Designing DAG-shaped Classifiers for Fast Triggers

Djalel Benbouzid
Vladimir Vava Gligorov
Michael Williams
Balázs Kégl

benbouzid @ lal.in2p3.fr
Laboratoire de l’Accélérateur Linéaire
Université Paris-Sud, CNRS/IN2P3

May, 17th 2013
Motivation
Fast Classification and Trigger Design

- High Level Triggers,
- Face detection,
- Web page ranking, ...

- Imbalanced data
- Real-time classification
- Limited budget
- Multi-class

Designing DAG-shaped classifiers for fast triggers
Fast Classification and Trigger Design

- High Level Triggers,
- Face detection,
- Web page ranking, ...
- Imbalanced data
- Real-time classification
- Limited budget
- Multi-class
What might come to mind...

**Adaboost**  (Freund and Schapire 1995) ¹
**Good properties** both theoretical and practical
**Multi-class** flavors like Adaboost.MH
**Versatile** can be used with a plethora of base learners
**Anytime** Direct control over over the complexity

---

Adaboost.MH

- Iteratively constructs a pool of base classifiers
- For \( K \) classes, returns a score function of the form

\[
f(x) = \sum_{i=1}^{T} \alpha_i h_i(x) \in \mathbb{R}^K
\]

Prediction: \( \hat{\ell} = \arg \max_{\ell} f_\ell(x) \)

- Usually not applicable to real-time classification (big \( T \))
- Representation
Adaboost.MH

- Iteratively constructs a pool of base classifiers
- For $K$ classes, returns a score function of the form

$$f(x) = \sum_{i=1}^{T} \alpha_i h_i(x) \in \mathbb{R}^K$$

Prediction: $\hat{\ell} = \arg \max_{\ell} f_\ell(x)$

- Usually not applicable to real-time classification (big $T$)
- Representation
Adaboost.MH

- Iteratively constructs a pool of base classifiers
- For $K$ classes, returns a score function of the form

$$f(x) = \sum_{i=1}^{T} \alpha_i h_i(x) \in \mathbb{R}^K$$

Prediction: $\hat{\ell} = \arg \max_{\ell} f_\ell(x)$

- Usually not applicable to real-time classification (big $T$)
- Representation
Cascade classifiers
Chaining classifiers

Designing DAG-shaped classifiers for fast triggers
Chaining classifiers

Designing DAG-shaped classifiers for fast triggers
Chaining classifiers

Designing DAG-shaped classifiers for fast triggers
**Chaining classifiers**

Designing DAG-shaped classifiers for fast triggers
But... 

- Hand-tuning of the hyper-parameters.
- No early classification for signal.
- The margin information is lost.
- No straightforward extension to multi-class classification.
Cascade improvements

- Viola & Jones cascade (IJCV 2004)
- SoftCascade (Bourdev et. al CVPR 2005)
- FCBoost (Saberian et. al. NIPS 2010)
- SVMBoost (Xu et. al. ICML 2013)
MDDAG
Markov Decision Directed Acyclic Graph
Example (trailer...)

Designing DAG-shaped classifiers for fast triggers
Example (trailer...)
Q-value | Feature $h_4$

Actions
- Skip
- Eval
- Quit

Q value

$Q(f(x))$
The diagram illustrates a model with actions and features. The graph shows the Q-values for different features $h_6$ as a function of $f(x)$. The graph includes curves for different actions: Skip, Eval, Quit, and an additional curve for another action. The Q-values range from 0.0 to 1.2.
\begin{align*}
f(x) &= Q(x) = \langle \mathbf{h}_7, \mathbf{w} \rangle \\
	ext{Actions} &= \{\text{Skip}, \text{Eval}, \text{Quit}\}
\end{align*}
Q-value | Feature $h_8$

Actions
- Skip
- Eval
- Quit

$f(x)$

-0.2 -0.16 -0.12 -0.08 -0.04 0.0 0.04 0.08 0.12 0.16 0.2

Q value

Eval Quit Skip Eval Skip Quit

Q-value | Feature $h_8$
The figure shows a graph with nodes and edges indicating a sequence of actions. The graph includes a transition diagram with labeled states and transitions. The upper part of the graph represents a network with nodes connected by lines, indicating a path or sequence of actions. The lower part of the graph is a plot of the Q-value against a feature, labeled as $h_9$. The plot includes shaded regions representing different actions: Skip, Eval, and Quit. The x-axis represents the feature value $f(x)$, and the y-axis represents the Q-value.
MDDAG : The setup
The setup

**Given** Sequence of *K*-class base classifiers

\((h_1, \ldots, h_N)\),

\[ h_j : \mathcal{X} \rightarrow \mathbb{R}^K, j = 1, \ldots, N \]

**Goal** Learn an agent \( \pi \) which takes decisions

\{EVAL, SKIP, QUIT\}

**Learning** Episodic undiscounted Markov Decision Process
What the agent knows... (1)

The score function

\[ f^{(j)}(x) = \sum_{j' = 1}^{j} b_{j'}(x) h_{j'}(x) \in \mathbb{R}^K \]

\[ b_{j}(x) = \begin{cases} 
1 & \text{if the feature } j \text{ was evaluated} \\
0 & \text{otherwise} 
\end{cases} \]

Two top ranked labels

\[ \ell_{1}^{(j)}(x) = \arg \max_{\ell} f^{(j-1)}(x) \]

\[ \ell_{2}^{(j)}(x) = \arg \max_{\ell, \ell \neq \ell_{1}^{(j)}(x)} f^{(j-1)}(x) \]

Their score difference

\[ \Delta^{(j)}(x) = \max_{\ell} f^{(j-1)}(x) - \max_{\ell, \ell \neq \ell_{1}^{(j)}(x)} f^{(j-1)}(x) \]
For a given instance

\[
( j, (\ell_1^{(j)}, \ell_2^{(j)}), \Delta^{(j)} )
\]

- base classifier index
- winning labels
- score difference
What the agent knows… (2)

For a given instance

\[
\left( j, \ell_1^{(j)}, \ell_2^{(j)}, \Delta^{(j)} \right)
\]

\begin{align*}
\text{base classifier index} & \quad \text{winning labels} & \quad \text{score difference}
\end{align*}

The agent \( \pi \)

\[
\pi\left( j, \ell_1^{(j)}, \ell_2^{(j)}, \Delta^{(j)} \right) \mapsto \{ \text{EVAL, SKIP, QUIT} \}
\]

\[
\text{state descriptor} \quad \text{actions}
\]
Learning from interaction

- The agent takes actions and receives rewards.
- **QUIT** action reward \( \propto L(f, (x, \ell)) \)
- Penalize **Eval** action: \( r_t = -\beta, \quad 0 < \beta < 1 \)
- Control of the accuracy / complexity trade-off.
- The agent maximizes \( \varrho = \mathbb{E}_x \left\{ \sum_{t=1}^T r(t) \right\} \), \( r_t \in \mathbb{R} \)

### Objective function

\[
\varrho = \mathbb{E}_{(x, \ell) \sim \mathcal{D}} \left\{ -L(f, (x, \ell)) - \beta \sum_{j=1}^N b_j(x) \right\}.
\]
What motivates the agent

Multi-class 0-1 loss

\[
L_{\Pi}(f, (x, \ell)) = \mathbb{I} \left\{ f_\ell(x) - \max_{\ell' \neq \ell} f_{\ell'}(x) < 0 \right\}
\]

Multi-class exponential loss

\[
L_{\text{EXP}}(f, (x, \ell)) = \exp \left( \sum_{\ell' \neq \ell}^{K} f_{\ell'}(x) - f_\ell(x) \right),
\]

Feature costs by varying $\beta$

Asymmetric classification
What the agent learns

Estimate the value of taking a given action in a given state

Radial Basis Functions
Simple discretization
Data-dependent classification
A toy example
A toy example
A toy example
Related work

- Póczos et. al. (ICML, 2009)
- Gao and Koller (NIPS, 2011)
- Dulac-Arnold et. al. (Machine Learning, 2012)
Benchmarks
Object detection benchmarks

Viola-Jones dataset with FP rate = 0.02

DPed dataset with FP rate = 0.1
LHCb Toy Data (1)

Test set:

2body 1674 candidates, 1674 events
3body 4892 candidates, 1968 events
4body 9390 candidates, 1945 events
bkgd 1635092 candidates, 43252 events

Average evaluation cost: 5.418

2body 18.092
3body 13.551
4body 11.357
bkgd 5.389

Error:

- Adaboost – 6 base classifiers: 0.09523
- MDDAG – 5.42 base classifiers: 0.0456
- Adaboost – 100 base classifiers: 0.03858
LHCb Toy Data (2)
LHCb Toy Data (2)

2body vs background

2body vs 3body
Conclusion
Conclusion and next steps

- A sparse classifier of the form of a Directed Acyclic Graph.
- Adapt to many \textit{(multi-class)} loss functions.
- Data-dependent classification.
Conclusion and next steps

- A sparse classifier of the form of a Directed Acyclic Graph.
- Adapt to many (multi-class) loss functions.
- Data-dependent classification.

- Next: go further with triggers
- Features have acquisition cost
- Learn to manage the cost budget
Thank you
Path clustering

MNIST

All labels

only 2s

Designing DAG-shaped classifiers for fast triggers
Path clustering

All labels

Designing DAG-shaped classifiers for fast triggers
Path clustering