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-Cho So[3]

available on `http://appsrv.cse.cuhk.edu.hk/~jjhu/koil/KOIL_slide.pdf`

[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

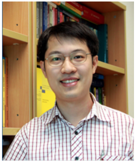
1. Web Intelligence and Social Computing Lab at CUHK (WISC Lab)
2. Shenzhen Key Laboratory in Shenzhen Research Institute at CUHK

(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

Online Learning

1. Definition of Online learning
    - learn from the streaming data
    - update the model adaptively from the data stream
2. Properties
    - process the data one by one
    - update the model in each iteration
    - approximate the learning performance of the batch-train methods

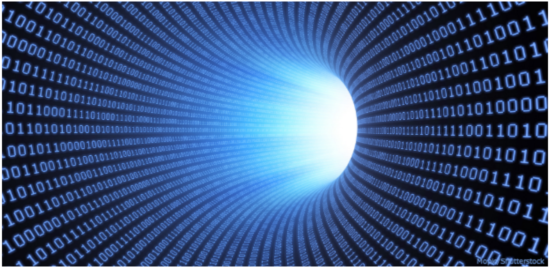


Figure : Rutrell Yasin, Amazon Kinesis does heavy-lifting on streaming, big data

Imbalanced Data & Cost-sensitive Learning

1. Properties:
   - uneven data distribution
   - No. of samples in one class $<$ No. of samples in the other class
2. Problems:
   - Accuracy: inappropriate
   - Misclassification costs for possitive and negative samples are not the same.

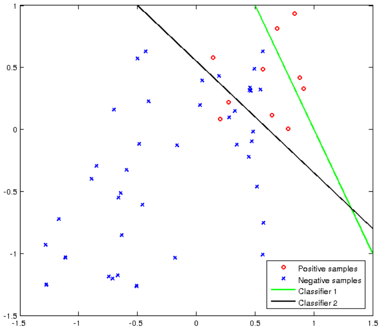
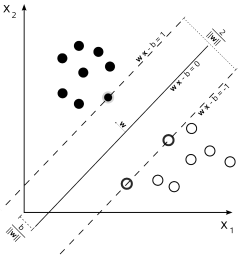


Figure : Imbalanced data

Support Vector Machine [Cortes 1995]

1. SVM maps the instance $\mathbf{x}$ to the Reproducing Kernel Hilbert Space
$$\phi : \mathbf{x} \longmapsto \phi(\mathbf{x})$$
.
2. In RKHS, dot product of two elements:
$$\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_{\mathcal{H}} = k(\mathbf{x}_i, \mathbf{x}_j)$$
3. The objective of SVM is to maximize the margins of the hyperplane in RKHS.



KOIL: Definitions & Notations

1. non-linear decision function $f : \mathbb{R}^d \to \mathbb{R}$
2. a sequence of imbalanced feature-labeled pair instances $\{\mathbf{z}_t = (\mathbf{x}_t, y_t) \in \mathcal{Z}, t \in [T]\}$, where $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, $\mathbf{x}_t \in \mathcal{X} \subseteq \mathbb{R}^d$, $y_t \in \mathcal{Y} = \{-1, +1\}$ and $[T] = \{1, \ldots, T\}$.
3. $f(\mathbf{x})$ can be calculated by

$$\langle f(\cdot), k(\mathbf{x}, \cdot)\rangle_{\mathcal{H}} = f(\mathbf{x}). \tag{1}$$

4. Assumption: positive class (minority) & negative class (majority)
5. $N_k^{\tilde{y}}(\mathbf{z})$: the set of the k-nearest neighbors of $\mathbf{z}$ and have the label of $\tilde{y}$.

Related Work

1. Online Learning with Kernels: minimize the hinge loss function

$$\min_{f} \ell_h(f, \mathbf{x}, y) := \max(0, 1 - yf(\mathbf{x})) \quad (2)$$

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

2. Online Linear AUC Maximization: minimize the AUC-based loss function
   - Online AUC Maximization (OAM) [Zhao 2011]

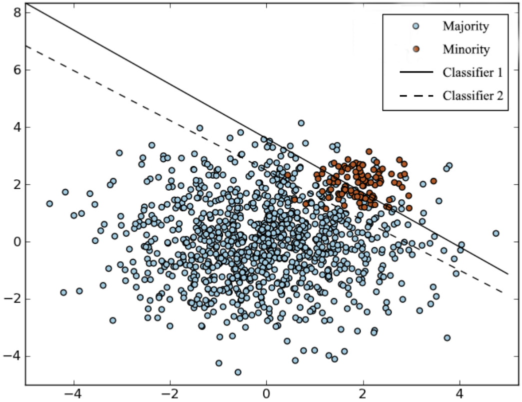
$$\min_{w} \ell_h(w, \mathbf{x}^+, \mathbf{x}^-) := \max(0, 1 - w \cdot (\mathbf{x}^+ - \mathbf{x}^-)) \quad (3)$$

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

$$\min_{w} \ell_h(w, \mathbf{x}^+, \mathbf{x}^-) := (1 - w \cdot (\mathbf{x}^+ - \mathbf{x}^-))^2 \quad (4)$$

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?



KOIL: Formulation

1. $\mathcal{K}^+$ and $\mathcal{K}^-$: the information of positive and negative SVs respectively, where $|B^+| = |B^-|$.

$$\mathcal{K}^+.\mathcal{A} := \{\alpha_i^+\}_{i=1}^{|B^+|}, \quad \mathcal{K}^+.\mathcal{B} := \{\mathbf{z}_i \,|\, y_i = +1\}_{i=1}^{|B^+|} \tag{5}$$

$$\mathcal{K}^-.\mathcal{A} := \{\alpha_i^-\}_{i=1}^{|B^-|}, \quad \mathcal{K}^-.\mathcal{B} := \{\mathbf{z}_i \,|\, y_i = -1\}_{i=1}^{|B^-|}. \tag{6}$$

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

$$f(\mathbf{x}) = \sum_{\substack{\alpha_i^+ \in \mathcal{K}^+.\mathcal{A} \\ \mathbf{x}_i^+ \in \mathcal{K}^+.\mathcal{B}}} \alpha_i^+ k(\mathbf{x}_i^+, \mathbf{x}) + \sum_{\substack{\alpha_j^- \in \mathcal{K}^-.\mathcal{A} \\ \mathbf{x}_j^- \in \mathcal{K}^-.\mathcal{B}}} \alpha_j^- k(\mathbf{x}_j^-, \mathbf{x}), \tag{7}$$

KOIL: AUC Optimization

1. Given the positive dataset $D^+ = \{\mathbf{z}_i | y_i = +1\}$ and the negative dataset $D^- = \{\mathbf{z}_j | y_j = -1\}$, the AUC is measured as:

$$AUC(f) = \frac{\sum_{i=1}^{|D^+|} \sum_{j=1}^{|D^-|} \mathbb{I}[f(\mathbf{x}_i^+) - f(\mathbf{x}_j^-) > 0]}{|D^+||D^-|} \quad (8)$$

$$= 1 - \frac{\sum_{i=1}^{|D^+|} \sum_{j=1}^{|D^-|} \mathbb{I}[f(\mathbf{x}_i^+) - f(\mathbf{x}_j^-) \leq 0]}{|D^+||D^-|}$$

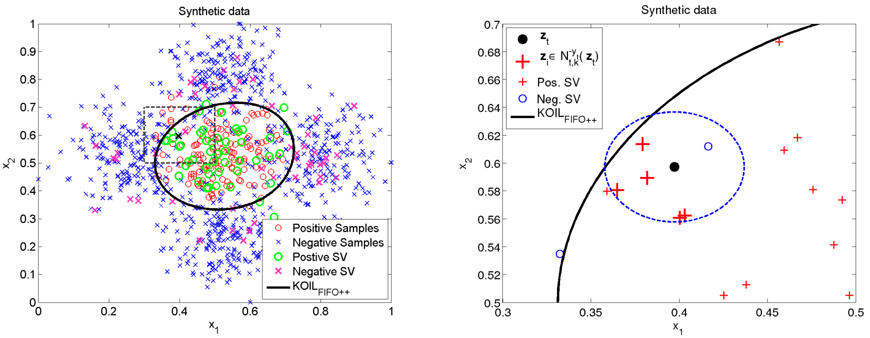
   where $\mathbb{I}[\pi]$ is the indicator function.

2. Maximizing AUC equals to minimizing $\sum_{i=1}^{|D^+|} \sum_{j=1}^{|D^-|} \mathbb{I}[f(\mathbf{x}_i^+) - f(\mathbf{x}_j^-) \leq 0]$

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

$$\ell_h(f, \mathbf{z}, \mathbf{z}') := \frac{|y - y'|}{2} \left[1 - \frac{1}{2}(y - y')(f(\mathbf{x}) - f(\mathbf{x}'))\right]_+ \quad (9)$$

KOIL: Intuition



(a) (b)

1. Assign an initial weight to $\mathbf{z}_t$
2. Update the weight of SVs, which are KNN of $\mathbf{z}_t$ and have the opposite label $-y_t$.
3. does not affect the weight of SVs in the whole buffer

KOIL: Intuition – Update Kernel

Notation:

$\mathbf{z}_t^- := (\mathbf{x}_t, -1)$

$\mathbf{z}_t^+ := (\mathbf{x}_t, +1)$

Objective function:

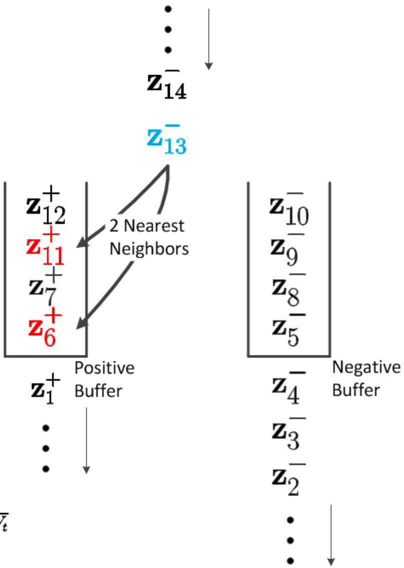
$$\hat{\mathcal{L}}(f, \mathbf{z}_t) = \frac{1}{2}\|f\|_{\mathcal{H}}^2 + C \sum_{\mathbf{z}_i \in N_k^{-y_t}(\mathbf{z}_t)} \ell_h(f, \mathbf{z}_t, \mathbf{z}_i)$$

Update Decision Function:

$$f_{t+1} := f_t - \eta \partial_f \hat{\mathcal{L}}(f, \mathbf{z}_t)|_{f=f_t}$$

Update Weight:

$$\alpha_{i,t} = \begin{cases} \eta C y_t |V_t|, & i = t \\ (1-\eta)\alpha_{i,t-1} - \eta C y_t, & \forall i \in V_t \\ (1-\eta)\alpha_{i,t-1}, & \forall i \in I_t^{y_t} \cup \overline{V_t} \end{cases}$$



KOIL: Problem for online learning with kernel

1. What if the fixed-size buffers are full?

KOIL: Problem for online learning with kernel

1. What if the fixed-size buffers are full?
    1. Reservoir Samping (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 Samping (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

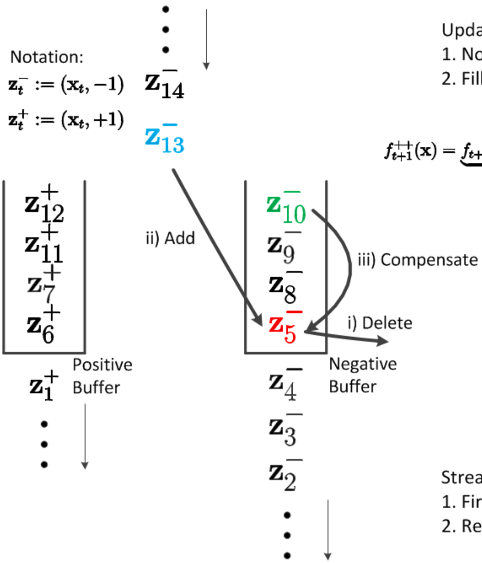
1. What if the fixed-size buffers are full?
   1. Reservoir Samping (RS)
   2. First-In-First-Out (FIFO)
2. What if we directly remove the SV from the buffer?
   1. information loss
   2. compensation scheme for information loss

KOIL: Intuition – Update Buffers

Notation:

$\mathbf{z}_t^- := (\mathbf{x}_t, -1)$

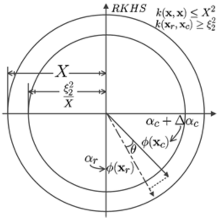
$\mathbf{z}_t^+ := (\mathbf{x}_t, +1)$



Update Buffers:

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

$$f_{t+1}^{++}(\mathbf{x}) = \underbrace{f_{t+1}(\mathbf{x}) - \alpha_r k(\mathbf{x}_r, \mathbf{x})}_{\text{Removal}} \underbrace{+\Delta\alpha_c \cdot k(\mathbf{x}_c, \mathbf{x})}_{\text{Compensation}}$$



Stream oblivious policies:

1. First-In-First-Out (FIFO)
2. Reservoir Sampling (RS)

KOIL: Update Kernel

1. Minimize the *instantaneous regularized risk of AUC*.

$$\min_f \mathcal{L}(f_t, \mathbf{z}_t) = \frac{1}{2}\|f_t\|^2_{\mathcal{H}} + C\sum_{i=1}^{t-1} \ell_h(f_t, \mathbf{z}_t, \mathbf{z}_i) \tag{10}$$

2. Minimize the *localized instantaneous regularized risk of AUC* (Reduce the effect of outliers):

$$\min_f \hat{\mathcal{L}}(f_t, \mathbf{z}_t) = \frac{1}{2}\|f_t\|^2_{\mathcal{H}} + C\sum_{\mathbf{z}_i \in N_k^{-y_t}(\mathbf{z}_t)} \ell_h(f_t, \mathbf{z}_t, \mathbf{z}_i) \tag{11}$$

3. Stochastic Gradient Descent: update f_t in each iteration

$$f_{t+1} := f_t - \eta \partial_f \hat{\mathcal{L}}(f, \mathbf{z}_t)|_{f=f_t} \tag{12}$$

4. Updating rule for the kernel weights:

$$\alpha_i = \begin{cases} \eta C y_t \sum_{\mathbf{z}_j \in N_k^{-y_t}(\mathbf{z}_t)} \mathbb{I}[\phi(\mathbf{z}_t, \mathbf{z}_j) < 1 \wedge y_t \neq y_j], & i = t \\ (1-\eta)\alpha_i - \eta C y_t, & \forall i, \mathbf{z}_i \in N_k^{-y_t}(\mathbf{z}_t) \\ (1-\eta)\alpha_i, & \text{otherwise} \end{cases} \tag{13}$$

KOIL: Update Budget

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

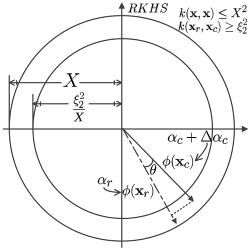
$$\hat{f}_{t+1}(\mathbf{x}) = f_{t+1}(\mathbf{x}) - \alpha_r k(\mathbf{x}_r, \mathbf{x}) \tag{14}$$

2. Compensate the loss by adding $\Delta\alpha_c$:

$$\begin{aligned} f_{t+1}^{++}(\mathbf{x}) &= \hat{f}_{t+1}(\mathbf{x}) + \Delta\alpha_c \cdot k(\mathbf{x}_c, \mathbf{x}) \\ &= \underbrace{f_{t+1}(\mathbf{x}) - \alpha_r k(\mathbf{x}_r, \mathbf{x})}_{\text{Removal}} \underbrace{+\Delta\alpha_c \cdot k(\mathbf{x}_c, \mathbf{x})}_{\text{Compensation}} \end{aligned} \tag{15}$$

3. By Eq. (15), we have

$$\Delta\alpha_c = \alpha_r \frac{k(\mathbf{x}_r, \mathbf{x})}{k(\mathbf{x}_c, \mathbf{x})} \approx \alpha_r. \tag{16}$$



Theoretical Analysis

Lemma 1 (Norm of f)

Suppose for all $\mathbf{x} \in \mathbb{R}^d$, $k(\mathbf{x}, \mathbf{x}) \leq X^2$, where $X > 0$. Let ξ_1 be in $[0, X]$, such that $k(\mathbf{x}_t, \mathbf{x}_i) \geq \xi_1^2$, $\forall \mathbf{z}_i = (\mathbf{x}_i, y_i) \in N_t^{-y_t}(\mathbf{z}_t)$. 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 $\ell : \mathcal{H} \times \mathcal{Z} \times \mathcal{Z} \to [0, U]$ defined by Eq. (9), we can determine the bound by

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

Theorem (Regret bound of KOIL)

Suppose for all $\mathbf{x} \in \mathbb{R}^d$, $k(\mathbf{x}, \mathbf{x}) \leq X^2$, *where* $X > 0$. *Let* ξ_1 *be in* $[0, X]$, *such that* $k(\mathbf{x}_t, \mathbf{x}_i) \geq \xi_1^2$, $\forall\, \mathbf{z}_i = (\mathbf{x}_i, y_i) \in N_t^{-y_t}(\mathbf{z}_t)$. *Given* $k > 0, C > 0, \eta > 0$ *and a bounded convex loss function* $\ell : \mathcal{H} \times \mathcal{Z} \times \mathcal{Z} \to [0, U]$ *for* f_t *updated by Eq. (12), with* $f_1 = 0$, *we have*

$$R_T \leq \frac{\|f^*\|_{\mathcal{H}}^2}{2\eta} + \eta C k \sum_{t=1}^{T} \Big((U-1) + (k+1) C (X^2 - \xi_1^2) \Big). \tag{19}$$

Moreover, assume that $\forall i \in I_t^+ \cup I_t^-$, $|\alpha_{i,t}| \in [0, \gamma\eta]$ *and* $k(\mathbf{x}_r, \mathbf{x}_c) \geq \xi_2^2$ *with* $0 < \xi_2 \leq X$ *for any replaced support vector* $\mathbf{x}_r$ *and compensated support vector* $\mathbf{x}_c$ *at any trial. With* $f_1^{++} = 0$ *and* f_t^{++} *updated by Eq. (15), we have*

$$R_T^{++} \leq R_T + T\left(4\gamma C k \sqrt{(X^2 - \xi_2^2)(X^2 - \xi_1^2)} + 2\gamma^2 (X^2 - \xi_2^2) \right). \tag{20}$$

Set η to be $O(\frac{1}{\sqrt{T}})$, $R_T \sim O(\sqrt{T})$, as tight as the standard regret bound.

Experiment Setup

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];
- "$\text{KOIL}_{\text{RS}++}$": our proposed kernelized online imbalanced learning algorithm with fixed budgets updated by RS++.
- "$\text{KOIL}_{\text{FIFO}++}$": our proposed kernelized online imbalanced learning algorithm with fixed budgets updated by FIFO++.

Benchmark Datasets

Table : Summary of the benchmark datasets.

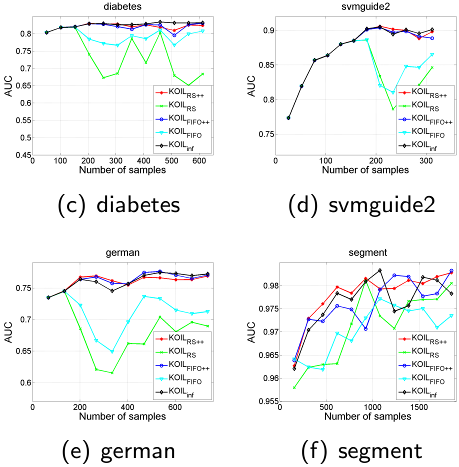
Dataset	Samples	Dimensions	T^{-}/T^{+}
sonar	208	60	1.144
australian	690	14	1.248
heart	270	13	1.250
ionosphere	351	34	1.786
diabetes	768	8	1.866
glass	214	9	2.057
german	1000	24	2.333
svmguide2	391	20	2.342
segment	2310	19	6.000
satimage	4435	36	9.687
vowel	528	10	10.000
letter	15000	16	26.881
poker	25010	10	47.752
shuttle	43500	9	328.546

AUC Measure on Benchmark Dataset

Table : Average AUC performance (mean±std) on the benchmark datasets, •/○ (-) indicates that both/one of $\text{KOIL}_{\text{RS}++}$ and $\text{KOIL}_{\text{FIFO}++}$ are/is significantly better (worse) than the corresponding method (pairwise *t*-tests at 95% significance level).

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



(c) diabetes (d) svmguide2

(e) german (f) segment

Figure : Average AUC performance of KOIL.

Experiment: Effect of Buffer Size

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

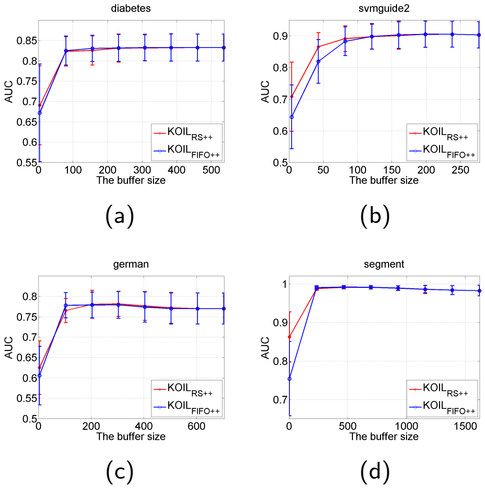


Figure : Average AUC of KOIL for buffer sizes.

Experiment: Effect of k

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

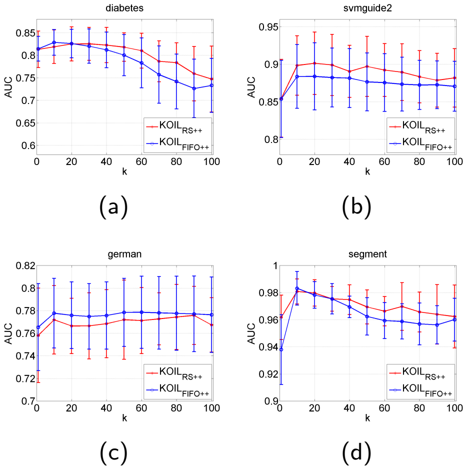


Figure : Average AUC of KOIL with different k.

Conclusion

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
6. Experiments on benchmark and synthetic datasets.

Thanks!

References

Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks." Machine learning 20.3 (1995): 273-297.

Kivinen, Jyrki, Alexander J. Smola, and Robert C. Williamson. "Online learning with kernels." IEEE Transactions on Signal Processing, 52.8 (2004): 2165-2176.

Cavallanti, Giovanni, Nicol Cesa-Bianchi, and Claudio Gentile. "Tracking the best hyperplane with a simple budget perceptron." Machine Learning 69.2-3 (2007): 143-167.

Dekel, Ofer, Shai Shalev-Shwartz, and Yoram Singer. "The forgetron: A kernel-based perceptron on a budget." SIAM Journal on Computing 37.5 (2008): 1342-1372.

Orabona, Francesco, Joseph Keshet, and Barbara Caputo. "The projectron: a bounded kernel-based perceptron." Proceedings of the 25th international conference on Machine learning. ACM, 2008.

Zhao, Peilin, Rong Jin, Tianbao Yang, and Steven C. Hoi. "Online AUC maximization." In Proceedings of the 28th International Conference on Machine Learning (ICML-11), pp. 233-240. 2011.

Gao, Wei, Rong Jin, Shenghuo Zhu, and Zhi-Hua Zhou. "One-Pass AUC Optimization." arXiv preprint arXiv:1305.1363 (2013).

Rosenblatt, Frank. "The perceptron: a probabilistic model for information storage and organization in the brain." Psychological review 65.6 (1958): 386.