



ELSEVIER

Pattern Recognition Letters xxx (2002) xxx–xxx

Pattern Recognition
Letters

www.elsevier.com/locate/patrec

Distributed learning with bagging-like performance

Nitesh V. Chawla^a, Thomas E. Moore^a, Lawrence O. Hall^{a,*},
Kevin W. Bowyer^b, Philip Kegelmeyer^c, Clayton Springer^c

^a Department of Computer Science and Engineering, University of South Florida, 4202 East Flower Avenue, Tampa, FL 33620 USA

^b Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556 USA

^c Sandia National Laboratories, Biosystems Research Department, P.O. Box 969, MS 9951, Livermore, CA 94551-0969, USA

Received 11 December 2001; received in revised form 3 June 2002

Abstract

Bagging forms a committee of classifiers by bootstrap aggregation of training sets from a pool of training data. A simple alternative to bagging is to partition the data into disjoint subsets. Experiments with decision tree and neural network classifiers on various datasets show that, given the same size partitions and bags, disjoint partitions result in performance equivalent to, or better than, bootstrap aggregates (bags). Many applications (e.g., protein structure prediction) involve use of datasets that are too large to handle in the memory of the typical computer. Hence, bagging with samples the size of the data is impractical. Our results indicate that, in such applications, the simple approach of creating a committee of n classifiers from disjoint partitions each of size $1/n$ (which will be memory resident during learning) in a distributed way results in a classifier which has a bagging-like performance gain. The use of distributed disjoint partitions in learning is significantly less complex and faster than bagging.

© 2002 Published by Elsevier Science B.V.

Keywords: Distributed learning; Bagging; Large data sets; Ensembles; Multiple classifiers

1. Introduction

Many data mining applications use data sets that are too large to be handled in the memory of the typical computer (Shafer et al., 1996; Darlington et al., 1997; Chan and Stolfo, 1993; Provost et al., 1999; Moore and Lee, 1998; Bowyer et

al., 2000; Hall et al., 1999, 2000; Oates and Jensen, 1998). One possible approach is to subsample the data in some manner (Provost et al., 1999; Breiman, 1999). However, it can be difficult a priori to know how to subsample so that accuracy is not affected. Also, recent work by Perlich et al. (in press) has shown that classifier accuracy tends to increase with more training data even for large data sets. Another possible approach is to partition the original data into smaller subsets, and form a committee of classifiers (Chan and Stolfo, 1993; Provost and Hennessy, 1996). One advantage of this approach is that the partition size can simply be set at whatever amount of the original

* Corresponding author. Tel.: +813-974-4195; fax: +813-974-5456.

E-mail addresses: chawla@csee.usf.edu (N.V. Chawla), tmooore4@csee.usf.edu (T.E. Moore), hall@csee.usf.edu (L.O. Hall), kwb@cse.nd.edu (K.W. Bowyer), wpk@ca.sandia.gov (P. Kegelmeyer), csprin@ca.sandia.gov (C. Springer).

41 data can be conveniently handled on the available
 42 system. Another advantage is that the committee
 43 potentially has better accuracy than a single clas-
 44 sifier constructed on all the data.

45 In its typical form, bagging involves random
 46 sampling with replacement from the original pool
 47 of training data to create “bags” of data for a
 48 committee of thirty to one hundred classifiers.
 49 Bagging has been shown to almost always result in
 50 equal or (usually) improved performance over a
 51 single classifier created on all of the original data
 52 (Breiman, 1996; Quinlan, 1996; Bauer and Kohavi,
 53 1999). The success of bagging suggests that it
 54 might be a useful approach to create accurate
 55 classifiers for large data sets. We define *large* data
 56 sets as those which do not fit in the memory of a
 57 typical scientific computer. However, experience
 58 with bagging has primarily been in the context of
 59 *small* data sets. If the original data set will not fit in
 60 the main memory of the typical computer, then
 61 none of the thirty or more bags one might create
 62 will fit. This raises the question of which particu-
 63 lars of the bagging approach are essential in the
 64 context of large data sets. We consider the ques-
 65 tion of whether simple partitioning of the training
 66 data into tractable-size subsets can produce an
 67 ensemble of classifiers with accuracy equal to that
 68 of a single classifier built on all of the data. In this
 69 work, we show that simple partitioning of a large
 70 original data set into disjoint subsets results in
 71 better performance than creating bags of the same
 72 size. In other words, it is the committee of classi-
 73 fiers that is essential and bootstrap aggregation to
 74 form the individual classifiers is not essential.
 75 Further, it is straightforward to create one or
 76 several different disjoint partitions of data and the
 77 process is rapid.

78 Classifiers can be built in a distributed way on
 79 disjoint partitions. Each of n classifiers can be
 80 learned on a separate processor in parallel. Fur-
 81 ther, for large enough partitions (bags) the per-
 82 formance of the resulting classifier will meet or
 83 exceed that of a classifier built on all the data. The
 84 time required to build an ensemble of n classifiers
 85 which are learned independently on n processors
 86 without communication will be the longest time
 87 required to learn a single classifier on a processor.

In Section 2, we discuss related work. In Section 88
 3, we describe our experimental setup for small 89
 data sets from the UC Irvine repository (Blake and 90
 Merz, 1998), as well as the setup for experiments 91
 with a mid-size and large data set from our own 92
 research. We also describe the base classifiers and 93
 computing systems used in the experiments. Sec- 94
 tion 4 contains the experimental results. Finally, 95
 Section 5 contains a discussion of the results and 96
 conclusions that can be drawn. 97

2. Literature review 98

Bagging (Breiman, 1996) has been shown to 99
 improve classifier accuracy. Bagging basically 100
 combines models learned on different samplings of 101
 a given dataset. According to Breiman, bagging 102
 exploits the instability in the classifiers, since per- 103
 turbing the training set produces different classi- 104
 fiers using the same learning algorithm. Quinlan 105
 (1996) experimented with bagging on various da- 106
 tasetes and found that bagging substantially im- 107
 proved accuracy. However, the experiments were 108
 performed on *small* datasets, the largest one being 109
 20,000 examples. 110

Domingos (1997) empirically tested two alter- 111
 native theories supporting bagging: (1) bagging 112
 works because it approximates Bayesian model 113
 averaging or (2) it works because it shifts the 114
 priors to a more appropriate region in the decision 115
 space. The empirical results showed that bagging 116
 worked possibly because it counter-acts the in- 117
 herent simplicity bias of the decision trees. That is, 118
 with M different bags, M different classifiers are 119
 learned, and together their output is more complex 120
 than that of the single learner. 121

In (Street and Kim, 2001) an ensemble of clas- 122
 sifiers are built from training data which is treated 123
 as a stream. Each classifier is trained on a fixed 124
 amount of data from the stream. The size of the 125
 ensemble is fixed at 25 classifiers. Classifiers 126
 “compete” for entry into the ensemble based on 127
 their accuracy and diversity. This approach allows 128
 an ensemble classifier to be built from an unlimited 129
 amount of training data. It also facilitates building 130
 an ensemble classifier which might be built on data 131
 with temporal dependencies, where the concept to 132

133 be modeled may vary over time. In their experi-
134 ments, the ensemble classifier was usually slightly
135 less accurate than a classifier which was built with
136 as much data as the current ensemble had used. In
137 our work larger training sets are used.

138 In (Breiman, 1999) classifiers were built on
139 small randomly chosen subsets of an overall
140 training set. This approach can deal with ex-
141 tremely large training datasets, but requires many
142 classifiers because it uses somewhere around 800
143 examples per classifier in the discussed experi-
144 ments. Also, the process of continually selecting
145 small subsets can be computationally problematic
146 for very large datasets. This is different from the
147 approach we are looking at, in which the number
148 of training samples may be as large as main
149 memory space available.

150 Chan and Stolfo (1995) compared arbiter and
151 combiner strategies by applying a learning algo-
152 rithm to disjoint subsets of data. An arbiter
153 scheme uses a learned representation of which
154 classifier to choose given an example and a com-
155 biner takes classifier outputs of a test example as
156 input and produces a classification. The described
157 experiments showed that the arbiter strategy can
158 sometimes better sustain the accuracy compared to
159 the classifier learned on the entire data set. The
160 combiner strategy showed a drop in accuracy with
161 the increase in the number of subsets, which can be
162 attributed to the lack of information content in the
163 small subsets. However, a few cases resulted in an
164 improvement in accuracy. We are interested in
165 disjoint subsets of larger original data sets than in
166 (Chan and Stolfo, 1995) and so there is reason to
167 expect that accuracy can be maintained.

168 Chan and Stolfo (1996) relaxed their definition
169 of strict disjoint subsets by allowing a small
170 amount of overlap across the subsets. On the da-
171 taset DNA Splice Junction with 3190 examples
172 and protein coding region with 20,000 examples, it
173 was found that overlapping did not bring any gain
174 to their meta-learning strategy. Each classifier
175 trained on a disjoint set is biased towards its own
176 set, and when these classifiers are combined a
177 protocol of knowledge sharing is established, and
178 each individual classifier's bias is reduced. Again,
179 we are interested in large data sets relative to those
180 considered in this work.

Domingos (1996) describes how a specific-to- 181
general rule induction system (RISE) was sped up 182
by applying it to disjoint training sets. This al- 183
lowed the time required for learning to become 184
linear in the number of examples. The resulting 185
rule based classifiers were voted (with some 186
weighting) in an approach very similar to bagging. 187
The major difference was that the size of each 188
training data set was much smaller than the orig- 189
inal. On a set of seven data sets from the Irvine 190
repository using disjoint partitions of between 100 191
and 500 examples they found that the resulting 192
voting performance was generally as good as or 193
better than applying RISE to all the data. 194

Dietterich (2000) describes how an ensemble of 195
decision trees was built from a single unmodified 196
training set. Diversity in the trees was obtained by 197
randomly choosing a test at each internal node 198
from among the top k tests (ranked by information 199
gain, with k typically 20). Each tree was generally 200
suboptimal, but when voted as an ensemble they 201
provided a bagging-like and sometimes better ac- 202
curacy gain. This result seems to suggest that 203
bagging-like performance can be obtained from a 204
set of diverse classifiers (in the sense that they 205
make errors on different examples) which may 206
each be suboptimal (in the sense that they are not 207
as good as a classifier built on all the data without 208
any manipulation), but similar in accuracy. 209

Hall et al. (2000) learned decision trees using 210
disjoint partitions of data and then combined the 211
classifiers. It was found that when using a conflict 212
resolution strategy for combining rules, the accu- 213
racy usually did not decrease for a small number of 214
partitions, at least on the datasets tested. Our 215
current work is similar to this, but focuses on 216
comparison of bagging-like approaches to simple 217
partitioning of large datasets. 218

Provost et al. (1999) found that subsampling the 219
data gave the same accuracy as the entire dataset 220
at much lower computational cost. They analyzed 221
“progressive sampling” methods—progressively 222
increasing the sample size until the model accuracy 223
no longer improved. It was found that adding 224
more training instances did not help the accuracy 225
of the classifier, and after some number of in- 226
stances the performance of the classifier plateaus. 227
As pointed out later in the discussion section, our 228

229 results indicate that simple subsampling to pro-
230 duce one smaller training set is not a profitable
231 strategy for the larger datasets that we consider.
232 However, more complicated subsampling strate-
233 gies may be useful.

234 3. Experiments

235 Three sets of experiments were performed. The
236 first uses five small datasets, representative of
237 those commonly used in pattern recognition and
238 machine learning research. It compares four ap-
239 proaches to creating a committee of N classifiers,
240 with each classifier created using $(1/N)$ th of the
241 training data. The performance of the approaches
242 is also compared to that of *true bagging*—bags of
243 the same size as the pool of training data, ran-
244 domly sampled with replacement. The point of this
245 first set of experiments is to isolate the essential
246 factor(s) leading to good performance in the
247 committee of classifiers. The second set of experi-
248 ments uses a mid-size dataset of almost 230,000
249 examples. The same four approaches are evaluated
250 on this data set. The point is to verify that the
251 pattern of performance results observed with
252 smaller data sets holds with a larger data set.

253 Based on the first two sets of experiments, the
254 disjoint partitioning approach is identified as of-
255 fering equivalent performance for a given size of
256 partition/bag. It is also the simplest of the ap-
257 proaches considered. The last experiment uses a
258 large dataset of approximately 3.6 million exam-
259 ples to investigate the degree of performance im-
260 provement that the disjoint partitioning approach
261 can achieve over a classifier built on all the original
262 data.

263 3.1. Variations of partitioning and bagging

264 We investigated four different approaches to
265 creating a committee of classifiers from an original
266 data set (see Fig. 1 for an illustration). One ap-
267 proach is to simply randomly partition the original
268 data into N disjoint partitions of size $(1/N)$ th
269 of the original data. Thus the union of the N training
270 sets is identical to the original data. Results of this

original data set:

A B C D E F G H I J K L M N O P

Disjoint partitions (random order of data)

A B C D E F G H I J K L M N O P D

Small Bags (replication within and across):

A C H L B P L P D I O H K C F K SB

No Replication Small Bags:

A C H L O P L N D I O H K C F P NRSB

Disjoint Bags (no replication across, larger):

A B C D C E F G H E I J K L J M N O P O DB

Fig. 1. Four approaches to a committee of classifiers.

271 approach are labeled with ‘D’ (for “disjoint”) on
272 the graphs.

273 The second approach is to create N bags of size
274 $(1/N)$ th of the data. Each bag is created inde-
275 pendently, by random sampling is done with re-
276 placement, so the union of the training sets is
277 generally not the same as the original data. This
278 approach is labeled ‘SB’ (for “small bags”) on the
279 graphs. Comparison of the SB performance versus
280 that of disjoint partitions shows whether the ran-
281 dom replication of data elements results in any
282 inherent advantage.

283 The third approach is like small bags, but
284 sampling without replacement for each individual
285 bag. When sampling the individual bags without
286 replacement, elements of the original data do not
287 repeat within a bag, but may repeat across bags.
288 This approach is labeled ‘NRSB’ (for “no-repli-
289 cation small bags”) on the graphs.

290 The fourth approach begins with the disjoint
291 partitions. Then, independently for each partition,
292 a number of its elements are randomly selected
293 with replacement to be added to the partition to
294 form a “disjoint bagged” training set. Thus the
295 union of the training sets is a superset of the
296 original data; all elements of the original data
297 appear, plus some random replications. The
298 number of added elements is equal to the average
299 number of repeated elements in a bag in the “small
300 bags.” Thus a bag used in this approach is slightly
301 larger than $(1/N)$ th of the original data. The
302 amount of “extra” data included decreases as the

303 bag size decreases. Results of this approach are
304 labeled ‘DB’ (for “disjoint bagged”) on the graphs.
305 Comparison of the results of this approach to the
306 results of disjoint partitions looks again at whether
307 the random replication of data elements results in
308 any inherent advantage, but through the effect of
309 allowing larger bag size.

310 In addition to the above four approaches, we
311 also ran “true bagging” on each of the five small
312 datasets. By “true bagging” we mean creating M
313 bags, each of the size of the original data, inde-
314 pendently using random sampling with replace-
315 ment. True bagging is expected to out-perform
316 committees of classifiers formed using smaller
317 bags, but the point is to provide a baseline per-
318 formance comparison for the other approaches.

319 For each experiment in which voting is used,
320 ties are broken by assigning the example to the
321 largest class (in the training set) participating in
322 the tie.

323 3.2. Datasets

324 Three of the small data sets are from the UCI
325 repository (Blake and Merz, 1998), one is from the
326 ELENA project,¹ and one is from our own re-
327 search. The mid-size dataset comes from the
328 problem of predicting the secondary structure of
329 proteins. It is part of the training data set used
330 with a neural network that won the CASP-3 sec-
331 ondary structure prediction contest (Jones, 1999).
332 We concatenated his train set one and test set one
333 as our overall training set. This dataset contains
334 almost 230,000 elements. Each amino acid in a
335 protein can have its structure labeled as helix (H),
336 coil (C), or sheet (E). The features for a given
337 amino acid are twenty values in the range -17 to
338 17 , representing the log likelihood of the amino
339 acid being any one of twenty basic amino acids.
340 Using a window of size 15 centered around the
341 target amino acid and an extra bit for each win-
342 dow to indicate an N or C terminus (beginning or
343 end of chain) gives a feature vector of size 315.

344 Our large dataset also comes from the protein
345 database (PDB) (Berman et al., 2000) used in the
346 CASP contests (Lawrence Livermore National
347 Laboratories, 2001). For 18,098 protein chains
348 taken from the PDB, there are a total of 3,679,152
349 amino acids for structure prediction. Using a
350 window of size seventeen centered around the
351 target amino acid (without an extra N/C terminus
352 bit), we have a feature vector of size 340. This
353 training data takes from 1.3 to 30 GB to store,
354 depending on how feature values are encoded (e.g.
355 signed char, integer, or float). The test data for the
356 experiments with the large dataset consists of a
357 separate set of data. It is all protein chains entered
358 into the PDB from July 11 2000 to July 28 2000,
359 that are based on X-ray crystal structures with
360 resolution of three angstroms or finer. There were
361 146 chains entered in this time frame, made up of
362 38,423 amino acids. All results are reported on a
363 per amino acid basis.

364 The size and class distribution of the datasets
365 are summarized in Table 1. Note that the experi-
366 ments include both two-class (Mammography,
367 Phoneme) and multi-class (Letter, PenDigits, Sat-
368 Image) datasets. They also include datasets that
369 are approximately balanced (Letter, PenDigits)
370 and those that are skewed (Mammography, Pho-
371 neme, SatImage).

372 The four approaches to creating a committee of
373 classifiers, plus true bagging, were applied to each
374 of the small datasets. The number of bags/parti-
375 tions was varied from one to eight. Given the
376 modest size of the datasets, creating bags/parti-
377 tions of less than $(1/8)$ th the original size appears
378 to begin to starve the classifiers for training data.
379 For the experiments on the small and mid-size
380 datasets, the reported results are calculated from
381 10-fold cross-validation.

382 3.3. Base classifiers and computing systems

383 For the experiments on the small and mid-size
384 datasets, release eight of the C4.5 decision tree
385 system (Quinlan, 1992) and the cascade correlation
386 neural network learning code (Fahlman and
387 Lebiere, 1990) were run on standard SUN work-
388 stations and a 24-node Beowulf cluster computer
389 (Mimir). Mimir consists of 900 Mhz Athlon pro-
390 cessors.

¹ ftp.dice.ucl.ac.be in the directory pub/neural-nets/ELENA/
databases.

Table 1
Data sets sizes and class distributions

Letter dataset (UCI)—20,000 samples in 26 classes					
A:789	B:766	C:736	D:805	E:768	F:775
G:773	H:734	I:755	J:747	K:739	L:761
M:792	N:783	O:753	P:803	Q:783	R:758
S:748	T:796	U:813	V:764	W:752	X:787
Y:786	Z:734				
Phoneme dataset (ELENA)—5404 samples in two classes					
0:3818	1:1586				
PenDigits dataset (UCI)—10,992 samples in ten classes					
0:1143	1:1143	2:1141	3:1055	4:1144	5:1055
6:1056	7:1142	8:1055	9:1055		
SatImage dataset (UCI)—6435 samples in six classes					
1:1533	2:703	3:1358	4:626	5:707	7:1508
Mammography dataset—11,183 samples in two classes					
1:10,923	2:260				
Jones' PDB dataset—227,260 samples in three classes					
H:75,455		C:100,909		E:50,896	
PDB dataset—3,619,461 samples in three classes					
H:1,254,335		C:1,537,261		E:827,865	

390 cessors each with 512 Mb of memory and pro-
391 cessors are connected with 100 Bt Ethernet. The
392 neural network was only applied to the small data
393 sets due to its extremely long training times and
394 larger memory requirements.

395 The one run of the large dataset to produce a
396 single classifier was done on a single node of a 64-
397 processor SGI IRIX64 with 32 GB of main
398 memory at Sandia National Labs, also using the
399 standard C4.5 release eight. Creating the one de-
400 cision tree on the large dataset took approximately
401 *thirty days* on the SGI and was not attempted with
402 the much slower neural network learning code.

403 The experiments using partitions of the large
404 dataset were run on the DOE's "ASCI Red" par-
405 allel supercomputer (Sandia National Labs, 1998).
406 The ASCI Red has 4640 compute nodes, each
407 containing two Pentium III processors sharing 256
408 MB of memory. The processors run a version of
409 the UNIX operating system. The system is based
410 on a distributed-memory mesh architecture, and is
411 capable of 3.15 TeraFLOPS. These experiments
412 used a version of C4.5 modified with MPI calls for
413 parallel execution. The parallel structure of this
414 version of C4.5 is quite simple. The disjoint par-

titions are loaded into the different compute node 415
memories and each compute node independently 416
grows a decision tree. The parallel computation to 417
create eight decision trees on one-eighths of the 418
large dataset takes approximately 8 h; that is, eight 419
processors running in parallel for 8 h. It is possible 420
to create 32 distributed trees built on 1/8 size 421
partitions of the data in approximately 10 h. 422

4. Results 423

4.1. C4.5 on small data sets 424

Figs. 2–6 summarize the experimental compar- 425
ison of the different approaches on the small da- 426
taset detailed in Table 1 for C4.5. The plots 427
compare the performance of two, four, six, and 428
eight disjoint partitions (D) to that of C4.5 on the 429
complete data set, and to classifier committees 430
formed using the other three approaches (DB, SB, 431
NRSB). Results are shown as the average paired 432
difference across the 10 folds in the 10-fold cross- 433
validation, with standard error indicated. Note the 434
zero mean accuracy mid-line; if a point is above 435

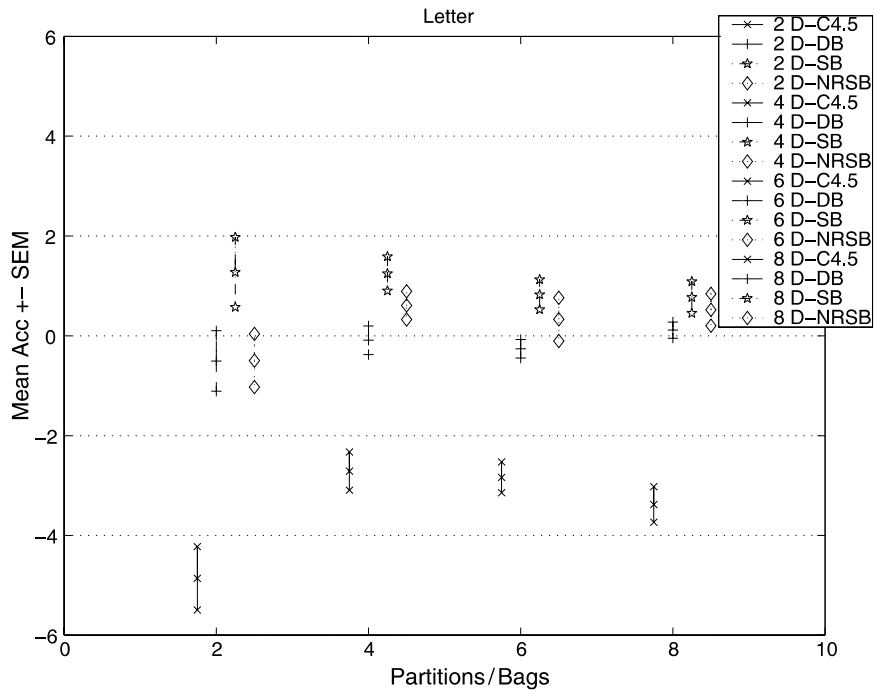


Fig. 2. Comparison on Letter dataset with C4.5.

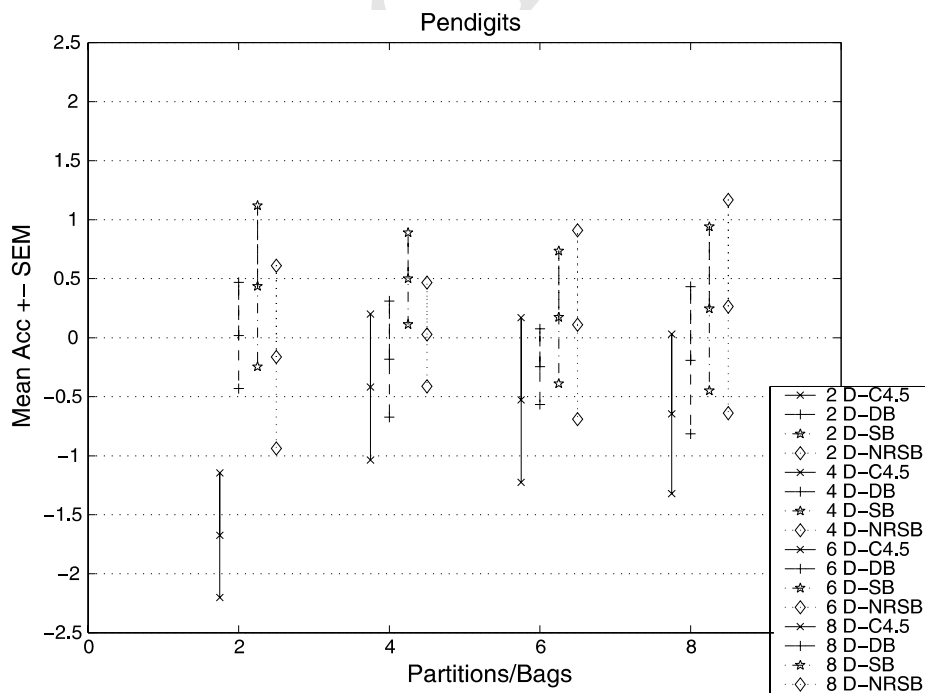


Fig. 3. Comparison on PenDigits dataset with C4.5.

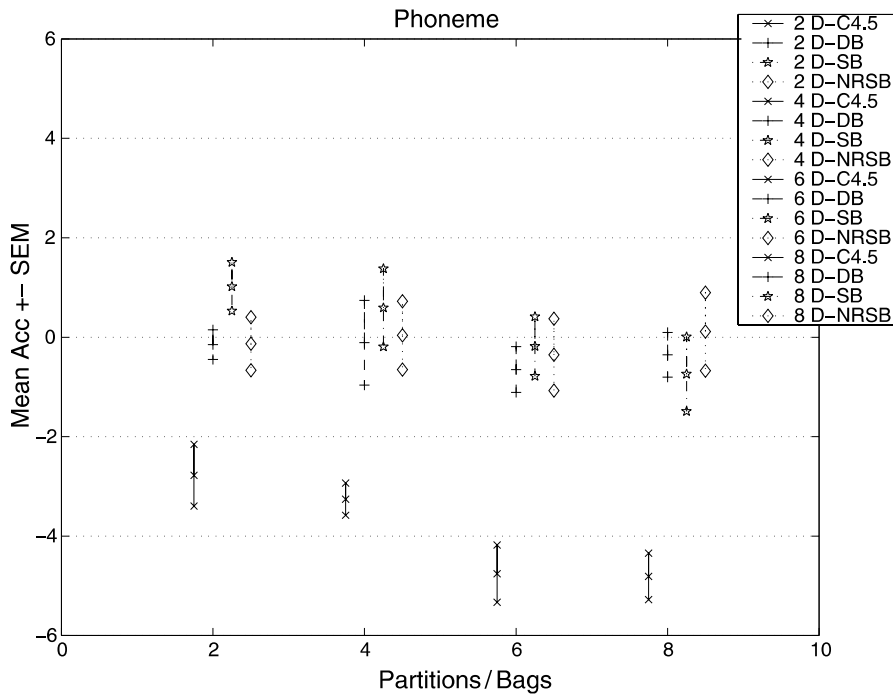


Fig. 4. Comparison on Phoneme dataset with C4.5.

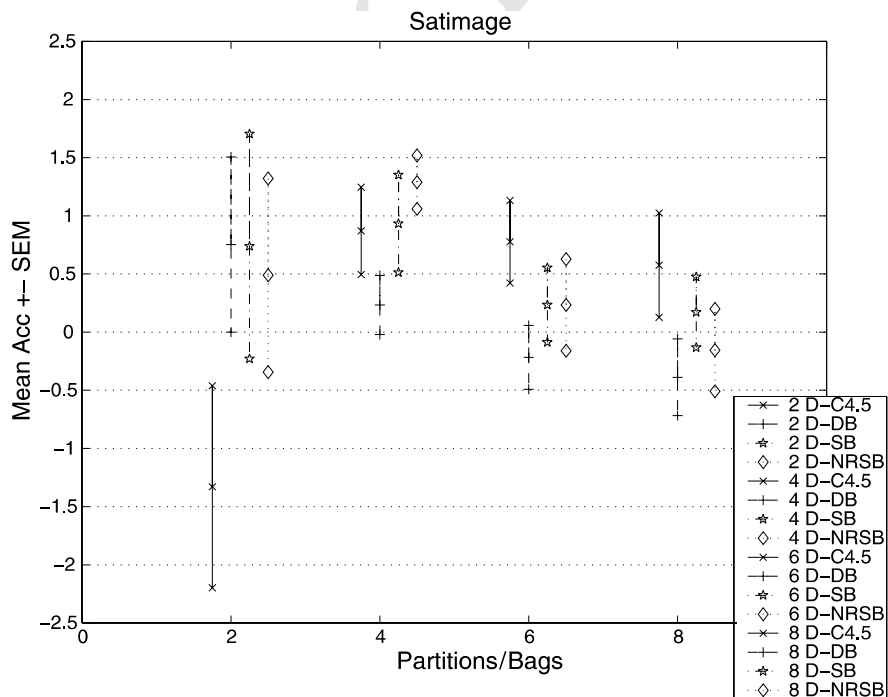


Fig. 5. Comparison on SatImage dataset with C4.5.

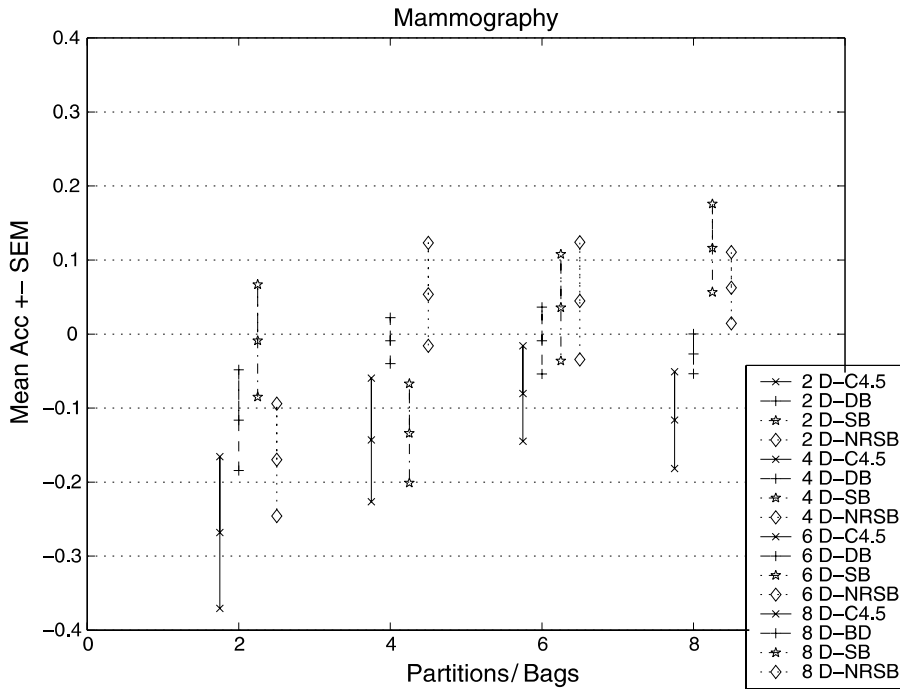


Fig. 6. Comparison on Mammography dataset with C4.5.

436 the line, then the first of the alternatives examined
437 is superior. If the point is below, then the second
438 was best. And the extent to which the error bar
439 spans the mid-line is the extent to which the results
440 are inconclusive.

441 As an example, the first cluster of four data
442 points on the plot in Fig. 2 represents the results
443 for a committee of two classifiers on the Letter
444 data set. The first point is the difference between a
445 committee of two disjoint partitions and C4.5
446 trained on all of the data; note that the committee
447 of two classifiers performs significantly worse. The
448 second point is the difference between a committee
449 formed using two disjoint partitions versus a
450 committee using two disjoint bags (DB), the third
451 point is two disjoint partitions versus two small
452 bags (SB), and the fourth point is two disjoint
453 partitions versus no-replication small bags
454 (NRSB).

455 From examining the sequence of plots it is clear
456 that disjoint partitions in a number of instances
457 beat small bags. It appears to make little difference
458 whether the small bags are created by sampling

459 with or without replacement. The ensembles cre-
460 ated from “bagged disjoint” appear to generally
461 perform slightly better than those created from
462 simple disjoint partitions, but then the training sets
463 for the individual decision trees have repeated ex-
464 amples which give some examples greater
465 “weight”.

466 Because it uses constant-size bags as the num-
467 ber of classifiers in the committee grows larger,
468 “true bagging” should naturally outperform any
469 of the four approaches. Data points for “true
470 bagging” performance are given in Table 2. Tables
471 3 and 4 show the accuracy results obtained by
472 learning four classifiers each built from 1/4 of a

Table 2
Data points for “true bagging” results

Dataset	C4.5	50 bags	75 bags	100 bags
Phoneme	86.50	89.15	89.02	88.15
SatImage	86.30	90.89	90.86	90.84
PenDigits	96.57	98.42	98.43	98.36
Mammography	98.50	98.76	98.79	98.79
Letter	88.10	93.54	93.65	93.80

Table 3
Four partitions/bags accuracy in percentage using C4.5

Dataset	D	SB	DB	NRSB
Phoneme	83.25	82.66	83.36	83.22
SatImage	87.24	86.31	87.00	85.95
PenDigits	96.15	95.65	96.33	96.12
Mammography	98.37	98.51	98.38	98.32
Letter	85.44	84.19	85.525	84.83

Table 4
Four partitions/bags accuracy in percentage using CC

Dataset	D	SB	DB	NRSB
Phoneme	83.75	84.06	83.92	83.59
SatImage	89.43	88.97	89	89.26
PenDigits	90.89	91.01	91.73	90.86
Mammography	97.78	98.19	98.43	98.02
Letter	82.15	81.88	81.96	82.17

473 disjoint four partition utilizing decision trees and
474 neural networks, respectively. Each value in the
475 table is an average over a 10-fold cross-validation.
476 The results from a four partition are representa-

477 tive, as more partitions cause greater “data star-
478 vation” in some of the small domains. Bagging
479 with 50–100 bags clearly outperforms the small
480 ensembles. However, the point is that true bagging
481 is simply not a practical option for large datasets.
482 For the example large dataset here, true bagging
483 would require creating about 50 classifiers, each
484 training on a data set of the *same* size as the
485 original data. Recall that creating one classifier on
486 all the data took 30 days on a substantial SGI
487 system. When the dataset is too large to handle
488 conveniently in the memory of the typical com-
489 puter, the dataset must be broken into some
490 number of practically sized, though not “small,”
491 chunks. The question addressed here is whether
492 there is any advantage in creating the practically
493 sized chunks using some bagging-like approach, or
494 whether simple partitioning is sufficient.

4.2. Cascade correlation on small datasets 495

496 Figs. 7–11 summarize the experimental com-
497 parison of the different approaches on the small

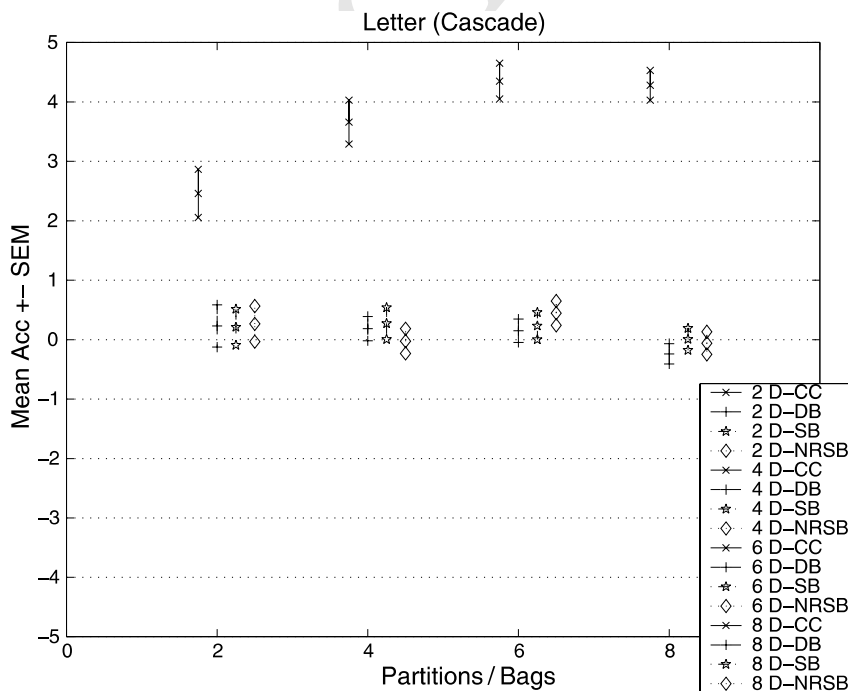


Fig. 7. Comparison on Letter dataset with cascade correlation.

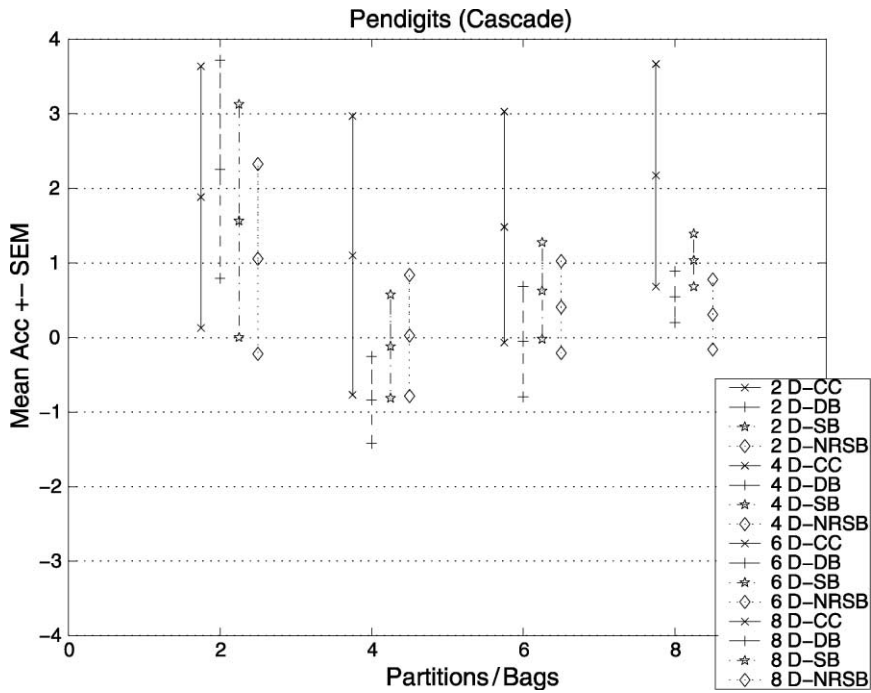


Fig. 8. Comparison on PenDigits dataset with cascade correlation.

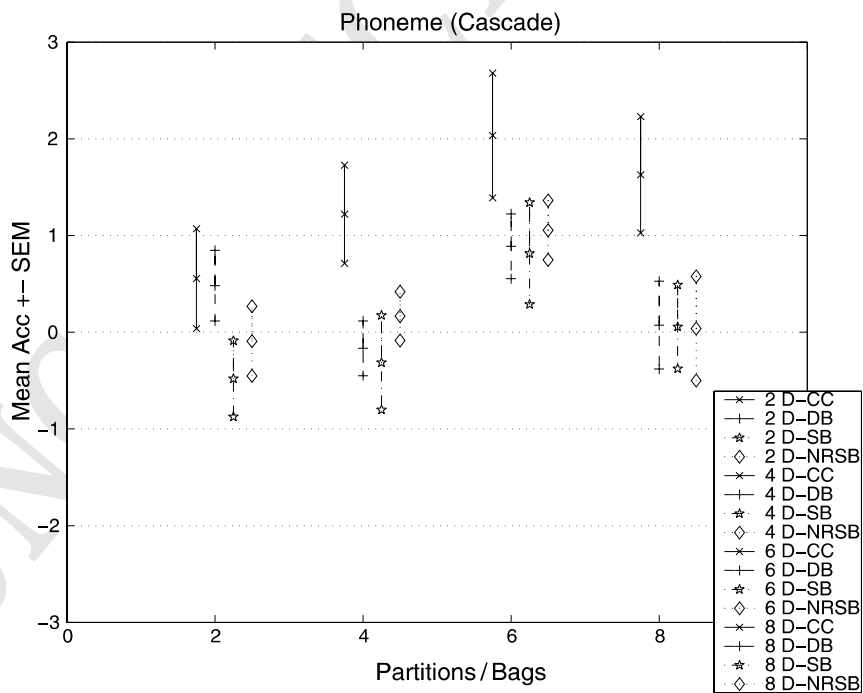


Fig. 9. Comparison on Phoneme dataset with cascade correlation.

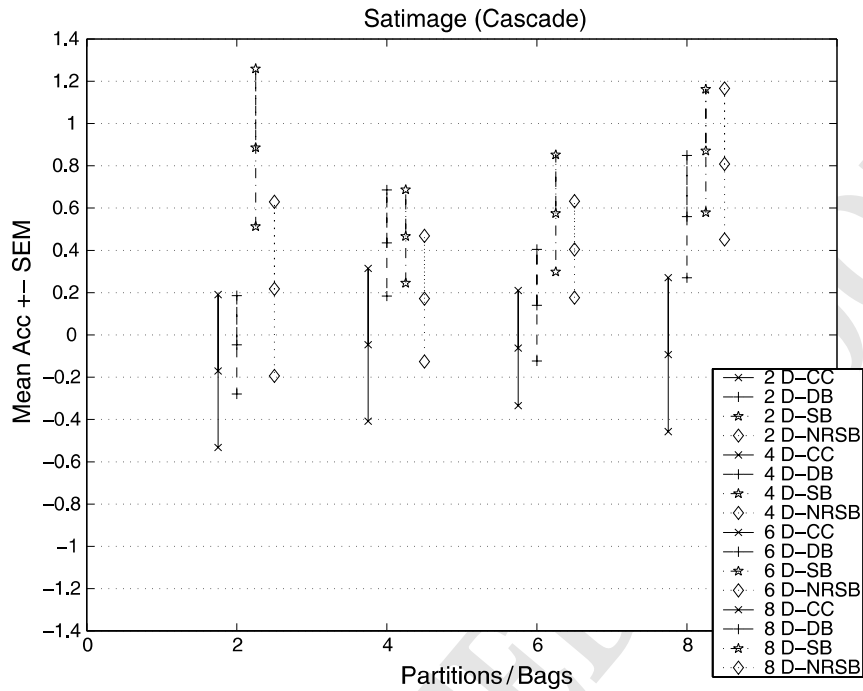


Fig. 10. Comparison on SatImage dataset with cascade correlation.

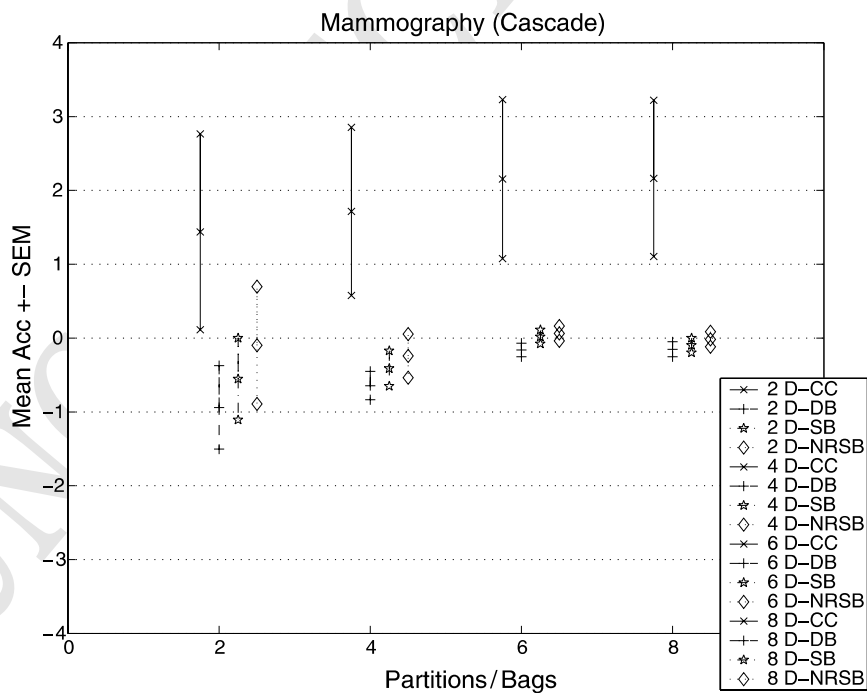


Fig. 11. Comparison on Mammography dataset with cascade correlation.

498 datasets detailed in Table 1 using cascade corre-
499 lation. The data was not normalized. The results
500 are similar to those obtained with C4.5. The dis-
501 joint partitions are generally as good or better than
502 small bags. The exception is that they lose to small
503 bags for a four partition on the Mammography
504 dataset and to DB at every partition in the
505 Mammography data set. The Mammography da-
506 taset is highly skewed with a small minority class
507 and the repeated examples in the disjoint bag data
508 sets seemed to make an important, if minor, dif-
509 ference in accuracy. The other exception is on the
510 PenDigits dataset where the DB approach is better
511 for a four partition.

512 It is interesting that the ensemble of neural
513 networks is generally no worse than learning from
514 all the data even when partitioning up the small
515 data into eight subsets. In fact, the ensemble is
516 always better than a single neural network for the
517 Letter data set. In general, the ensembles built
518 from disjoint partitions are no worse than those
519 built from bags of the same size for cascade cor-
520 relation neural networks.

4.3. Results on mid-size data set

521

522 The comparison of the four approaches on the
523 mid-size dataset are shown with C4.5 in Fig. 12.
524 The results are the means of paired differences and
525 the standard error from a 10-fold cross-validation
526 from 4 to 16 disjoint partitions. In all cases the
527 ensemble classifier learned from disjoint partitions
528 resulted in a significantly better classifier than ap-
529 plying C4.5 to all of the data.

530 Again, we see that simple disjoint partitioning
531 offers excellent performance in comparison to the
532 other options. In particular, the “small bags” ap-
533 proach performs poorly. Only the “bagged dis-
534 joints,” with its slightly larger number of examples
535 in each bag, offers any hint of performance im-
536 provement over disjoint partitions. The “bagged
537 disjoint” have the same number of distinct ex-
538 amples with a few repeat examples which essen-
539 tially gives the repeat examples greater weight.

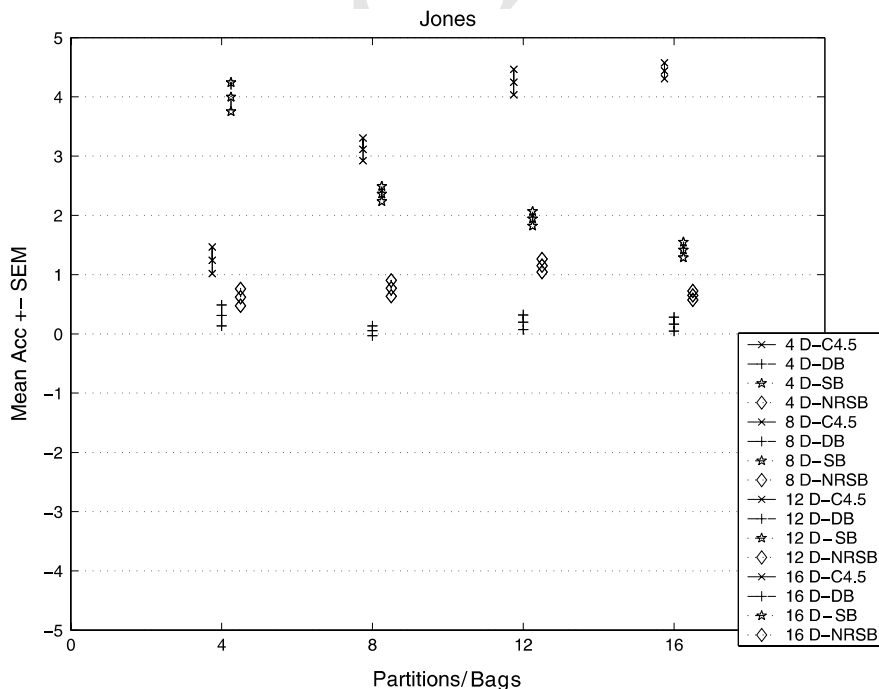


Fig. 12. Results on (Jones') "medium" PDB dataset.

540 4.3.1. Timing

541 We did a detailed comparison of the time re-
542 quired to create an ensemble on the Beowulf
543 cluster for the Jones data set. Table 5 shows the
544 average and standard deviation over 10 trials. For
545 disjoint partitioning for a four partition, there is
546 an average of a 9.4 times speed up over learning a
547 decision tree on all the data, which increases to
548 26.6 times for an eight partition, 46.1 times for a
549 16 partition and 71.3 times for a 24 partition.
550 While the speed gain is impressive, it is actually
551 true that for very large data sets that do not fit in
552 main memory all the data cannot be practically
553 used unless a distributed approach is pursued.

554 4.4. Partitioning results on large dataset

555 Fig. 13 compares the results of creating one
556 decision tree on all of the large dataset versus using
557 a committee of N classifiers, for $N = 8, 16, 24,$ and
558 32. All of the committees were formed using dis-
559 joint partitions of size $(1/8)$ th of the large PDB
560 dataset. This size partition just fills the memory of
561 the compute nodes on the ASCI Red. The $N = 8$

Table 5

Jones dataset: average and standard deviation of CPU time (longest time taken to learn a decision tree on a partition/bag by a processor) in seconds for 10-folds

Approach	Average time	Standard deviation
C4.5	9094.38	137.11
4-D	963.61	14.42
4-SB	990.61	17.04
4-NRSB	967.96	21.22
8-D	342.172	8.46
8-SB	352.04	8.98
8-NRSB	339.51	5.45
12-D	197.19	5.52
12-SB	202.55	3.89
12-NRSB	196.98	4.72
16-D	127.48	3.6
16-SB	130.62	1.32
16-NRSB	128.50	2.7

Convention = D: disjoint; SB: small bag; NRSB: no-replication small bag. For instance, four-dimensional means four disjoint partitions.

point represents a straightforward disjoint parti- 562
tioning, as was used with the smaller datasets. For 563

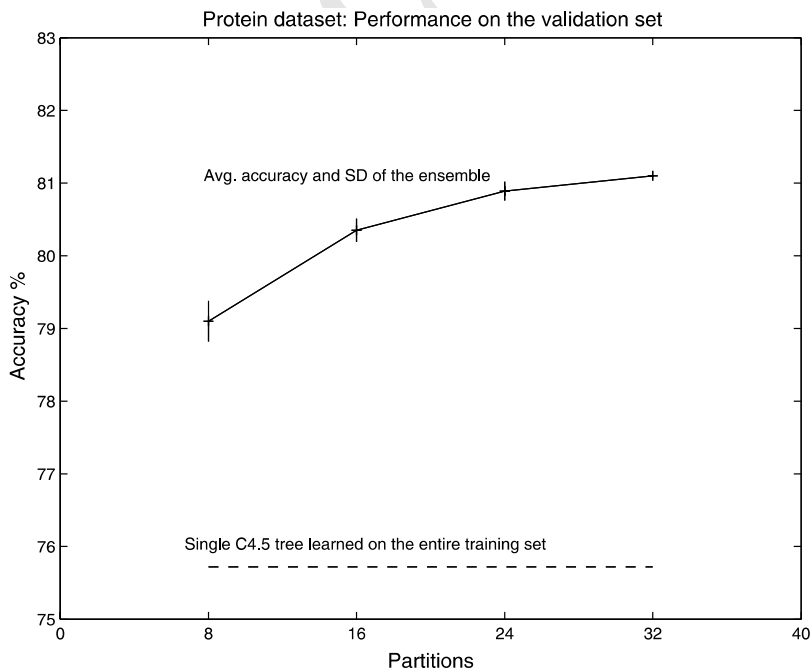


Fig. 13. Partitioning results for large PDB dataset.

564 the committees of 16, 24, 32, and 40 classifiers, we
 565 varied the earlier methodology to make use of
 566 multiple different partitions of the dataset. For the
 567 committees of 16, 24, 32, and 40 classifiers, mul-
 568 tiple different partitions of the dataset were used.
 569 For example, to create a set of sixteen classifiers,
 570 eight classifiers trained on a different eight parti-
 571 tion of the data were added to those created on the
 572 original eight-partition. We report an average and
 573 standard deviation over five trials of different en-
 574 sembles of trees of the appropriate size.

575 The average accuracy per amino acid of a single
 576 classifier trained on (1/8)th of the large dataset is
 577 71.21%. A single decision tree created using all the
 578 data performs substantially better than this,
 579 75.72% versus 71.21%. At the same time, an av-
 580 erage committee of eight classifiers created on
 581 (1/8)ths of the data performs substantially better
 582 than a single tree created on all the data, 79.1%
 583 versus 75.72%.

584 It took approximately 30 days to create a single
 585 tree from all the data and an average of 10 h to
 586 create an ensemble of from 8 to 32 trees. We did
 587 some experiments to determine the relative speeds
 588 of the SGI machine vs. an ASCII Red processor
 589 and found the ratio to be 1.92 for the same size
 590 decision tree learning problem. So, the Red pro-
 591 cessor is significantly faster. Normalizing for pro-
 592 cessor speed, the distributed learner can be created
 593 approximately 37 times faster than learning from
 594 all the data.

595 5. Conclusions and discussion

596 The results support several important conclu-
 597 sions. The overall conclusion is that datasets too
 598 large to handle practically in the memory of the
 599 typical computer are appropriately handled by
 600 simple partitioning to form a committee of classi-
 601 fiers. More specifically, a committee created using
 602 disjoint partitions can be expected to perform at
 603 least as well as a committee created using the same
 604 number and size of bootstrap aggregates (“bags”).
 605 Also, the performance of the committee of classi-
 606 fiers can be expected to exceed that of a single
 607 classifier built from all the data.

608 The following considerations may provide in-
 609 sight into the pattern of results. Practical factors
 610 aside, one generally wants (a) each classifier in a
 611 committee to be formed using as much data as
 612 possible, and (b) the size of the committee to be as
 613 large as possible. Practical considerations typically
 614 (a) limit the amount of data that can be used in
 615 training a single classifier, and (b) limit the size of a
 616 classifier committee. If the data set is large enough,
 617 or the memory limit small enough, then parti-
 618 tioning into N disjoint subsets gives a reasonable
 619 size committee and this approach should suffice. If
 620 the N disjoint partitions result in too small of a
 621 committee, then the data set may be partitioned
 622 multiple times to increase committee size, as we
 623 did in Section 4.4. A typical result from parti-
 624 tioning the data multiple times is shown in Table 6
 625 for the “small” Letter data set. The data set is
 626 broken into a two partition five different ways. A
 627 10-fold cross-validation is done, meaning that the
 628 test set is left out and then the training set parti-
 629 tioned into halves (eight different ways). From the
 630 table, it is clear that accuracy continues to increase
 631 and with 16 partitions we are approaching the
 632 accuracy of pure bagging (93.8% for 100 bags),
 633 considerably outperforming a C4.5 generated de-
 634 cision tree from the whole data set.

635 Results obtained here seem to support the po-
 636 sition that bagging results depend simply on ob-
 637 taining a diverse set of classifiers (Breiman, 1996;
 638 Chawla et al., 2001; Dietterich, 2000; Domingos,
 639 1996). Building classifiers on disjoint partitions of
 640 the data provides a set of classifiers that meet this
 641 requirement. Each individual classifier performs
 642 similarly, but correctly classifies a (partially) dif-
 643 ferent set of examples.

Table 6
 Multiple two partitions on the Letter data set for a 10-fold
 cross-validation using C4.5

Number of two partitions	Accuracy	Standard deviation
1	83.645	1.078
2	89.225	0.522
3	91.21	0.589
4	91.885	0.461
5	92.26	0.427

C4.5 is 88.1% accurate in a 10-fold CV.

644 Some researchers have suggested that many
645 large-data-set problems can be solved using only a
646 fraction of the data, perhaps by simple subsam-
647 pling. However, recent work has suggested that all
648 the data (even for very large training data sets)
649 may result in the maximal accuracy classifier
650 (Perlich et al., in press). Classical pattern recog-
651 nition would suggest that this question is more
652 appropriately viewed in terms of the density of
653 training sample population in the feature space,
654 rather than simply the size of the dataset. There is
655 excess data only when (parts of) feature space are
656 densely populated. The fact that the average
657 (1/8)th partition of our large dataset had perfor-
658 mance of 71.21%, whereas a single classifier
659 trained on all the data gave 75.72%, indicates that
660 the original data could not be profitably subsam-
661 pled in a simple way. Given that the problem has a
662 340-dimension feature space, this is perhaps not
663 surprising, as even 3.6 million examples can result
664 in a sparse population of such a space.

665 Acknowledgements

666 This work was supported in part by the United
667 States Department of Energy through the Sandia
668 National Laboratories LDRD program and ASCI
669 VIEWS Data Discovery Program, contract num-
670 ber DE-AC04-76DO00789. A portion of the work
671 was presented at CVPR 2001. Thanks to Robert
672 Banfield for his help with experimental work.
673 Thanks also to the referees for their insightful
674 comments and guidance.

675 References

676 Bauer, E., Kohavi, R., 1999. An empirical comparison of voting
677 classification algorithms: Bagging, boosting and variants.
678 Machine Learning 36 (1,2).
679 Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat,
680 T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E., 2000. The
681 protein data bank. Nucleic Acids Research 28, 235–242,
682 Available from <<http://www.pdb.org/>>.
683 Blake, C.L., Merz, C.J., 1998. UCI repository of machine
684 learning databases. Available from <<http://www.ics.uci.edu/~mllearn/MLRepository.html>>.
685
686 Bowyer, K.W., Chawla, N.V., Moore Jr., I.E., Hall, L.O.,
687 Kegelmeyer, W.P., 2000. A parallel decision tree builder for

mining very large visualization datasets. In: IEEE System,
Man, and Cybernetics Conference, pp. 1888–1893. 688
Breiman, L., 1996. Bagging predictors. Machine Learning 24 689
(2), 123–140. 690
Breiman, L., 1999. Pasting bites together for prediction in large 692
data sets. Machine Learning 36 (1–2), 85–103. 693
Chan, P., Stolfo, S., 1993. Towards parallel and distributed 694
learning by meta-learning. In: Working Notes AAAI 695
Workshop on Knowledge Discovery in Databases, pp. 696
227–240. 697
Chan, P., Stolfo, S., 1995. Learning arbiter and combiner trees 698
from partitioned data for scaling machine learning. In: Proc. 699
Intl. Conf. on Knowledge Discovery and Data Mining, pp. 700
39–44. 701
Chan, P., Stolfo, S., 1996. Scaling learning by meta-learning 702
over disjoint and partially replicated data. In: Ninth Florida 703
Artificial Intelligence Research Symposium, pp. 151–155. 704
Chawla, N., Moore Jr., T.E., Bowyer, K.W., Hall, L.O., 705
Springer, C., Kegelmeyer, W.P., 2001. Bagging is a small- 706
data-set phenomenon. In: IEEE Conf. on Computer Vision 707
and Pattern Recognition, pp. 684–689. 708
Darlington, J., Guo, Y., Sutiwaraphun, J., To, H., 1997. 709
Parallel induction algorithms for data mining. In: Advances 710
in Intelligent Data Analysis Reasoning about Data, Second 711
Internat. Symposium, IDA-9 7. Proc., pp. 437–445. 712
Dietterich, T., 2000. An experimental comparison of three 713
methods for constructing ensembles of decision trees: 714
Bagging, boosting, and randomization. Machine Learning 715
40 (2), 139–158. 716
Domingos, P., 1996. Using partitioning to speed up specific-to- 717
general rule induction. In: Proc. AAAI-96 Workshop on 718
Integrating Multiple Learned Models. AAAI Press, Port- 719
land, OR, pp. 29–34. 720
Domingos, P., 1997. Why does bagging work? A Bayesian 721
account and its implications. In: Proc. Third Internat. Conf. 722
on Knowledge Discovery and Data Mining. 723
Fahlman, S.E., Lebiere, C., 1990. The cascade-correlation 724
learning architecture. In: Advances in Neural Information 725
Processing Systems 2. Morgan Kaufmann, Los Altos, CA. 726
Hall, L.O., Bowyer, K.W., Kegelmeyer, W.P., Moore Jr., T.E., 727
Chao, C., 2000. Distributed learning on very large data sets. 728
In: ACM SIGKDD Workshop on Distributed and Parallel 729
Knowledge Discovery. 730
Hall, L.O., Chawla, N.V., Bowyer, K.W., Kegelmeyer, W.P., 731
1999. Learning rules from distributed data. In: ACM 732
SIGKDD Workshop on Large-Scale Parallel Data Mining 733
Systems. 734
Jones, D.T., 1999. Protein secondary structure prediction based 735
on decision-specific scoring matrices. Journal of Molecular 736
Biology 292, 195–202. 737
Lawrence Livermore National Laboratories, 2001. Protein 738
structure prediction center. Available from <<http://predictioncenter.llnl.gov/>>. 739
740
Moore, A.W., Lee, M.S., 1998. Cached sufficient statistics for 741
efficient machine learning with large datasets. Journal of 742
Artificial Intelligence Research 8, 67–91. 743

- 744 Oates, T., Jensen, D., 1998. Large datasets lead to overly
745 complex models: an explanation and a solution. In: Proc.
746 Fourth Internat. Conf. on Knowledge Discovery and Data
747 Mining, August.
- 748 Perlich, C., Provost, F., Simonoff, J., 2002. Tree induction vs.
749 logistic regression: a learning-curve analysis. *J. Machine*
750 *Learning Res.*, in press.
- 751 Provost, F.J., Hennessy, D.N., 1996. Scaling up: Distributed
752 machine learning with cooperation. In: Proc. 13th National
753 Conf. on Artificial Intelligence, AAAI'96, pp. 74–79.
- 754 Provost, F.J., Jensen, D., Gates, T., 1999. Efficient progressive
755 sampling. In: Proc. Fifth ACM SIGKDD Internat. Conf.
756 on Knowledge Discovery and Data Mining, pp. 23–32.
- Quinlan, J.R., 1996. Bagging, boosting, and C4.5. In: Proc. 757
Thirteenth National Conf. on Artificial Intelligence, pp. 758
725–730. 759
- Quinlan, J.R., 1992. C4.5: Programs for Machine Learning. 760
Morgan Kaufmann, San Mateo, CA. 761
- Sandia National Labs, 1998. ASCI RED, the world's first 762
teraops supercomputer. Available from <[http://www.san-](http://www.sandia.gov/ASCI/Red/)
763 [dia.gov/ASCI/Red/](http://www.sandia.gov/ASCI/Red/)>. 764
- Shafer, J., Agrawal, R., Mehta, M., 1996. Sprint: A scalable 765
parallel classifier for data mining. In: Proc. 22nd VLDB 766
Conf., Mumbai (Bombay), India, pp. 1–12. 767
- Street, W.N., Kim, Y., 2001. A streaming ensemble algorithm 768
(SEA) for large-scale classification. In: Provost, F., Srikant, 769
R. (Ed.), KDD'01, San Francisco, CA, pp. 377–382. 770