IEEE ICDM Brussels 10-13 Dec © 2012

Healing Truncation Bias : Self-weighted Truncation framework for Dual Averaging

* Hidekazu Oiwa

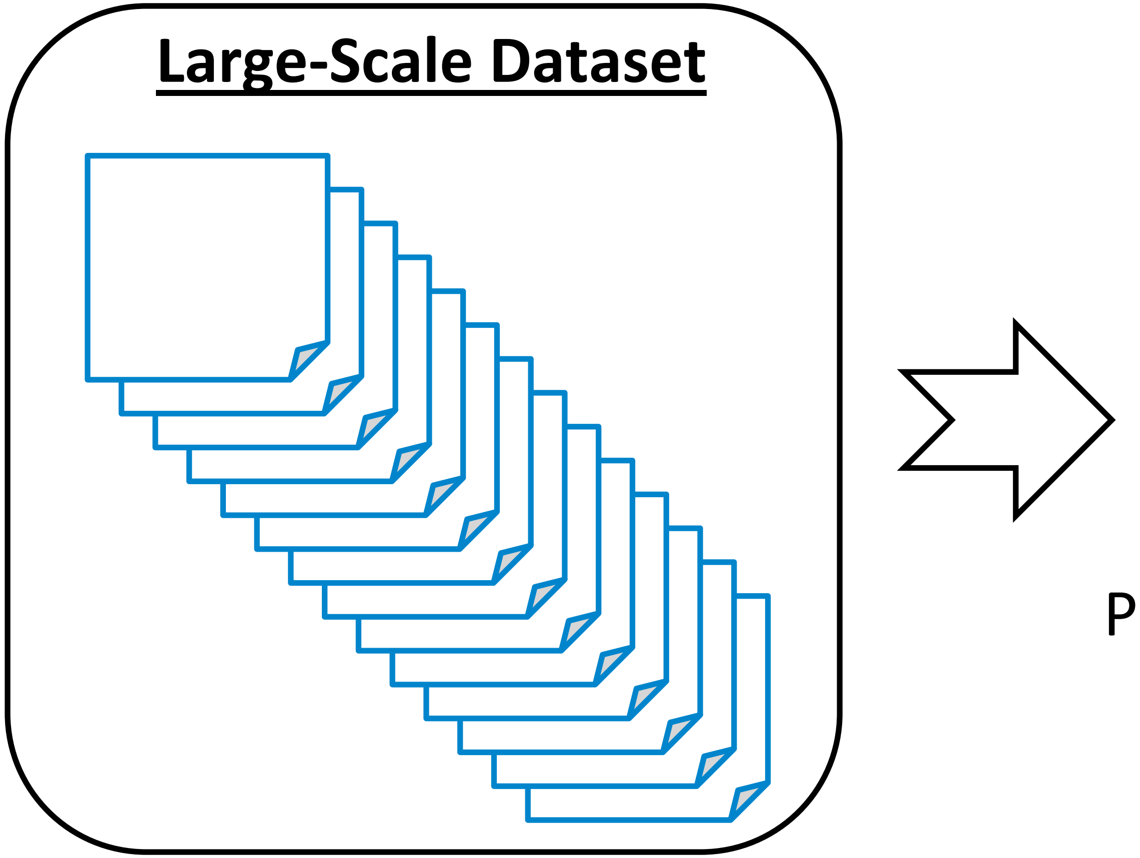
Shin Matsushima

Hiroshi Nakagawa

The University of Tokyo

Objective : Large-Scale Learning

- Learn parameters $\mathbf{w}$ from large-scale dataset
  - Predict <u>Output</u> y from <u>Input</u> $\mathbf{x}$ by $\langle \mathbf{w}, \mathbf{x} \rangle$
  - Assume data size / dim. are very large



<u>Linear Predictor</u>

$\mathbf{w}$

Predict y by $\langle \mathbf{w}, \mathbf{x} \rangle$

Optimization Problem

Empirical Risk Minimization

$$\mathbf{w}^* = \underset{\mathbf{w} \in \mathcal{W}}{\operatorname{argmin}} \sum_{t=1}^{T} \ell_t(\mathbf{w})$$

Convex Loss function

$$\ell_t(\cdot) : \mathbf{W} \rightarrow \Re_+$$

Evaluate predictability

Ex. Hinge Loss

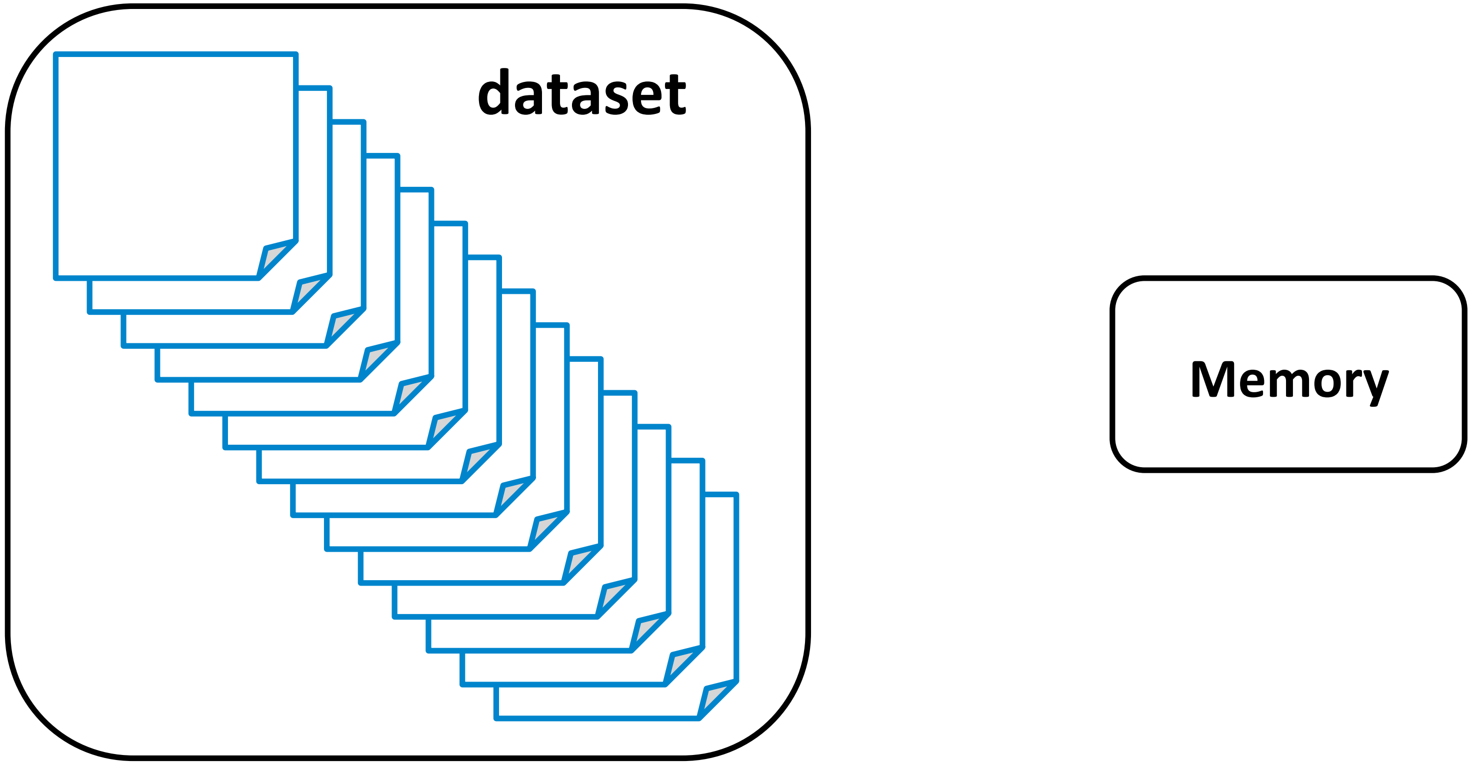
$$\ell_t(\mathbf{w}) = [1 - y_t \langle \mathbf{x}_t, \mathbf{w} \rangle]_+$$

Log-Loss

$$\ell_t(\mathbf{w}) = \log(1 + e^{-y_t \langle \mathbf{x}_t, \mathbf{w} \rangle})$$

2 Challenges in large-scale learning

Large-Scale Learning : Challenge 1



Data Size >>> Memory Size

Data loading time may be dominant in classical optimization methods

[Yu+, 2010]

Large-Scale Learning : Challenge 2

$$\mathbf{w} = \{2.5, 1.2, -1.1, \ldots, \ldots, \ldots, \ldots, \ldots, \ldots, \ldots, \ldots, \ldots, \ldots\}$$

$$\langle \mathbf{w}, \mathbf{x}_i \rangle$$

Dimension is large

$$\mathbf{x}_i = \{0, 2, 1, \ldots, \ldots, \ldots, \ldots, \ldots, \ldots, \ldots, \ldots, \ldots, \ldots\}$$

Inner-product calculation becomes very costly

$\langle \mathbf{w}, \mathbf{x} \rangle$ ⇐ Make inner-product faster!

So…, Sparse Online Learning!

- Sparse Online Learning is a combination of Online Learning and L1-Regularization

Online Learning

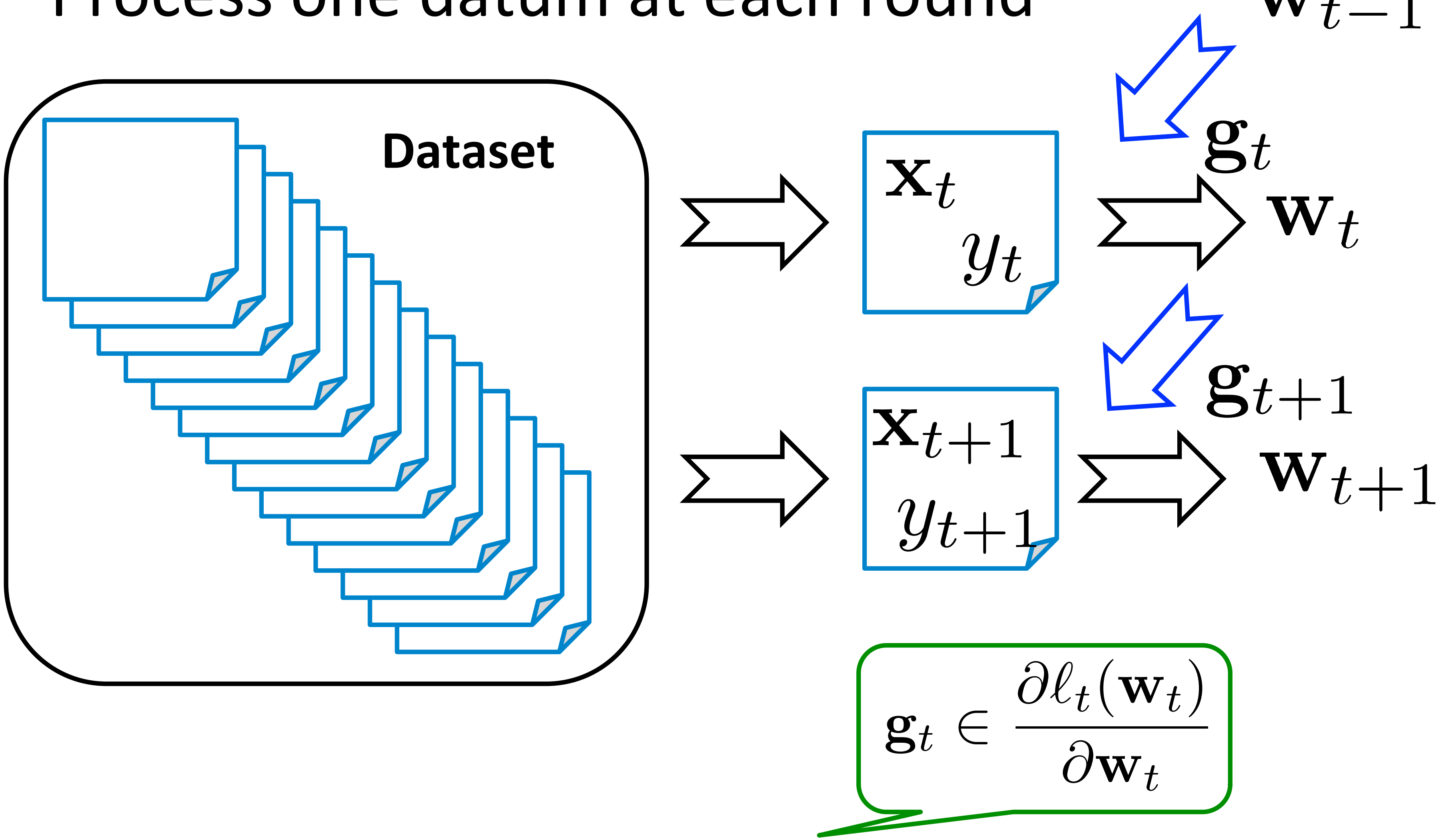
- Smaller data-loading count
- Robust for data redundancy

L1-Regularization

- Faster inner-product
- Robust for feature redundancy

Online Learning

- Process one datum at each round



First-order derivative of convex loss functions is used

L1-regularization

- Sparsify weight vector
  - Component is truncated if not helpful for prediction
- Formulation

$\Phi(\mathbf{w}) = \lambda ||\mathbf{w}||_1$ where λ : parameter interpolating losses and L1

$$\mathbf{w} = (2.5, 1.2, -1.1, 0.8, 0.1, \ldots, \ldots, \ldots, \ldots$$

$$\mathbf{w} = (1.5, 0.2, -0.1, 0.0, 0.0, \ldots, \ldots, \ldots, \ldots$$

Truncated components are not used

=> Faster Prediction and Reduce redundant features

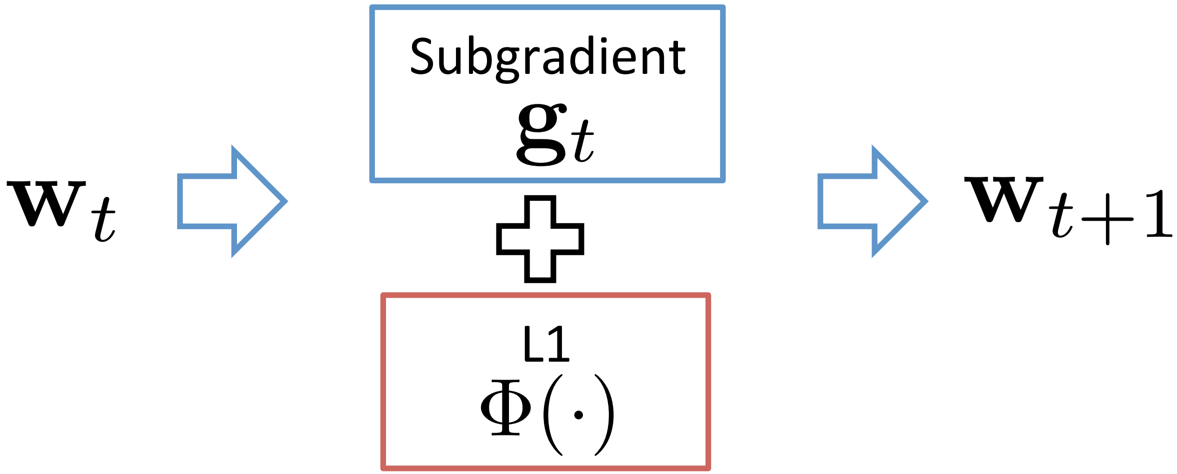
Previous Work

Sparse Online Learning

- RDA [Xiao, 2009]
- COMID [Duchi+, 2010]
- FTPRL [McMahan+, 2010]

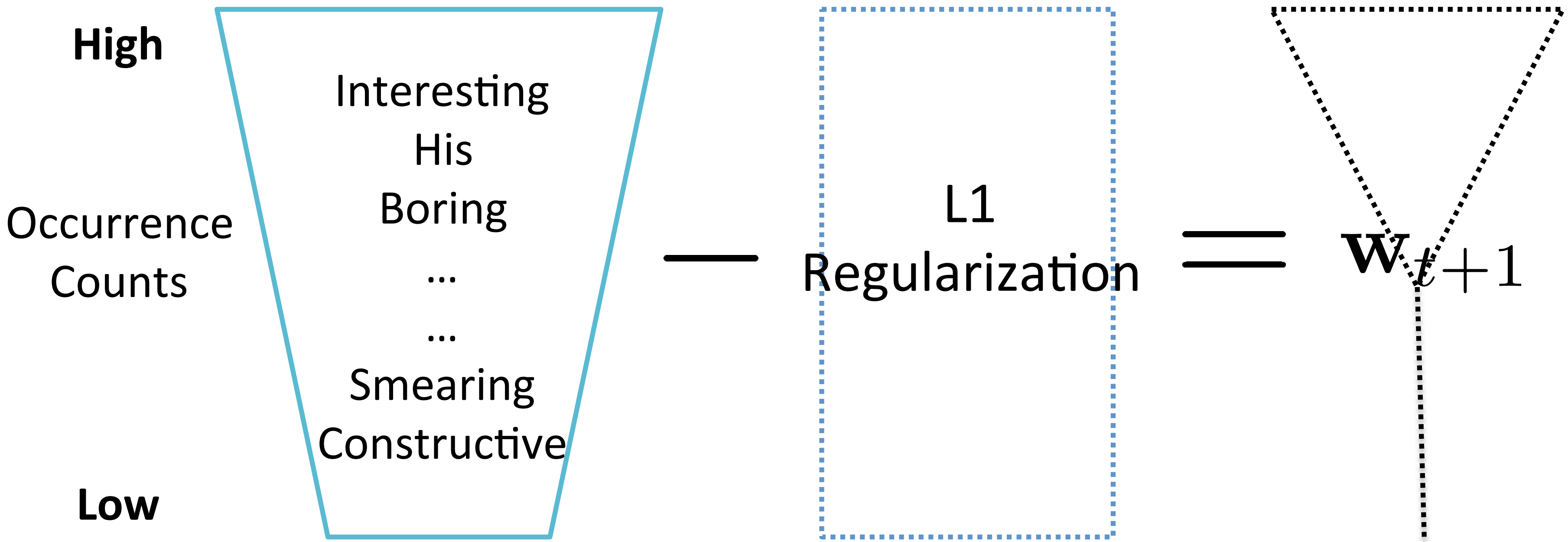
RDA is a state-of-the-art framework.

(In our experiments, RDA outperforms other methods)



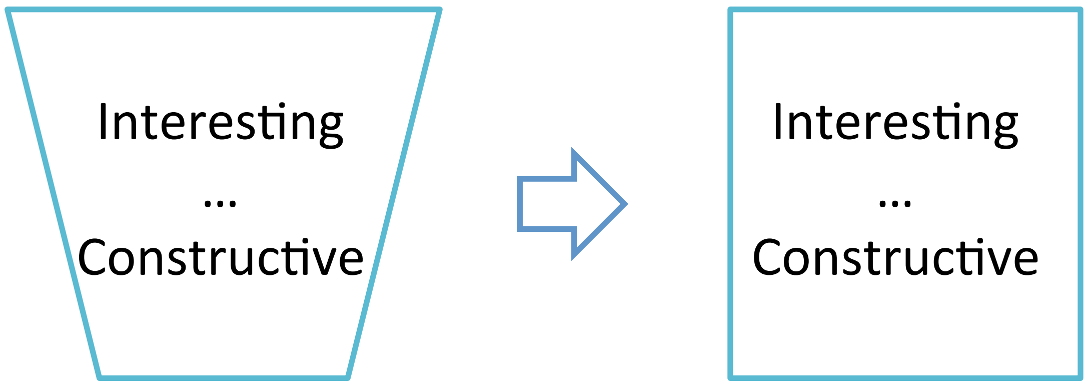
Truncation Bias

- Heterogeneity among features makes bias
  - Truncation ignores feature info.
  - Crucial features are truncated if
    - low-frequency
    - Small value range



Truncation Bias in Online Learning make the problem more complex

- Truncation Bias in Batch Learning
  - Scaling each feature by scanning all data once

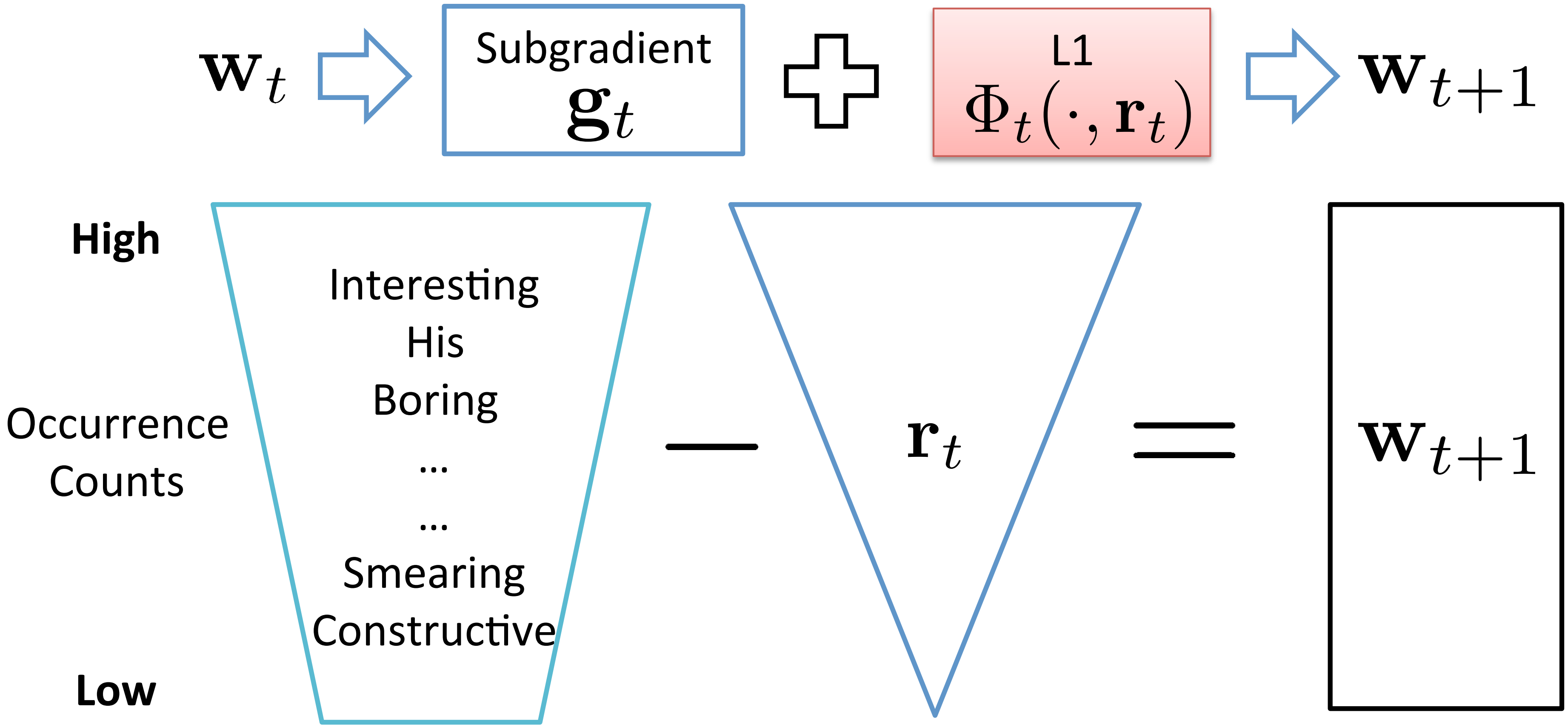


- Truncation Bias in Online Learning
  - Cannot scan all data, cannot count occurrences of features
  - Dynamic scaling leads to inconsistency prediction
    - If weight vector and input are the same, $\langle \mathbf{w}, g_i(\mathbf{x}) \rangle \neq \langle \mathbf{w}, g_j(\mathbf{x}) \rangle$

Our Approach [1/2]

Self-weighted Truncation framework for RDA

- Introduce self-weighted vector $\mathbf{r}_t$
  - Integrate $\mathbf{r}_t$ for healing truncation bias

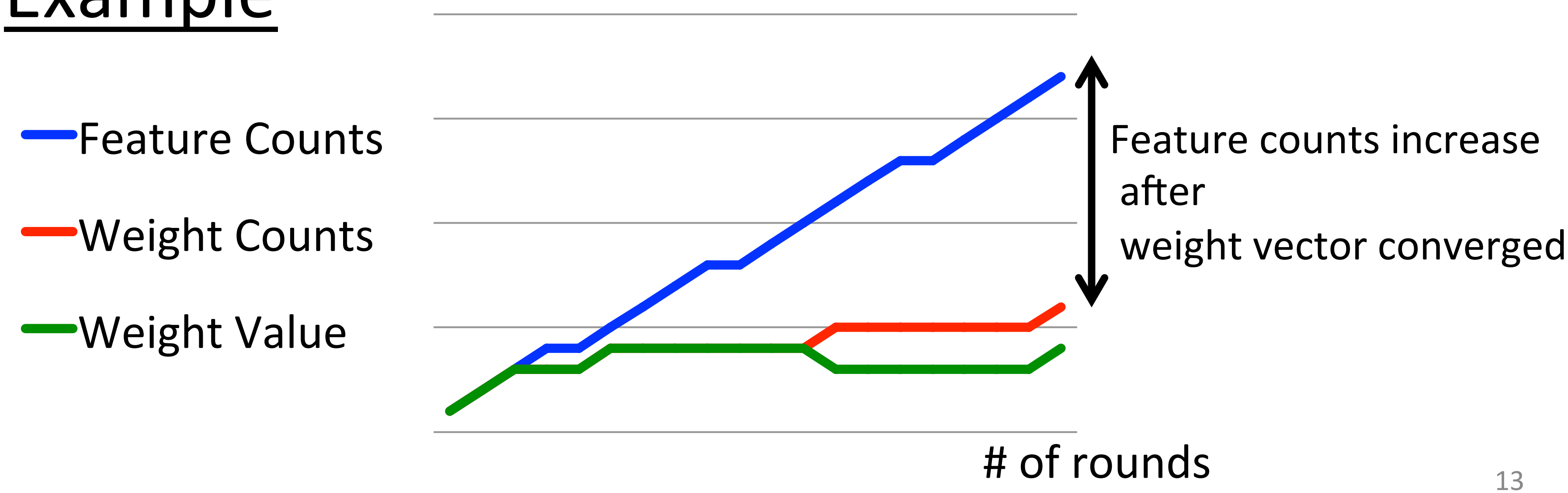


Our Approach [2/2]

Self-weighted Truncation framework for RDA

- $\mathbf{r}_t$ is based on Subgradient not original feature
  - Collecting feature info. is not good approach!
  - Value range of $\mathbf{w}_t$ depends more on update frequency than on feature counts

Example

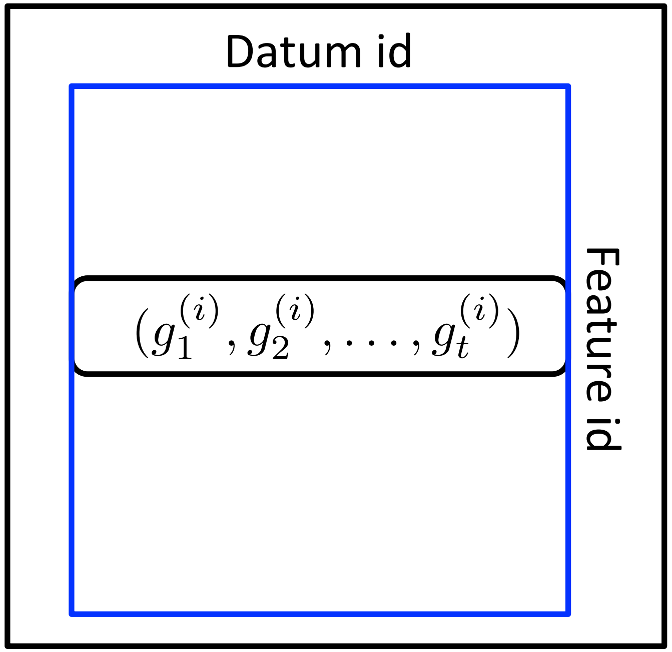


Self-weighted Truncation framework [1/2]

Define $\mathbf{r}_t$

$$r_t^{(i)} = r_{t,q}^{(i)} = \sqrt[q]{\sum_{\tau=1}^{t} |g_\tau^{(i)}|^q}$$

where $q > 0$



Update frequency of feature i is low ⇨ Few number of nonzero components $(g_1^{(i)}, g_2^{(i)}, \ldots, g_t^{(i)})$ ⇨ $r_t^{(i)}$ becomes small

Computational complexity of updating $\mathbf{r}_t$: $O($*Nonzero elements of* $\mathbf{g}_t)$

Self-weighted Truncation framework [2/2]

Reformulate L1-regularization

$$\Phi_t(\mathbf{w}_t) = \lambda \|\mathbf{R}_t \mathbf{w}_t\|_1$$

$$s.t. \quad \mathbf{R}_t = \begin{bmatrix} r_t^{(1)} & 0 & \cdots & 0 \\ 0 & r_t^{(2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & r_t^{(d)} \end{bmatrix}$$

Adaptive Truncation via Update Frequency

Algorithm : Extension to RDA (STDA)

$$w_{t+1}^{(i)} = \begin{cases} 0 & v_t^{(i)} \leq 0 \\ -\text{sign}(\bar{g}_t^{(i)}) \dfrac{t v_t^{(i)}}{\beta_t} & \text{otherwise} \end{cases} \qquad v_t^{(i)} = |\bar{g}_t^{(i)}| - \lambda \bar{r}_t^{(i)}$$

Theoretical Analysis

	STDA	RDA
Computational Complexity	$O(d)$	$O(d)$
Regret Upper Bound	$O(\sqrt{T})$	$O(\sqrt{T})$

d : # of non-zero elems.

T : # of data

Regret : $$\sum_{t=1}^{T} (\ell_t(\mathbf{w}_t) + \Phi(\mathbf{w}_t)) - \inf_{\mathbf{w}} \left(\sum_{t=1}^{T} (\ell_t(\mathbf{w}) + \Phi(\mathbf{w})) \right)$$

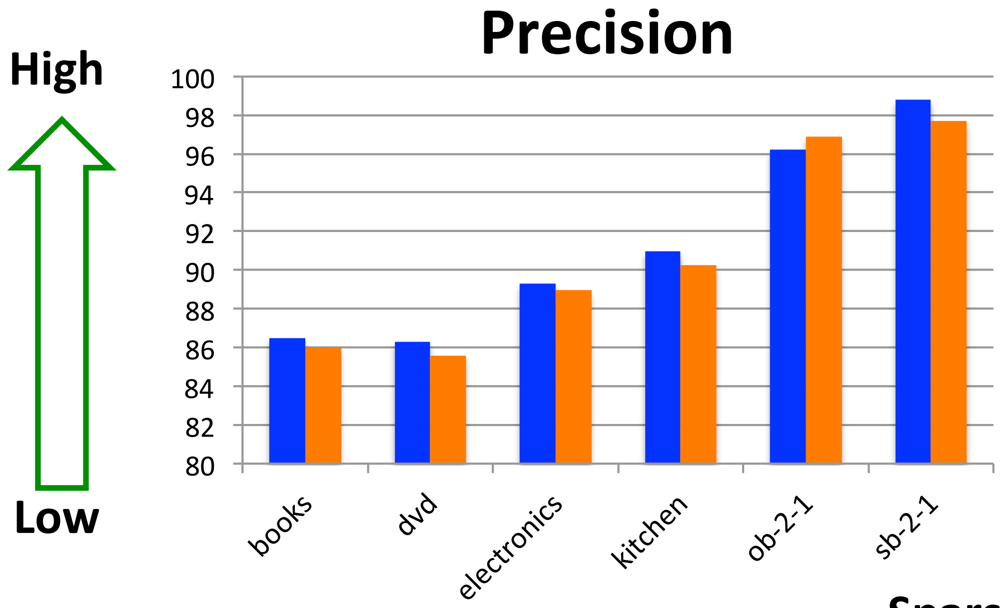
Experiments Overview

- Classification in 6 datasets
  - Comparison1 : vs. Original RDA
  - Comparison2 : vs. Self-weighted based on feature
  - Self-weighted parameter q is set to ∞
    - If $q \geq 2$, obtained almost similar results

# of iteration : 20
10-fold CV to set λ

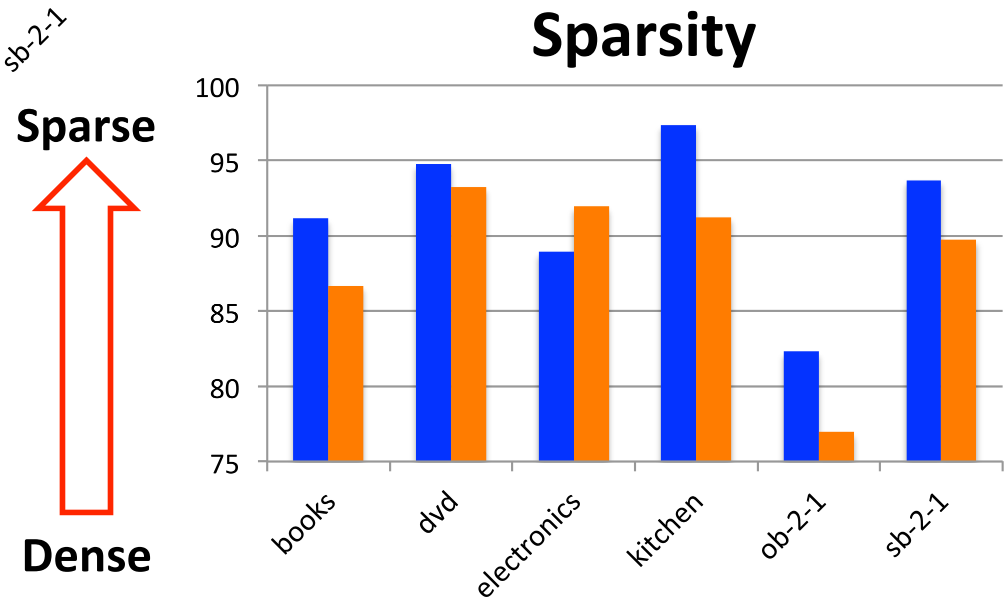
	# of data	# of features	task
books	4,465	332,440	Sentiment
dvd	3,586	282,900	Sentiment
electronics	5,681	235,796	Sentiment
kitchen	5,945	205,665	Sentiment
ob-2-1	1,000	5,942	News
sb-2-1	1,000	6,276	News

Comparison 1 : vs. Original RDA



Our framework

Original RDA



In 4 datasets out of 6 datasets,
Our framework obtain more precise model with more sparsity

Comparison of Important features

- Dataset : books (Sentiment Analysis)

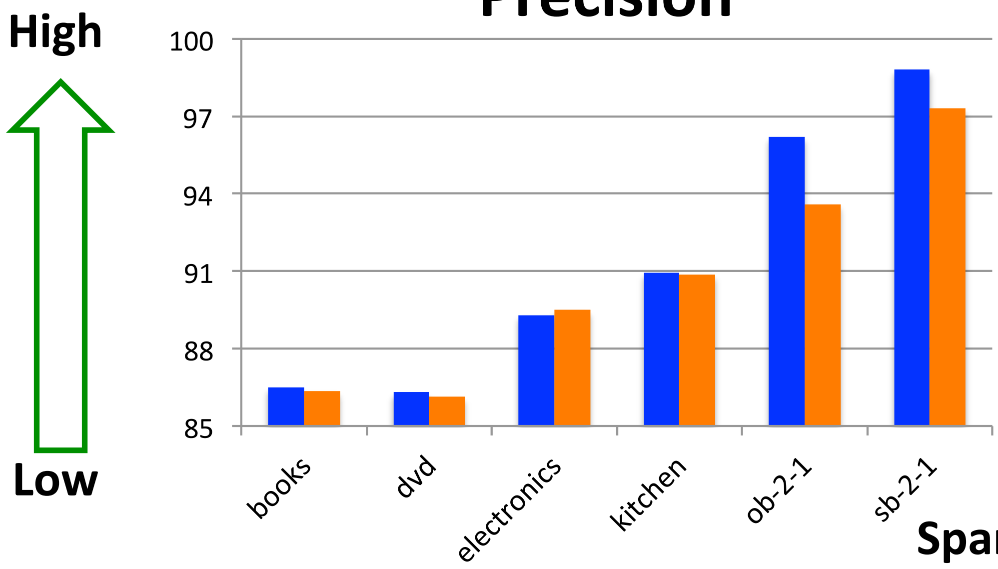
Our framework	**Original RDA**
"some interesting" (117)	"his" (1491)
"a constructive" (101)	"more" (877)
"be successful" (64)	"time" (1161)
"was blatantly" (29)	"almost" (376)
"smearing" (30)	"say" (2407)

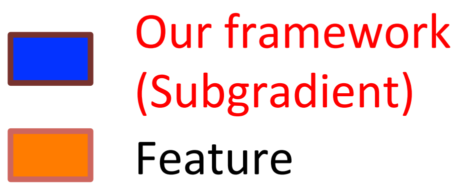
() : Occurrence Counts

Our framework obtain helpful but rare features that conventional algorithms cannot retain

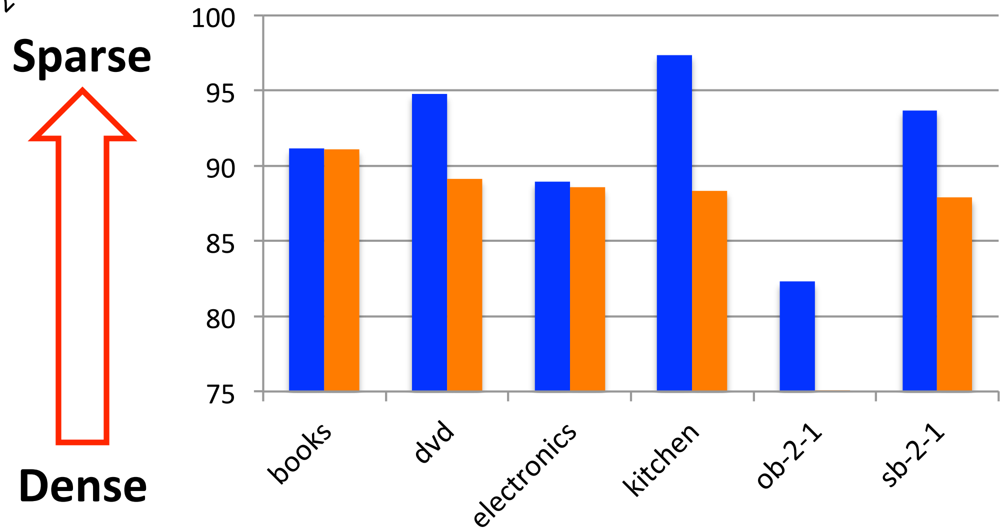
Comparison 2
vs. feature-based framework

Precision





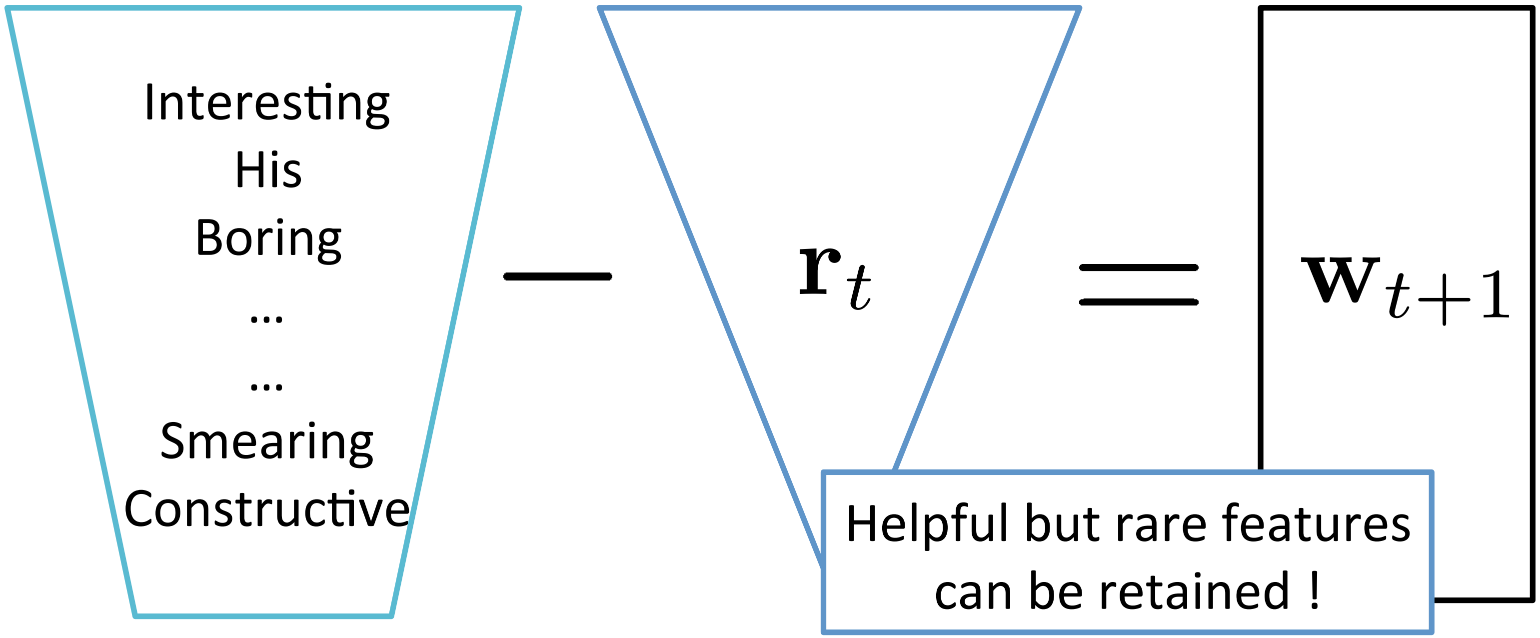
Sparsity



In 5 datasets out of 6 datasets,
Our framework obtain more precise model with more sparsity

Conclusion

- Propose Self-weighted Truncation framework
  - Healing truncation bias on the fly by Subgradients



- Guarantee theoretical bound
- Show experimental results
- Other experiments and analyses are in our paper!