

**ACTIVE LEARNING IN GP:  
THE DYNAMIC SUBSET SELECTION  
FAMILY**

Malcolm Heywood

# CONTEXT

- Computational expense of inner loop
  - Case of multi-class classification
    - $\#Evals = \#Class \times \#Trials \times \#Gen \times Pop\_Size \times |TD|$
  - where
    - $\#Class \rightarrow$  Num. Classes - 1
    - $\#Trials \rightarrow$  Num. Population initializations or 'Trials'
    - $\#Gen \rightarrow$  Num. Generations or stop criterion
    - $Pop\_Size \rightarrow$  Num. Individuals in the population
    - $|TD| \rightarrow$  Num. Exemplars in a training partition

# SOME ILLUSTRATIVE FIGURES

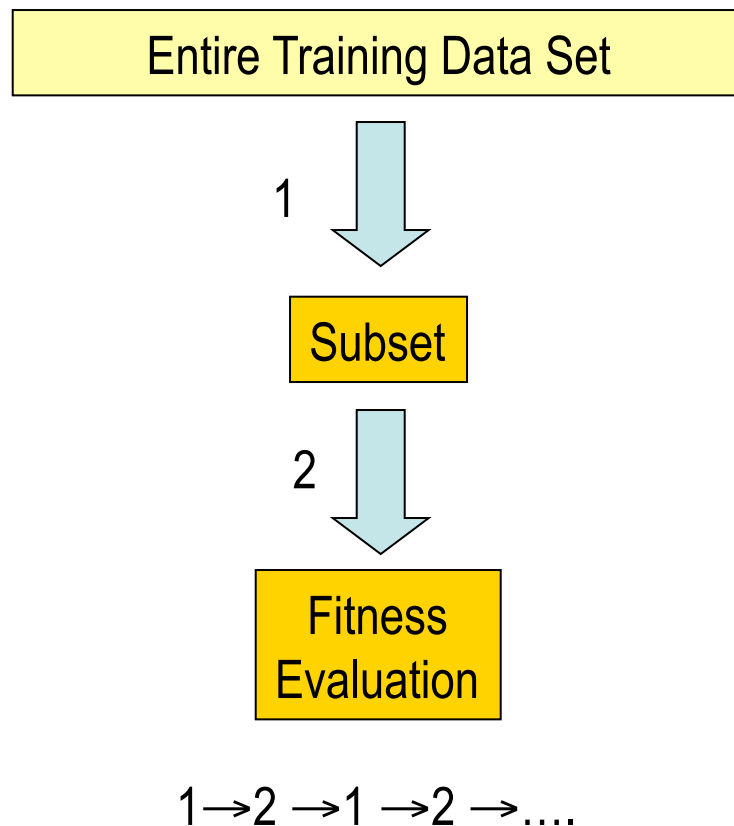
Pop\_Size = 1,000; #Trials = 50; #Generations = 50

Dataset	#Class	TD	#Eval
Census	2	196,294	490,735,000,000
Thyroid	3	3,709	18,545,000,000
KDD'99	5	145,584	1,455,840,000,000
Shuttle	7	43,500	652,500,000,000

# WHAT CAN YOU CONTROL?

- Decreasing
  - #Trials?
  - #Generations?
  - Pop Size?
- Training partition
  - Are all exemplars equally significant?
    - Can the useful subset of exemplars be identified dynamically?
- Case of fixed subset of exemplars (100)
  - Number of evaluations now a **constant**
    - Census 'speedup' = 1,963 x's faster or 4 orders of magnitude
    - Thyroid 'speedup' = 74 x's faster or 2 orders of magnitude...

# SIMPLE SUBSET SELECTION (WEISS AND PROVOST 03)



- Select up to subset size with uniform probability
  - Correlated with **Accuracy**
- Enforce class balance heuristic
  - Correlated with **AUC**
- Model derived under Decision Tree
  - Applicability to GP?

# SIMPLE SUBSET SELECTION PLUS AUC APPROXIMATION

(DOUCETTE AND HEYWOOD 08)

- Wilcoxon-Mann-Whitney (WMW) estimator for AUC
  - Provides direct estimate of AUC
  - Expense mitigated by SSS

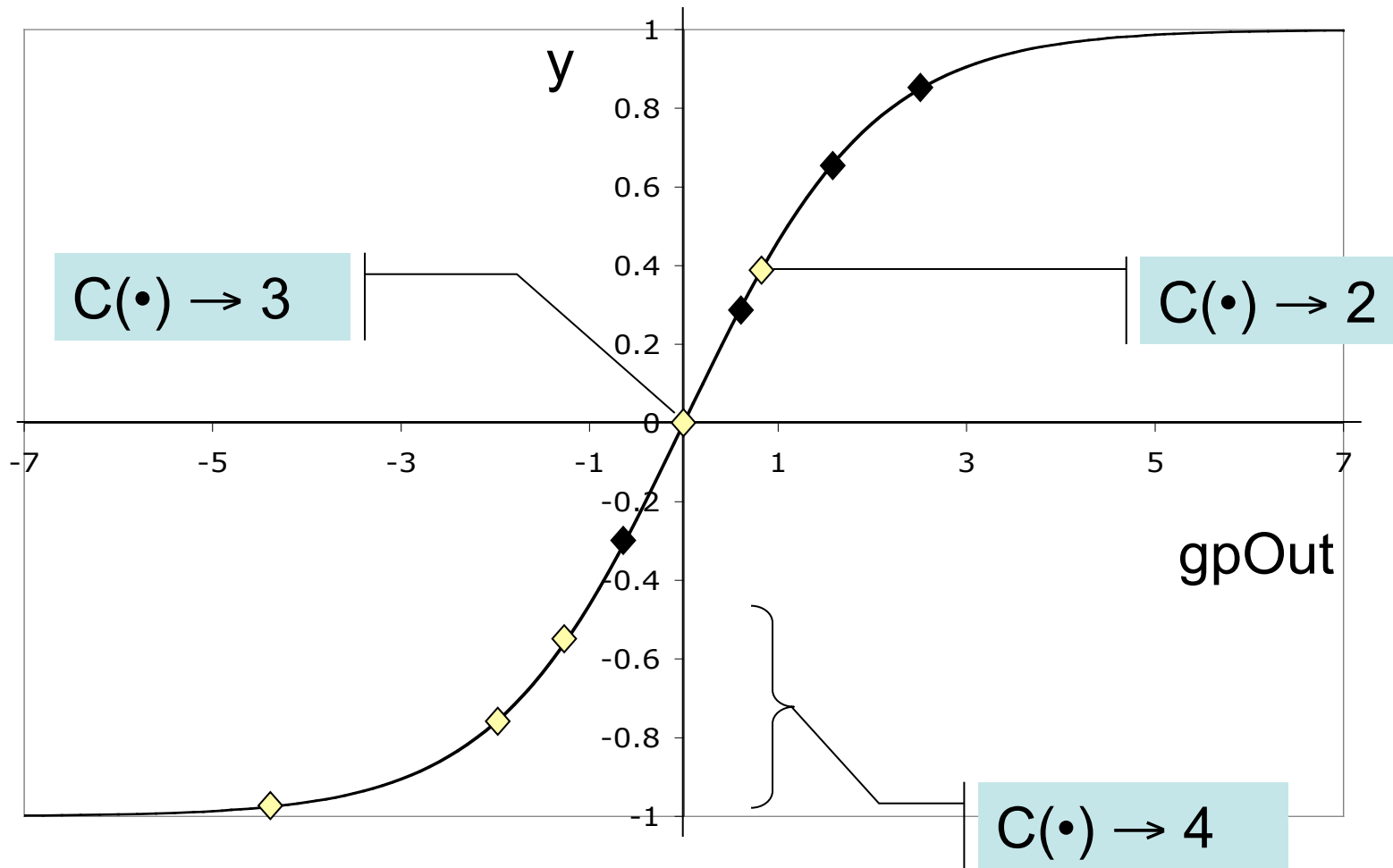
$$\text{WMW} = \sum_{j \in |P|} \sum_{i \in |N|} C(y, P(j), N(j))$$

IF  $y(a) > y(b)$   
THEN  $C(y, a, b) = 1$  ELSE  $C(y, a, b) = 0$

– Where

- P and N are the positive and negative class exemplars from the current SSS partition;
- $y(x)$  is the activation function (typically sigmoid or threshold)

# ESTIMATING WMW STATISTIC



# DYNAMIC SUBSET SELECTION (1)

## (GATHERCOLE AND ROSS 94)

- Exemplar 'difficulty' and 'age' heuristics
  - Exemplar age
    - IF exemplar(i) NOT selected
    - THEN Age(i) += 1
    - ELSE Age(i) = 0
  - Exemplar difficulty
    - Difficulty(i) += error(i)
    - Where 'error(i)' is the interaction between exemplar(i) and individual under evaluation
      - E.g. under the classification domain,
        - » Difficulty(i)  $\propto$  count of incorrect classifications

# DYNAMIC SUBSET SELECTION (2)

- Select exemplars probabilistically,

$$\text{Exemplar}(i).\text{weight} = \text{Diff}(i)^a + \text{Age}(i)^b$$

$$P(\text{select}) = \frac{\text{Exemplar}(i).\text{weight} \times SS}{\sum_{k \in TS} \text{Exemplar}(k).\text{weight}}$$

– Where

- ‘a’ and ‘b’ establish the relative weight of Difficulty and Age (1 and 3.5);
- SS is the subset size

- Potential drawbacks

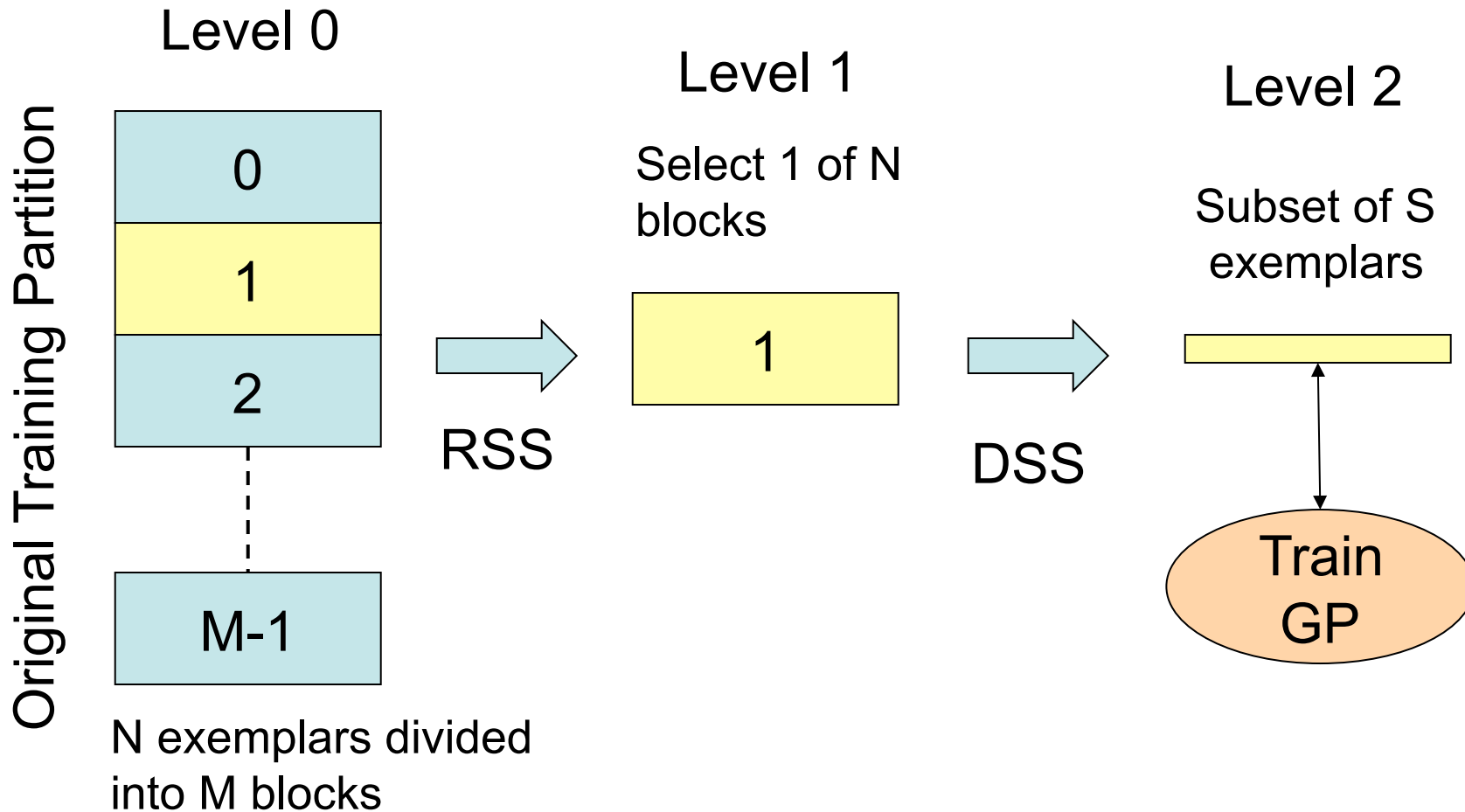
- Overhead in maintaining a record of ‘Exemplar.weight’
- No class balancing

# HIERARCHICAL DSS (1)

## (SONG *ET AL* 05)

- Divide training partition into stratified ‘blocks’
  - Block < cache capacity
- Select block with uniform probability
- DSS determines Subset for training epoch
  - Age and Difficulty data recorded at block
  - Trade off in number Subsets sampled per block
    - Over learning on block content
    - Versus
    - Cache consistency
- Problems under very imbalanced data

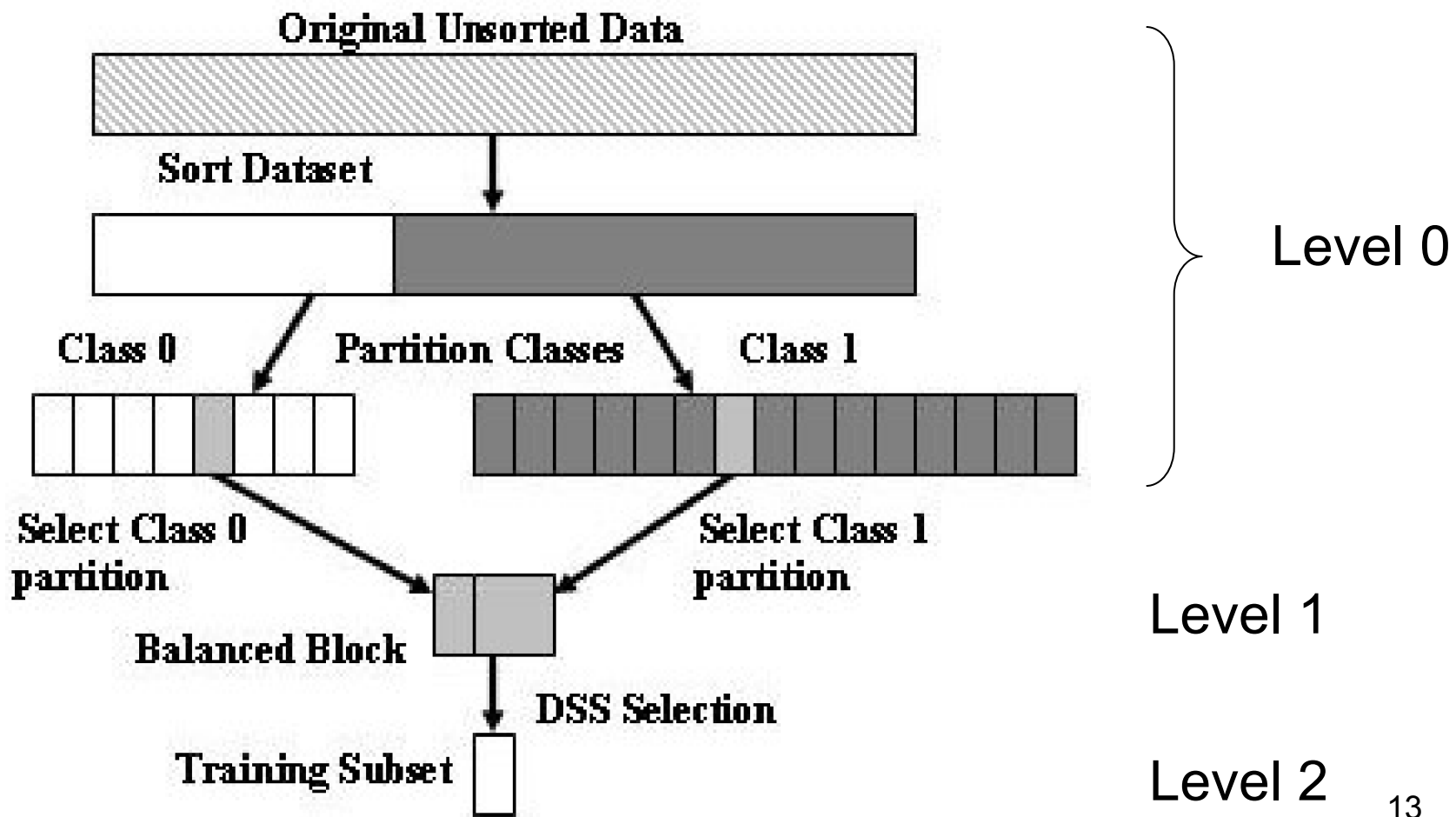
# HIERARCHICAL DSS (2)



1. Stratify training data
2. While (stop criterion == False)
  1. Select Block with uniform probability
  2. While (stopDSSselect == False)
    1. IF (DSSiteration MOD (SubsetFreq))
      1. THEN DSS subset selection
    2. Select candidate for S.S. tournament
    3.  $\forall$  Members  $\in$  Tournament
      1. Establish fitness over content of subset
      2. Update exemplar difficulty
    4. Identify Parents
    5. Apply search operators
    6.  $\forall$  exemplars  $\in$  Block  $\&\& \notin$  SS
      1. exemplar.Age++
  3. Update Block.Error
    1. stopDSSselect = MaxDSSiter  $\times$  Block.Error

# BALANCED BLOCK H-DSS (1)

(CURRY *ET AL* 07)



# BALANCED BLOCK H-DSS (2)

## PARTITION SELECTION

- Partition difficulty and age

$$P(\text{SelectPartition}(i)) = \frac{\text{Part.weight}(i)}{\sum_j \text{Part.weight}(j)}$$

$$\text{Part.weight} = \frac{\text{diff} \times \text{Part.diff}(i)}{\sum_j \text{Part.diff}(j)} + \frac{\text{age} \times \text{Part.age}(i)}{\sum_j \text{Part.age}(j)}$$

– Where

- diff is an a priori fixed weighting of ‘difficulty’ relative to age
- Age = 1 - diff

# BALANCED BLOCK H-DSS (3)

## PARTITION DIFFICULTY

- Part.diff(i)  $\rightarrow$  exponentially weighted avg.
  - Take value from previous participation
$$\text{TempP.diff}(i, 0) = \text{Part.diff}(i - 1)$$
  - Update exemplar-wise
$$\text{TempP.diff}(i, t) = \alpha \text{exemplarDiff}(t) + (1 - \alpha) \text{TempP.diff}(i, t - 1)$$
  - Update Part.diff
$$\text{Part.diff}(i) = \text{TempP.diff}(i, t)$$

# BALANCED BLOCK H-DSS (4)

## SUBSET SAMPLING LIMIT

- Subset Sampling Limit (level 2)
  - Sample  $\propto$  weighted error of partitions comprising a block
    - $0.5 \text{ MaxSamples} \sum_{p \in \text{Block}} E_p(i_p - 1)$
    - Where
      - $i_p$  is the partition selection count
      - $E_p$  is the normalized partition error  $[0, 1]$
      - MaxSamples defines sample limit for worst case error

# BENCHMARKING H-DSS VARIANTS

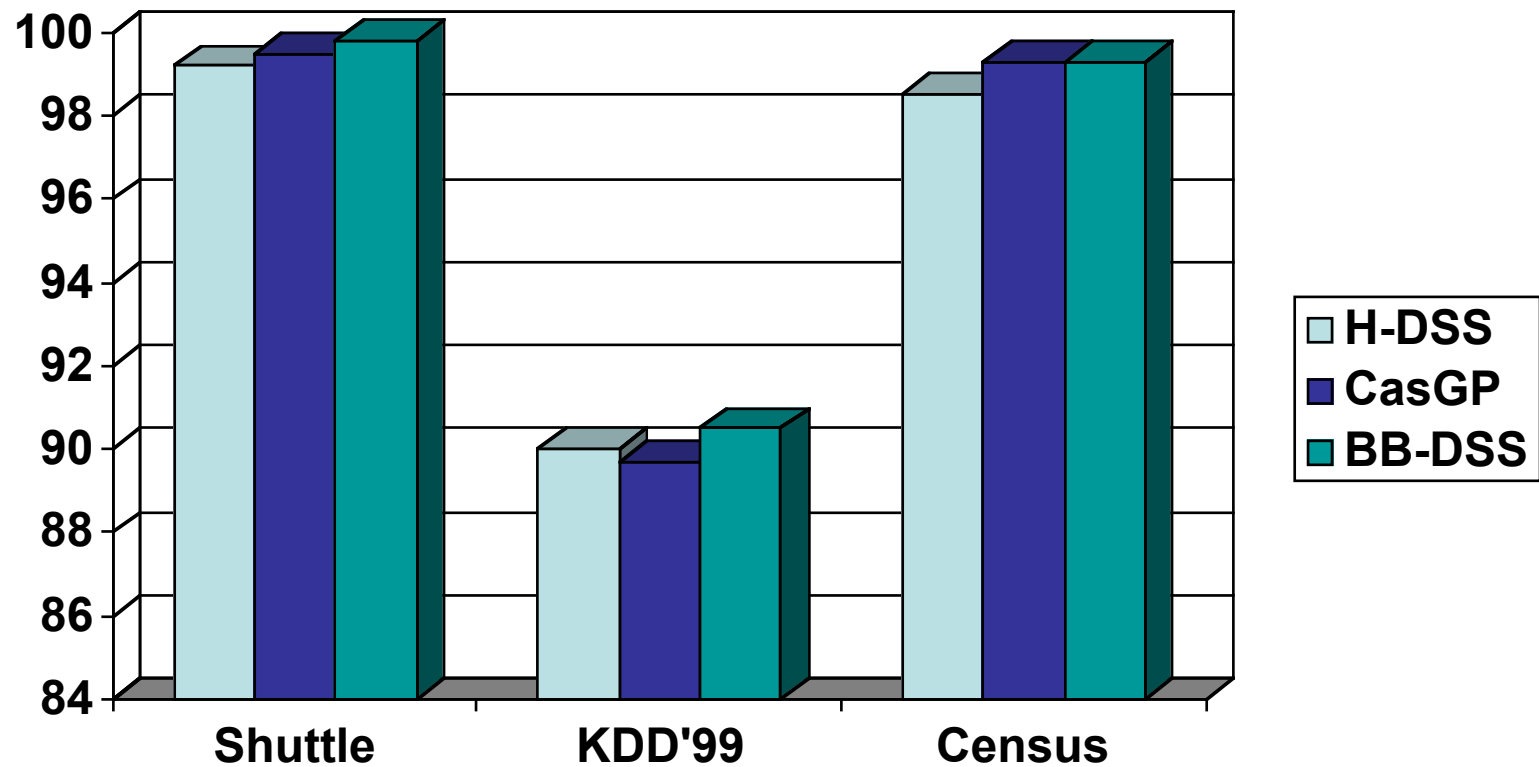
- H-DSS versus BB-DSS versus CasGP
- CasGP
  - Iteratively applies H-DSS
  - Augment input of next layer with
    - All application attributes
    - Output from previous layers
    - Only train 1 layer at a time
  - Multiple restarts per layer

# DATA SETS

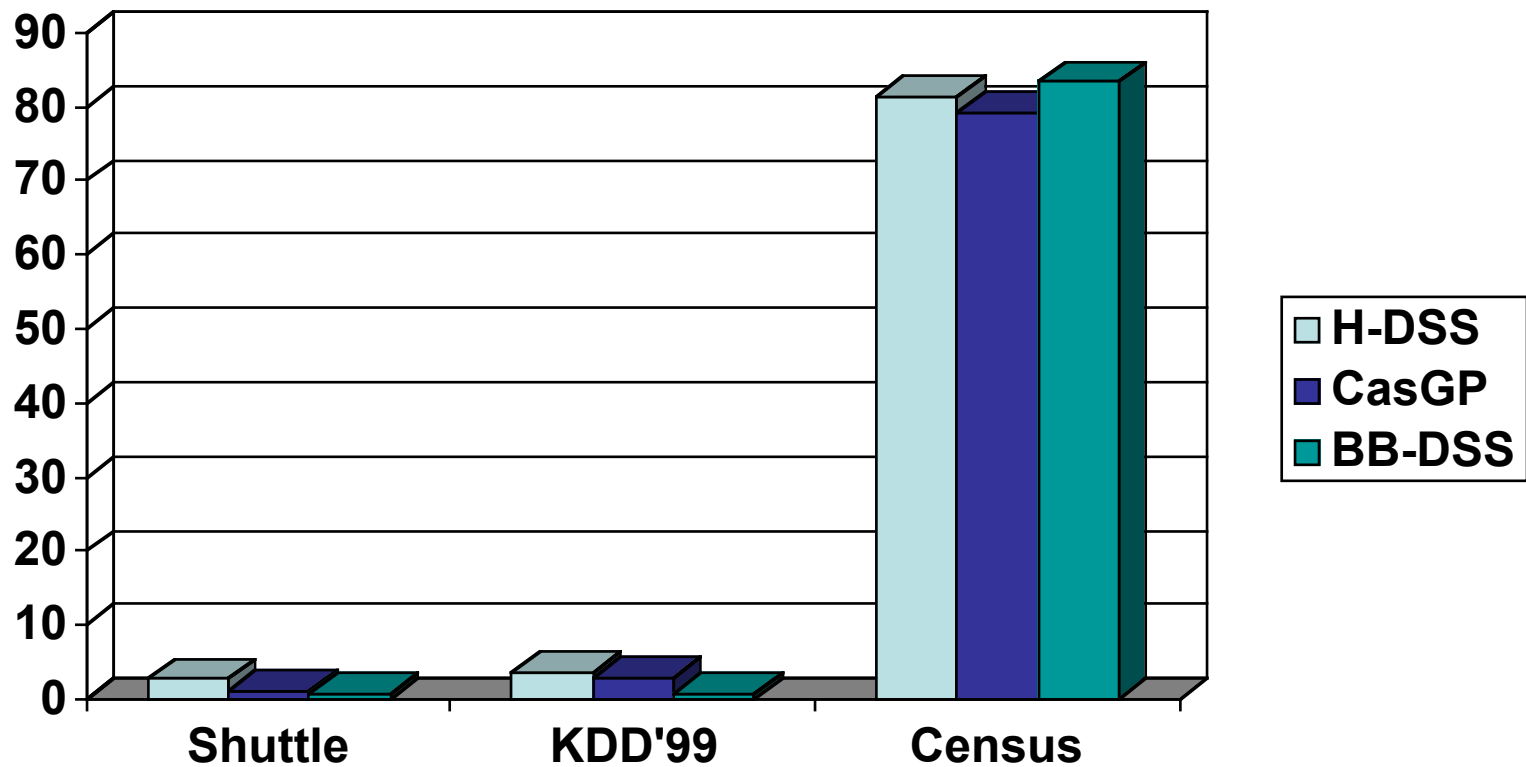
## LARGE AND UNBALANCED

Data set	Shuttle Train(test)	KDD'99 Train(test)	Census Train(test)
Class 1	9,392 (3,022)	97,249 (60,577)	5,479 (2,683)
Class 0	34,108 (11,478)	396,744 (250,424)	89,651 (44,708)
Total	43,500 14,500	493,993 (311,001)	95,130 (47,391)

# MEDIAN DETECTION RATE

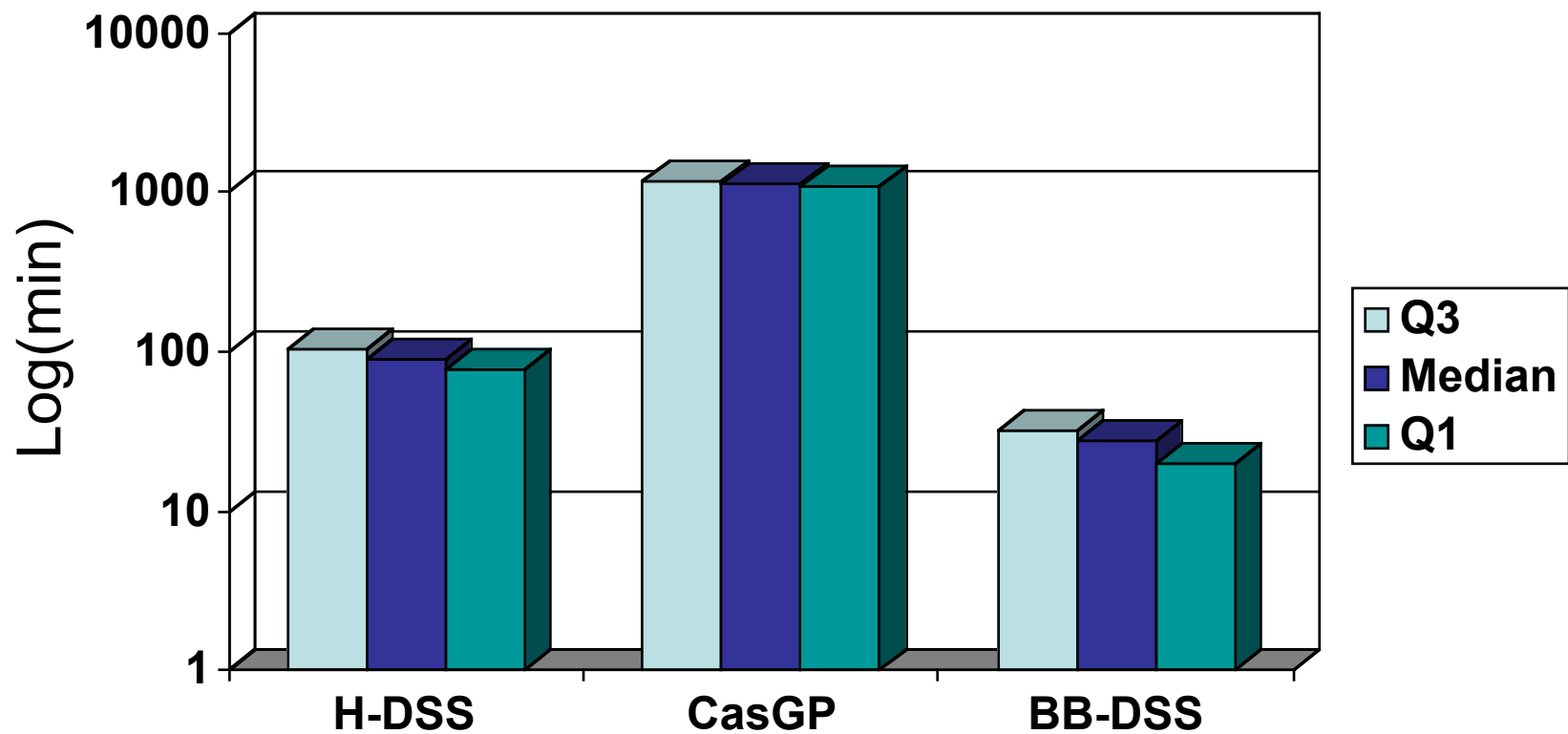


# MEDIAN FALSE POSITIVE RATE



# KDD'99 CPU TRAINING TIME

## USER TIME (MINUTES)



# CONCLUSION

- Strong performance from simple models
  - Uniform (exemplar) selection
  - Class balancing heuristic
- Provides computational leverage for more complex cost function
- DSS family
  - Stronger model of exemplar selection

# REFERENCES

- Weiss and Provost 2003
  - Learning when Training Data are Costly: The effect of Class Distribution on Tree Induction.
  - Journal of Machine Learning Research 19
- Doucette and Heywood 08
  - GP Classification under Imbalanced Datasets
  - European Conference on Genetic Programming
- Gathercole and Ross 94
  - Dynamic training Subset Selection for Supervised Learning in Genetic Programming
  - Parallel Problem Solving from Nature III
- Song *et al* 05
  - Training Genetic Programming on Half a Million Patters
  - IEEE Transactions on Evolutionary Computation
- Curry *et al* 07
  - Scaling Genetic Programming to Large Datasets using Hierarchical Dynamic Subset Selection
  - IEEE Transactions on Systems, Man, and Cybernetics