**Problem description:** “Feature selection at training time”

**Goal**  Fast classification  (no waste of processing)  

---

**Complex classification:**  problem too hard to plan full automation  

⇒ confidence estimation is needed (for rejection)  

---

**Several types of features are available,**  

from raw data to high-level features  

---

**Low-level features are much faster to be computed**  

---

**Low-level features are sufficient to identify well some particular classes**

---

**Application**  Classification of scanned documents  mainly for mailroom automation  

---

**Cost parameter**  $c$  

---

## A cascade decision tree algorithm  

**Inputs** one test sample $x$  

---

$N$ “group-of-features extractors” $f_i$  

---

ranked by increasing CPU needs  

---

$N$ confidence-rated classifiers $C_i$ (*)  

---

each trained on first features $\{f_k\}_{k=1..i}$

---

(*) assume that  the more input features there are, the more accurate the classification is  

---

**Params**  

confidence thresholds $\{\tau_i\}_{i=1..N}$

---

confidence correction functions $\{\tau_i\}_{i=1..N}$

---

**Outputs**  

predicted class $\hat{c}$  

---

confidence $s(\hat{c}|x)$  

---

for $i = 1$ to $N$ do  

---

Compute group of features $f_i(x)$

---

Compute outputs of classifier $C_i$ on $\{f_k\}_{k=1..i}$

---

predicted class $\hat{c}_i$ and confidence $s_i$  

---

if $i = N$ or $s_i \geq \tau_i$ then  

---

return $\hat{c} = \hat{c}_i$, $s(\hat{c}|x) = \tau_i(s_i)$

---

end if  

---

end for  

---

**Costs and optimization**

**Confidence thresholds $\{\tau_i\}_{i=1..N-1}$**

---

We define three types of additive costs:  

---

| $C_0 \geq 0$ | TRUE | LESS features |
| $C^- > 0$ | FALSE | LESS features |
| $C^+ = 1$ | FALSE | MORE features |

---

**Loss function to minimize**  

Sum of these costs  

---

**Optimization method**  Grid search on (quantized) possible thresholds combinations  

---

**Confidence correction functions $\{\tau_i\}_{i=1..N}$**

---

Non-parametric approach (may overfit?) vs. Parametric approach (how much engineered?)

---

**Conclusions**

---

* The proposed testing cascade strategy relies on some properties of the core classifier.  

---

* Empirical results are mixed.  

---

* What are the alternative algorithms to select features at test time?