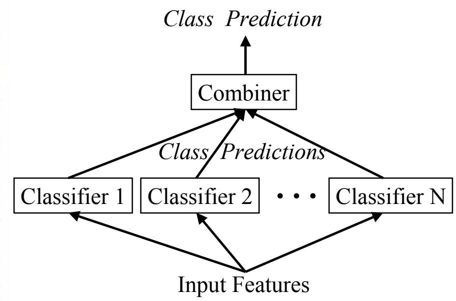
Ensemble Learning

- what is an ensemble?
- why use an ensemble?
- selecting component classifiers
- selecting combining mechanism
- some results

A Classifier Ensemble



Key Ensemble Questions

Which components to combine?

- different learning algorithms
- same learning algorithm trained in different ways
- same learning algorithm trained the same way

How to combine classifications?

- majority vote
- weighted (confidence of classifier) vote
- weighted (confidence in classifier) vote
- learned combiner

What makes a good (accurate) ensemble?

Why Do Ensembles Work?

Hansen and Salamon, 1990

If we can assume classifiers are random in predictions and accuracy > 50%, can push accuracy arbitrarily high by combining more classifiers

Key assumption: classifiers are independent in their predictions

- not a very reasonable assumption
- more realistic: for data points where classifiers predict with > 50% accuracy, can push accuracy arbitrarily high (some data points just too hard)

What Makes a Good Ensemble?

Krogh and Vedelsby, 1995

Can show that the accuracy of an ensemble is mathematically related:

$$\hat{E} = \overline{E} - D$$

$\hat{E}$ is the error of the entire ensemble

$\overline{E}$ is the average error of the component classifiers

D is a term measuring the diversity of the components

Effective ensembles have accurate and diverse components

Ensemble Mechanisms - Components

- Separate learning methods
  - not often used
  - very effective in certain problems (e.g., protein folding, Rost and Sander, Zhang)
- Same learning method
  - generally still need to vary something externally
    - exception, some good results with neural networks
  - most often, data set used for training varied:
    - Bagging (Bootstrap and Aggregate), Breiman
    - Boosting, Freund & Schapire
      - Ada, Freund & Schapire
      - Arcing, Breiman

Ensemble Mechanisms - Combiners

- Voting
- Averaging (if predictions not 0,1)
- Weighted Averaging
  - base weights on confidence in component
- Learning combiner
  - Stacking, Wolpert
    - general combiner
  - RegionBoosting, Maclin
    - piecewise combiner

Bagging

Varies data set

Each training set a *bootstrap* sample

bootstrap sample - select set of examples (with replacement) from original sample

Algorithm:

for k = 1 to *#classifiers*

train′ = bootstrap sample of train set

create classifier using *train′* as training set

combine classifications using simple voting

Weak Learning

Schapire showed that a set of weak learners (learners with > 50% accuracy, but not much greater) could be combined into a strong learner

Idea: weight the data set based on how well we have predicted data points so far

- data points predicted accurately - low weight
- data points mispredicted - high weight

Result: focuses components on portion of data space not previously well predicted

Boosting - Ada

Varies weights on training data

Algorithm:

for each data points: weight w_i to *1/#datapoints*

for k = 1 to *#classifiers*

generate $classifier_k$ with current weighted train set

ε_k = sum of w_i's of misclassified points

$\beta_k = 1 - \varepsilon_k / \varepsilon_k$

multiply weights of all misclassified points by β_k

normalize weights to sum to 1

combine: weighted vote, weight for $classifier_k$ is $\log(\beta_k)$

Q: what to do if $\varepsilon_k = 0.0$ or $\varepsilon_k > 0.5$?

Boosting - Arcing

Sample data set (like Bagging), but probability of data point being chosen weighted (like Boosting)

m_i = #number of mistakes made on point i by previous classifiers

probability of selecting point i :

$$prob_i = \frac{1 + m_i^{\,4}}{\sum_{j=0}^{N} 1 + m_j^{\,4}}$$

Value 4 chosen empirically

Combine using voting

Some Results - BP, C4.5 Components

Dataset	C4.5	BP	BagC4	BagBP	AdaC4	AdaBP	ArcC4	ArcBP
letter	14.0	18.0	7.0	10.5	4.1	5.7	3.9	4.6
segment	3.7	6.6	3.0	5.4	1.7	3.5	1.5	3.3
promoter	12.8	5.3	10.6	4.0	6.8	4.5	6.4	4.6
kr-vs-kp	0.6	2.3	0.6	0.8	0.3	0.4	0.4	0.3
splice	5.9	4.7	5.4	3.9	5.1	4.0	5.3	4.2
breastc	5.0	3.4	3.7	3.4-	3.5	3.8-	3.5	4.0-
housev	3.6	4.9	3.6	4.1	5.0-	5.1-	4.8-	5.3-

Some Theories on Bagging/Boosting

Error = Bayes Optimal Error + Bias + Variance

Bayes Optimal Error = noise error

Theories:

- Bagging can reduce variance part of error
- Boosting can reduce variance AND bias part of error
- Bagging will hardly ever increase error
- Boosting may increase error
- Boosting susceptible to noise
- Boosting's increases *margins*

Combiner - Stacking

Idea:

- generate component (level 0) classifiers with part of the data (half, three quarters)
- train combiner (level 1) classifier to combine predictions of components using remaining data
- retrain component classifiers with all of training data

In practice, often equivalent to voting

Combiner - RegionBoost

- Train "weight" classifier for each component classifier
- "weight" classifier predicts how likely point will be predicted correctly
- "weight" classifiers: k-Nearest Neighbor, Backprop
- Combiner, generate component classifier prediction and weight using corresponding "weight" classifier
- Small gains in accuracy