Classifier Selection

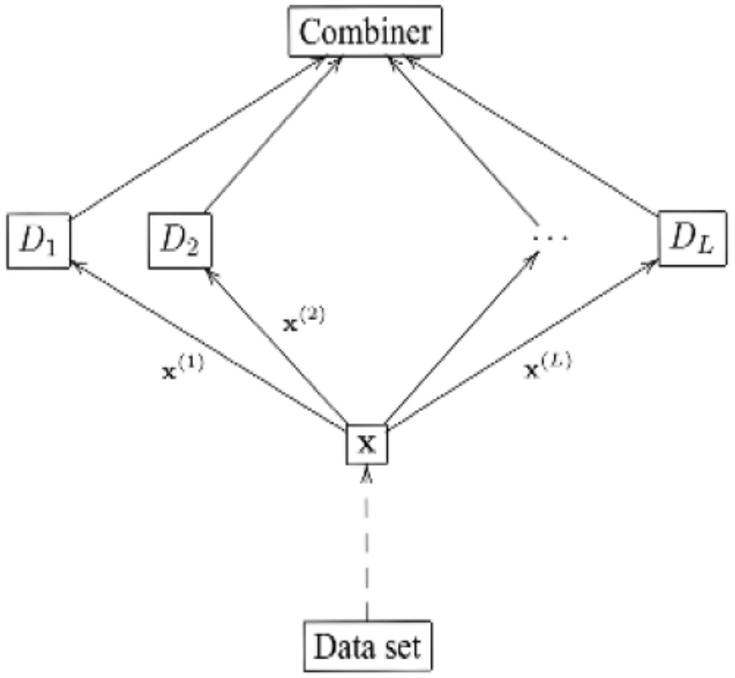
Nicholas Ver Hoeve
Craig Martek
Ben Gardner

Classifier Ensembles

Assume we have an ensemble of classifiers with a well-chosen feature set.

We want to optimize the competence of this system. Simple enhancements include:

- Improve/train each classifier
- Add or remove classifiers if the modification increases accuracy
- Improve Combiner



Classifier Selection

Using the classifier ensemble model as given, high, consistent accuracy on each classifier is generally preferred.

However, consider the idea that some classifiers excel at differentiating between certain *subspaces* of the input vector domain; but whose *overall* accuracy may be lacking.

That is, assume a classifier can have a *domain of expertise* which is less than the entire feature space.

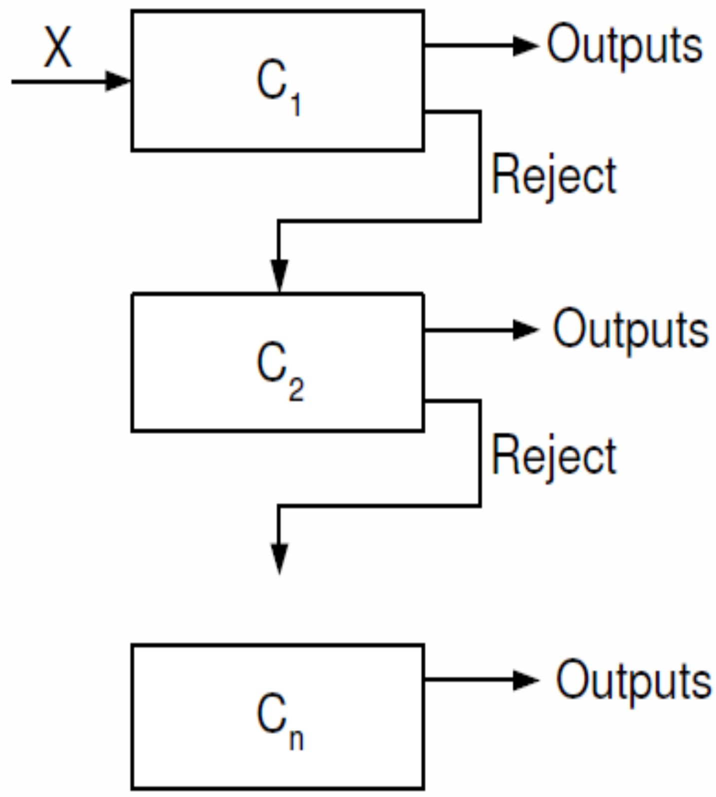
Classifier Selection

To take advantage of classifiers' "domains of expertise", we can:

- Rely on the combiner to detect when this occurs based on the class labels it receives on input
  - Possible if rejection is allowed or degrees of confidence are used
  - Normally, combiner cannot see input **x** directly
  - Due to the canonical ensemble structure, all classifiers, (including poor classifiers for the region) receive and classify the input- even if the result is unused
- Modify the ensemble structure, for example:
  - Use a *Cascade Structure* ; to be discussed
  - Use *Selection Regions* ; to be discussed

Cascade Classifiers

- Excellent for real time systems
  - Typically classifies 'easy' inputs in less time
  - Majority of inputs use only a few classifiers
- Permits additional 'fail-safety' in exceptional cases that may be too slow to run for all inputs



Classifier Selection

We can estimate the confidence of a classifier in terms of posterior probability with the following equation:

If the classifier outputs are reasonably well-calibrated estimates of the posterior probabilities, that is, $d_{i,j} = \hat{P}(\omega_j|\mathbf{x}, D_i)$, then the confidence of classifier $D_i \in \mathcal{D}$ for object $\mathbf{x}$ can be measured as

$$C(D_i|\mathbf{x}) = \max_{j=1}^{c} \hat{P}(\omega_j|\mathbf{x}, D_i) \tag{6.1}$$

Aside from the statement itself, also of note is that the domain of **x** is now D_i, that is, *not* the entire feature space.

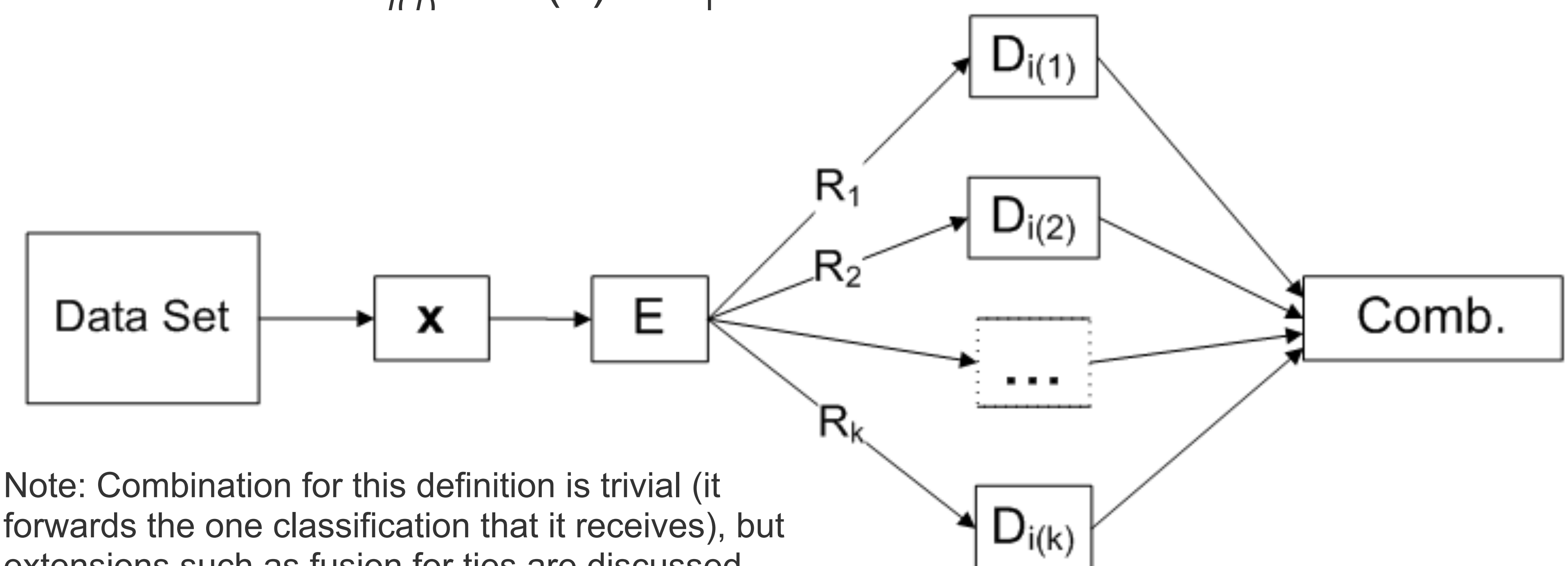
- Posterior probabilities have always depended on **x**; however we previously assumed non-biased **x** for fairness in the experiment.
  - The preliminary assignment of **x** to a classifier can introduce a favorable bias.

Preliminary Questions

- How do we build the individual classifiers?
- How do we evaluate the competence of classifiers for a given **x**? If several classifiers tie as the most competent candidates, how do we break the tie?
- Once the competences are found, what selection strategy will we use?
  - The standard strategy is to select the most competent classifier and take its decision
    - But if several tie for highest competence, do we take one decision or shall we fuse their decisions?
    - When is it beneficial to select one classifier to label **x** when we should be looking for a fused decision?

Selection Regions

- Assume we have a set of classifiers
  - $D = \{D_{1,} D_{2, \ldots,} D_L\}$
- Let $\mathbf{R}^n$ be divided into K *selection regions* (also called *regions of competence*) called $\{R_{1,} R_{2, \ldots,} R_k\}$
- Let E map each input **x** to its corresponding Region R_j
  - $E : \mathbf{x} \rightarrow R_{j,}$ where R_j is the region for which $D_{i(j)}$ is applied
- Feed **x** into $D_{i(j)}$ iff $E(\mathbf{x}) = R_j$



Note: Combination for this definition is trivial (it forwards the one classification that it receives), but extensions such as fusion for ties are discussed later.

Selection Regions

Let D^* be the classifier with the highest average accuracy among the elements of $\mathcal{D}$ over the whole feature space $\Re^n$. Denote by $P(D_i|R_j)$ the probability of correct classification by D_i in region R_j. Let $D_{i(j)} \in \mathcal{D}$ be the classifier responsible for region $R_j, j = 1, \ldots, K$. The overall probability of correct classification of our classifier selection system is

$$P(correct) = \sum_{j=1}^{K} P(R_j) P(D_{i(j)}|R_j) \tag{6.2}$$

where $P(R_j)$ is the probability that an input $\mathbf{x}$ drawn from the distribution of the problem falls in R_j. To maximize $P(correct)$, we assign $D_{i(j)}$ so that

$$P(D_{i(j)}|R_j) \geq P(D_t|R_j),\ \forall\, t = 1, \ldots, L \tag{6.3}$$

Ties are broken randomly. From Eqs. (6.2) and (6.3),

$$P(correct) \geq \sum_{j=1}^{K} P(R_j) P(D^*|R_j) = P(D^*) \tag{6.4}$$

Selection Regions

- From the previous equation, the ensemble is at least as accurate as the most accurate classifier.
  - True for any partition of the feature space
  - We must be careful to select the most accurate classifier for each region- this is often not easy
- Partitioning can decrease runtime by supporting classifiers that are not always needed (compared with the option of running classifiers that may sometimes be ignored)
  - Important to point out because the canonical ensemble with rejection dominates any Selection-Region System
    - That is, we can construct an ensemble with rejection that has the same output

*by modifying each classifier to always reject if the input is beyond its 'region'

Dynamic Competence Estimation

- Estimation is done during classification
- Decision-independent
  - Do not need label output by classifier for input
- Decision-dependent
  - Label for input by all classifiers is known

Direct k-nn

- Decision-independent
  - Accuracy of classifier on k-nn of input
- Decision-dependent
  - Use k-nn of input labeled with same class
- Competence is accuracy on these neighbors

Distance-based k-nn

- Uses actual output of classifiers
- Decision-independent

$$C(D_i|\mathbf{x}) = \frac{\sum_{\mathbf{z}_j \in N_{\mathbf{x}}} P_i(l(\mathbf{z}_j)|\mathbf{z}_j)(1/d(\mathbf{x}, \mathbf{z}_j)}{\sum_{\mathbf{z}_j \in N_{\mathbf{x}}} (1/d(\mathbf{x}, \mathbf{z}_j)}$$

- Decision-dependent

$$C(D_i|\mathbf{x}) = \frac{\sum_{\mathbf{z}_j} P_i(s_i|\mathbf{z}_j)1/d(\mathbf{x}, \mathbf{z}_j)}{\sum_{\mathbf{z}_j} 1/d(\mathbf{x}, \mathbf{z}_j)}$$

Potential Functions

- Decision-independent

$$\phi(\mathbf{x}, \mathbf{z}_j) = \frac{g_{ij}}{1 + \alpha_{ij}(d(\mathbf{x}, \mathbf{z}_j))^2} \qquad \phi(\mathbf{x}, \mathbf{z}_j) = \frac{g_{ij}}{1 + \alpha_{ij}(d(\mathbf{x}, \mathbf{z}_j))^2}$$

- g_{ij} is 1 if D_i recognizes $\mathbf{z}_j$ correctly, -1 if not
- α gives the contribution to the field of $\mathbf{z}_j$

15 nearest neighbors of input **x** ($\mathbf{z}_j$ = distance)

Object ($\mathbf{z}_j$)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
True label ($l(\mathbf{z}_j)$)	2	1	2	2	3	1	2	1	3	3	2	1	2	2	1
Guessed label (s_i)	2	3	2	2	1	1	2	2	3	3	1	2	2	2	1

- Direct k-nn
  - Decision-independent $= 0.666$
  - Decision-dependent (ω_2, k=5) = 0.8
- Distance-based k-nn
  - Decision-independent ≈ 0.7
  - Decision-dependent (ω_2) ≈ 0.95

Diversity

- A Dynamic Classifier Selection Method to Build Ensembles using Accuracy and Diversity
- Measure accuracy and diversity

$$DF_{i,k} = \frac{N^{00}}{N^{11} + N^{10} + N^{01} + N^{00}}$$

- Select most accurate classifiers, then most diverse of those
- Use a fusion method

Tie-breaking

- If all classifiers agree on a label, choose it
- Otherwise, calculate accuracy of classifiers
- If a label can be picked by the most accurate or a plurality of tied classifiers, choose that
- Next highest confidence is used to break tie
- Random amongst tied labels if we get this far

TABLE 6.1 Tie-Break Examples for the Dynamic Classifier Selection Model for an Ensemble of Nine Classifiers.

	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9
Class labels	1	2	1	2	3	2	3	1	2
Competences	0.8	0.7	0.7	0.6	0.6	0.6	0.4	0.3	0.2
Final label: **1** (single most competent classifier)									
Class labels	1	2	2	2	3	2	3	1	2
Competences	0.7	0.7	0.7	0.6	0.6	0.6	0.4	0.3	0.2
Final label: **2** (majority of the competence-tied classifiers)									
Class labels	1	2	2	2	3	2	3	1	2
Competences	0.7	0.7	0.6	0.5	0.5	0.5	0.4	0.3	0.2
Final label: **2** (competence-tie resolved by the second most competent classifier)									
Class labels	1	2	1	1	2	2	3	1	2
Competences	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7
Final label: **1** (random selection between competence-tied labels)									
Class labels	1	2	2	2	3	2	3	1	2
Competences	0.7	0.7	0.6	0.6	0.6	0.6	0.4	0.3	0.2
Final label: **1** (random selection between competence tied-labels (item 5))									

Regions of Competence

- Dynamic Estimation of Competence might be too computationally demanding.
- Instead of identifying most competent classifier for input **x** (local), identify classifier for the region **x** falls in.
- Needs reliable estimates of competence across regions to perform well.
- Most competent classifier is picked for each region.
- Region Assignment has larger effect on accuracy than competence estimation technique.

Clustering

- Used to ensure each region has sufficient data.
- Method 1: Clustering and selection
  - Splits feature space into K regions.
  - Finds K clusters (defining regions) and cluster centroids.
  - For input **x**, find most competent classifier for closest cluster.
- Method 2: Selective Clustering
  - Splits feature space into more clusters; smaller regions.
  - Splits data set into positive examples (Z^+) and negative examples (Z^-) for each classifier.
  - One cluster in Z^+ for each class (total c), K_i clusters in Z^-.
  - **x** placed in region with closest center (Mahalanobis distance) and classified by most competent classifier.

Clustering and Selection

Clustering and selection (training)

1. Design the individual classifiers $D_1, \ldots, D_L$ using the labeled data set $\mathbf{Z}$. Pick the number of regions K.
2. Disregarding the class labels, cluster $\mathbf{Z}$ into K clusters, $C_1, \ldots, C_K$, using, e.g., the K-means clustering procedure [2]. Find the cluster centroids $\mathbf{v}_1, \ldots, \mathbf{v}_K$ as the arithmetic means of the points in the respective clusters.
3. For each cluster C_j, (defining region R_j), estimate the classification accuracy of $D_1, \ldots, D_L$. Pick the most competent classifier for R_j and denote it by $D_{i(j)}$.
4. Return $\mathbf{v}_1, \ldots, \mathbf{v}_K$ and $D_{i(1)}, \ldots, D_{i(K)}$.

Fig. 6.2 *Training of the clustering and selection method.*

Clustering and selection (operation)

1. Given the input $\mathbf{x} \in \Re^n$, find the nearest cluster center from $\mathbf{v}_1, \ldots, \mathbf{v}_K$, say, $\mathbf{v}_j$.
2. Use $D_{i(j)}$ to label $\mathbf{x}$.

Fig. 6.3 *Operation of the clustering and selection method.*

Selection or Fusion?

- Recurring theme: competences of the regions need to be reliable enough
    - Otherwise can overtrain and generalize poorly
- Can run statistical tests (paired t-test) to determine whether classifier for specific region is significantly better than other classifiers
- Can determine difference in accuracy needed to be significant for different sample sizes and accuracies.

Selection or Fusion?

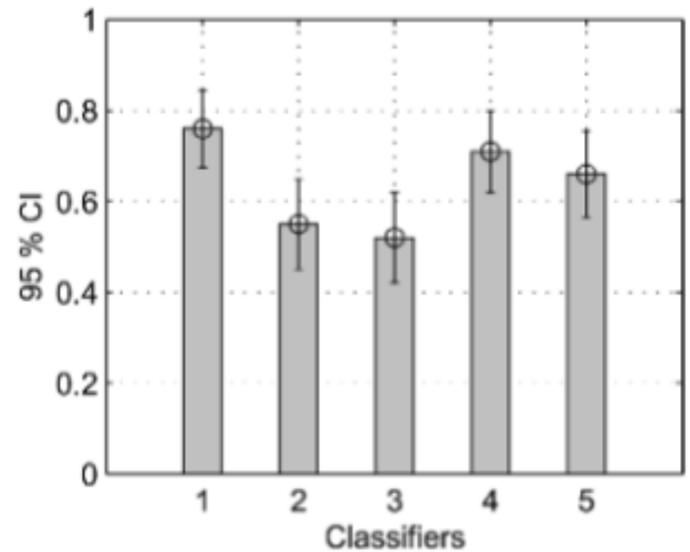


Fig. 6.4 95 percent confidence intervals (*CI*) for the five classifiers.

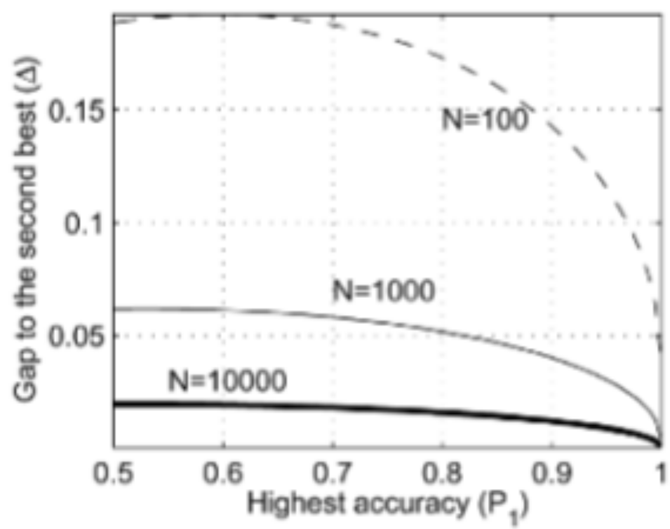


Fig. 6.5 *The difference Δ between the best and the second best classification accuracies in region R_j guaranteeing that the 95 percent CI of the two do not overlap.*

Mixture of Experts (ME)

- Uses a separate classifier that determines the "participation" of classifiers for determining class label of **x**
- Gating Network
  - input: **x**
  - output: $p_1(\mathbf{x})$, $p_2(\mathbf{x})$, ..., $p_L(\mathbf{x})$
    - $p_i(x)$ = probability that Di is the most competent expert for input **x**
- Selector chosen based on $p_i(x)$'s.
  - Stochastic selection, Winner takes all, Weighted
- Training the ME model
  - Gradient descent, Expectation Maximization

Mixture of Experts (ME)

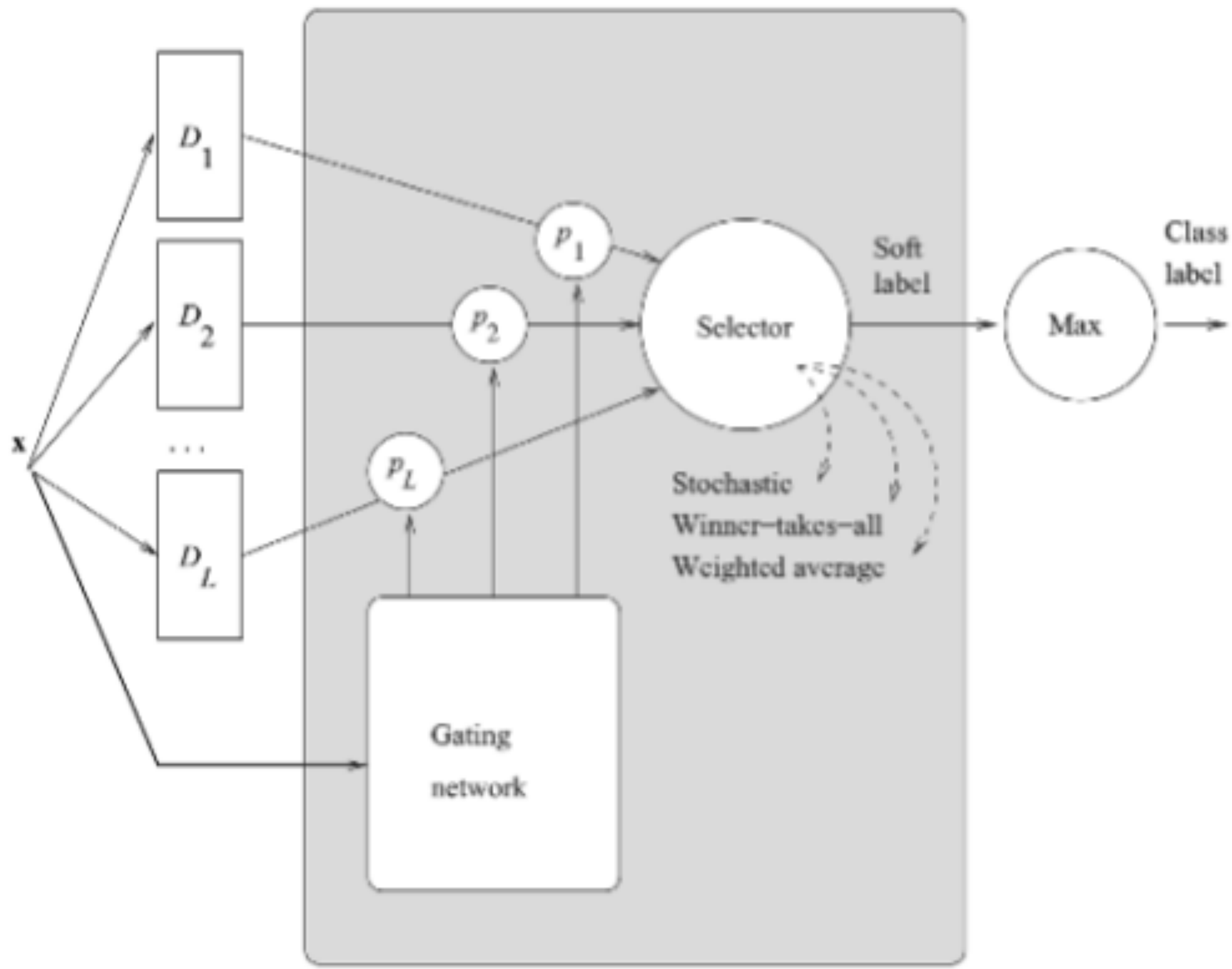


Fig. 6.6 *Mixture of experts.*

References

- K. Woods, W.P. Kegelmeyer, and K. Bowyer. Combination of multiple classifiers using local accuracy estimates. *IEEE Transactions on Pattern Analysis and Machine Intelligence* , 19:405-410, 1997.
- A. Santana, R. Soares, A. Canuto, and M. de Souto. A Dynamic Classifier Selection Method to Build Ensembles using Accuracy and Diversity. *Ninth Brazilian Symposium on Neural Networks* , pp. 36-41, 2006.
- L.Oliveira, A. Britto Jr., R. Sabourin. Improving Cascading Classifiers with Particle Swarm Optimization. *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)* , 2005

Questions?