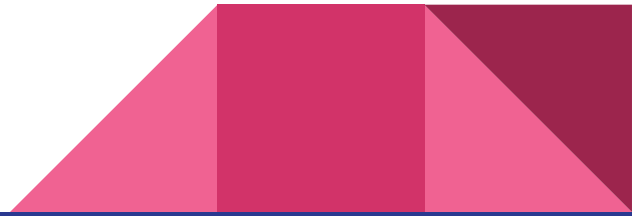
Team no. 65
Reduction from Cost-sensitive Multiclass Classification to one-vs-one Binary Classification

- Ashwin P (20162040)
- Damodar Dukle (20162055)
- Ravi Prakash (20162028)
- Varun Mundale (20162011)

Mentor - Vatika Harlalka

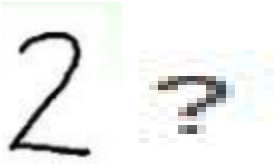
Background

- Multiclass classification
- One-vs-one classification
- Cost-sensitive classification



Cost-Sensitive Multiclass classification:

Which digit did you write ?



1 (one)

2(two)

3 (three)

4 (four)

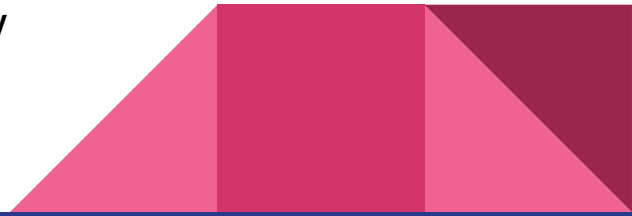
A Multiclass classification problem of classifying instances into one of the more than two classes

Performance evaluation:

- ZIP code recognition:
  1: wrong 2: right 3: wrong 4: wrong
- Check value recognition:
  1: one-rupee mistake 2: no mistake 3: one-rupee mistake 4: two-rupee mistake
- Evaluation by writing similarity:
  1: not very similar 2: very similar 3: somewhat similar 4: a silly prediction

Different applications:

Evaluate miss-predictions differently



Cost Vector:

Cost vectors **c**: a row of cost components

- Absolute cost for digit 2: **c** = (1, 0, 1, 2)
- Regular classification cost for label 2: $c_c^{(2)}$ = (1, 0, 1, 1)
- An HIV infected patient:
  = (HIV infected: 0, viral fever: 100, healthy: 10000)

c

Reduction of multiclass classification to multiple binary classification problems:

- **One vs rest:** The one vs rest strategy involves training a single classifier per class, with the samples of that class as positive samples and all other samples as negative
- **One vs one**: one trains K (K − 1) / 2 binary classifiers for a K-way multi class problem; each receives the samples of a pair of classes from the original training set, and must learn to distinguish these two classes. At prediction time, a voting scheme is applied: all K (K − 1) / 2 classifiers are applied to an unseen sample and the class that got the highest number of "+1" predictions gets predicted by the combined classifier

Cost generation procedure:

Randomized proportional (RP) cost generation procedure that was used by Beygelzimer et al (2005) is used here.

In particular, we generate the cost vectors from a cost matrix C(y,k) that does not depends on x. The diagonal entries C(y,y) are set as 0 and each of the other entries C(y,k) is a random variable sampled uniformly from

$$\left[0, 2000 \frac{|\{n: y_n = k\}|}{|\{n: y_n = y\}|}\right]$$

Metrics:

Simple Accuracy

No. of correctly classified samples / Total no. of samples

Cost-sensitive Accuracy

1- (total cost of misclassification) / (total maximum misclassification cost)

Mean cost(Expected cost)

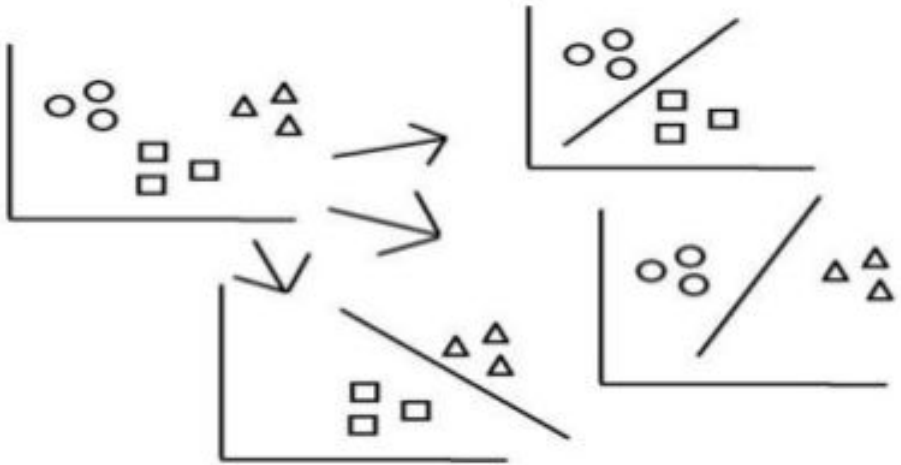
Average (total cost of misclassification) over k iterations.

Standard deviation

S.D (total cost of misclassification) over k iterations.

One-vs-one classification (OVO):

In a one vs one strategy we would divide the data sets into pairs (like shown below) and then do classification.Note that one hypothesis separates only two classes irrespective of the other classes.



OVO Performance

Dataset	OVO			
	OVO Weighted Accuracy	OVO Simple Accuracy	OVO Mean cost	OVO Cost SD
IRIS	0.9715	0.9552	51.2019	79.2083
Digits	0.9956	0.9926	8.6974	10.7017
Glass	0.9519	0.4907	567.7087	129.89
Vowels	0.8451	0.759	308.4642	61.4984
zoo	0.9997	0.998	2.543	11.0848
breast	0.9337	0.8993	135.3452	120.6874
vehicle	0.8277	0.6533	296.6505	43.7233
yeast	0.9949	0.5788	764.3932	203.3211

CSOVO Algorithm:

I. Training Set Expansion and Weighting

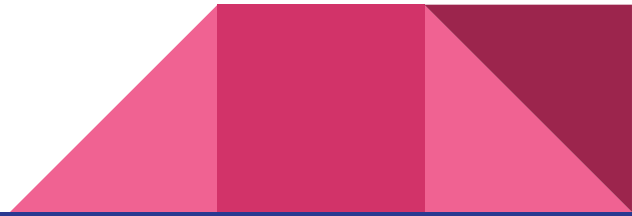
1. Obtain NK training examples Sw

   Convert each sample (x,label,cost vector) to (x,relabel,weight) for each class

   To remove ambiguity put only those examples

2. Weighted classification algorithm to obtain gw.

3. Return gw



CSOVO Algorithm (contd.):

II . Weighted Binary classification

Decomposition of multiclass problem to K(K-1)/2 binary classification subtasks.

For every i,j classes

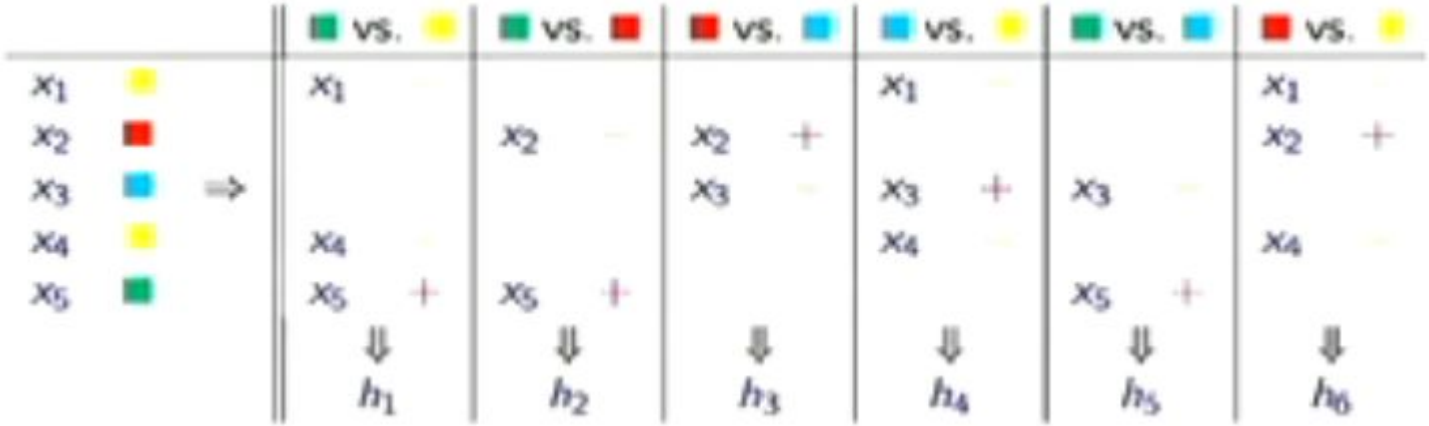
III. Prediction

Return g(x) =maximum l such that sample is classified in favour of l.

CSOVO Performance:

Dataset	CSOVO			
	CSOVO Weighted Accuracy	CSOVO Simple Accuracy	CSOVO Mean cost	CSOVO Cost SD
IRIS	0.9963	0.6618	5.659	0.9266
Digits	0.9532	0.6962	93.8455	12.4001
Glass	0.9815	0.3805	154.2676	22.7826
Vowels	0.7283	0.1689	537.9881	30.2003
zoo	0.9996	0.9173	6.1601	3.877
breast	0.984	0.8755	26.5592	9.5169
vehicle	0.85	0.3823	221.9028	61.2436
yeast	0.9944	0.4167	289.8259	21.4859

Weighted All Pairs:



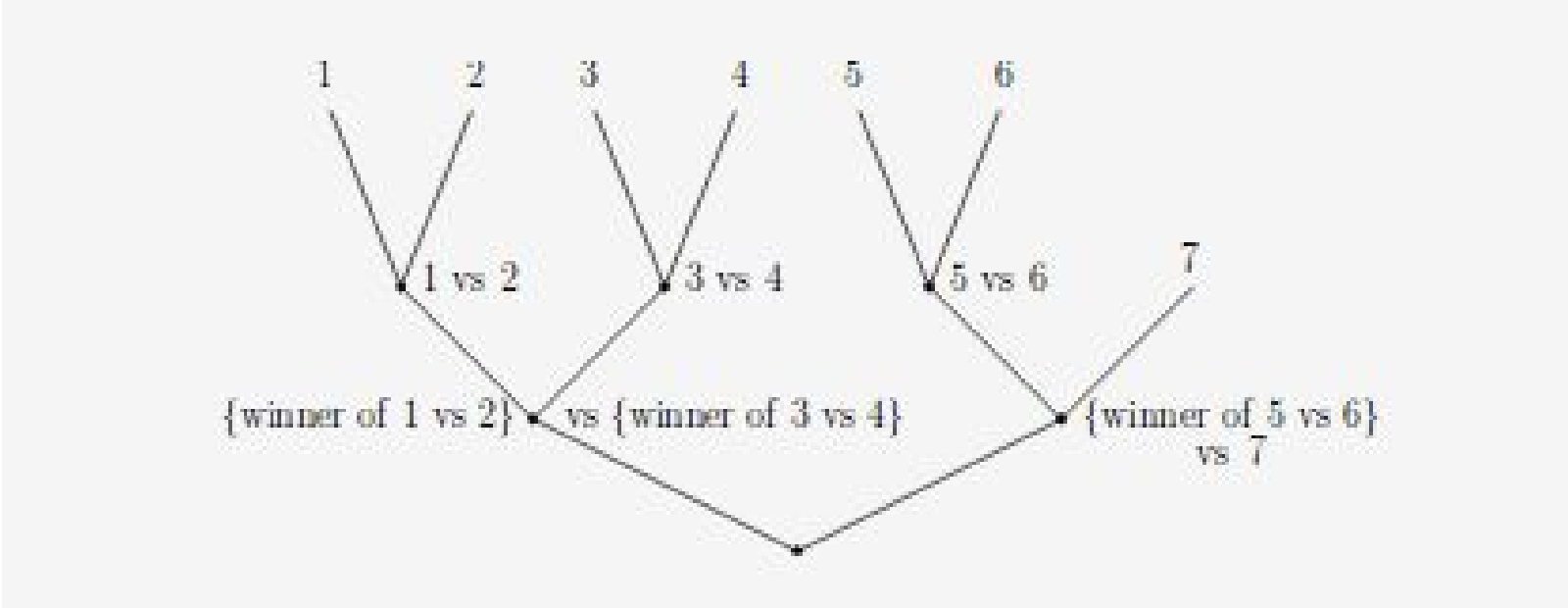
$$\mathcal{S}_b^{(i,j)} = \left\{ \left(\mathbf{x}_n, \underset{\ell=i \text{ or } j}{\operatorname{argmin}} \, \mathbf{c}_n[\ell] \, , \left| \mathbf{v}_n[i] - \mathbf{v}_n[j] \right| \right) \right\}$$

where $\mathbf{v}_n[i] = \int_{c_{\min}}^{\mathbf{c}_n[i]} \frac{1}{|\{k : \mathbf{c}_n[k] \leq t\}|} dt$

Performance of WAP:

Dataset	WAP			
	WAP Weighted Accuracy	WAP Simple Accuracy	WAP Mean cost	WAP Cost SD
IRIS	0.9822	0.946	32.8651	25.6876
Digits	0.9584	0.7191	84.5049	9.454
Glass	0.9594	0.3638	589.8356	104.8734
Vowels	0.775	0.1391	449.7351	26.3484
zoo	0.9955	0.9173	6.532	3.6502
breast	0.9268	0.9094	122.796	39.996
vehicle	0.9128	0.4662	139.4282	49.297
yeast	0.9961	0.4094	348.1403	41.459

All-Pair-Filter Tree:



Kernel Perceptron:

- A kernel machine is a classifier which stores a number of its samples(m) and corresponding weights and makes decision based on them when classifying an unknown sample.
- Here alpha is misclassification count
- In practice we replace dot(x[i],x)
- with Kernel function K(x[i],x)

$$\hat{y} = \operatorname{sgn}(\mathbf{w}^{\mathsf{T}}\mathbf{x})$$
$$= \operatorname{sgn}\left(\sum_{i}^{n} \alpha_i y_i \mathbf{x}_i\right)^{\mathsf{T}} \mathbf{x}$$
$$= \operatorname{sgn}\sum_{i}^{n} \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{x})$$

CSOVO on different classifiers

CSOVO Performance for various classifiers under digits dataset				
Classifier Algorithm	**Average Cost Weighted Accuracy**	**Average Simple Accuracy**	**Average Mean Cost**	**Average Binary Classifier Accuracy**
Simple Perceptron	95.91%	63.19%	85.5009424923	86.30%
Voted Perceptron	94.17%	79.56%	118.129093928	90.19%
Kernel Perceptron	85.24%	28.22%	363.911210699	NA
SVM	73.73%	16.22%	520.426037693	70.11%

Comparison of Algorithms

Dataset	CSOVO	OVO	WAP
	Accuracy	Accuracy	Accuracy
IRIS	0.9963	0.9715	0.9822
Digits	0.9532	0.9956	0.9584
Glass	0.9815	0.9519	0.9594
Vowels	0.7283	0.8451	0.775
zoo	0.9996	0.9997	0.9955
breast	0.984	0.9337	0.9268
vehicle	0.85	0.8277	0.9128
yeast	0.9944	0.9949	0.9961
Dataset	CSOVO	OVO	WAP
	Mean	Mean	Mean
IRIS	5.659	51.2019	32.8651
Digits	93.8455	8.6974	84.5049
Glass	154.2676	567.7087	589.8356
Vowels	537.9881	308.4642	449.7351
zoo	6.1601	2.543	6.532
breast	26.5592	135.3452	122.796
vehicle	221.9028	296.6505	139.4282
yeast	289.8259	764.3932	348.1403

Cost-Weighted Neural Network

Dataset	CWNN		
	CWNN Weighted Accuracy	CWNN Simple Accuracy	CWNN Mean cost
IRIS	0.9579	0.9473	0.4462
Digits	0.6628	0.1644	3.4359
Glass	0.9424	0.3148	3.385
Vowels	0.6776	0.3508	3.4603
zoo	0.9846	0.6923	0.6299
breast	0.8214	0.6503	1.1166
vehicle	0.5132	0.2311	4.9581
yeast	0.9964	0.3153	2.3901

Thank you