Machine Learning with Big Data
CSL7110

Syllabus

Introduction: What is big data, Unreasonable effectiveness of data (1 lecture)
Streaming algorithms: Streaming Naive Bayes, Stream and sort (2 lectures)
Platforms for learning from big data: MapReduce, New Software Stack, Large Scale File System Organization (5 lectures)
Nearest Neighbour Search, Jaccardi Similarity of Sets, Similarity of Documents, Locality Sensitive Hashing, The Stream Data Model (4 lectures)
Randomized methods: Clustering, Hashing, Sketching, Scalable stochastic gradient descent (3 lectures)
Frequent Itemsets: The Market Basket Model, A-Priori Algorithm, Handling larger datasets in Main Memory, Limited-Pass Algorithms, Counting Frequent Items in a Stream (6 lectures)
Parameter Servers: Introduction, Abstraction, Parameter Cache Synchronization, Asynchronous execution, Model Parallel Examples (3 lectures)
Graph-based methods (6 lectures)
Page Rank, Topic Sensitive Page Rank, Approaches to Page Rank iteration, Link Spam, Semi-supervised learning, Scalable link analysis, Models for Recommendation Systems, Social Networks as Graphs (9 lectures)
Large-scale Machine Learning with CPUs and GPUs (3 lectures)

- Available online at: https://cse.iitj.ac.in/images/pdf/curriculum/CSE-Courses-Details.pdf

Timings

- Lectures (in hybrid mode)
  - Tue: 6-7:30 PM; Sat: 2-3 PM
- Institute’s attendance policy applicable to all the students.
- Regular students must attend in person.
- Need for an extra slot (?)

Marks distribution

- Assignments: 15-20%

- Quizzes: 10-15%

- Major Project: 25-30%

- Class Performance: 5%, with -ve marking

- Exams: 40-50%

Penalty for Malpractice(s)

- Reduction in grade
- ‘F’ grade
- Semester drop

Plagiarism

https://www.plagiarism.org/

Use of Google Classroom

- Use google classroom for all discussions related to the course, and prefer to post your comments/thoughts publicly. Avoid sending emails/private messages to the course-instructor/TAs.

- Everyone is encouraged to participate actively, and even respond to queries shared by others.

- For those joining online, use the email-id provided by the institute.

Any Comments?

Objectives of this course

- Challenges involving large datasets: practical knowledge and experience

- Scalable algorithms: Learning/Non-learning based

- Techniques to work with constraints on RAM and throughput time.

Prerequisites

- Basic probability, statistics, linear algebra and optimization.
- Data structures and Algorithms.
- Ability to understand, adapt, write and debug (complex, parallel) codes, beyond Matlab or R.
- Working knowledge of all the popular ML algorithms.
- One of the following courses on Machine Learning: ML/PRML/IML

Asymptotic analysis

- Measures number of operations as function of problem size.
- We use physical devices to store and read data using different operations (e.g., disk seeking, scanning, memory access, etc.).
- Different operations => Different costs
- Disk access is cheapest when you scan sequentially.

Memory access

L1 cache reference	0.5 ns
Branch mispredict	5 ns
L2 cache reference	7 ns
Mutex lock/unlock	100 ns
Main memory reference	100 ns
Compress 1K bytes with Zippy	10,000 ns
Send 2K bytes over 1 Gbps network	20,000 ns
Read 1 MB sequentially from memory	250,000 ns
Round trip within same datacenter	500,000 ns
Disk seek	10,000,000 ns
Read 1 MB sequentially from network	10,000,000 ns
Read 1 MB sequentially from disk	30,000,000 ns
Send packet CA->Netherlands->CA	150,000,000 ns

Any Comments?

Why large datasets are difficult to work with?

- Visualization
- Errors and biases
- Computational cost of learning
- Tradeoff in the accuracy of algorithms (best to worst)
- Large dataset => Heavy models (often)

ImageNet Classification Challenge

- One of the earliest million-scale (image) datasets
- Initially, 1000 classes, 1000 images per class, each image tagged with one class/category
- Classes are derived from the WordNet hierarchy.

ImageNet Classification Challenge

- ILSVRC’10
  - Winning entry: SIFT and LBP features with two non-linear encoding representations + (stochastic) one-vs-rest linear SVMs
  - Honourable mention: An improved Fisher vector representation along with PCA + one-vs-rest linear SVMs

ImageNet Classification Challenge

- ILSVRC'11
  - Winning entry: High-dimensional image signatures with compression using product quantization + one-vs-rest linear SVMs

ImageNet Classification Challenge

- ILSVRC'12
  - Winning entry: A deep CNN trained on RGB values, with 60 million parameters using an efficient GPU implementation and a novel hidden-unit dropout trick.
  - Second place: (Handcrafted Features) + Linear classifiers

Revisiting SVM

SVM Overview

- Training: Learn a weight vector per category/class/label.

- Testing: Evaluate all the weight vectors for a given test/unseen data point using dot product and output the one(s) with the maximum similarity.

$$\min_{\mathbf{w}_\ell} \left[||\mathbf{w}_\ell||_2^2 + C \sum_{i=1}^{N} \left(\max(0, 1 - s_{\ell_i} \mathbf{w}_\ell^T \mathbf{x}_i)\right)^2 \right]$$

A naive one-vs-rest approach

- Learn a weight vector per label; evaluate all the weight vectors for a given test point.

- Challenges while scaling to large datasets:
  - Training complexity
  - Model size
  - Prediction speed

A naive one-vs-rest approach

- Challenges while scaling to large datasets:
  - Training complexity
  - Model size
  - Prediction speed

- E.g.; Consider a dataset with ~1.6M features, ~325K labels, ~1.8M training points, ~587K test points
  - Training time: **96 days** (using an ad hoc application of Liblinear)
  - Model size: **870 GB** (learnt using L2 regularization in linear SVM)
  - Each prediction involves **325K dot products** in a **1.6M dimensional feature space**, followed by **sorting 325K values**.

A naive one-vs-rest approach

- Challenges while scaling to large datasets:
  - Training complexity
  - Model size
  - Prediction speed

- E.g.; Consider a dataset with ~1.6M features, ~325K labels, ~1.8M training points, ~587K test points
  - Training time: **96 days** (using an ad hoc application of Liblinear)
  - Model size: **870 GB** (learnt using L2 regularization in linear SVM)
  - Each prediction involves **325K dot products** in a **1.6M dimensional feature space**, followed by **sorting 325K values**.

- WikiLSHTC-325K

Any Comments?

DiSMEC*

Input: Training data $\mathcal{T} = \{(\mathbf{x}_1, \mathbf{y}_1) \ldots (\mathbf{x}_n, \mathbf{y}_n)\}$, input dimensionality D, label set $\{1 \ldots L\}$, $B = \lfloor \frac{L}{1000} \rfloor + 1$ and Δ

Output: Learnt matrix $\mathbf{W}_{D,L}$ in sparse format

1: Load single copy of input vectors $\mathbf{X} = \{\mathbf{x}_1 \ldots \mathbf{x}_n\}$ in the main memory $\triangleright$ *Refactor data without replication*

2: Load binary sign vectors $\mathbf{s}_l = \{+1, -1\}_{i=1}^{n}$ separately for each label in the main memory

3: **for** $\{b = 0; b < B; b++\}$ **do** $\triangleright$ *1st parallelization*

4:     #pragma omp parallel for private(ℓ) $\triangleright$ *2nd parallelization*

5:     **for** $\{l = b \times 1000; l \leq (b+1) \times 1000; l++\}$ **do**

6:         Using $(\mathbf{X}, \mathbf{s}_l)$, train weight vector $\mathbf{w}_\ell$ on a single core

7:         Prune ambiguous weights in $\mathbf{w}_\ell$ $\triangleright$ Model reduction

8:     **end for**

9:     **return** $\mathbf{W}_{D,1000}$ $\triangleright$ *Learnt matrix for a batch on one node*

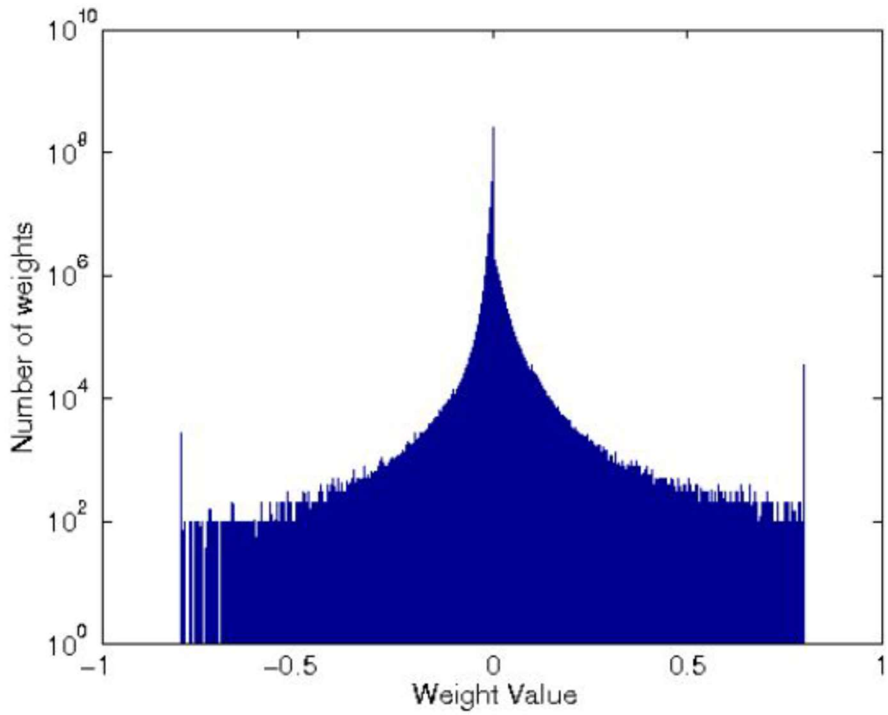
10: **end for**

11: **return** $\mathbf{W}_{D,L}$ $\triangleright$ *Learnt matrix from all the nodes*

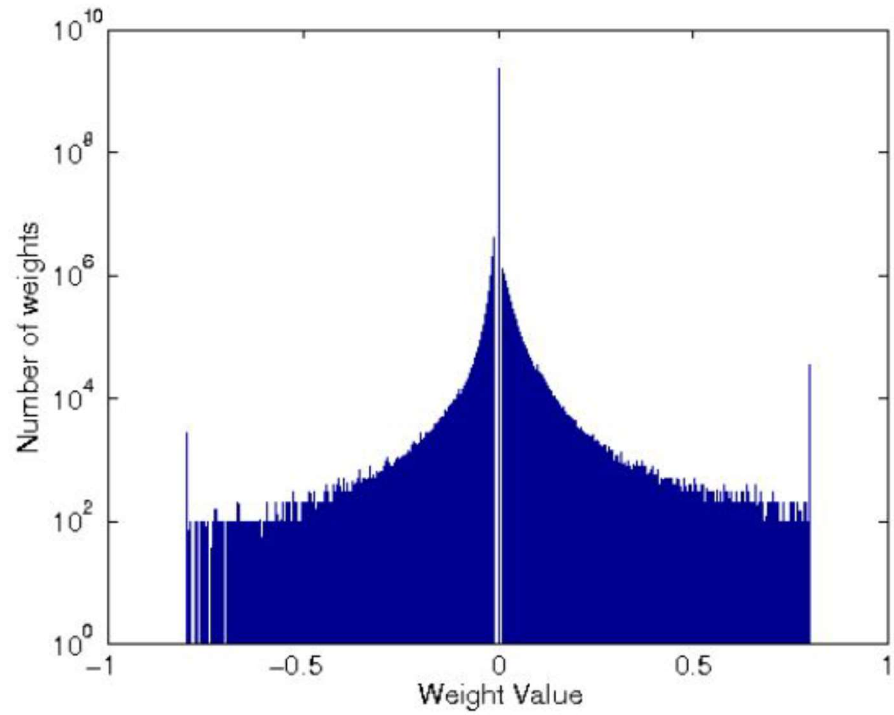
* R. Babbar, and B. Schölkopf, ***DiSMEC - Distributed Sparse Machines for Extreme Multi-label Classification***, in *WSDM*, 2017

DiSMEC Algorithm

- The parameter “delta” essentially tunes the model’s behavior between the two extremes of L2 and L1 regularization.



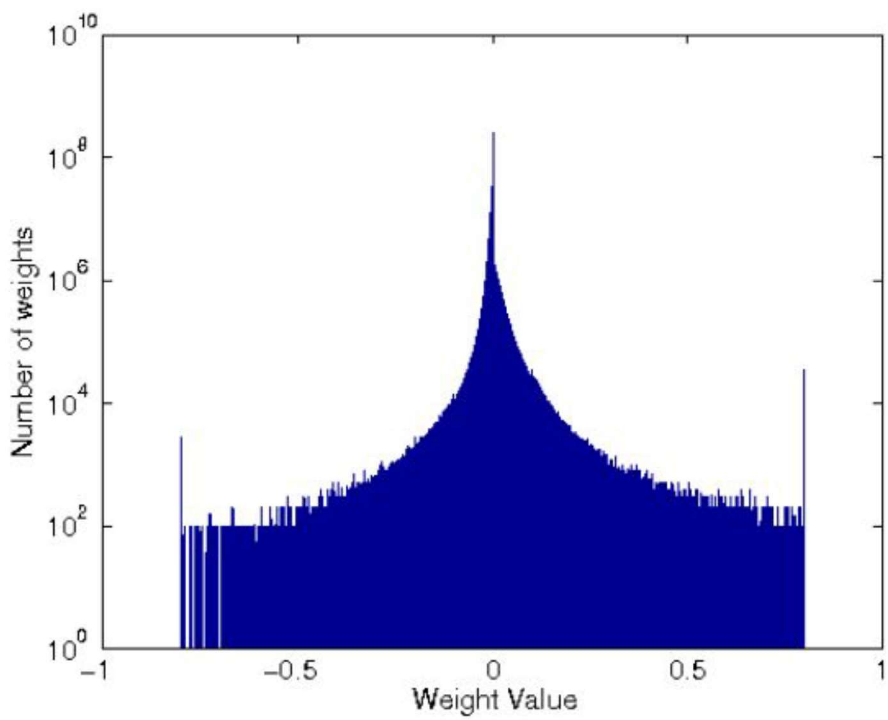
(a) Distribution of weights before pruning



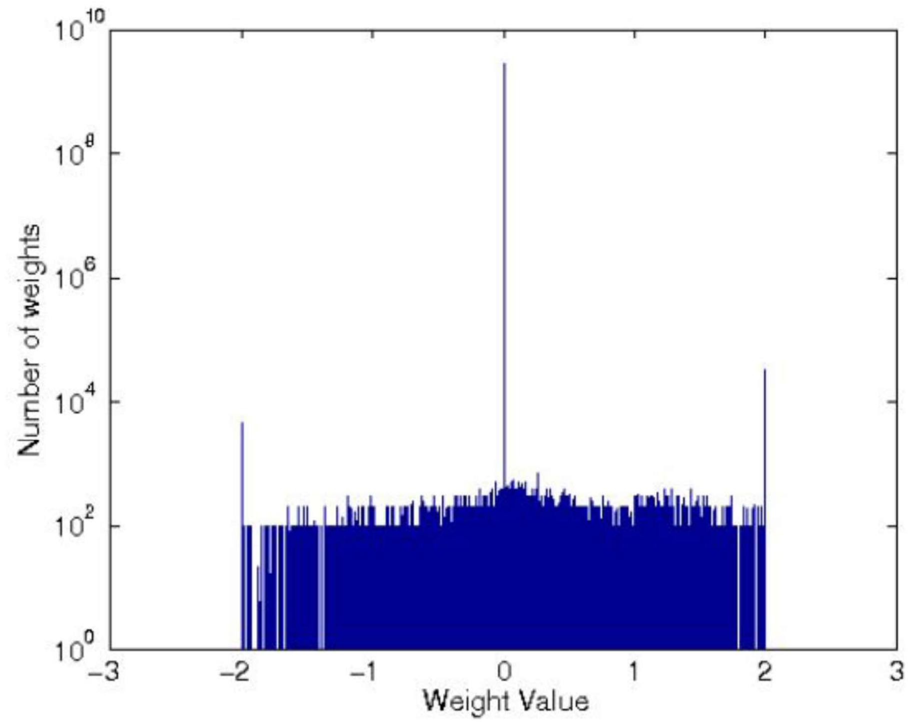
(b) Distribution of weights after pruning

DiSMEC Algorithm

- The parameter “delta” essentially tunes the model’s behavior between the two extremes of L2 and L1 regularization.



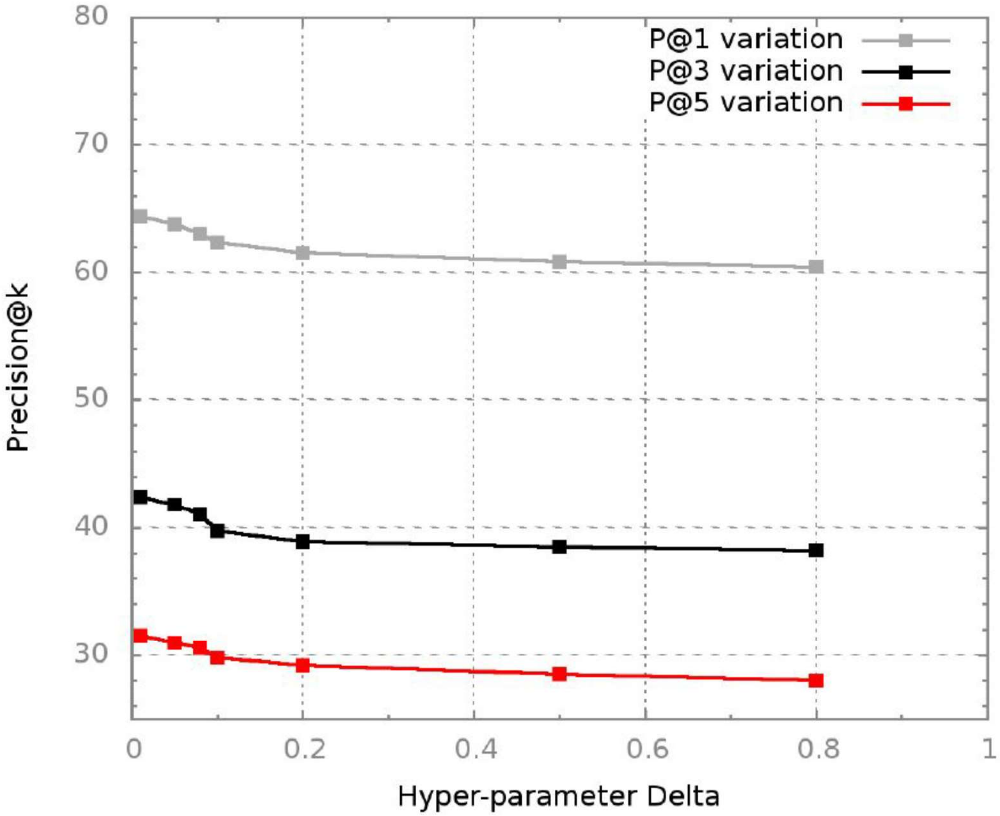
(a) Distribution of weights before pruning

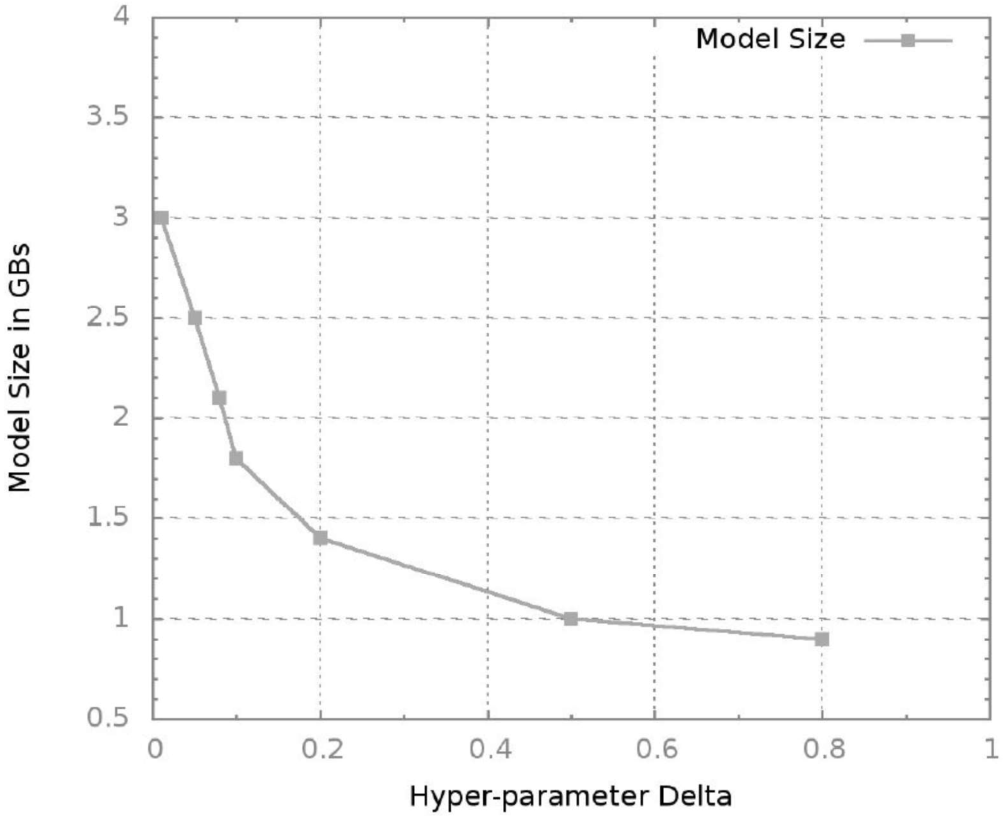


(a) Distribution of weights for l_1-regularization

DiSMEC Algorithm

- The parameter "delta" essentially tunes the model's behavior between the two extremes of L2 and L1 regularization.





Advantages of DiSMEC

- Doubly parallelized architecture: Efficient utilization of multi-node multi-core architectures.
- Explicit model sparsity induction (not learned!).
- Batch training => Distributed storage of models => Real-time prediction.
- Example: WikiLSHTC-325K dataset (~1.6M features, ~325K labels, ~1.8M training points, ~587K test points):
  - Training time: 6 hours on 400 cores and 3 hours on 1000 cores
  - Model size: 3 GB (few billion parameters less)
  - Prediction time: 3 ms

Any Comments?