Pr_{q_j}(\Sigma^d)$
- $T[i + 1, j, 0] = \sum_{q_k \in Q} T[i, k, 0] \cdot \delta(q_k, w_{i+1}, q_j)$

Iteration

- $T[i + 1, j, d + 1] = ??$
Edit Automaton

Figure 1: Deterministic automaton for string ab
EDP with Edit Automaton

- Combine the PFA $A$ with the Edit Automaton for $w$
- Distribution and Edit distance within the same model
- Decode the machine to solve $d_{edit}(w, A)$
EDP with Edit Automaton

- Combine the PFA $\mathcal{A}$ with the Edit Automaton for $w$
- Distribution and Edit distance within the same model
- Decode the machine to solve $d_{edit}(w, \mathcal{A})$

Bad news

- Number of states seemed to be exponential with respect to $w$ and $\Sigma$
Randomized algorithm

We are given a PFA $\mathcal{A}$, a string $w$ and two values $\epsilon > 0, \delta > 0$

Randomized version
Is there an algorithm which in time polynomial in $|\mathcal{A}|, |w|, \frac{1}{\epsilon}, \frac{1}{\delta}$ computes a value $v$ such that, with probability at least $1 - \delta$,

$$|v - d_{edit}(w, \mathcal{A})| \leq \epsilon?$$
Randomized algorithm

We are given a PFA $\mathcal{A}$, a string $w$ and two values $\epsilon > 0, \delta > 0$

Randomized version
Is there an algorithm which in time polynomial in $|\mathcal{A}|, |w|, \frac{1}{\epsilon}, \frac{1}{\delta}$ computes a value $\nu$ such that, with probability at least $1 - \delta$,

$$|\nu - d_{edit}(w, \mathcal{A})| \leq \epsilon?$$

PAC framework

- Extract $n$ samples of the distribution and approximate the solution
- How many samples do we need to ensure previous conditions?
Bounds

- Let $X_1, \ldots, X_N$ be $N$ random strings sampled $iid$ using a PFA $A$.
- Let $Y_i = d_{edit}(u, X_i)$.
- Let $Y$ be the random variable $\frac{1}{N} \sum_{i=1}^{N} Y_i$.
- Let $Z$ be the event $|Y - E(Y)| \geq \epsilon$.

The goal is to bound $Pr[Z]$. 
Bounds

Concentration bounds

- Problem: they deal with bounded random variables
Bounds

Concentration bounds

- Problem: they deal with bounded random variables

Solution

- Define $M = \max |X_i|$ and let $m$ be an arbitrary number
- $Pr[Z] = Pr[Z \cap M \leq m] + Pr[Z \cap M > m]$
- We bound separately:
  - $Pr[Z \cap M \leq m] \leq Pr[Z | M \leq m]$
  - $Pr[Z \cap M > m] \leq Pr[M > m]$
Bounds

- By Concentration bounds (Hoeffding’s inequality)
  - $Pr[Z \mid M \leq m] \leq 2e^{-\frac{2\epsilon^2 N}{(|u|+m)^2}}$ since $0 \leq Y_i \leq |u| + m$

- By Union Bound
  - $Pr[M > m] \leq N Pr[|X| > m] \leq Ne^{-c_A m}$
By Concentration bounds (Hoeffding’s inequality)

- \( Pr[Z \mid M \leq m] \leq 2e^{-2\epsilon^2 N \frac{m}{(\|u\| + m)^2}} \) since \( 0 \leq Y_i \leq \|u\| + m \)

By Union Bound

- \( Pr[M > m] \leq NPr[|X| > m] \leq Ne^{-c_A m} \)

It follows that:

\[
Pr[|Y - E(Y)| \geq \epsilon] \leq 2e^{-2\epsilon^2 N \frac{m}{(m + \|u\|)^2}} + Ne^{-c_A m}
\]

We can now bound each term by \( \frac{\delta}{2} \).
Bounds

On one hand,

\[ 2e^{\frac{-2\epsilon^2 N}{(m + |u|)^2}} \leq \frac{\delta}{2} \]

\[ \log 2 + \frac{-2\epsilon^2 N}{(m + |u|)^2} \leq \log(\delta) - \log 2 \]

\[ N \geq \frac{(m + |u|)^2(2 \log 2 - \log(\delta))}{2\epsilon^2} \] (1)

\[ m + |u| \leq \sqrt{\frac{2N\epsilon^2}{2 \log 2 - \log(\delta)}} \]

\[ m \leq \epsilon \sqrt{\frac{2N}{2 \log 2 - \log(\delta)}} - |u| \] (2)
On the other hand,

\[ Ne^{-c_A m} \leq \frac{\delta}{2} \]

\[ \frac{N}{e^{mc_A}} \leq \frac{\delta}{2} \]

\[ e^{mc_A} \geq \frac{2N}{\delta} \]

\[ m \geq \frac{\log 2N - \log(\delta)}{c_A} \]  \hspace{1cm} (3)

\[ N \leq \frac{\delta e^{mc_A}}{2} \]  \hspace{1cm} (4)
Bounds

We want the following to hold:

\[
\frac{(m + |u|)^2 (2 \log 2 - \log(\delta))}{2\epsilon^2} \leq \frac{\delta e^{mc_A}}{2}
\]

or

\[
\frac{\log 2N - \log(\delta)}{c_A} \leq \epsilon \sqrt{\frac{2N}{2 \log 2 - \log(\delta)}} - |u|
\]  
(5)
At the end of the day...

\[ N \geq \frac{1}{2} \left( \frac{1}{Kc_A} + \sqrt{\frac{1}{K^2c_A^2} + \frac{1}{K} \cdot \left( \frac{\log \frac{1}{\delta}}{c_A} + |u| \right)^4} \right) \]

with \( K = \epsilon \sqrt{\frac{1}{2 \log 2 \delta}} \)
At the end of the day...

\[ N \geq \frac{1}{2} \left( \frac{1}{Kc_A} + \sqrt{\frac{1}{K^2 c_A^2} + \frac{1}{K} \cdot \left( \frac{\log \frac{1}{\delta}}{c_A} + |u| \right) } \right)^4 \]

with \( K = \epsilon \sqrt{\frac{1}{2 \log 2 - \log \delta}} \)

EDP is PAC-learnable
Solving MSP

Use EDP to approximate MSP

**Heuristics**

- From the most probable string to a EDP local minimum:
  - Iteratively querying neighborhood strings
  - Iteratively moving to the most repeated edit operation
  - ...
  - ...
To do

- Justification of the use of a randomized algorithm
- Experimentation
  - Which heuristic performs better?
  - Median string is more profitable when minimizing Edit distance
Computing the median string of a distribution

Jorge Calvo-Zaragoza  Colin de la Higuera  Jose Oncina

Pattern Recognition and Artificial Intelligence Group

December 3, 2014