Stumping Along a Summary
Exploration & Exploitation Challenge 2011

Christophe Salperwyck\(^1\)  Tanguy Urvoy\(^2\)

\(^1\)Orange Labs / Lille University
\(^2\)Orange Labs

ICML 2011
Outline

1. Problem Statement

2. Approach
   - Online Summary
   - Base Predictors
   - Online Combination of Predictors

3. Experiments and Results
   - Adobe Dataset (Challenge)
   - Orange Portal Dataset

4. Conclusion and Discussion
Online content selection problem

General problem statement

- a web site offers different items (news, ads, books...)
- in order to maximize the number of clicks:
  - select for each visitor the item she is most likely to click on
- instance described by **visitor profile features + item features + mixed features**
  - gps position, time, navigation history, query...
  - item id, topic, creation time, content keywords...
  - cosine similarity, gps distance...
- prediction/learning/selection must be done online:
  - limited time
  - limited memory
- rare and noisy events

Our formalization: only order matters

**online (bipartite) ranking with partial feedback**

- ranking + scarcity \(\rightarrow\) optimize AUC rather than accuracy
- partial feedback \(\rightarrow\) risk of being stuck in suboptimal predictive models
Problem Statement

Online content selection

Off-line evaluation of online content selection systems

- online evaluation: costly
- rejection sampling from logs: unbiased estimator, but too many rejections

This challenge simulation setting

- choose both profile and item from a batch of 6 consecutive log instances
- less realistic than rejection sampling ⇒ no rejection
- maybe enough for comparison of algorithms

From our own experiments with Orange portal logs

- some profile features are very discriminant
  - e.g. pageid: orange home page click rate = 5 × other pages
  - only learn to select home page vs. other pages
- what we did to reduce this bias: take logs from a single page (web-mail)
- maybe better: select batch of instances with “similar” visitor profiles
  - same time, same visited pages. . .
  - close gps positions etc.
Outline

1 Problem Statement

2 Approach
   - Online Summary
   - Base Predictors
   - Online Combination of Predictors

3 Experiments and Results
   - Adobe Dataset (Challenge)
   - Orange Portal Dataset

4 Conclusion and Discussion
Motivation for a Summary

Challenge constraints

- limited time: 100ms/round for predicting and learning
- limited memory: 1.7GB of memory
- keep Garbage Collector quiet 😊

We wanted a summary with small memory footprint

- fast updates
- fast queries
Numerical features summary

Assuming a static distribution, we want to answer questions like:

- what are the click/¬click counts for feature $x_i \in [a, b]$?

GK merge-prune algorithm: online equal-freq histogram

Greenwald M., Khanna S. *Space-efficient online computation of quantile summaries*. SIGMOD’01

- this algorithm maintains an array of tuples $< v_i, g_i, \Delta_i >$ where:
  - $v_i$ is a value seen in the stream
  - $g_i$ is the number of values seen between $v_{i-1}$ and $v_i$
  - $\Delta_i$ is the maximum possible error on $g_i$

- properties:
  - equal-freq, online, bounded memory
  - insensitive to data order and/or data distribution
  - strong error guarantees ($\epsilon$-approximate quantile summary)
Nominal features summary

Problem: the number of nominal values is not supposed to be bounded
- for instance: client ids, cookies or city names
- we do not need to keep rare values: e.g. outdated cookies or small cities

Solution: hashing
- hash nominal values into a fixed number of buckets
- count click/¬click for each bucket
- may be done with several hashing functions to reduce error:
Outline

1. Problem Statement

2. Approach
   - Online Summary
   - Base Predictors
   - Online Combination of Predictors

3. Experiments and Results
   - Adobe Dataset (Challenge)
   - Orange Portal Dataset

4. Conclusion and Discussion
Base Predictors: probability estimation trees

- **Output**: click-rate estimate
- **Split criterion**: Gini impurity
- **Numerical features**:
  - simple decision stumps (depth=1)
    
    \[
    \text{estimate} = \begin{cases} 
    \text{left estimate} & \text{if } x_i \leq \theta \\
    \text{right estimate} & \text{else}
    \end{cases}
    \]
  - deeper univariate trees
    - gradual unfolding: add new leaves with data arrival
- **Nominal features**:
  - hash-stump: hash and treat as continuous
  - best vs. all: best value/others
- **Univariate trees**: work with a single feature
  - global summary → full-rebuild from summary at each step
- **Multivariate trees**: combine several features
  - one summary per leaf → unfolding is a definitive action
Outline

1. Problem Statement

2. Approach
   - Online Summary
   - Base Predictors
   - Online Combination of Predictors

3. Experiments and Results
   - Adobe Dataset (Challenge)
   - Orange Portal Dataset

4. Conclusion and Discussion
Online combination of predictors

- Output linear combination of predictions
- Intuition: whack bad predictors
  - exponential weighting
  - online linear ranker
    - set weight to optimize bipartite ranking loss (i.e. AUC)

- Simple averaging: optimizes nothing
- Online bagging
Experiments

Adobe dataset (phase 1 of challenge)
- 3M instances (i.e. 500K batches of 6)
- 120 features: 99 continuous / 21 nominals (last nominal feature is "option id")
- click rate: $\sim 0.0024$

Orange portal web-mail dataset
- 6M instances (i.e. 1M batches of 6)
- 33 nominal features
- click rate: $\sim 0.0005$
Outline

1. Problem Statement

2. Approach
   - Online Summary
   - Base Predictors
   - Online Combination of Predictors

3. Experiments and Results
   - Adobe Dataset (Challenge)
   - Orange Portal Dataset

4. Conclusion and Discussion
Adobe dataset with perfect player

Experiments and Results

Adobe Dataset (Challenge)

Christophe Salperwyck, Tanguy Urvoy (Orange labs)

Stumping along a summary

July 2011
Adobe dataset without perfect player

Experiments and Results
Adobe Dataset (Challenge)

Christophe Salperwyck, Tanguy Urvoy (Orange labs)
Stumping along a summary
Deviation from static model
Utree vs. full dataset discretization (MODL)
Outline

1. Problem Statement

2. Approach
   - Online Summary
   - Base Predictors
   - Online Combination of Predictors

3. Experiments and Results
   - Adobe Dataset (Challenge)
   - Orange Portal Dataset

4. Conclusion and Discussion
Orange portal dataset with perfect player

Orange portal (web-mail) click rewards

Orange portal (web-mail) click-rates (exponential smoothing)
Orange dataset with perfect player

![Graph 1: Orange portal (web-mail) click rewards [NO PERFECT]]

Random 1
Random 2
UTree
best

![Graph 2: Orange portal (web-mail) click-rates [NO PERFECT]]

Christophe Salperwyck, Tanguy Urvoy (Orange labs)

Stumping along a summary
Thank you for this challenge!

Experiment summary

- interesting simulation scheme: how many items/batches?
- combination of GK merge-prune and hashing provides a good stream summary to rely on for click prediction
- surprisingly (unfortunately?) simple decision stumps seem to be competitive base predictors

Why are decision stumps so good?

- maybe because of their stability?
- maybe stump-induced partial order lets some room for exploration?
Experiment summary (for Adobe dataset)

- GK-100: Greenwald & Khanna (100 tuples), CMS-1: Count min-sketch (1 hash)
- 1R: single rules $x_i \in ]a, b] ? 1 : 0$
- HStump: hash then split like stump
- RF: random forests, UTree: univariate tree
- AVG: averaging, OZA: Oza bagging (instance weighting)
- multivariate predictors typed in red

<table>
<thead>
<tr>
<th>name</th>
<th>summary</th>
<th>base-predictors</th>
<th>combi.</th>
<th>score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td></td>
<td></td>
<td></td>
<td>1127 ± 41</td>
</tr>
<tr>
<td>LinRank</td>
<td>GK-4/CMS-1</td>
<td>1R (proba. output)</td>
<td>LRank-120</td>
<td>~ 1600</td>
</tr>
<tr>
<td>LinRank</td>
<td>GK-4/CMS-1</td>
<td>1R (0/1 output)</td>
<td>LRank-120</td>
<td>~ 1800</td>
</tr>
<tr>
<td>RF</td>
<td>GK-100/CMS-1</td>
<td>20 trees (10 feat.)</td>
<td>AVG-20</td>
<td>~ 1900</td>
</tr>
<tr>
<td>Stumps1VA</td>
<td>GK-100/CMS-1</td>
<td>Stump/1 vs. all</td>
<td>AVG-120</td>
<td>~ 2000</td>
</tr>
<tr>
<td>UTree</td>
<td>GK-100/CMS-1</td>
<td>UTree/HStump</td>
<td>AVG-120</td>
<td>~ 2030</td>
</tr>
<tr>
<td>Cnrs</td>
<td></td>
<td></td>
<td></td>
<td>~ 2040</td>
</tr>
<tr>
<td>Simple stumps</td>
<td>GK-100/CMS-1</td>
<td>Stump/HStump</td>
<td>AVG-120</td>
<td>~ 2050</td>
</tr>
<tr>
<td>Best</td>
<td>GK-100/CMS-1</td>
<td>800 stumps + 200 RF</td>
<td>OZA-1K</td>
<td>~ 2080</td>
</tr>
<tr>
<td>Inria</td>
<td></td>
<td></td>
<td></td>
<td>~ 2200</td>
</tr>
<tr>
<td>Perfect</td>
<td></td>
<td></td>
<td></td>
<td>6678</td>
</tr>
</tbody>
</table>
Adobe dataset (end zoom)

Adobe click rewards [END ZOOM]

Adobe click-rates [END ZOOM]
Utree learning