Anomaly Detection in Large Data

Chandresh Kumar Maurya
Research scholar

Department of Computer Science & Engineering
Indian Institute of Technology, Roorkee,
Haridwar-247667, INDIA

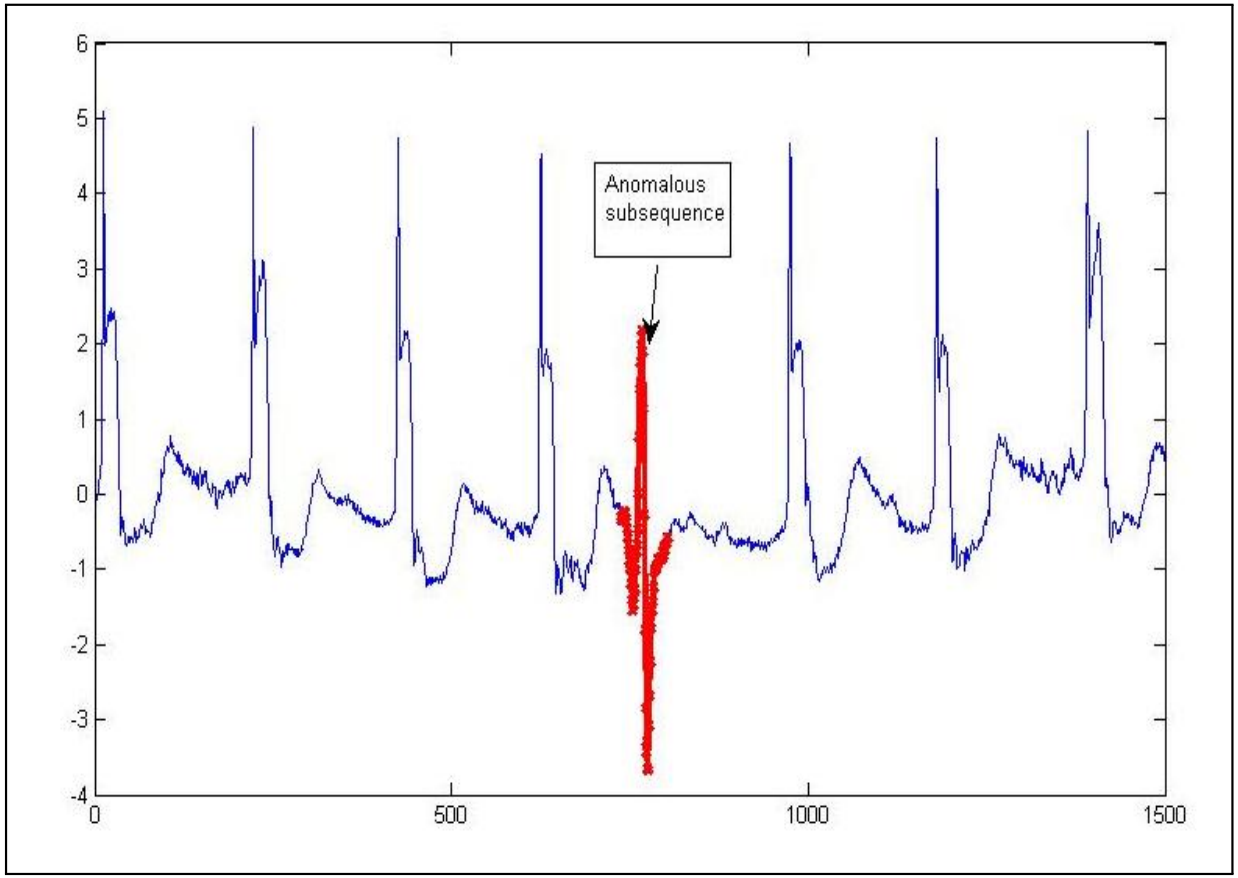
May 8, 2016

Outline

Introduction

What is Anomaly Detection?

Anomaly found in EEG data shown in red color



Contd...

Anomaly in Image data

Figure : Source: V. Mahadevan et al. *CVPR, 2010*

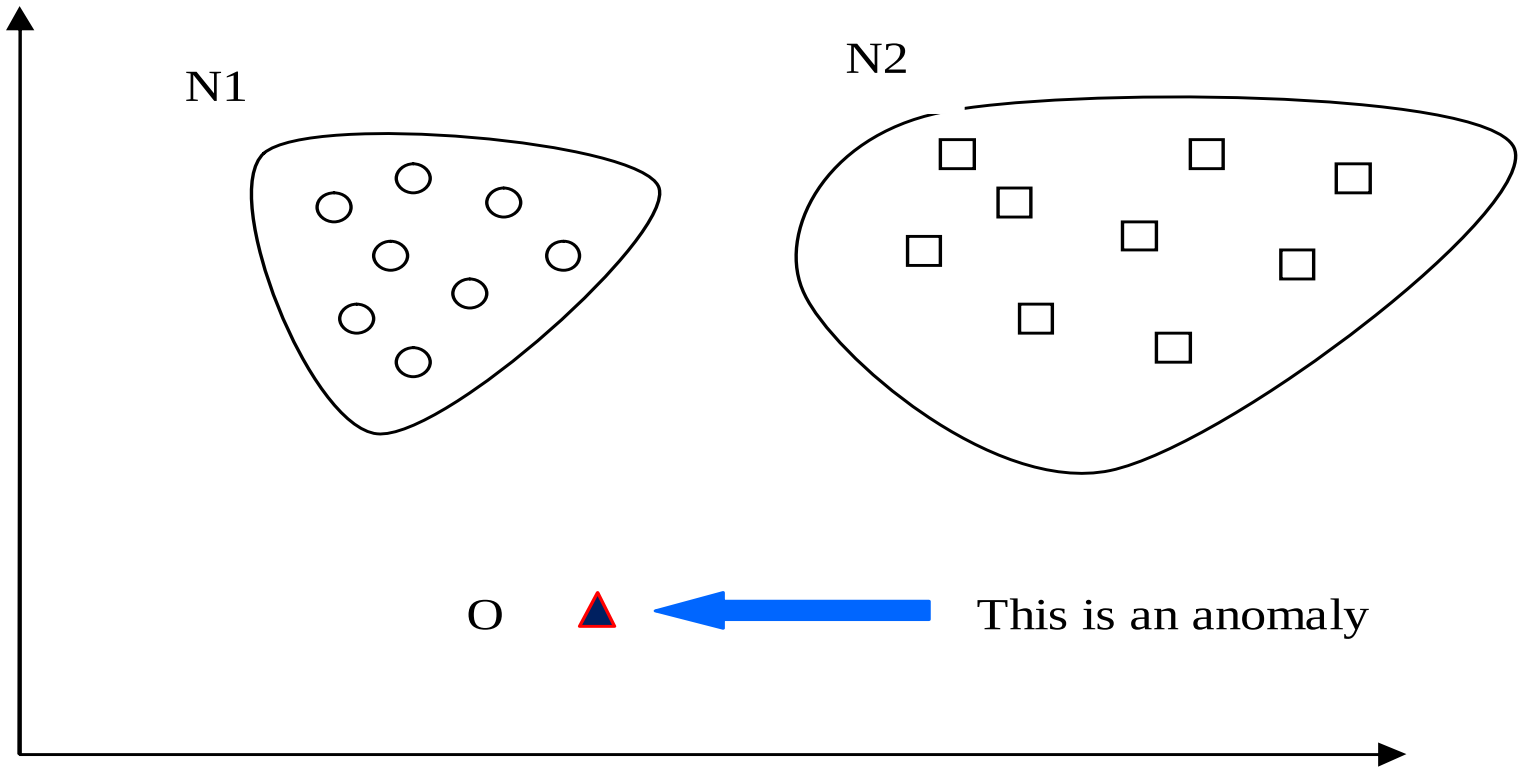
What is large Data ?

Examples–

- Image Data (>1Millions pixels)
- Social Media Data (12 terabytes of tweets everyday)
- Sensor Data
- Video Surveillance Data
- Medical Data (e.g. Gene expression data)
- Flight Navigation data
- Power plant data and Many more . . .

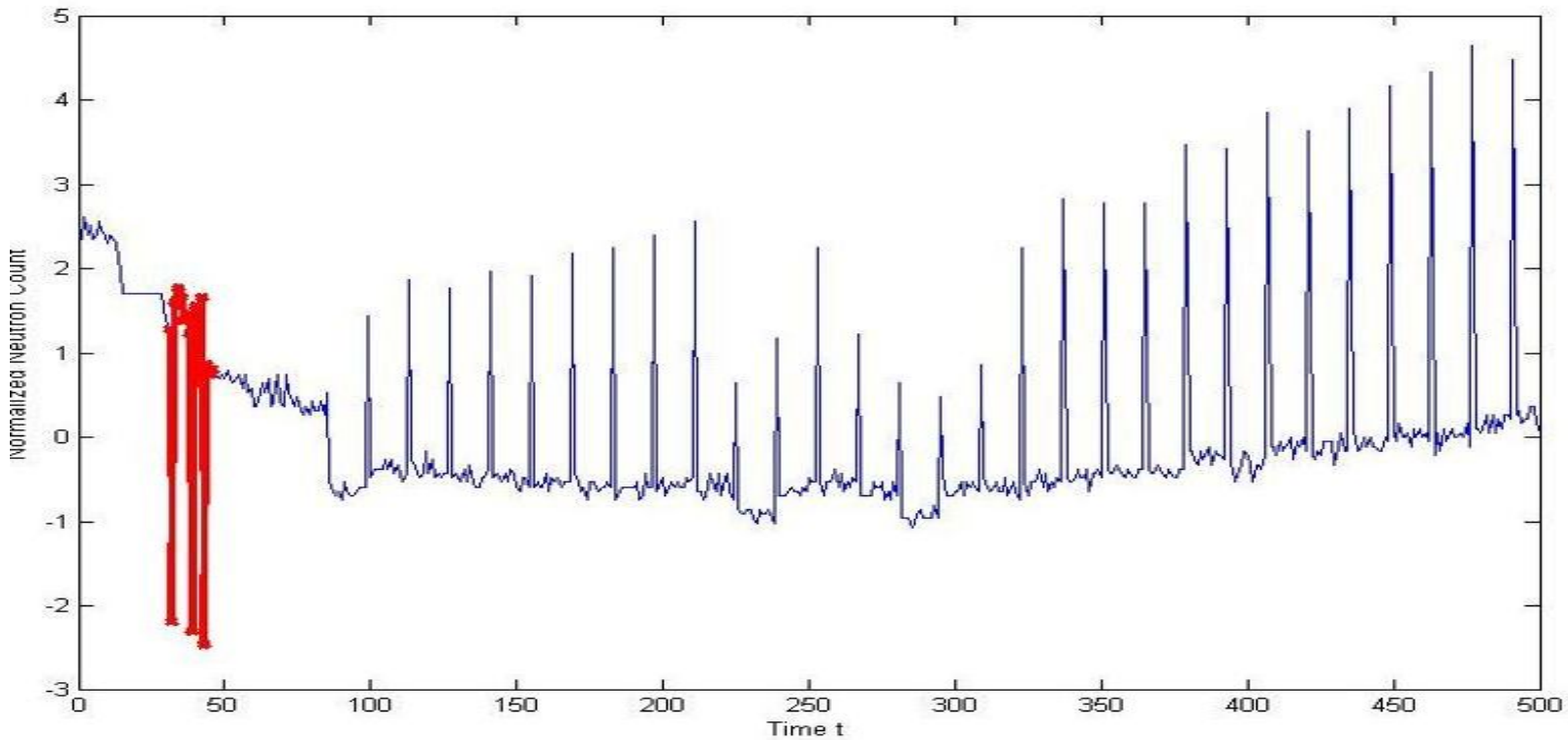
Types of Anomalies

Point Anomaly



Types of Anomalies

Subsequence Anomaly



Types of Anomalies

Contextual Anomaly

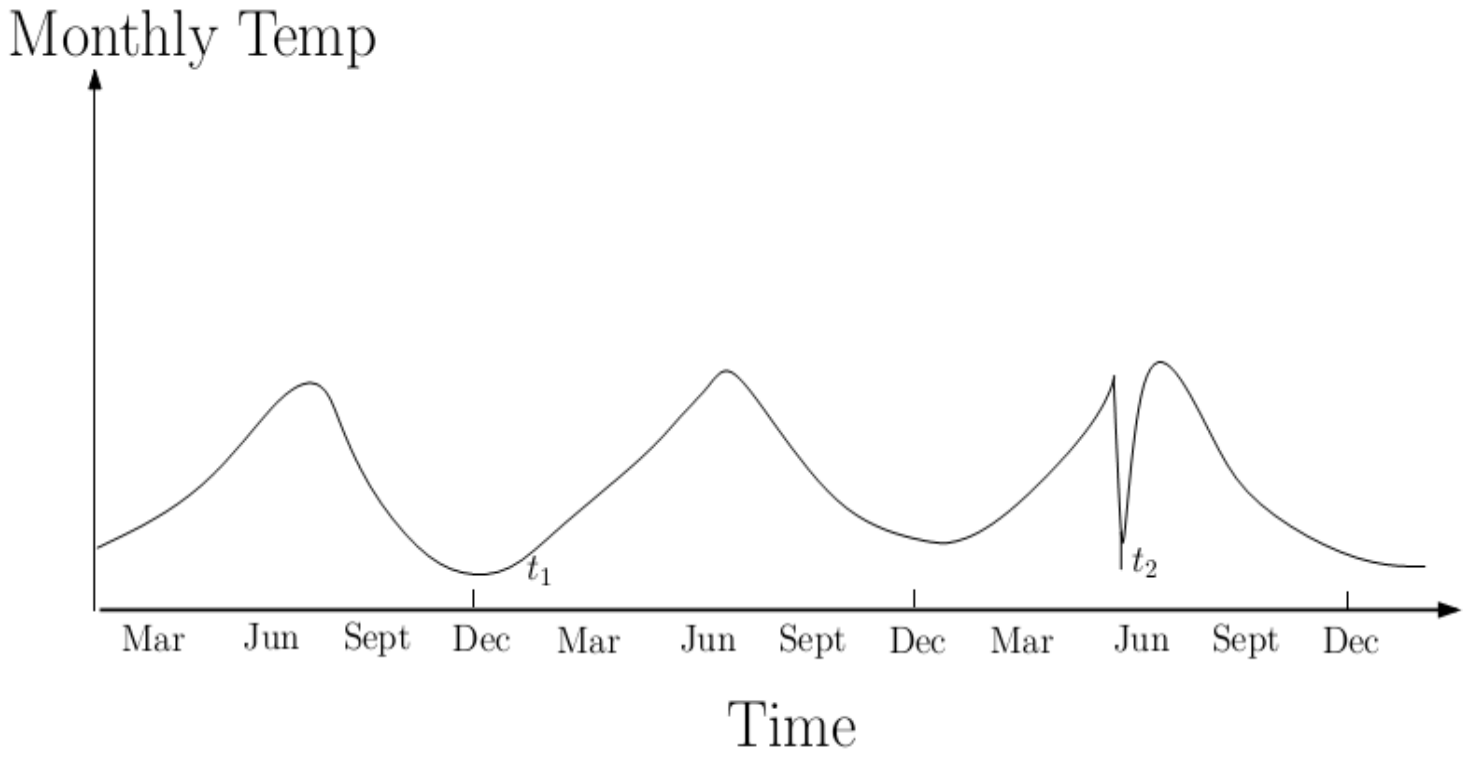


Figure : Source: Chandola et al. *ACM* , 2009

Why anomaly detection in Large data?

Three aspects to look at:–

- **Volume** –Convert 350 billion annual meter readings to better predict power consumption.
- **Velocity** – Scrutinize 5 million trade events created each day to identify potential fraud .
- **Variety** – Large data can be structured and unstructured. For example, sensor, online transaction data, audio, video, click stream etc.

Applications

- Business
- Healthcare
- Computer Security
- Plant Monitoring
- Surveillance
- Satellite Imagery and many more . . .

Approaches to Anomaly Detection

Based on Data Labels

- **Supervised :** – Training data is available for both normal as well as anomalous class.
- **Semi-Supervised :** – Training data has labeled instances only for normal class.
- **Unsupervised :** – Normal instances are available in abundant whereas anomalous instances are rare.

Output of Anomaly Detection Approaches

- Label
- Score

Literature Review- Traditional Approaches

- Classification-Based
- Statistical-Model Based
- Clustering-Based
  - Proximity-Based
  - Density-Based
- Others: Information Theory Based
- Spectral Theory Based

Drawbacks in Traditional approaches :– The Big Picture

Minor

- Statistical techniques –underlying distribution
- Proximity-based and density-based – appropriate metric
- Clustering based techniques – optimization for reducing quadratic time
- Information-theoretic –appropriate measure of information?
- Spectral techniques –embedding dimension?

In addition to above :–

Major

- High dimension , heterogeneity, noisy, streaming, distributed data ?

Recent Approaches to Anomaly Detection

- Non Parametric Techniques
- Multiple Kernel Learning
- Non-Negative Matrix Factorization
- Random Projection Pursuit
- Ensemble based Methods

Research Gaps.....

Major issues with Large data

- Scalability
- Sparsity
- Heterogeneity
- Streaming
- Distributed

Filling the dots with Modern Approaches

- Non-Parametric—Scalability, Heterogeneity, Sparsity, Streaming, Distributed)
- MKL —Scalability, Heterogeneity,Sparsity, Streaming, Distributed
- NNMF—Scalability, Sparsity, Streaming, Distributed, Heterogeneity
- Random Projection——Scalability, Sparsity, Streaming, Distributed, Heterogeneity)

Proposed Research Objectives

Given heterogeneous, sparse, high-dimensional, streaming, distributed data(large data), problem is to identify anomalies efficiently.

Research Objectives

- Anomaly detection in streaming data
- Anomaly detection in streaming, sparse, high dimension data
- Anomaly detection in sparse, high dimension, distributed data

Research Gaps.....Fill the dots

Solving major issues of large data for anomaly detection

- Online Learning $\rightarrow$ Algorithm 1 and 2
- Distributed Learning$\rightarrow$ Algorithm 3 and 4

Benchmark Datasets

Table : Summary of data sets used in the experiment

Dataset	Balance	#Train	# Test	#Features	Sparsity (%)	#Pos:Neg
news20	False	6,598	4,399	1,355,191	99.9682	1:6.0042
rcv1	True	10,000	10,000	47,236	99.839	1:1.0068
url	False	8,000	2,000	3,231,961	99.9964	1:10.0041
realsim	False	56,000	14,000	20,958	99.749	1:1.7207
gisette	False	3,800	1,000	5,000	0 .857734	1:11.667
webspam	False	8,000	2,000	16,609,143	99.9714	1:15
w8a	false	40,000	14,951	300	95.8204	1:26.0453
ijcnn1	false	48,000	91,701	22	39.1304	1:9.2696
covtype	false	2,40,000	60,000	54	0	1:29.5344
pageblocks	false	3,280	2,189	10	0	1::11.1933

Real Datasets

- KDDCup 2008 data set (1,02,887 samples, 117 features)
- Nuclear power plant data set (10,000 samples, 48 features)

Contributions

- Algorithm 1 and 2 solve anomaly detection problem in large-scale streaming data.
- Algorithm 3 and 4 solve anomaly detection problem in large-scale distributed data.
- Algorithm 5 is an application of existing algorithm for anomaly detection in nuclear power plant data.

Online Learning: A motivating example

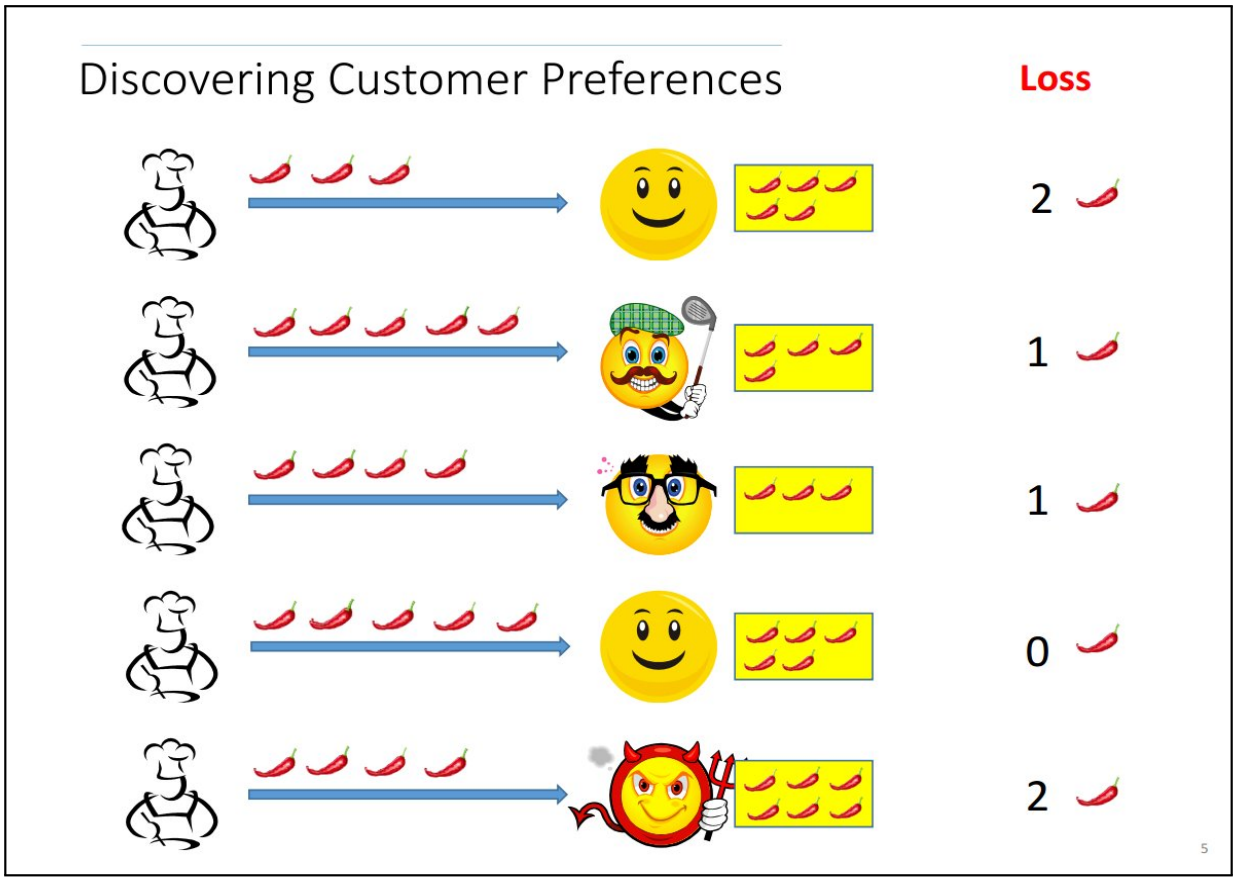


Image credit: Dr. Purushottam Kar, IIT Kanpur

Online Learning: Formalization

Repeat:

- Receive instance: x_t (such as customer profile and his/her transaction history)
- Choose a model w_t predict: $\hat{y}_t = sign(w_t^T x_t)$ (fraudulent or not)
- Receive correct label: y_t (Was transaction indeed genuine?)
- Suffer loss: $\ell(y_t, \hat{y}_t)$ (penalty for wrong prediction)
- Update model w_t

End

Goal: Want to minimize the cumulative number of mistakes.

Our general approach

- Represent anomaly detection problem as class-imbalance learning problem.
- Wish to maximize Gmean metric (why not accuracy?)
- Why maximize *Gmean*? Because *Gmean* is robust to class-imbalance problem [Kubat and Matwin, 1997]

$$Gmean = \sqrt{sensitivity \times specificity} \tag{1}$$

where $sensitivity$ and $specificity$ denote accuracy on positive and negative examples respectively and defined as:

$$sensitivity = \frac{TP}{TP + FN}, specificity = \frac{TN}{TN + FP}$$

Contd...

Lemma

Maximizing Gmean as given in (1) is equivalent to minimizing the following objective:

$$\sum_{y_t=+1} \frac{N}{P} I_{(y_t f_t(\mathbf{x_t})<0)} + \sum_{y_t=-1} \frac{P - FN}{P} I_{(y_t f_t(\mathbf{x_t})<0)} \tag{2}$$

Proof.

Proof can be found in the paper. □

Objective 1 is non-convex (how?). Convex-relaxation is required.

Contd...

A suitable convex surrogate for (1) is hinge loss. So use slightly modified hinge loss.

$$\ell(\mathbf{w};(\mathbf{x},y)) = \max\left(0, \rho - y\mathbf{w}^T\mathbf{x}\right) \tag{3}$$

where,

$$\rho = \left(\frac{N}{P}\right) I_{(y=1)} + \left(\frac{P-FN}{P}\right) I_{(y=-1)}$$

Loss in (3) upper bounds the loss given in lemma (1). Using loss (3) in online setting, we get **PAGMEAN** algorithm.

Contd...

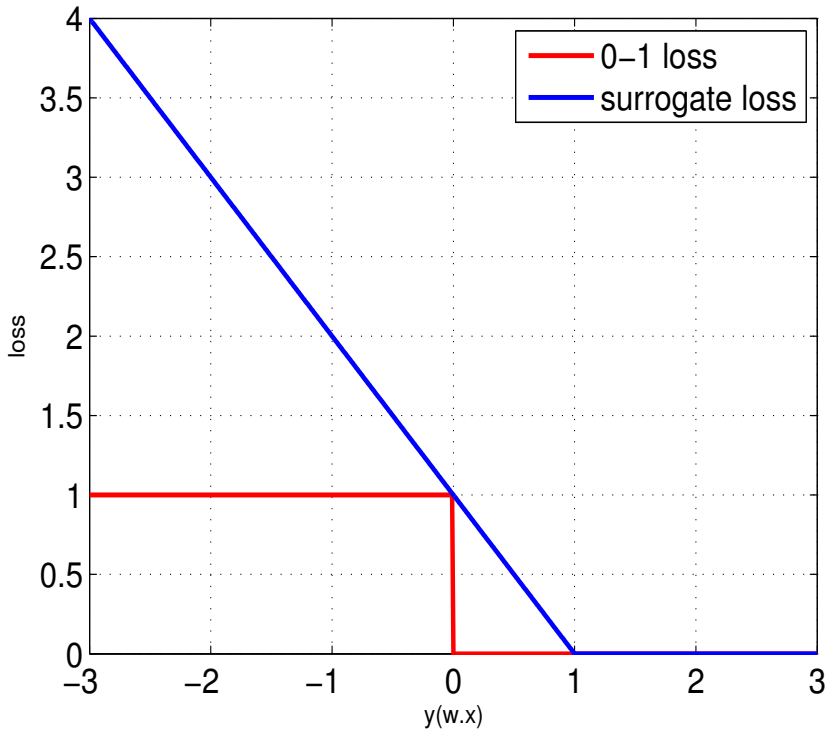


Figure : Modified Hinge loss

Algorithm 3: Distributed Sparse Class Imbalance Learning (DSCIL)

Want to solve the following convex optimization problem in a distributed setting:

$$\underset{\mathbf{w}}{\text{minimize}} \quad \frac{1}{2m}\sum_{i=1}^{m} C_{y_i}\ell(\mathbf{w};\mathbf{x}_i,y_i) + \lambda\|\mathbf{w}\|_1 \tag{6}$$

where, $\ell : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}_+$ is the non-negative convex loss function, $\mathbf{w}$ is the model parameter that we wish to learn; $(\mathbf{x}_i, y_i)$ are example-label pair and $\|\cdot\|_1$ denotes L_1 norm. C_{y_i} denotes the cost associated with the label y_i.

Algorithm 3: Distributed Sparse Class Imbalance Learning (DSCIL)

Loss function

$$\ell(y_i f_i(\mathbf{x}_i)) = C_{y_i} \max\left(0, 1 - y_i f_i(\mathbf{x}_i)\right)^2 \tag{7}$$

where,

$$C_{y_i} = \begin{cases} C_+ & y_i = +1 \\ C_- & y_i = -1 \end{cases}$$

Algorithm 3: Distributed Sparse Class Imbalance Learning (DSCIL)

Algorithm 1: DSCIL

Parameter: ρ

Initialize: $\mathbf{w}_i = \mathbf{0}, \forall i = 1, ..., N, \mathbf{z} = \mathbf{0}, \mathbf{u} = \mathbf{0}$

for $t := 1, ..., T$ **do**

$$\mathbf{w}_i^{t+1} = \underset{\mathbf{w}_i}{\operatorname{argmin}} \quad \ell_i(Y_i \circ X_i \mathbf{w}_i) + \frac{\rho}{2}\|\mathbf{w}_i - \mathbf{z}^t - \mathbf{u}^t\|_2^2 \quad \forall i = 1, ..., N$$

$$\mathbf{z}^{t+1} = \underset{\mathbf{z}}{\operatorname{argmin}} \quad \lambda\|\mathbf{z}\|_1 + \frac{N\rho}{2}\|\mathbf{z} - \bar{\mathbf{w}}^{\mathbf{t+1}} - \bar{\mathbf{u}}^t\|_2^2$$

$$\mathbf{u}_i^{t+1} = \quad \mathbf{u}_i^k + \mathbf{w}_i^{k+1} - \mathbf{z}^{k+1} \tag{9}$$

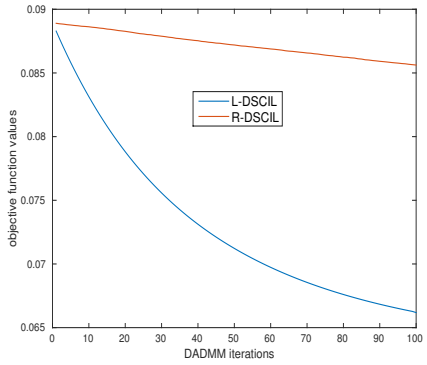
end

Output: $\mathbf{w}^{T+1}$

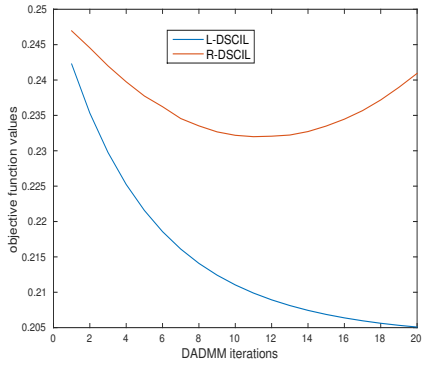
Algorithm 3: DSCIL experimental testbed and setup

- Evaluation metric *Gmean*, *cost*, *speedup*.
- Compared with CSSCD and CSRCD
- All the parameters were found by grid search.

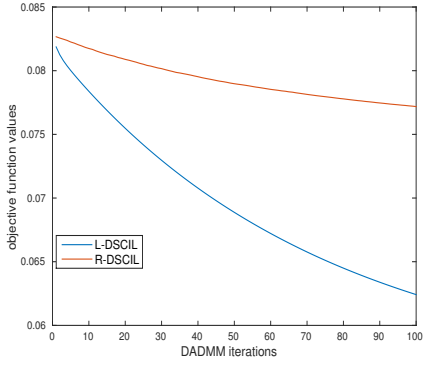
Algorithm 3: DSCIL convergence



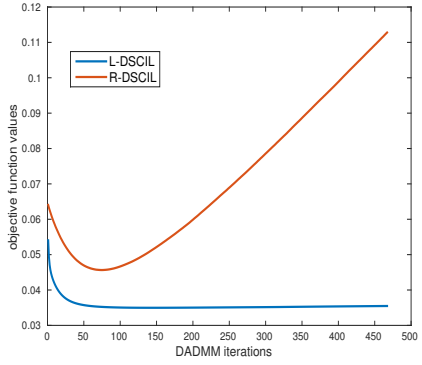
(a) ijcnn1



(b) rcv1



(c) pageblocks



(d) w8a

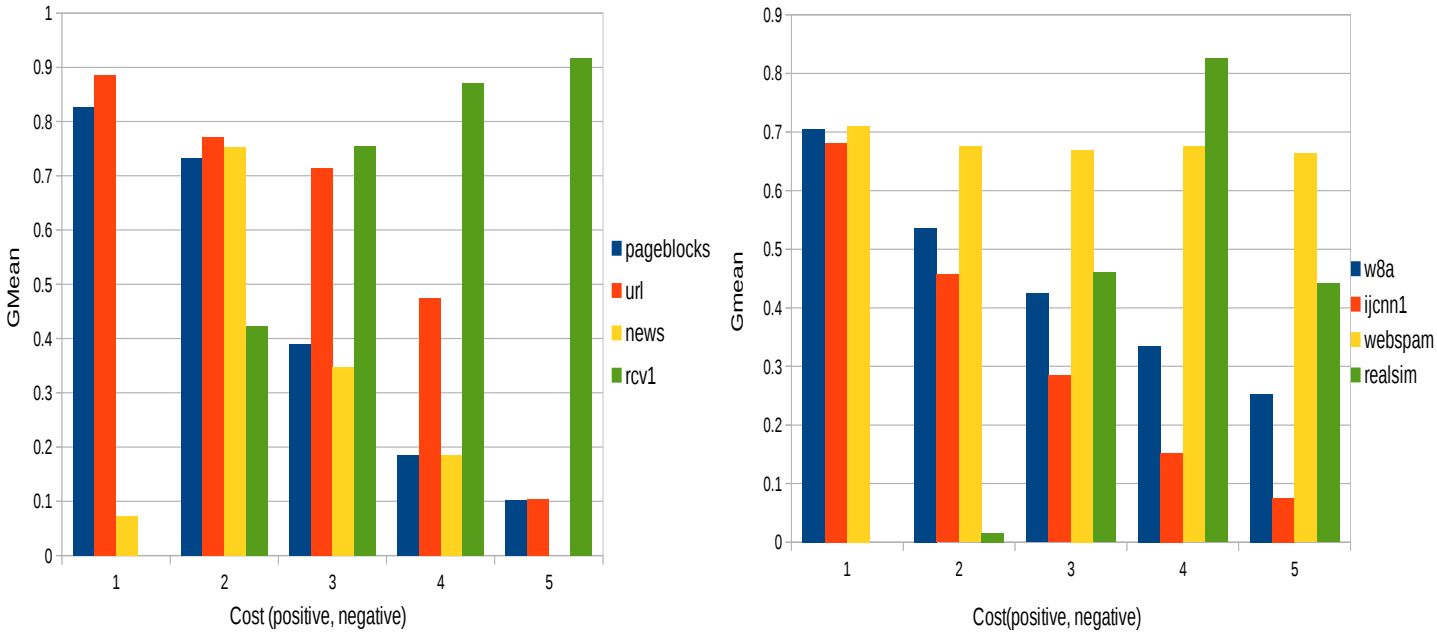
Figure : Objective Function vs DADMM iterations over benchmark data sets. (i) ijcnn1 (ii) rcv1 (iii) pageblocks (iv) w8a

Algorithm 3: DSCIL Performance Comparison with respect to Gmean

TABLE 2: Performance comparison of CSFSOL, L-DSCIL, R-DSCIL and CSSCD over various benchmark data sets

Algorithm	news						rcv1				
	Accuracy	Sensitivity	Specificity	Gmean	Sum		Accuracy	Sensitivity	Specificity	Gmean	Sum
CSFSOL	0.966356	0.982759	0.966137	0.974412	0.974448		0.957	0.960148	0.953312	0.956724	0.95673
L-DSCIL	0.992271	0.568966	0.997927	**0.753516**	0.783446		0.9175	0.925116	0.908578	**0.916809**	0.916847
CSSCD	0.451466	1	0.444137	0.666436	0.722069		0.8989	0.871548	0.930945	0.900757	0.901246
R-DSCIL	0.991134	0.396552	0.999079	**0.629433**	0.697815		0.8697	0.792215	0.960478	0.872299	0.8763465
Algorithm	url						webspam				
	Accuracy	Sensitivity	Specificity	Gmean	Sum		Accuracy	Sensitivity	Specificity	Gmean	Sum
CSFSOL	0.9725	0.994505	0.970297	0.982327	0.982401		0.9925	0.952	0.9952	0.97336	0.9736
L-DSCIL	0.9385	0.824176	0.949945	**0.884829**	0.8870605		0.9705	0.528	1	0.726636	0.764
CSSCD	0.947	0.967033	0.944994	0.95595	0.956014		0.988	0.808	1	0.898888	0.904
R-DSCIL	0.944	0.791209	0.959296	0.871208	0.8752525		0.885	0.968	0.879467	**0.922672**	0.923733
Algorithm	gisette						realsim				
	Accuracy	Sensitivity	Specificity	Gmean	Sum		Accuracy	Sensitivity	Specificity	Gmean	Sum
CSFSOL	0.813	0.896	0.73	0.808752	0.813		0.8825	0	1	0	0.5
L-DSCIL	0.954	0.926	0.982	**0.953589**	0.954		0.812071	0.900304	0.800324	**0.848843**	0.850314
CSSCD	0.945	0.918	0.972	0.944614	0.945		0.9105	0.33617	0.986969	0.576012	0.66157
R-DSCIL	0.5	0	1	0	0.5		0.824286	0.90152	0.814002	**0.856644**	0.857761
Algorithm	ijcnn1						w8a				
	Accuracy	Sensitivity	Specificity	Gmean	Sum		Accuracy	Sensitivity	Specificity	Gmean	Sum
CSFSOL	0.831932	0.607668	0.855475	0.721002	0.731571		0.970637	0.0330396	1	0.181768	0.51652
L-DSCIL	0.868126	0.576217	0.89877	0.719643	0.737493		0.975587	0.682819	0.984755	**0.820006**	0.833787
CSSCD	0.83456	0.827824	0.835267	0.831537	0.831546		0.97231	0.563877	0.9851	0.745302	0.774489
R-DSCIL	0.718258	0.982438	0.690525	**0.823649**	0.836482		0.978597	0.348018	0.998344	0.589442	0.673181
Algorithm	covtype						pageblocks				
	Accuracy	Sensitivity	Specificity	Gmean	Sum		Accuracy	Sensitivity	Specificity	Gmean	Sum
CSFSOL	0.908817	0.86642	0.910199	0.88804	0.888309		0.793056	0.662069	0.81306	0.73369	0.737564
L-DSCIL	0.953617	0.699578	0.961897	0.820318	0.830737		0.919598	0.706897	0.95208	**0.820379**	0.829488
CSSCD	0.968433	0	1	0	0.5		0.887163	0.751724	0.907846	0.826105	0.829785
R-DSCIL	0.968433	0	1	0	0.5		0.90772	0.344828	0.993681	0.585362	0.669254

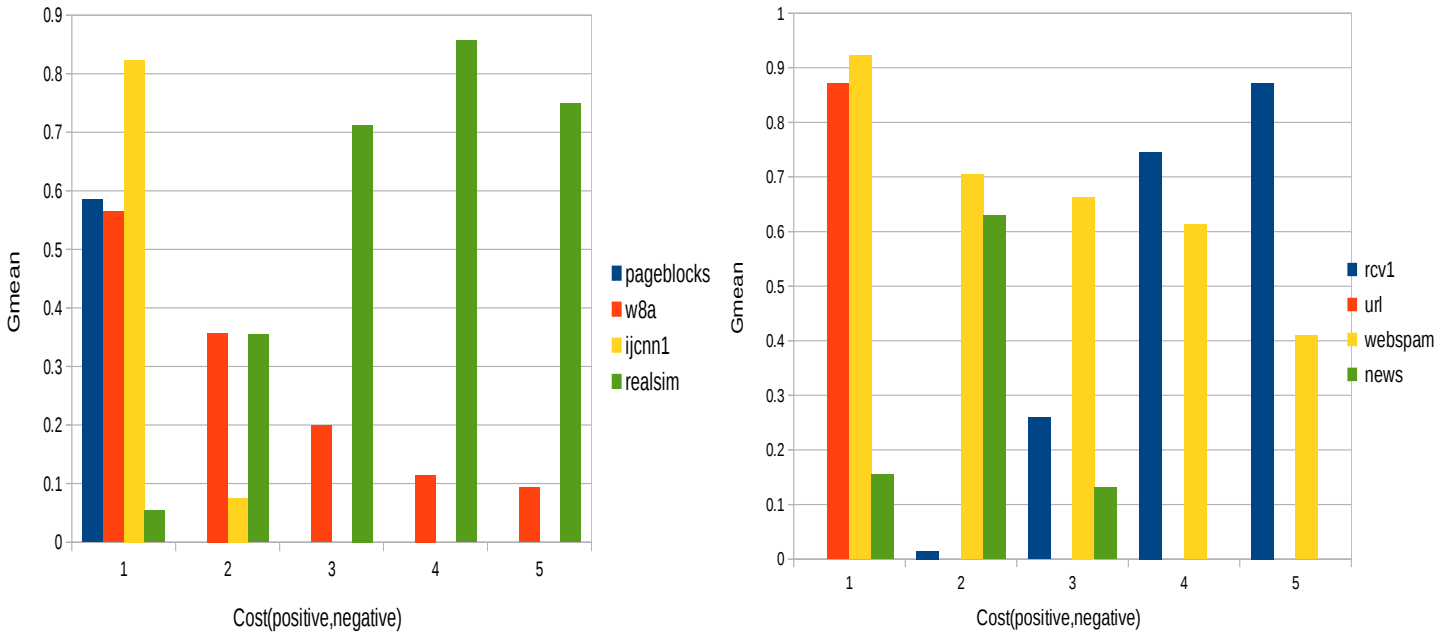
Algorithm 3: DSCIL Gmean versus Cost



(a) (b)

Figure : Gmean versus Cost over various data sets for L-DSCIL algorithm. Cost is given on the x-axis where each number denotes cost pair such that 1={0.1,0.9}, 2={0.2,0.8}, 3={0.3,0.7}, 4={0.4,0.6}, 5={0.5,0.5}

Algorithm 3: DSCIL Gmean versus Cost



(a) (b)

Figure : Gmean versus Cost over various data sets for R-DSCIL algorithm. Cost is given on the x-axis where each number denotes cost pair such that 1={0.1,0.9}, 2={0.2,0.8}, 3={0.3,0.7}, 4={0.4,0.6}, 5={0.5,0.5}

Algorithm 3: L-DSCIL speedup measurement

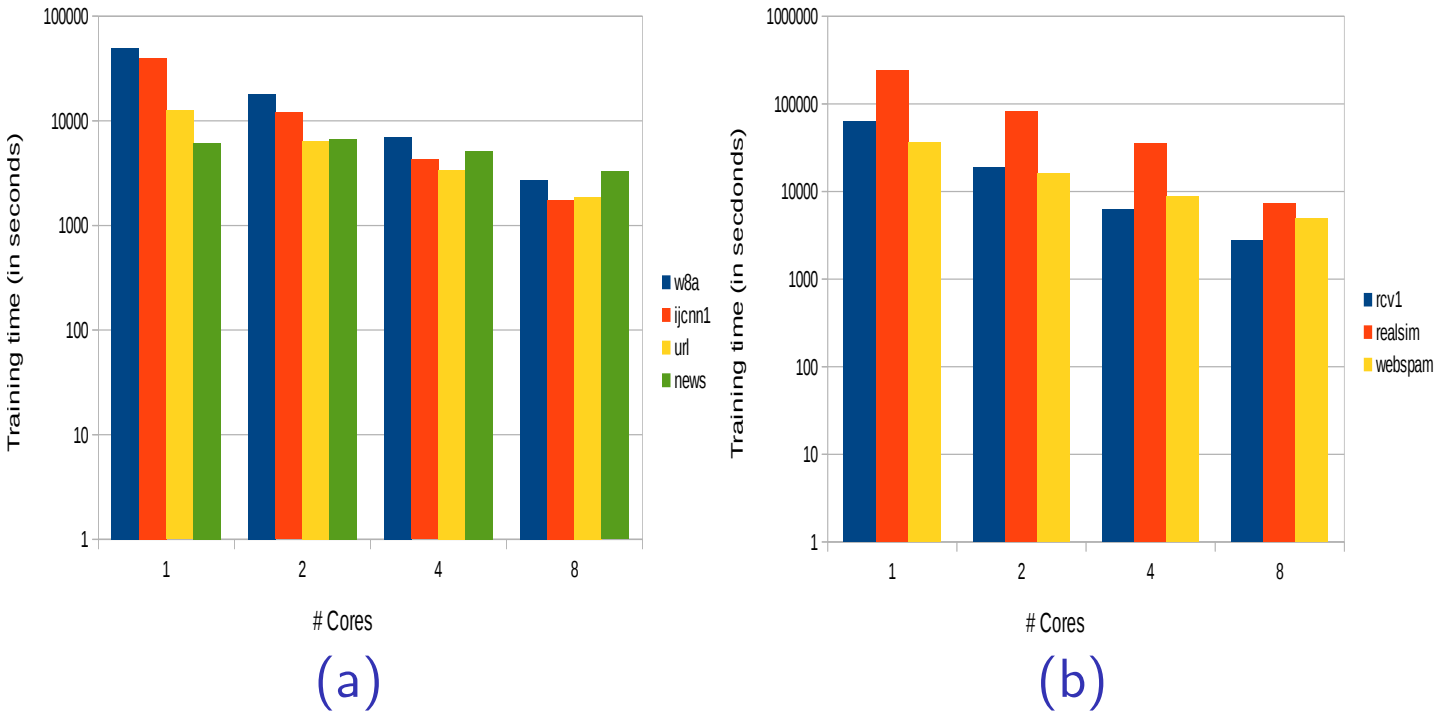


Figure : Training time versus number of cores to measure the speedup of L-DSCIL algorithm. Training time in both the figures is on the log scale.

Algorithm 3: R-DSCIL speedup measurement

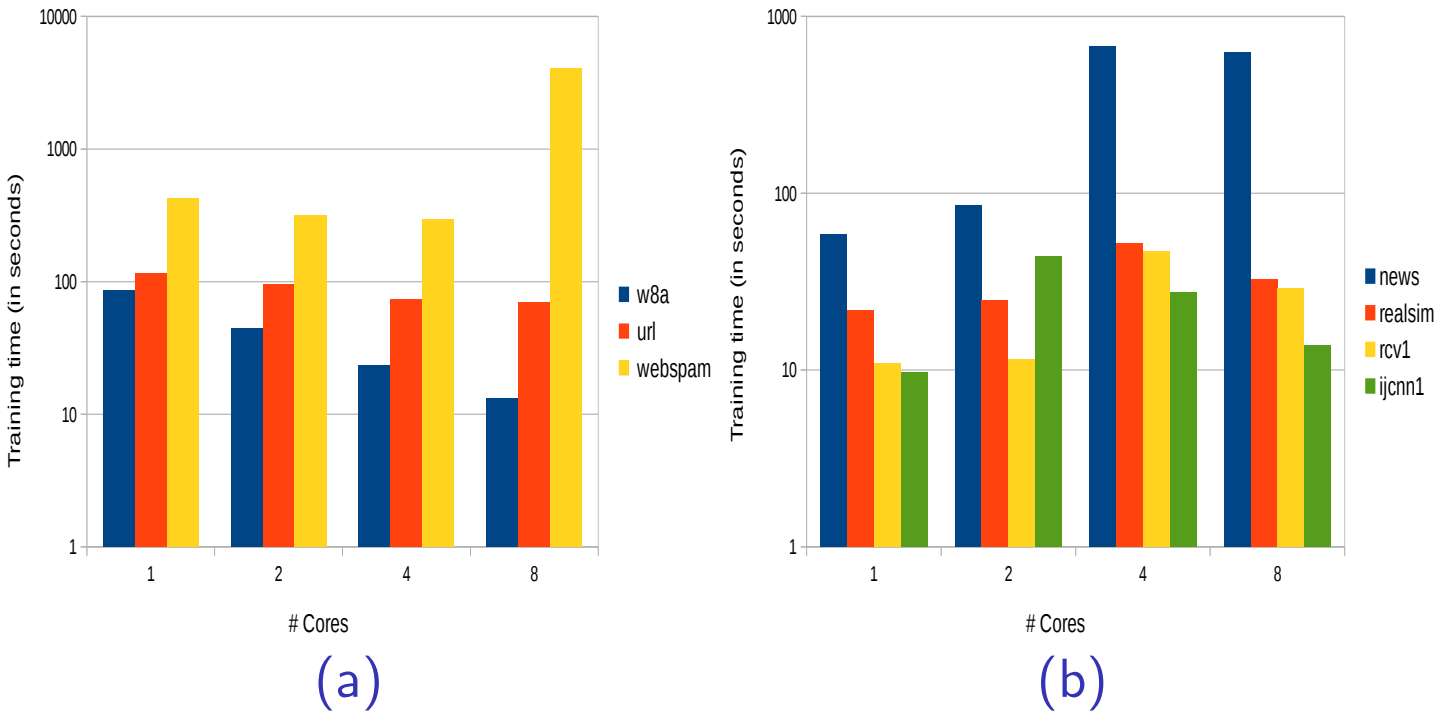


Figure : Training time versus number of cores to measure the speedup of R-DSCIL algorithm. Training time in Figure (a) is on the log scale.

Algorithm 3: DSCIL number of Cores versus Gmean

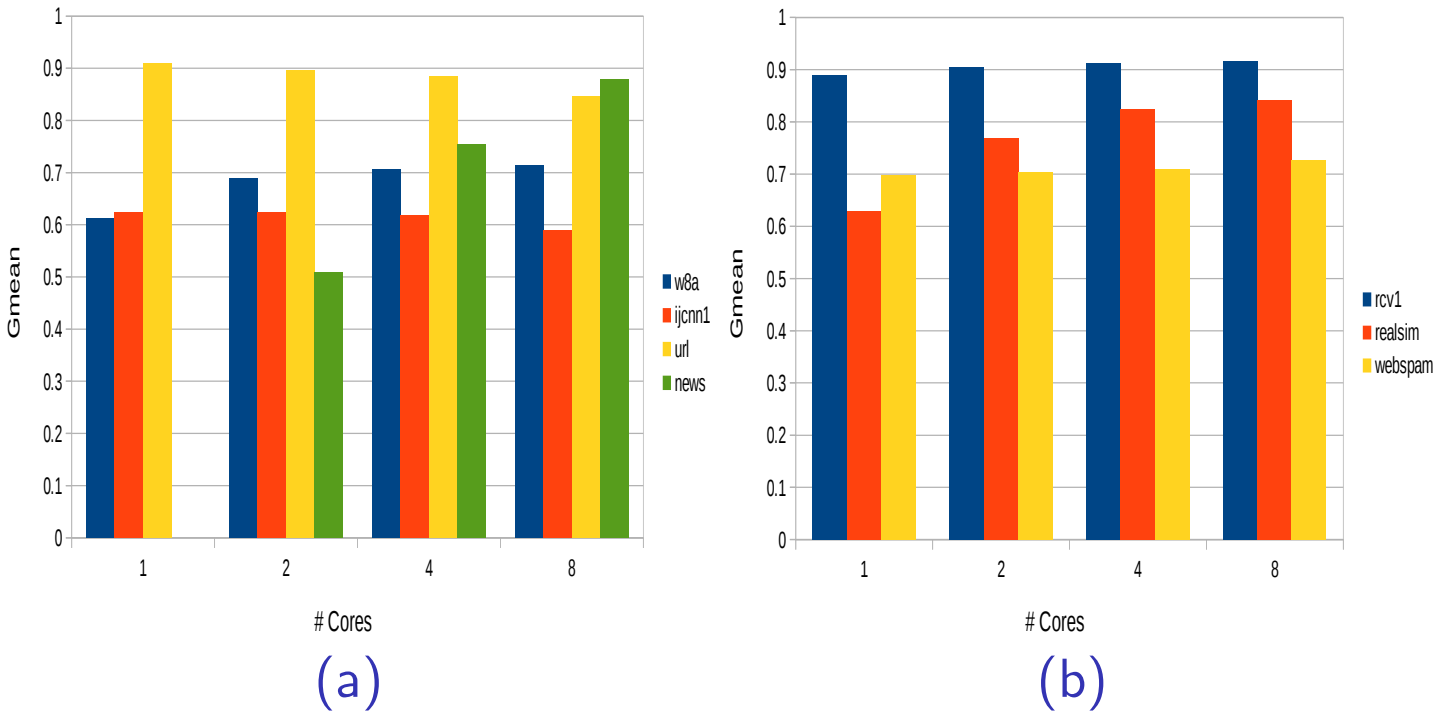


Figure : Effect of varying number of cores on *Gmean* in L-DSCIL algorithm.

Algorithm 3: DSCIL number of Cores versus Gmean

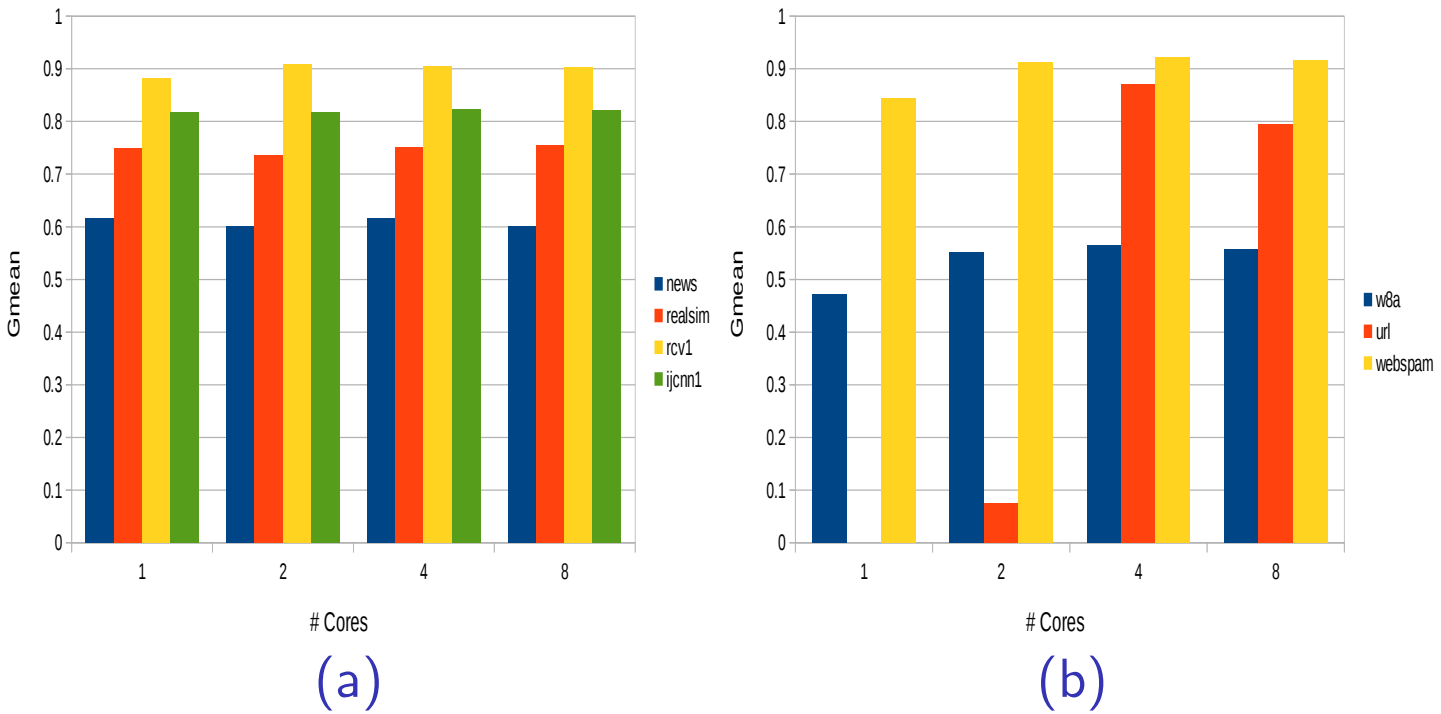


Figure : Effect of varying number of cores on *Gmean* in R-DSCIL algorithm.