Classifier Cascade for Minimizing Feature Evaluation Cost

Minmin Chen, Zhixiang (Eddie) Xu, Kilian Q. Weinberger, Olivier Chapelle, Dor Kedem
Machine Learning for Real-World Applications
Learning with Test-Time Budget

- Test efficiency
- Speed
- Precision

- Features varying in cost
- Millions/billions of executions per day
- Milliseconds per test

Minmin Chen, Zhixiang (Eddie) Xu, Kilian Weinberger, Olivier Chapelle, Dor Kedem

Classifier Cascade for Minimizing Feature Evaluation Cost
Showcase Application: Web Search Ranking
Gradient Boosted Regression Tree (GBRT)

- All 8 winners out of the 1,055 teams used GBRT or its variants

![Graphical representation of GBRT trees with test cost and precision at 5 relevant documents in top 5 returns]

[Friedman, 2001; Chapelle and Chang, 2011]
Research Goal

Keep the precision, reduce the test cost.

Post processing classifiers

Precision@5

Test cost

faster
Setup

\[ D = \{(x_1, y_1), (x_1, y_1), \cdots, (x_1, y_1)\} \in \mathcal{R}^d \times \{0, 1, 2, 3, 4\} \]

1 unit \( \approx 0.1 \) microsecond
Test Cost of an Ensemble

\[ H \]

\[ \alpha = 0.05 \]

\[ \alpha = 0.05 \]

\[ \alpha = 0.05 \]

\[ \alpha = 0.05 \]

\[ \alpha = 0.05 \]
Feature Evaluation Costs

\[ c = 1 + 5 + 20 + 1 + 20 + 1 + 5 + 0 \]

**A feature becomes free after it is computed for the first time**

**Evaluation cost**

**Feature cost**
Feature Evaluation Costs

\[ H \]

\[ \alpha = 0.05 \]

\[ c = 52 \]

\[ e = 1 \]

\[ \alpha = 0.05 \]

\[ c = 326 \]

\[ e = 1 \]

\[ \alpha = 0.05 \]

\[ c = 0 \]

\[ e = 1 \]

Minmin Chen, Zhixiang (Eddie) Xu, Kilian Weinberger, Olivier Chapelle, Dor Kedem

Classifier Cascade for Minimizing Feature Evaluation Cost
Solution 1: single stage sparse solution

Re-weight the weak classifiers of the ensemble to allow some classifiers to be removed
Single Stage Sparse Solution

\[ H' \]

\[ \alpha_1 = 0.05 \]
\[ \alpha_2 = 0.05 \]
\[ \alpha_3 = 0.052 \]
\[ \alpha_t = 0.05 \]
\[ \beta \rho = 0.0505 \]
\[ \alpha_2 = 0.05 \]
Single Stage Sparse Solution

\[ \mathcal{H}' \]

\[ \beta_1 = 0.1 \]
\[ \beta_2 = 0 \]
\[ \beta_3 = 0.02 \]
\[ \beta = 0.08 \]
\[ \beta_{T-1} = 0.05 \]
\[ \beta_T = 0 \]

\[ \mathbf{x}_i \]

\[ \text{Loss} \]
\[ \frac{1}{2n} \sum_{i=1}^{n} \left( \sum_{t=1}^{T} \beta_t h_t(x_i) - y_i \right)^2 \]

\[ \text{Exact cost} \]
\[ \sum_{t=1}^{T} e_t \delta(\beta_t \neq 0) \]
\[ \sum_{t=1}^{d} \delta \left( \sum_{t=1}^{T} F_{\alpha,t} \delta(\beta_t \neq 0) \right) \]

\[ \text{Approximated cost} \]
\[ \sum_{t=1}^{T} e_t |\beta_t| \]
\[ \sum_{t=1}^{T} c_\alpha \sum_{t=1}^{T} F_{\alpha,t} \beta_t^2 \]

Non-continuous 0-1 function

Minmin Chen, Zhixiang (Eddie) Xu, Killian Weinberger, Olivier Chapelle, Dor Kedem

Classifier Cascade for Minimizing Feature Evaluation Cost

Sunday, April 22, 12
Single Stage Sparse Solution

Loss
\[
\min_{\beta} \frac{1}{2n} \sum_{i=1}^{n} \left( \sum_{t=1}^{T} \beta_t h_t(x_i) - y_i \right)^2
\]

Evaluation cost
\[
\sum_{t=1}^{T} e_t |\beta_t|
\]

Feature cost
\[
\sum_{t=1}^{T} c_\alpha \sqrt{\sum_{t=1}^{T} F_{\alpha t} \beta_t^2}
\]

Reg.
\[
\rho |\beta|
\]

Non-differentiable

Variational method
(Boyd and Vandenberghe, 2004; Chapelle and Keerthi, 2004)

\[
\min_{\beta} \sqrt{g(\beta)} = \min_{\beta, \sigma > 0} \frac{1}{2} \left[ \frac{g(\beta)}{\sigma} + \sigma \right]
\]

\[
\sigma^* = \sqrt{g(\beta)}
\]
Single Stage Sparse Solution

\[
\min_{\beta, \sigma, \gamma} \frac{1}{2n} (H\beta - Y)^2 + \frac{1}{2} \left[ \beta^\top \Sigma \beta + \sum_t (\lambda e_t + \rho) \sigma_t \right] + \frac{1}{2} \left[ \beta^\top \Gamma \beta + \sum_\alpha \lambda c_\alpha \eta_\alpha \right]
\]

- Loss
- Evaluation cost + Reg.
- Feature cost

**Alternating optimization**

- fix $\sigma, \gamma$
- fix $\beta$

\[\sigma^* = \sqrt{g(\beta)}\]

**jointly convex**

**exact !**

- Fast training: 5,000 weak classifiers, 150,000 inputs, 13 seconds
Experimental results

Can we do better?

- Some expensive features are required to achieve high precision
- The dataset is highly class-skewed (most documents are irrelevant)
Solution 2: Cascade classifiers

Classify

easy inputs with cheap features

and

difficult inputs with expensive features.
Classifier Cascade [Viola and Jones, 2002]
Classifier Cascade

[Viola and Jones, 2002]

early exit

$\mathcal{H}_1(x) < \theta_1$
Classifier Cascade [Viola and Jones, 2002]

\[ \mathcal{H} \]

\[ \mathcal{H}_1 \]

\[ \mathcal{H}_2 \]

- early exit \( \mathcal{H}_1(x) < \theta_1 \)
- early exit \( \mathcal{H}_2(x) < \theta_2 \)

Minmin Chen, Zhixiang (Eddie) Xu, Kilian Weinberger, Olivier Chapelle, Dor Kedem
Classifier Cascade [Viola and Jones, 2002]

$\mathcal{H}$

$\mathcal{H}_1$

early exit
$\mathcal{H}_1(x) < \theta_1$

$\mathcal{H}_2$

early exit
$\mathcal{H}_2(x) < \theta_2$

$\mathcal{H}_K$

early exit

Minmin Chen, Zhixiang (Eddie) Xu, Kilian Weinberger, Olivier Chapelle, Dor Kedem
How to order the trees? **Greedy? (cheap trees first)**

Joint optimization on the ordering and the weights

[How to order the trees? Greedy? (cheap trees first)]

Joint optimization on the ordering and the weights

[Dundar et al. 2007; Raykar et. al, 2010]

**myopic!**

Minmin Chen, Zhixiang (Eddie) Xu, Kilian Weinberger, Olivier Chapelle, Dor Kedem

Classifier Cascade for Minimizing Feature Evaluation Cost

Sunday, April 22, 12
Joint Loss of the Cascade

Loss

\[
\frac{1}{2n} \sum_{k=1}^{K} \sum_{i=1}^{n} q_k^i \left( \sum_{t} \beta_t^k h_t(x_i) - y_i \right)^2
\]

Evaluation cost

\[
\sum_{t=1}^{T} e_t \sqrt{\sum_{k=1}^{K} (\beta_t^k d_k)^2}
\]

Feature cost

\[
\sum_{\alpha=1}^{d} c_{\alpha} \sqrt{\sum_{k=1}^{K} \sum_{t=1}^{T} (F_{\alpha t} \beta_t^k d_k)^2}
\]
Cronus: Cyclic Optimization

- Initialization:

  
  1
  
  $X_1$  
  
  $\beta^1, \theta_1$

  early exit

  2
  
  $X_2$

  $\beta^2, \theta_2$

  early exit

  \ldots

  K

  $X_K$

  $\beta^K, \theta_K$

  early exit

  similar to the single stage formulation

  jointly convex and closed form update
Single Stage Sparse Solution

\[
\min_{\beta, \sigma, \gamma} \frac{1}{2n} (H\beta - Y)^2 + \frac{1}{2} \left[ \beta^\top \Sigma \beta + \sum_t (\lambda e_t + \rho) \sigma_t \right] + \frac{1}{2} \left[ \beta^\top \Gamma \beta + \sum_\alpha \lambda c_\alpha \eta_\alpha \right]
\]

Loss\hspace{1cm}Evaluation cost + Reg.\hspace{1cm}Feature cost

Jointly convex

\[\sigma^* = \sqrt{g(\beta)}\]

Fast training: 5,000 weak classifiers, 150,000 inputs, 13 seconds
Cronus: Cyclic Optimization

- Re-optimization:

\[
\begin{align*}
X_1 & \xrightarrow{\beta^1, \theta_1} X_2 \\
X_2 & \xrightarrow{\beta^2, \theta_2} \cdots \\
X_K & \xrightarrow{\beta^K, \theta_K}
\end{align*}
\]
Cronus: Cyclic Optimization

- Re-optimization:

1. $X_1$, $\beta^1, \theta_1$
2. $X_2$, $\beta^2, \theta_2$
3. $X_K$, $\beta^K, \theta_K$

early exit

Minmin Chen, Zhixiang (Eddie) Xu, Kilian Weinberger, Olivier Chapelle, Dor Kedem

Classifier Cascade for Minimizing Feature Evaluation Cost
Cronus: Cyclic Optimization

- Re-optimization:

1. \( X_1 \) with \( \beta^1, \theta_1 \)
2. \( X_2 \) with \( \beta^2, \theta_2 \)
   - Early exit
3. \( X_K \) with \( \beta^K, \theta_K \)
   - Early exit
Experimental Results

700 features \((c = 1 \sim 200)\)
5,000 weak classifiers \((e = 1)\)
n\(\text{Tr} = 141,597\)  \(n\text{Te} = 146,769\)

Precision@5 = \(\frac{\text{relevant documents in top 5 returns}}{5}\)

\(\text{Test cost} \times 10^4\)

- **Early.exit**
  - \(s = 1.0, s = 0.6, s = 0.2\)
  - [Cambazoglu et al., 2010]

- **GBRT**
  - [Friedman et al., 2001]

- **AND-OR**
  - [Dundar et al., 2007]

- **Soft cascade**
  - [Raykar et al., 2010]

Minmin Chen, Zhixiang (Eddie) Xu, Kilian Weinberger, Olivier Chapelle, Dor Kedem

Classifier Cascade for Minimizing Feature Evaluation Cost
Experimental Results

- Early stages use primarily cheap features;
- Expensive features are gradually extracted;
- A few expensive features are used in the early stages;
- Re-optimization rejects data points more aggressively.
Conclusion

- **Cronus, cyclic optimization to post-process classifiers**
  - effectively trades off prediction accuracy and runtime cost
  - globally optimizes the order of feature extraction and classifiers
  - stage-wise closed form updates
Cronus, cyclic optimization to post-process classifiers

- effectively trades off prediction accuracy and runtime cost
- globally optimizes the order of feature extraction
- stage-wise closed form updates

Conclusion
Conclusion

- **Cronus, cyclic optimization to post-process classifiers**
  - effectively trades off prediction accuracy and runtime cost
  - globally optimizes the order of feature extraction and classifiers
  - stage-wise closed form updates
Conclusion

- **Cronus, cyclic optimization to post-process classifiers**
  - effectively trades off prediction accuracy and runtime cost
  - globally optimizes the order of feature extraction and classifiers
  - **stage-wise closed form updates**

  *jointly convex*

  *closed form solution at each iteration*
Thank you!
Questions?