Indexed Block Coordinate Descent for Large-Scale Linear Classification with Limited Memory

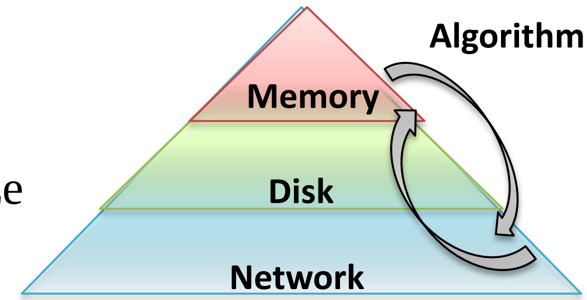
Ian E.H. Yen, Chun-Fu Chang, Ting-Wei Lin, Shan-Wei Lin, and Shou-De Lin

National Taiwan University

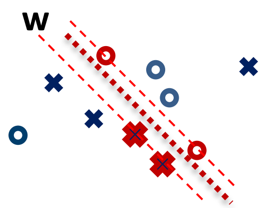
KDD 2013

Large-Scale Linear Classification

- Where is the bottleneck ?
  - I/O dominates often [Yu. 2010] [Chang. 2011]
  - More serious when memory size less than data size

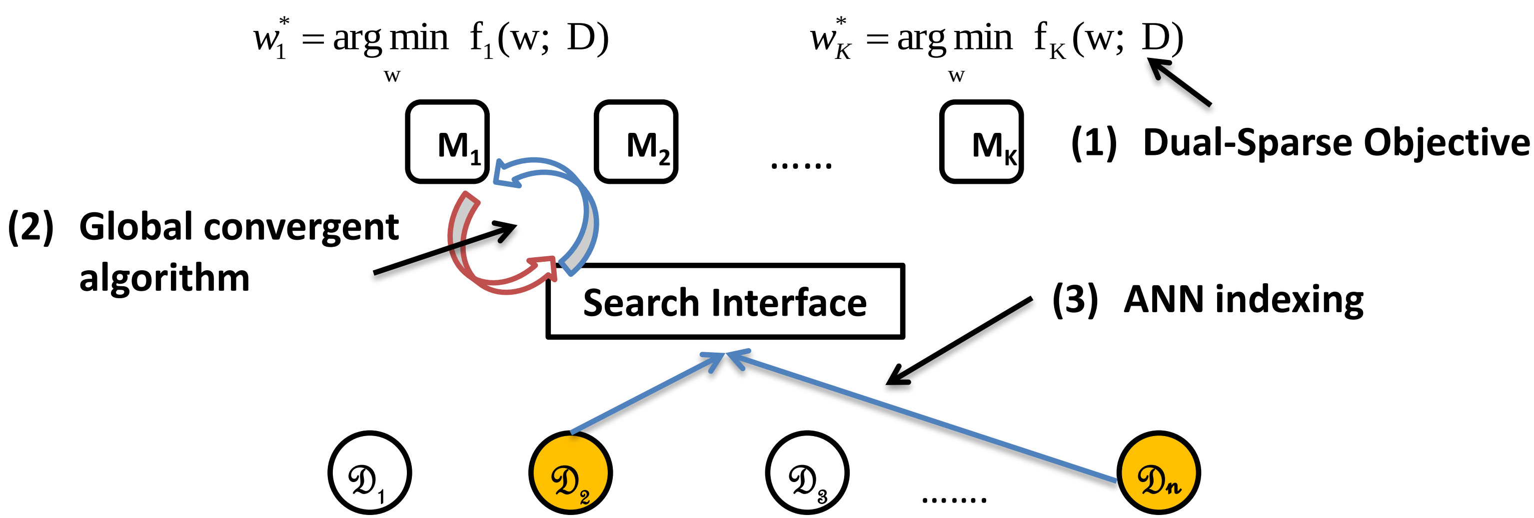


- Observation
  - Given large training data, usually a crucial subset of data is key to improve accuracy.
  - Referred as “dual-sparsity” in SVM literature.



- We can save a lot by reading only crucial samples into Memory
  - Challenge: Not known a priori
  - Our solution: Maintain index before Learning

Framework Overview

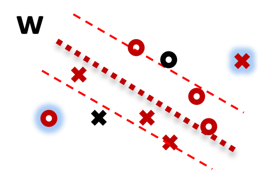


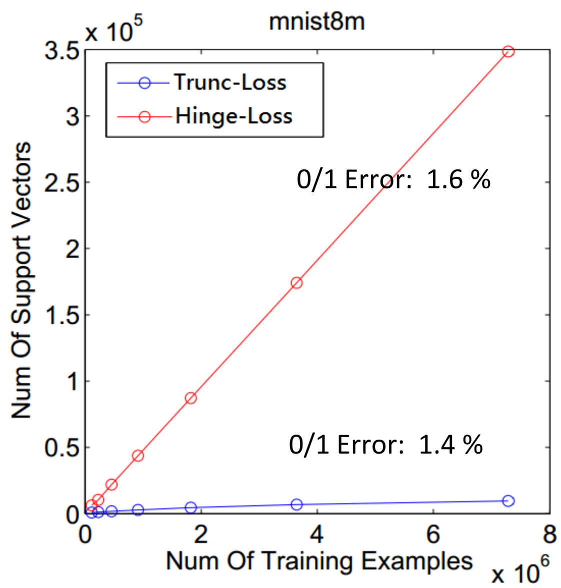
Outline

- Truncated-Loss for Sublinear Dual-Sparsity
  - Sequential Relaxation for Truncated-Loss
- Indexed Learning
  - Informative sample as Nearest Neighbor
  - Indexed Block Coordinate Descent
  - Solving block sub-problem
- Implementation of Indexing
- Experiments

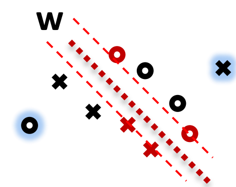
Improve Dual Sparsity from Linear to Sublinear --- Exploiting Truncated Loss

- Regular Support Vector Machine
  - |SV| linear to |data| for non-separable case

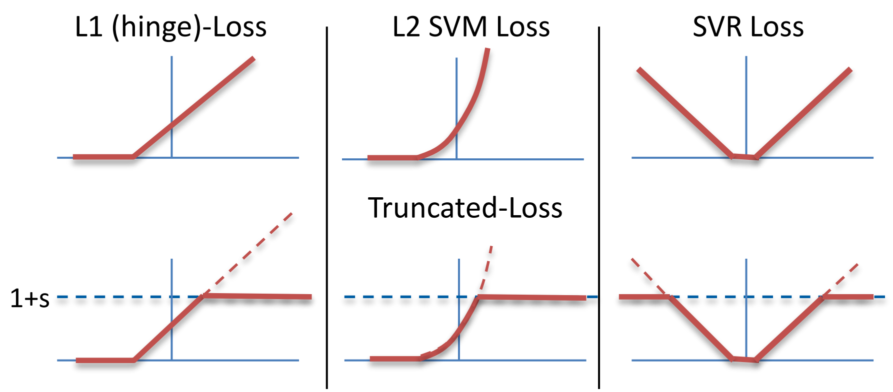




- General Truncated-Loss
  - We can modify any Convex Loss L(.) to Truncated-Loss R(.)=min{ L(.), 1+s }



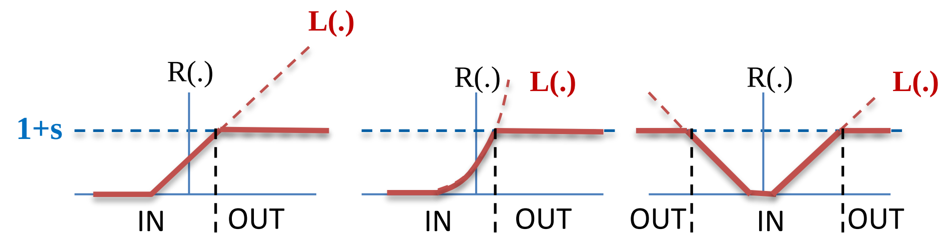
  - Pros: (1) Suppress influence of outliers.
    (2) |SV| sublinear to |data| empirically.
  - Cons: Non-convex problem
    ➔ CCCP for L1-loss [Collobert. 2006]
  - General Relaxation for Truncated-Loss



Sequential Relaxation for Truncated-Loss Problem

Truncated-Loss Problem:

$$\min_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{l} R(y_l, \boldsymbol{w}^T\boldsymbol{x}_l) \quad (1)$$



Majorization Minimization for Truncated-Loss

Initialize w^0 with convex loss L(.) learned on random sub-samples.

repeat

$$w^{t+1} = \arg\min_{w}\left\{ \frac{1}{2}\|w\|^2 + C\sum_{l\in \mathrm{IN}(w^t)} \mathrm{L}(y_l, w^T x_l) + C\sum_{l\in \mathrm{OUT}(w^t)} 1+s \right\} \quad (2)$$

until convergence of w^t

- Minimize (2) decreases objective (1).
- Reason: i. Outlier have loss R(.)=1+s, while non-outliers have loss R(.)=L(.).
  ii. Both 1+s and L(.) upper-bound R(.)=min{ L(.),1+s }.

➔ For each iteration, ignore OUT(w^t) and solve convex-loss L(.) on only IN(w^t).

Sequential Relaxation for Truncated-Loss Problem

$$\min_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_l R(y_l, \boldsymbol{w}^T\boldsymbol{x}_l) \quad (1)$$

Majorization Minimization for Truncated-Loss

Initialize w^0 with convex loss L(.) learned on random sub-samples.

repeat

$$w^{t+1} = \arg\min_{w}\left\{ \frac{1}{2}\|w\|^2 + C\sum_{l\in \mathrm{IN}(w^t)} \mathrm{L}(y_l, w^T x_l) + C\sum_{l\in \mathrm{OUT}(w^t)} 1+s \right\} \quad (2)$$

until convergence of w^t

Theorem 1: The sequence $\{\boldsymbol{w}^t\}_{t=0}^{\infty}$ produced by (2) converges to a stationary point of (1) with at least linear rate.

Proof: By reduction to Block Coordinate Descent on non-convex quadratic problem:

$$\begin{aligned} \min_{\boldsymbol{w},\boldsymbol{\xi},\boldsymbol{d}} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{l\in\mathcal{D}} d_l\xi_l + (1-d_l)(1+s) \\ s.t. \quad & y_l\boldsymbol{w}^T\boldsymbol{x}_l \le 1-\xi_l \\ & \xi_l \ge 0 \\ & 0 \le d_l \le 1, \quad l = 1..m \end{aligned}$$

between **w** and **d**.

Outline

- Truncated-Loss for Sublinear Dual-Sparsity
  - Sequential Relaxation for Truncated-Loss

- Indexed Learning
  - Informative sample as Nearest Neighbor
  - Indexed Block Coordinate Descent
  - Solving block sub-problem

- Implementation of Indexing

- Experiments

Block Coordinate Descent (BCD) in the Dual

Now we focus on L1-loss, L2-loss SVM problems:

$$w^{t+1} = \arg\min_{w} \left\{ \frac{1}{2}\|w\|^2 + C \sum_{l \in \mathbf{IN(w^t)}} L(y_l, w^T x_l) + C \sum_{l \in \mathbf{OUT(w^t)}} 1 + s \right\}$$

Block Coordinate Descent on the Dual:

$$\begin{aligned} \min_{\boldsymbol{\alpha} \in \mathbb{R}^N} \quad & f(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^T \bar{Q} \boldsymbol{\alpha} - \boldsymbol{e}^T \boldsymbol{\alpha} \\ s.t. \quad & 0 \le \alpha_l \le U, \quad l \in IN(\boldsymbol{w}^t) \\ & \alpha_l = 0, \quad l \in OUT(\boldsymbol{w}^t) \end{aligned}$$

Dual-Sparsity: The optimal solution α^* contains only |SV| << N non-zeros.

Shrinking: Iteratively eliminate non-active α_l from working set. [Joachims, 1998]

To avoid I/O in limited-memory case:

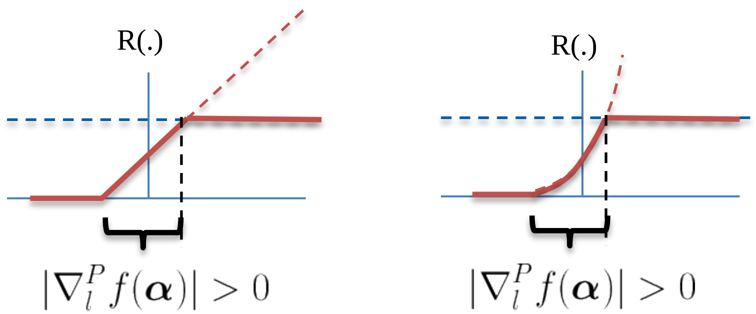
Caching: Read partition of data into memory, caching samples with active α_l. [Chang, 2011]

Indexing: Read only samples with most active α_l into memory via ANN Search Index.

Informative Samples as Nearest Neighbors

Samples with non-zero $\nabla_l^P f(\boldsymbol{\alpha})$:

$$\{l | L^{-1}(1+s) \leq y_l \boldsymbol{w}^T \boldsymbol{x}_l \leq 1\}$$



Standard ANN (similarity) search finds:

$$\underset{l}{argmax} \quad \hat{\boldsymbol{q}}^T \hat{\boldsymbol{x}}_l$$

Transform target samples as Nearest Neighbor in embedded space defined by V(.):

$$\begin{aligned} \underset{l}{argmin} \quad |\hat{\boldsymbol{w}}^T \hat{\boldsymbol{x}}_l| &= \underset{l}{argmin} \quad (\hat{\boldsymbol{w}}^T \hat{\boldsymbol{x}}_l)^2 \\ &= \underset{l}{argmin} \quad \boldsymbol{V}(\hat{\boldsymbol{w}})^T \boldsymbol{V}(\hat{\boldsymbol{x}}_l) \\ &= \underset{l}{argmax} \quad (-\boldsymbol{V}(\hat{\boldsymbol{w}}))^T \boldsymbol{V}(\hat{\boldsymbol{x}}_l) \end{aligned}$$

where V(.) is degree-2 polynomial feature expansion.

➔ Indexing data with product defined by $(\hat{\boldsymbol{x}}_i^T \hat{\boldsymbol{x}}_j)^2$, query with $-(\hat{\boldsymbol{w}}^T \hat{\boldsymbol{x}}_i)^2$.

Indexed Block Coordinate Descent

Algorithm 1 Indexed Block Coordinate Descent

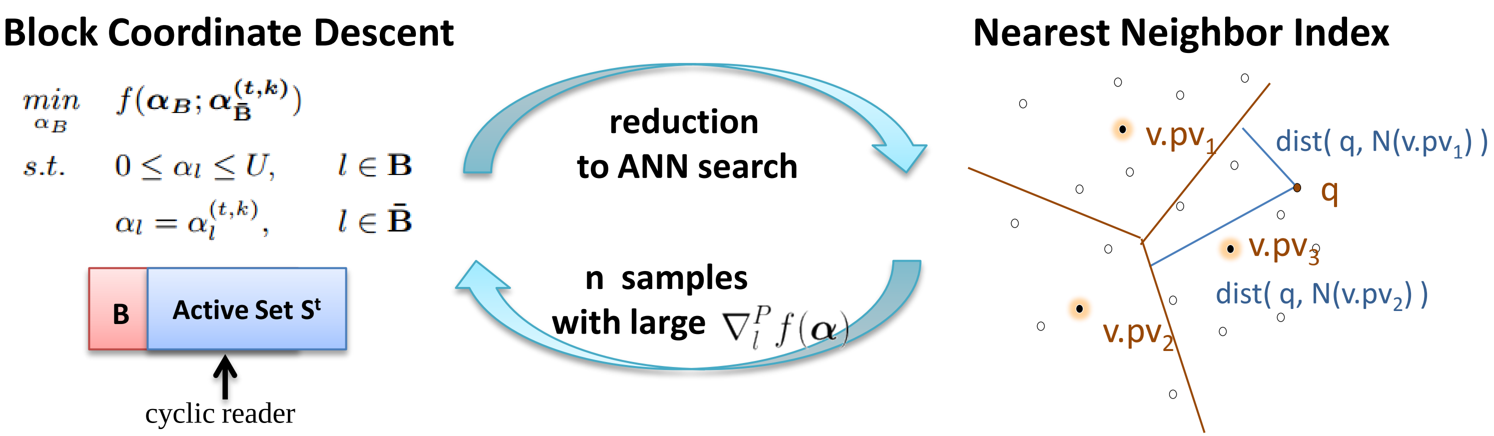
Input: $\boldsymbol{w}^{(t,0)} = \boldsymbol{w}^t$, $\mathbf{S}^{(0)} = \mathbf{S}^t \setminus OUT(\boldsymbol{w}^t)$
Output: $\boldsymbol{w}^{t+1} = \boldsymbol{w}^{(t,k)}$, $\mathbf{S}^{t+1} = \mathbf{S}^{(k)}$
repeat
  $(\mathbf{N}, n_e) \leftarrow$ queryIndex($\boldsymbol{w}^{(k)}$; $\boldsymbol{w}^t$, s, n)
  $\mathbf{B} \leftarrow [\ \mathbf{N},\ \mathbf{S}^{(k)}[\ r : r + n_e - |\mathbf{N}|\ ]\ ]$
  Solve block minimization problem (11) over $\mathbf{B}$.
  $\mathbf{S}^{(k+1)} \leftarrow [\ \mathbf{S}^{(k)},\ \mathbf{N}\ ]$
  $k \leftarrow k+1$; $r \leftarrow (r + n_e + 1) \bmod |\mathbf{S}^{(k)}|$
until problem (13) defined on $\mathbf{S}^{(k)}$ reach ϵ_S and $|\mathbf{N}| < n_e$

Cost each iteration:

$$T_{search}(n_e) + T_{opt}(|\mathbf{B}|), \quad n_e = \frac{n}{prec[n]}$$

Balance between these two terms:

Set $|\boldsymbol{B}| = n_e$ = # of explored.



- Global convergence to the solution of the Dual Problem.

Solving Block Sub-problems

Block Sub-problem (Dual)

$$\begin{aligned} \min_{\alpha_B} \quad & f(\boldsymbol{\alpha}_B; \boldsymbol{\alpha}_{\bar{\mathbf{B}}}^{(t,k)}) \\ s.t. \quad & 0 \le \alpha_l \le U, \qquad l \in \mathbf{B} \\ & \alpha_l = \alpha_l^{(t,k)}, \qquad l \in \bar{\mathbf{B}} \end{aligned}$$

Block Sub-problem (Primal)

$$\min_{\boldsymbol{w}} \quad \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{l\in\mathbf{B}} L(y_l \boldsymbol{w}^T \boldsymbol{x}_l) - \boldsymbol{w}^T \mathbf{v}_{\bar{\mathbf{B}}}$$

$$\mathbf{v}_{\bar{\mathbf{B}}} = \sum_{l\in\bar{\mathbf{B}}} \alpha_l^{(t,k)} y_l \boldsymbol{x}_l$$

They are standard Linear SVM problems. In this work, we empoly:

- L1 (hinge) loss ➔ Dual Coordinate Descent (DCD). [Heish, 2008]
- L2-Loss ➔ Trust-Region Quasi-Newton [Lin. 2008] and DCD.

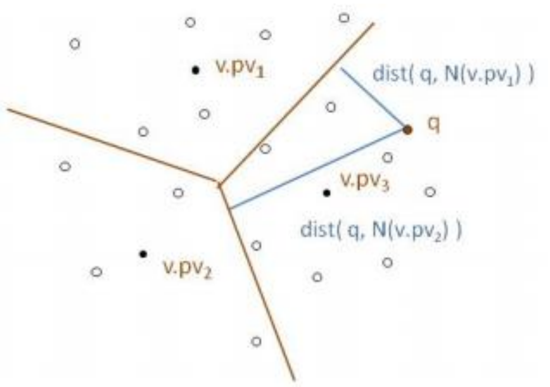
Outline

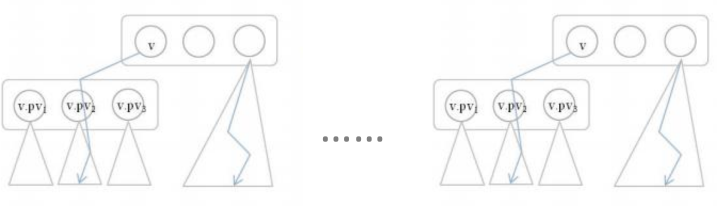
- Truncated-Loss for Sublinear Dual-Sparsity
  - Sequential Relaxation for Truncated-Loss
- Indexed Learning
  - Informative sample as Nearest Neighbor
  - Indexed Block Coordinate Descent
  - Solving block sub-problem
- Implementation of Indexing
- Experiments

Implementation of Indexing

- K-way Metric Tree
  - K reference points partition data into K subsets.
  - Recursively partitioning.
- Bias Reduction
  - Avoid bias to few reference points.
  - Bootstrap and build index for each random subsets.
- Incremental Search
  - Best-Bin-First search on each tree.
  - Traverse different trees in random order.

Metric Tree with $k = 3$.





Subset 1 Subset M

Outline

- Truncated-Loss for Sublinear Dual-Sparsity
  - Sequential Relaxation for Truncated-Loss
- Indexed Learning
  - Informative sample as Nearest Neighbor
  - Indexed Block Coordinate Descent
  - Solving block sub-problem
- Implementation of Indexing
- Experiments

Experiment – Data and Index

Table 1: Statistics of Data.

Dataset	#Samples	#Features	Storage (KB)
Covtype	581,012	54	69,516
Kddcup1999	4,898,431	126	725,180
PAMAP	3,850,505	104	2,198,880
Mnist8m	8,100,000	784	19,042,640

Table 2: Statistics of Index.

Dataset	Storage (KB)	Tree Size	Tree Width	Build Time (s)
Covtype	446,444	2,000	10	11
Kddcup1999	1,476,580	100,000	100	163
PAMAP	4,554,208	100,000	10	301
Mnist8m	20,704,784	10,000	10	1,539

- Feature scaled to [0,1].
- C=1, s=1 for all experiments.

Methods Compared:

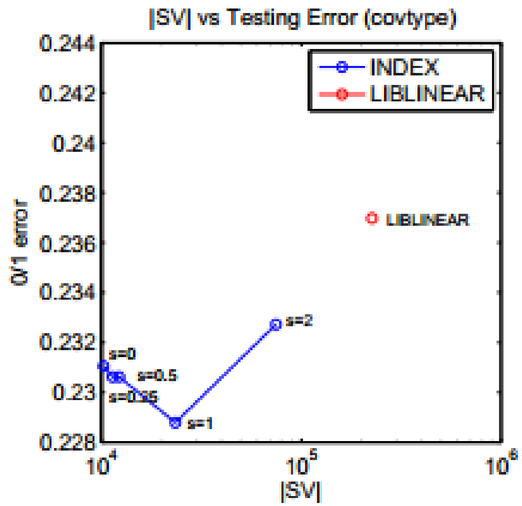
- Convex-Loss Solver
- Truncated-Loss Solver
- Indexed Truncated-Loss Solver

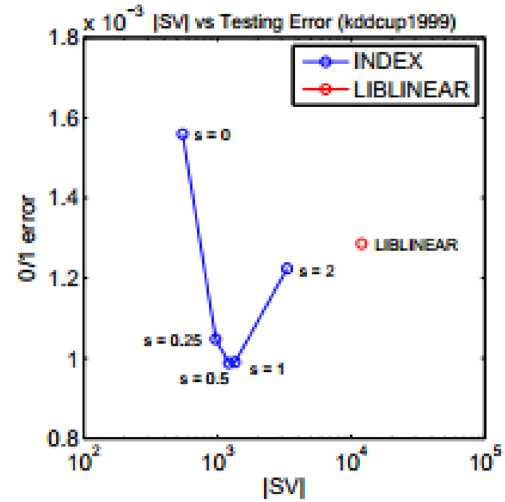
Solvers: (Liblinear, Pegasos)

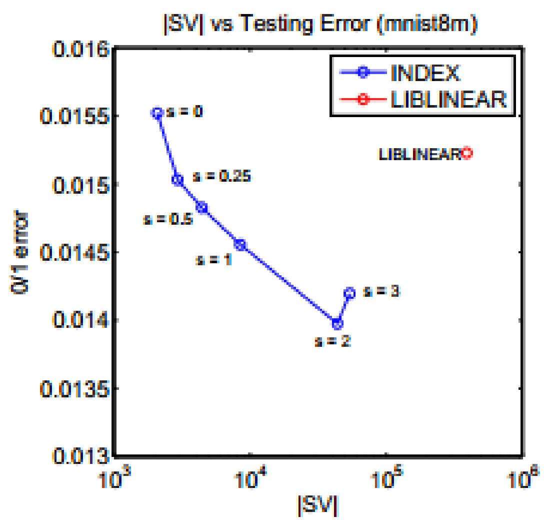
- L1-Loss (hinge-loss)
  -- Dual Coordinate Descent
  -- SGD (online)

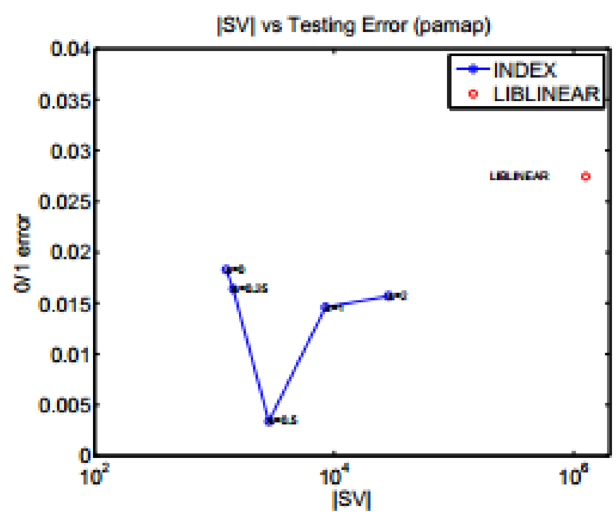
- L2-Loss
  -- Trust-Region Quasi-Newton
  -- Dual Coordinate Descent
  -- SGD (online)

|SV| vs. Truncated-Loss parameter (1+s)



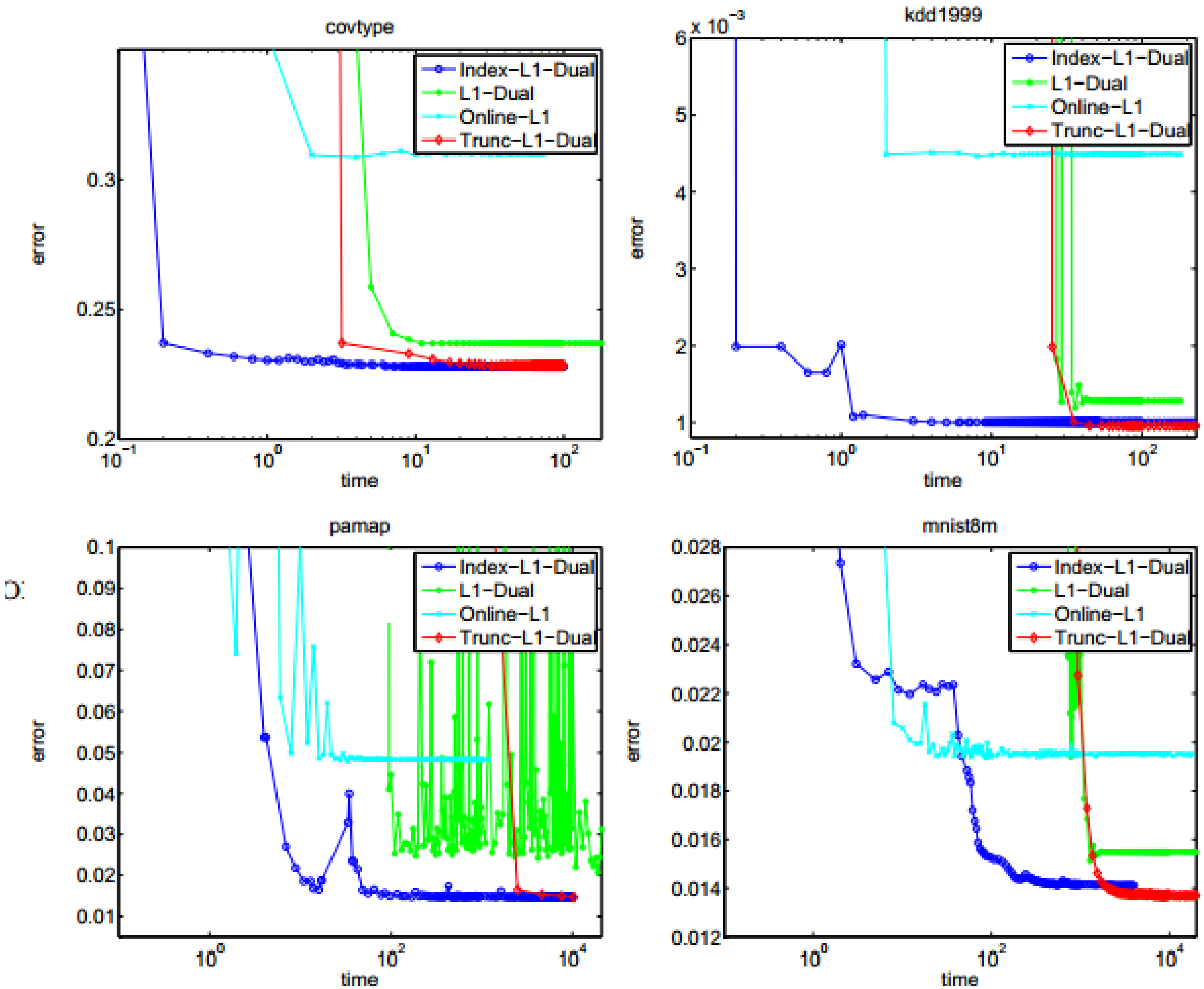






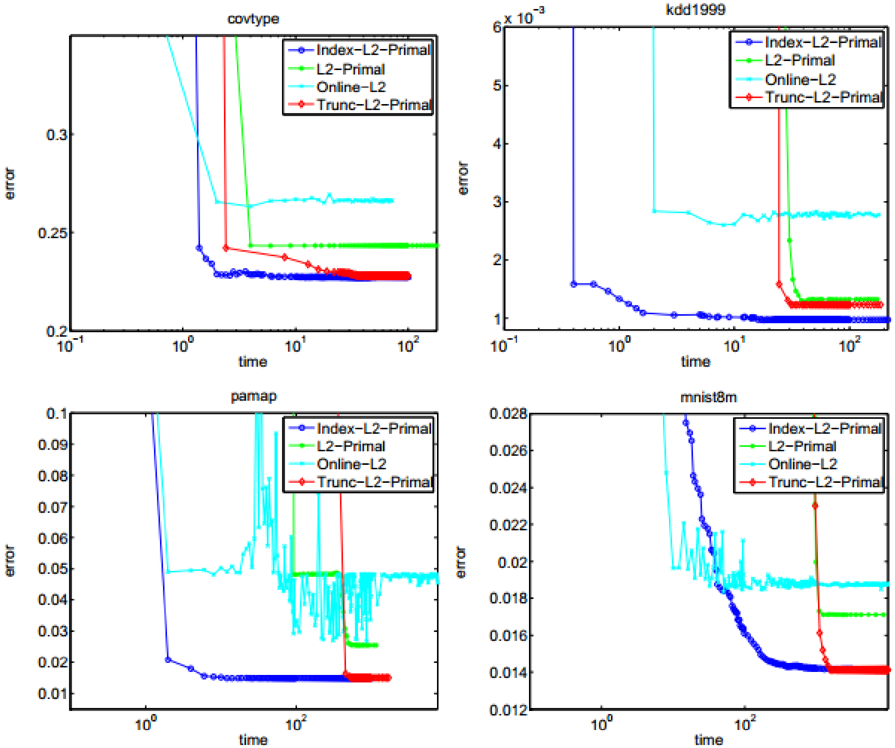
Testing Error vs. Time (log-scale)

L1-Loss, Dual Coordinate Descent

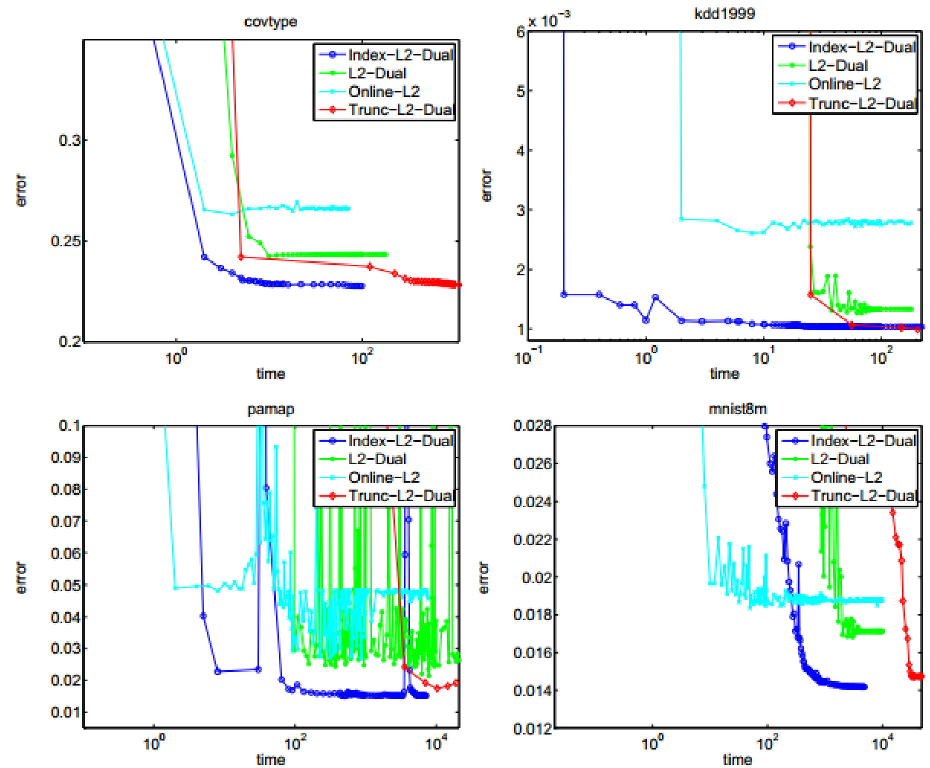


Testing Error vs. Time (log-scale)

L2-Loss, Primal Trust-Region Quasi-Newton

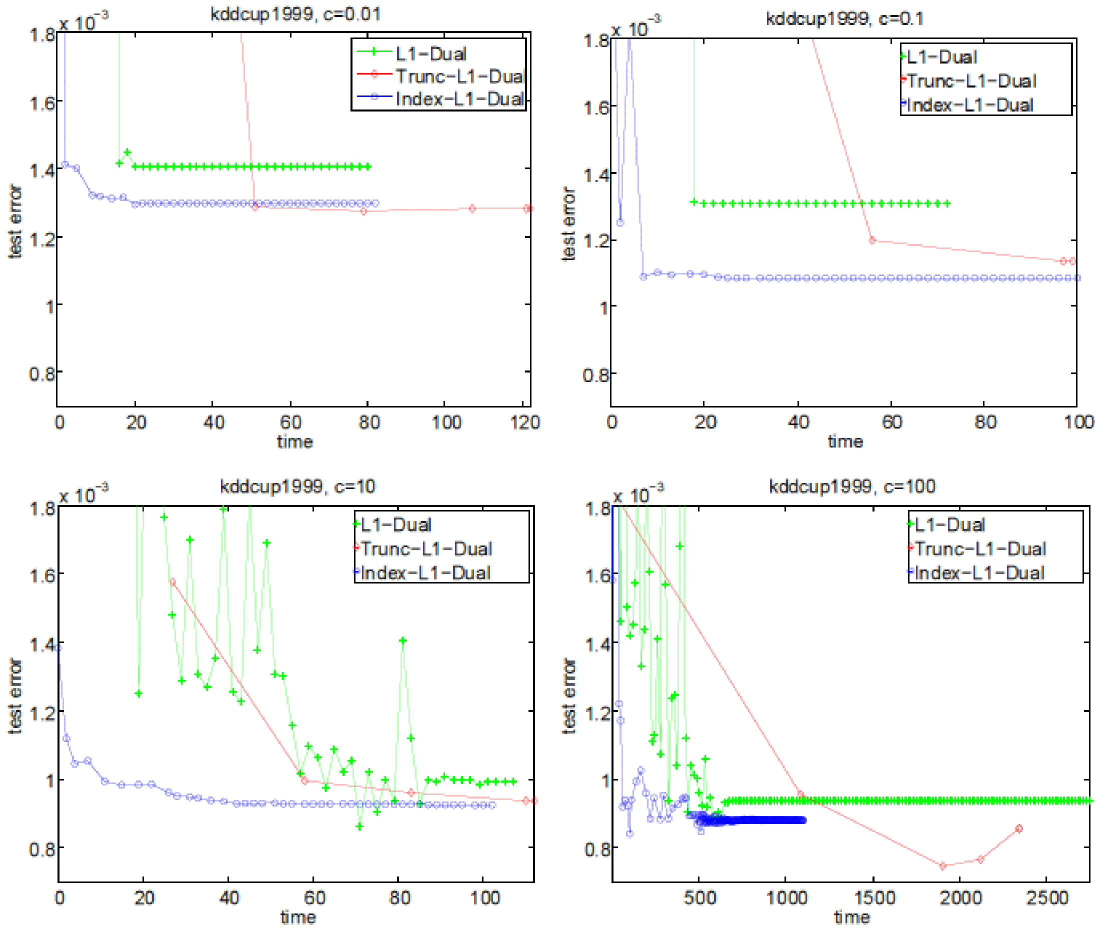


L2-Loss, Dual Coordinate Descent



C=0.01, 0.1, 10 and 100

L1-Loss, Dual Coordinate Descent



Conclusion

- The bottleneck of large-scale Linear Classification lies on time spent on disk/network I/O.
- In this work, we propose Indexed Block Coordinate Descent to solve Truncated-Loss SVM with both sublinear I/O and computation time.
- Our experiments show orders of magnitude speed up when one pre-built indexing structure to help solving optimization problem.
- This is especially useful when memory is limited, or there are lots of models (from different classes, parameters, or features) to be trained.

Thank You