Kernelized Online Imbalanced Learning with Fixed Budgets

Junjie Hu$^{1,2}$, Haiqin Yang$^{1,2}$, Irwin King$^{1,2}$, Michael R. Lyu$^{1,2}$, and Anthony Man-Man-Cho So$^3$


1 Shenzhen Key Laboratory of Rich Media Big Data Analytics and Applications, Shenzhen Research Institute, The Chinese University of Hong Kong
2 Computer Science and Engineering, The Chinese University of Hong Kong
3 Systems Engineering and Engineering Management, The Chinese University of Hong Kong

{jjhu, hqyang, king, lyu}@cse.cuhk.edu.hk, manchoso@se.cuhk.edu.hk

January 27, 2015
Lab Introduction

1. Web Intelligence and Social Computing Lab at CUHK (WISC Lab)
2. Shenzhen Key Laboratory in Shenzhen Research Institute at CUHK

(a) Junjie Hu  
(b) Haiqin Yang

(c) Irwin King  
(d) Michael Lyu  
(e) Anthony So
Overview

1 Introduction

2 Related Work

3 The Proposed Method (KOIL)

4 Theoretical Analysis

5 Experiments

6 Conclusion
Definition of Online learning
- learn from the streaming data
- update the model adaptively from the data stream

Properties
- process the data one by one
- update the model in each iteration
- approximate the learning performance of the batch-train methods

Figure: Rutrell Yasin, Amazon Kinesis does heavy-lifting on streaming, big data
Properties:
- uneven data distribution
- No. of samples in one class < No. of samples in the other class

Problems:
- Accuracy: inappropriate
- Misclassification costs for positive and negative samples are not the same.

Figure: Imbalanced data
1. SVM maps the instance $x$ to the Reproducing Kernel Hilbert Space

$$\phi : x \mapsto \phi(x)$$

2. In RKHS, dot product of two elements:

$$\langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}} = k(x_i, x_j)$$

3. The objective of SVM is to maximize the margins of the hyperplane in RKHS.
1. **non-linear decision function** \( f : \mathbb{R}^d \rightarrow \mathbb{R} \)

2. a sequence of imbalanced feature-labeled pair instances \( \{z_t = (x_t, y_t) \in \mathcal{Z}, t \in [T]\} \), where \( \mathcal{Z} = \mathcal{X} \times \mathcal{Y} \), \( x_t \in \mathcal{X} \subseteq \mathbb{R}^d \), \( y_t \in \mathcal{Y} = \{-1, +1\} \) and \( [T] = \{1, \ldots, T\} \).

3. \( f(x) \) can be calculated by

\[
\langle f(\cdot), k(x, \cdot) \rangle_{\mathcal{H}} = f(x) \tag{1}
\]

4. Assumption: positive class (minority) & negative class (majority)

5. \( N_k^{\tilde{y}}(z) \): the set of the \( k \)-nearest neighbors of \( z \) and have the label of \( \tilde{y} \).
Related Work

1. Online Learning with Kernels: minimize the \textit{hinge loss} function

\[
\min_f \ell_h(f, x, y) := \max(0, 1 - yf(x))
\] (2)

- NORMA [Kivinen 2004]
- Randomized Budget Perceptron [Cavallanti 2007]
- Forgetron [Dekel 2008]
- Projectron [Orabona 2008]

2. Online Linear AUC Maximization: minimize the \textit{AUC-based loss} function

- Online AUC Maximization (OAM) [Zhao 2011]

\[
\min_w \ell_h(w, x^+, x^-) := \max(0, 1 - w \cdot (x^+ - x^-))
\] (3)

- One-Pass AUC Optimization (OPAUC) [Gao 2013]

\[
\min_w \ell_h(w, x^+, x^-) := (1 - w \cdot (x^+ - x^-))^2
\] (4)
Problems & Motivation

1. Deal with non-linear imbalanced data?
2. Pay more attention on minority class?
3. Update the decision smoothly and robustly?
4. Store fixed number of support vectors without information loss?
1. \( \mathcal{K}^+ \) and \( \mathcal{K}^- \): the information of positive and negative SVs respectively, where \( |B^+| = |B^-| \).

\[
\mathcal{K}^+ \cdot \mathcal{A} := \{ \alpha^+_i \}_{i=1}^{|B^+|}, \quad \mathcal{K}^+ \cdot \mathcal{B} := \{ z_i \mid y_i = +1 \}_{i=1}^{|B^+|} \tag{5}
\]

\[
\mathcal{K}^- \cdot \mathcal{A} := \{ \alpha^-_i \}_{i=1}^{|B^-|}, \quad \mathcal{K}^- \cdot \mathcal{B} := \{ z_i \mid y_i = -1 \}_{i=1}^{|B^-|}. \tag{6}
\]

2. Goal: to seek a decision function \( f \) in Eq. (7).

\[
f(x) = \sum_{\alpha^+_i \in \mathcal{K}^+ \cdot \mathcal{A}} \alpha^+_i k(x^+_i, x) + \sum_{\alpha^-_j \in \mathcal{K}^- \cdot \mathcal{A}} \alpha^-_j k(x^-_j, x), \tag{7}
\]
Given the positive dataset $D^+ = \{z_i | y_i = +1\}$ and the negative dataset $D^- = \{z_j | y_j = -1\}$, the AUC is measured as:

$$AUC(f) = \frac{\sum_{i=1}^{|D^+|} \sum_{j=1}^{|D^-|} I[f(x_i^+) - f(x_j^-) > 0]}{|D^+||D^-|}$$

$$= 1 - \frac{\sum_{i=1}^{|D^+|} \sum_{j=1}^{|D^-|} I[f(x_i^+) - f(x_j^-) \leq 0]}{|D^+||D^-|}$$

where $I[\pi]$ is the indicator function.

Maximizing AUC equals to minimizing

$$\sum_{i=1}^{|D^+|} \sum_{j=1}^{|D^-|} I[f(x_i^+) - f(x_j^-) \leq 0]$$

Replace the discrete indicator function $I[\pi]$ in Eq. (8) by the surrogate convex loss function in Eq. (9)

$$\ell_h(f, z, z') := \frac{|y - y'|}{2} \left[ 1 - \frac{1}{2} (y - y')(f(x) - f(x')) \right]_+$$
1. Assign an initial weight to $z_t$
2. Update the weight of SVs, which are KNN of $z_t$ and have the opposite label $-y_t$.
3. does not affect the weight of SVs in the whole buffer
Notation:
\[ z_t^- := (x_t, -1) \]
\[ z_t^+ := (x_t, +1) \]

Objective function:
\[
\hat{L}(f, z_t) = \frac{1}{2} \| f \|_F^2 + C \sum_{z_i \in N_{x_t}^{yt}(x_t)} \ell_h(f, z_i, z_t)
\]

Update Decision Function:
\[ f_{t+1} := f_t - \eta \partial_f \hat{L}(f, z_t) |_{f=f_t} \]

Update Weight:
\[
\alpha_{i,t} = \begin{cases} 
\eta C y_t | V_t |, & i = t \\
(1 - \eta) \alpha_{i,t-1} - \eta C y_t, & \forall i \in V_t \\
(1 - \eta) \alpha_{i,t-1}, & \forall i \in I_t^{yt} \cup V_t
\end{cases}
\]
What if the fixed-size buffers are full?
What if the fixed-size buffers are full?

1. Reservoir Sampling (RS)
2. First-In-First-Out (FIFO)
KOIL: Problem for online learning with kernel

1. What if the fixed-size buffers are full?
   1. Reservoir Sampling (RS)
   2. First-In-First-Out (FIFO)

2. What if we directly remove the SV from the buffer?
KOIL: Problem for online learning with kernel

1. What if the fixed-size buffers are full?
   1. Reservoir Sampling (RS)
   2. First-In-First-Out (FIFO)

2. What if we directly remove the SV from the buffer?
   1. information loss
   2. compensation scheme for information loss
KOIL: Intuition – Update Buffers

Notation:
\[ z_t^- := (x_t, -1) \]
\[ z_t^+ := (x_t, +1) \]

Update Buffers:
1. Not filled: Add to buffer
2. Filled: i) Delete; ii) Add
   iii) Compensate

\[ f_{t+1}^{++}(x) = f_{t+1}(x) - \alpha_r k(x_r, x) + \Delta \alpha_c \cdot k(x_c, x) \]

Stream oblivious policies:
1. First-In-First-Out (FIFO)
2. Reservoir Sampling (RS)
KOIL: Update Kernel

1. Minimize the *instantaneous regularized risk of AUC*.
\[
\min_f \mathcal{L}(f_t, z_t) = \frac{1}{2} \| f_t \|_{\mathcal{H}}^2 + C \sum_{i=1}^{t-1} \ell_h(f_t, z_t, z_i)
\]  
   (10)

2. Minimize the *localized instantaneous regularized risk of AUC* (Reduce the effect of outliers):
\[
\min_f \hat{\mathcal{L}}(f_t, z_t) = \frac{1}{2} \| f_t \|_{\mathcal{H}}^2 + C \sum_{z_i \in N_k^{\neg y_t}(z_t)} \ell_h(f_t, z_t, z_i)
\]  
   (11)

3. Stochastic Gradient Descent: update \( f_t \) in each iteration
\[
f_{t+1} := f_t - \eta \partial_f \hat{\mathcal{L}}(f, z_t) \big|_{f=f_t}
\]  
   (12)

4. Updating rule for the kernel weights:
\[
\alpha_i = \begin{cases} 
\eta C y_t \sum_{z_j \in N_k^{\neg y_t}(z_t)} \mathbb{I}[\phi(z_t, z_j) < 1 \wedge y_t \neq y_j], & i = t \\
(1 - \eta) \alpha_i - \eta C y_t, & \forall i, z_i \in N_k^{\neg y_t}(z_t) \\
(1 - \eta) \alpha_i, & \text{otherwise}
\end{cases}
\]  
   (13)
KOIL: Update Budget

1. Remove SV via Reservoir Sampling (RS) or FIFO:

\[
\hat{f}_{t+1}(x) = f_{t+1}(x) - \alpha_r k(x_r, x)
\]  

(14)

2. Compensate the loss by adding \( \Delta \alpha_c \):

\[
f_{t+1}^{++}(x) = \hat{f}_{t+1}(x) + \Delta \alpha_c \cdot k(x_c, x)
\]

\[
= f_{t+1}(x) - \alpha_r k(x_r, x) + \Delta \alpha_c \cdot k(x_c, x)
\]

Removal \hspace{1cm} Compensation

(15)

3. By Eq. (15), we have

\[
\Delta \alpha_c = \alpha_r \frac{k(x_r, x)}{k(x_c, x)} \approx \alpha_r
\]  

(16)
Theoretical Analysis

**Lemma 1 (Norm of $f$)**

Suppose for all $x \in \mathbb{R}^d$, $k(x, x) \leq X^2$, where $X > 0$. Let $\xi_1$ be in $[0, X]$, such that $k(x_t, x_i) \geq \xi_1^2$, $\forall z_i = (x_i, y_i) \in N_t^{-y_t}(z_t)$. With $f_1 = 0$, we have

$$\|f_{t+1}\|_\mathcal{H} \leq Ck \sqrt{2X^2 - 2\xi_1^2}. \quad (17)$$

**Lemma 2 (pair-wise hinge loss bound)**

With the same assumption in Lemma 1 and the pair-wise hinge loss function $\ell : \mathcal{H} \times \mathcal{Z} \times \mathcal{Z} \rightarrow [0, U]$ defined by Eq. (9), we can determine the bound by

$$U = 1 + 2Ck(X^2 - \xi_1^2). \quad (18)$$
Theoretical Analysis

Theorem (Regret bound of KOIL)

Suppose for all $x \in \mathbb{R}^d$, $k(x, x) \leq X^2$, where $X > 0$. Let $\xi_1$ be in $[0, X]$, such that $k(x_t, x_i) \geq \xi_1^2$, $\forall z_i = (x_i, y_i) \in \mathcal{N}_{-yt}(z_t)$. Given $k > 0$, $C > 0$, $\eta > 0$ and a bounded convex loss function $\ell : \mathcal{H} \times \mathcal{Z} \times \mathcal{Z} \rightarrow [0, U]$ for $f_t$ updated by Eq. (12), with $f_1 = 0$, we have

$$R_T \leq \frac{\|f^*\|_H^2}{2\eta} + \eta C k \sum_{t=1}^{T} ((U-1)+(k+1)C(X^2-\xi_1^2)).$$

(19)

Moreover, assume that $\forall i \in I_t^+ \cup I_t^-$, $|\alpha_{i,t}| \in [0, \gamma \eta]$ and $k(x_r, x_c) \geq \xi_2^2$ with $0 < \xi_2 \leq X$ for any replaced support vector $x_r$ and compensated support vector $x_c$ at any trial. With $f_{1}^{++} = 0$ and $f_{t}^{++}$ updated by Eq. (15), we have

$$R_{T}^{++} \leq R_T + T \left(4\gamma C k \sqrt{(X^2-\xi_2^2)(X^2-\xi_1^2)+2\gamma^2(X^2-\xi_2^2)}\right).$$

(20)

Set $\eta$ to be $O\left(\frac{1}{\sqrt{T}}\right)$, $R_T \sim O(\sqrt{T})$, as tight as the standard regret bound.
1. All algorithms adopt the same setup.
2. the learning rate: $\eta = 0.01$
3. A 5-fold cross validation on the training data is applied to find the penalty cost $C \in 2^{-10:10}$.
4. For kernel-based methods, we use the Gaussian kernel and tune its parameter $\sigma \in 2^{-10:10}$ by a 5-fold cross validation on the training data.
Methods in Comparison

- “Perceptron”: the classical perceptron algorithm [Rosenblatt 1958];
- “OAM$_{seq}$”: an online linear AUC maximization algorithm [Zhao 2011];
- “OPAUC”: One-pass AUC maximization [Gao 2013];
- “NORMA”: online learning with kernels [Kivinen 2004];
- “RBP”: Randomized budget perceptron [Cavallanti 2007];
- “Forgetron”: a kernel-based perceptron on a fixed budget [Dekel 2008];
- “Projectron/Projectron++”: a bounded kernel-based perceptron [Orabona 2008];
- “KOIL$_{RS++}$”: our proposed kernelized online imbalanced learning algorithm with fixed budgets updated by RS++.
- “KOIL$_{FIFO++}$”: our proposed kernelized online imbalanced learning algorithm with fixed budgets updated by FIFO++.
## Benchmark Datasets

*Table*: Summary of the benchmark datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Samples</th>
<th>Dimensions</th>
<th>$T^- / T^+$</th>
</tr>
</thead>
<tbody>
<tr>
<td>sonar</td>
<td>208</td>
<td>60</td>
<td>1.144</td>
</tr>
<tr>
<td>australian</td>
<td>690</td>
<td>14</td>
<td>1.248</td>
</tr>
<tr>
<td>heart</td>
<td>270</td>
<td>13</td>
<td>1.250</td>
</tr>
<tr>
<td>ionosphere</td>
<td>351</td>
<td>34</td>
<td>1.786</td>
</tr>
<tr>
<td>diabetes</td>
<td>768</td>
<td>8</td>
<td>1.866</td>
</tr>
<tr>
<td>glass</td>
<td>214</td>
<td>9</td>
<td>2.057</td>
</tr>
<tr>
<td>german</td>
<td>1000</td>
<td>24</td>
<td>2.333</td>
</tr>
<tr>
<td>svmguide2</td>
<td>391</td>
<td>20</td>
<td>2.342</td>
</tr>
<tr>
<td>segment</td>
<td>2310</td>
<td>19</td>
<td>6.000</td>
</tr>
<tr>
<td>satimage</td>
<td>4435</td>
<td>36</td>
<td>9.687</td>
</tr>
<tr>
<td>vowel</td>
<td>528</td>
<td>10</td>
<td>10.000</td>
</tr>
<tr>
<td>letter</td>
<td>15000</td>
<td>16</td>
<td>26.881</td>
</tr>
<tr>
<td>poker</td>
<td>25010</td>
<td>10</td>
<td>47.752</td>
</tr>
<tr>
<td>shuttle</td>
<td>43500</td>
<td>9</td>
<td>328.546</td>
</tr>
</tbody>
</table>
## Table: Average AUC performance (mean±std) on the benchmark datasets, ●/○ (-) indicates that both/one of KOIL_RS++ and KOIL_FIFO++ are/is significantly better (worse) than the corresponding method (pairwise t-tests at 95% significance level).

<table>
<thead>
<tr>
<th>Data</th>
<th>KOIL_RS++</th>
<th>KOIL_FIFO++</th>
<th>Perceptron</th>
<th>OAM_seq</th>
<th>OPAUC</th>
<th>NORMA</th>
<th>RBP</th>
<th>Forgetron</th>
<th>Projectron</th>
<th>Projectron++</th>
</tr>
</thead>
<tbody>
<tr>
<td>sonar</td>
<td>.955±.028</td>
<td>.955±.028</td>
<td>.803±.083</td>
<td>.843±.056</td>
<td>.844±.077</td>
<td>.925±.044</td>
<td>.913±.032</td>
<td>.896±.054</td>
<td>.896±.049</td>
<td>.896±.049</td>
</tr>
<tr>
<td>australian</td>
<td>.923±.023</td>
<td>.922±.026</td>
<td>.869±.035</td>
<td>.925±.024</td>
<td>.923±.025</td>
<td>.919±.023</td>
<td>.911±.017</td>
<td>.912±.026</td>
<td>.923±.024</td>
<td>.923±.024</td>
</tr>
<tr>
<td>heart</td>
<td>.908±.040</td>
<td>.910±.040</td>
<td>.876±.066</td>
<td>.912±.040</td>
<td>.901±.043</td>
<td>.890±.051</td>
<td>.865±.043</td>
<td>.900±.053</td>
<td>.902±.038</td>
<td>.905±.042</td>
</tr>
<tr>
<td>ionosphere</td>
<td>.985±.015</td>
<td>.985±.015</td>
<td>.851±.056</td>
<td>.905±.041</td>
<td>.888±.046</td>
<td>.961±.016</td>
<td>.960±.030</td>
<td>.945±.031</td>
<td>.964±.025</td>
<td>.963±.027</td>
</tr>
<tr>
<td>diabetes</td>
<td>.826±.036</td>
<td>.830±.030</td>
<td>.726±.059</td>
<td>.827±.033</td>
<td>.805±.035</td>
<td>.792±.032</td>
<td>.828±.034</td>
<td>.820±.027</td>
<td>.832±.033</td>
<td>.833±.033</td>
</tr>
<tr>
<td>glass</td>
<td>.887±.053</td>
<td>.884±.054</td>
<td>.810±.065</td>
<td>.827±.064</td>
<td>.800±.074</td>
<td>.811±.077</td>
<td>.811±.071</td>
<td>.813±.075</td>
<td>.811±.070</td>
<td>.781±.076</td>
</tr>
<tr>
<td>german</td>
<td>.769±.032</td>
<td>.778±.031</td>
<td>.748±.033</td>
<td>.777±.027</td>
<td>.787±.026</td>
<td>.766±.032</td>
<td>.699±.038</td>
<td>.712±.054</td>
<td>.769±.028</td>
<td>.770±.024</td>
</tr>
<tr>
<td>svmguide2</td>
<td>.897±.040</td>
<td>.885±.043</td>
<td>.860±.037</td>
<td>.886±.045</td>
<td>.859±.050</td>
<td>.865±.046</td>
<td>.890±.038</td>
<td>.864±.045</td>
<td>.886±.044</td>
<td>.886±.045</td>
</tr>
<tr>
<td>segment</td>
<td>.983±.008</td>
<td>.985±.012</td>
<td>.875±.020</td>
<td>.919±.020</td>
<td>.882±.019</td>
<td>.910±.042</td>
<td>.969±.017</td>
<td>.943±.038</td>
<td>.979±.013</td>
<td>.978±.016</td>
</tr>
<tr>
<td>satimage</td>
<td>.924±.012</td>
<td>.923±.015</td>
<td>.700±.015</td>
<td>.755±.018</td>
<td>.724±.016</td>
<td>.914±.025</td>
<td>.899±.018</td>
<td>.892±.032</td>
<td>.910±.015</td>
<td>.904±.011</td>
</tr>
<tr>
<td>vowel</td>
<td>1.000±.000</td>
<td>1.000±.001</td>
<td>.848±.070</td>
<td>.905±.024</td>
<td>.885±.034</td>
<td>.996±.005</td>
<td>.968±.017</td>
<td>.987±.027</td>
<td>.982±.013</td>
<td>.994±.019</td>
</tr>
<tr>
<td>letter</td>
<td>933±.021</td>
<td>942±.017</td>
<td>.767±.029</td>
<td>.827±.021</td>
<td>.823±.018</td>
<td>.910±.027</td>
<td>.928±.011</td>
<td>.815±.102</td>
<td>.926±.016</td>
<td>.926±.015</td>
</tr>
<tr>
<td>poker</td>
<td>.681±.031</td>
<td>.693±.032</td>
<td>.514±.030</td>
<td>.503±.024</td>
<td>.509±.031</td>
<td>.577±.040</td>
<td>.501±.031</td>
<td>.572±.029</td>
<td>.675±.027</td>
<td>.675±.027</td>
</tr>
<tr>
<td>shuttle</td>
<td>.950±.040</td>
<td>.956±.021</td>
<td>.520±.134</td>
<td>.999±.000</td>
<td>.754±.043</td>
<td>.725±.053</td>
<td>.844±.041</td>
<td>.839±.060</td>
<td>.873±.063</td>
<td>.795±.063</td>
</tr>
</tbody>
</table>

| win/tie/loss | 14/0/0 | 9/4/1 | 12/1/1 | 13/1/0 | 12/2/0 | 13/1/0 | 11/3/0 | 10/4/0 |
1. RS/FIFO ↓ when the budget is full
2. RS++/FIFO++ approximate KOIL without removing SVs.

**Figure**: Average AUC performance of KOIL.
Experiment: Effect of Buffer Size

1. Stay unchanged when buffer size is large enough.
2. KOIL cannot learn well when buffer size is extremely small.

Figure: Average AUC of KOIL for buffer sizes.
Experiment: Effect of $k$

1. For noisy dataset, set $k$ small to avoid global effect
2. $k$ extremely small, KOIL cannot learn enough knowledge.

Figure: Average AUC of KOIL with different $k$. 
In this talk, we introduced the KOIL algorithm, which has the following properties:

1. AUC maximization for streaming data
2. Two fixed-size buffers
3. $k$-Nearest Neighbors to reduce the effect of noisy data
4. Loss compensation for support vector replacement in the buffers
5. Regret bound for KOIL and two lemmas
Thanks!


