Examples of model selection problems

- What is the best order of my polynomial model?
- Which neural net architecture gives the best generalization error?
- How many neighbors should I take in consideration in a nearest-neighbor algorithms?
- Should I use a linear model, a decision tree, a neural net, a local learning algorithms or a random guesser?
- Which of the 50 features are relevant for this problem?

Eager and lazy learners

A distinction between models in machine learning is between

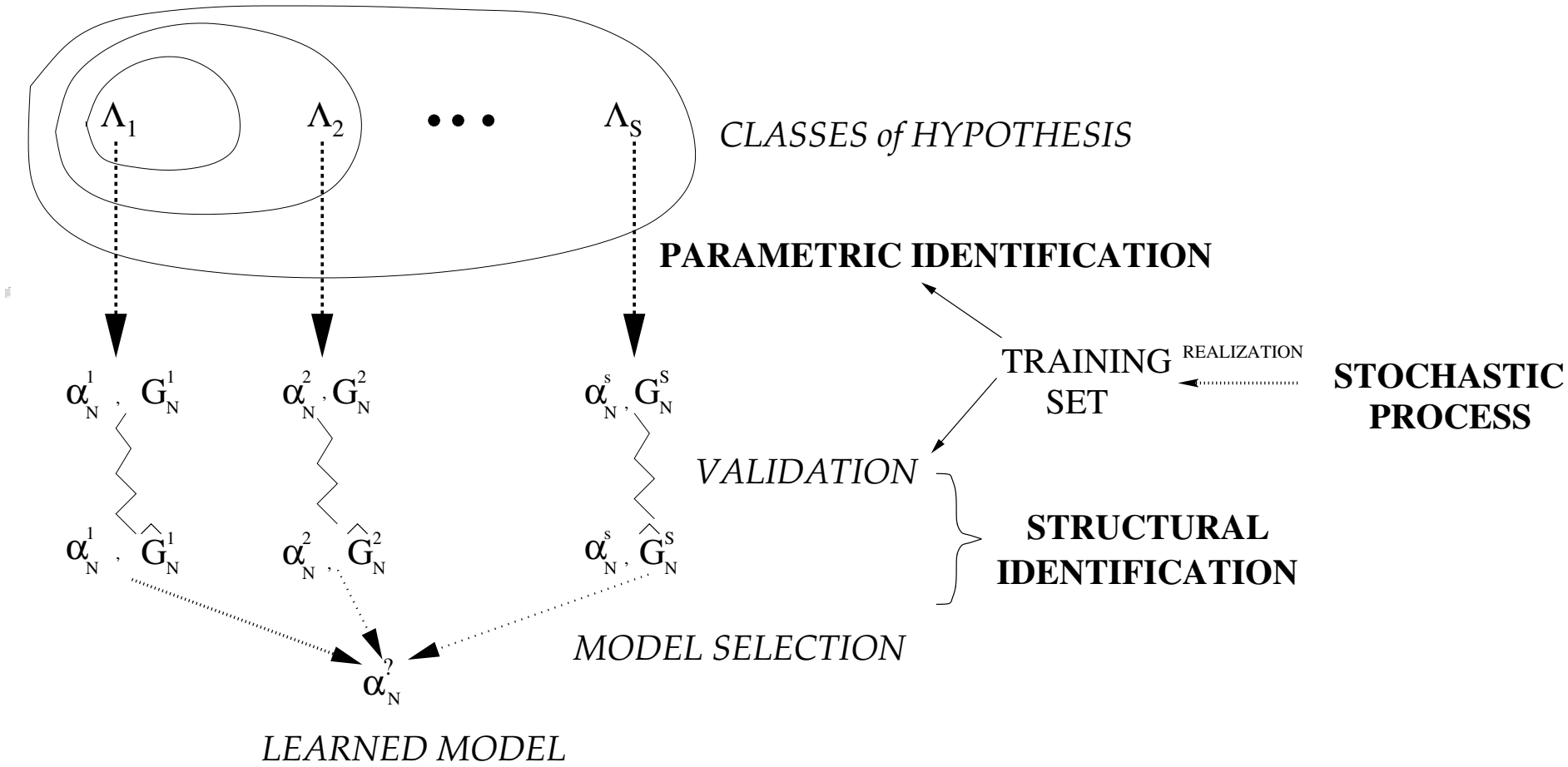
Eager models : they use the training set to calibrate the model parameters and then they discard the model. They typically demand an expensive identification procedure (whose cost if C) but allow a fast prediction ($P \approx 0$). Examples are neural networks.

Lazy models : they keep the training set stored in memory and when a prediction is required they access it and return the prediction. They typically have no identification procedure ($C \approx 0$) but a prediction time $P \neq 0$. Examples are nearest neighbor techniques and locally weighted regression.

Model selection

- Model selection concerns the final choice of the model structure in the set that has been proposed by model generation and assessed by model validation.
- In real problems, this choice is typically a subjective issue and is often the result of a compromise between different factors, like the quantitative measures, the personal experience of the designer and the effort required to implement a particular model in practice.
- Here we will consider only quantitative criteria. Two are the possible approaches:
  1. the *winner-takes-all* approach
  2. the *combination of estimators* approach.

Winner-takes-all



The best hypothesis is given by $h(\cdot, \alpha_N^{s^*})$ where

$$s^* = \arg \min_{s=1,\ldots,S} \widehat{\mathsf{MSE}}^s$$

This model is the one used for future predictions.

Searching for the best model

- Suppose that the number m of alternative models is very high.
- Winner-takes-all searches for the model among the m which should guarantee the least generalization error.
- This demand firsts the estimation of the generalization error for each model and then the selection.
- Selection is an optimisation problem and traditionally there have been a number of popular ways to search through a large collection of models.

The cost of assessing

- We have seen how cross-validation is a computationally expensive procedure.
- K-fold cross-validation techniques require the repetition of the identification procedure (having cost C) and the prediction procedure (having cost P) for K times.
- If $K = N$, we have leave-one-out and the cost becomes still higher.
- This is very expensive for all types of models (eager and lazy) but especially for eager models if $C >> P$.

The cost of searching

Brute force: given m models and a test set, they simply assess all the models on the test set and select the one with the lowest error. If the number of candidate models is large or if the number of testing points is high, this process is computationally expensive.

Descent methods: they treat the collection of models and their prediction error as a continuous and differentiable surface. They start at some point on the surface and proceed to descend in the direction that has less error. They are much faster than the brute force methods but they are prone to local minima: they can end up with a model that is arbitrarily worse than the best one. Also, sometimes, they cannot be used since there is no distance metric on the space of discrete models.

Non gradient-based methods: examples are genetic algorithms and simulated annealing. They are less sensible to the problem of local minima but they require a distance metric between models.

The Hoeffding's inequality

Theorem 1. *Let* $\mathbf{z}_1, \ldots, \mathbf{z}_N$ *be independent bounded random variables such that* $\mathbf{z}_i$ *falls in the interval* $[a_i, b_i]$ *with probability one. Let their sum be* $\mathbf{S}_N = \sum_{i=1}^N \mathbf{z}_i$*. Then for any* $\varepsilon > 0$ *we have*

$$P\left\{|S_N - E[\mathbf{S}_N]| > \varepsilon\right\} \leq 2\exp\left\{-2\varepsilon^2 / \sum_{i=1}^N (b_i - a_i)^2\right\}$$

Corollary 1. *If the variables* $\mathbf{z}_1, \ldots, \mathbf{z}_N$ *are independent and identically distributed, the following bound on the discrepancy between the sample mean* $\bar{z} = \frac{\sum_{i=1}^N z_i}{N}$ *and the expected value* $E[\mathbf{z}]$ *holds*

$$P\left\{|\bar{z} - E[\mathbf{z}]| > \varepsilon\right\} \leq 2\exp\left\{-2N\varepsilon^2/(b-a)^2\right\}$$

Assume that δ is a confidence parameter, that is we are $100(1-\delta)\%$ confident that the estimate $\bar{z}$ is within ε of the true expectation. It is possible to derive the expression

$$\varepsilon(N) = \sqrt{\frac{(b-a)^2 \log(2/\delta)}{2N}}$$

which measures with confidence $1-\delta$ how the sample mean $\bar{z}$, estimated on the basis of N points, is close to the expectation $E[\mathbf{z}]$. We can also determine the number of samples N necessary to obtain an accuracy ε and a confidence δ by using the relation

$$N > \frac{(b-a)^2 \log(2/\delta)}{2\varepsilon^2}$$

Hoeffding's bound is a general bound that only relies on the assumption that samples are drawn independently. Bayesian bounds are another example of statistical bounds which give tighter results under the assumption that samples are drawn from a normal distribution.

The racing technique

- An alternative technique called **racing** has been proposed by Maron and Moore in 1994.
- The idea is to test the various models in parallel, one point at a time.
- This technique tests the set of models in parallel, quickly discards the models that are clearly inferior and concentrates the computational effort on differentiating among the better models.
- The models which are significantly worse than the best ones are thrown out of the race and not tested again.

Statistical tests and racing

- A running average is maintained for each model's error.
- Using statistical bounds, like the Hoeffding's inequality, it is possible to compute the gap between the true generalization error and the test error computed on the basis of a growing number of test samples.
- The more test points that are seen, the tighter the estimated error is to the true error.

The racing algorithm

Consider a set of m candidate models and on a set D_N of N observations. At each iteration of the algorithm

1. we randomly select a point $\langle x_i, y_i \rangle$ from the set D_N,
2. we compute the leave-one-out error

$$\left(y_i - h_j(x_i, \alpha_N^{-i})\right)^2$$

   for the m models $h_j(\cdot, \alpha_N)$, $j = 1, \ldots, m$
3. we update the estimate

$$\widehat{\mathsf{MSE}}_{\mathsf{loo}}^{(j)} = \frac{\sum_{i=1}^{N_{\mathsf{ts}}} \left(y_i - h_j(x_i, \alpha_N^{-i})\right)^2}{N_{\mathsf{ts}}}$$

   of each model

The racing algorithm (II)

3. Using the Hoeffding's bound we calculate the interval of confidence of the true MSE error of the jth model

$$\text{Prob}\left\{\left|\text{MSE}^{(j)} - \widehat{\text{MSE}}_{\text{loo}}^{(j)}\right| > \epsilon\right\} < 2e^{-2N_{\text{ts}}\epsilon^2}/B^2 = \delta$$

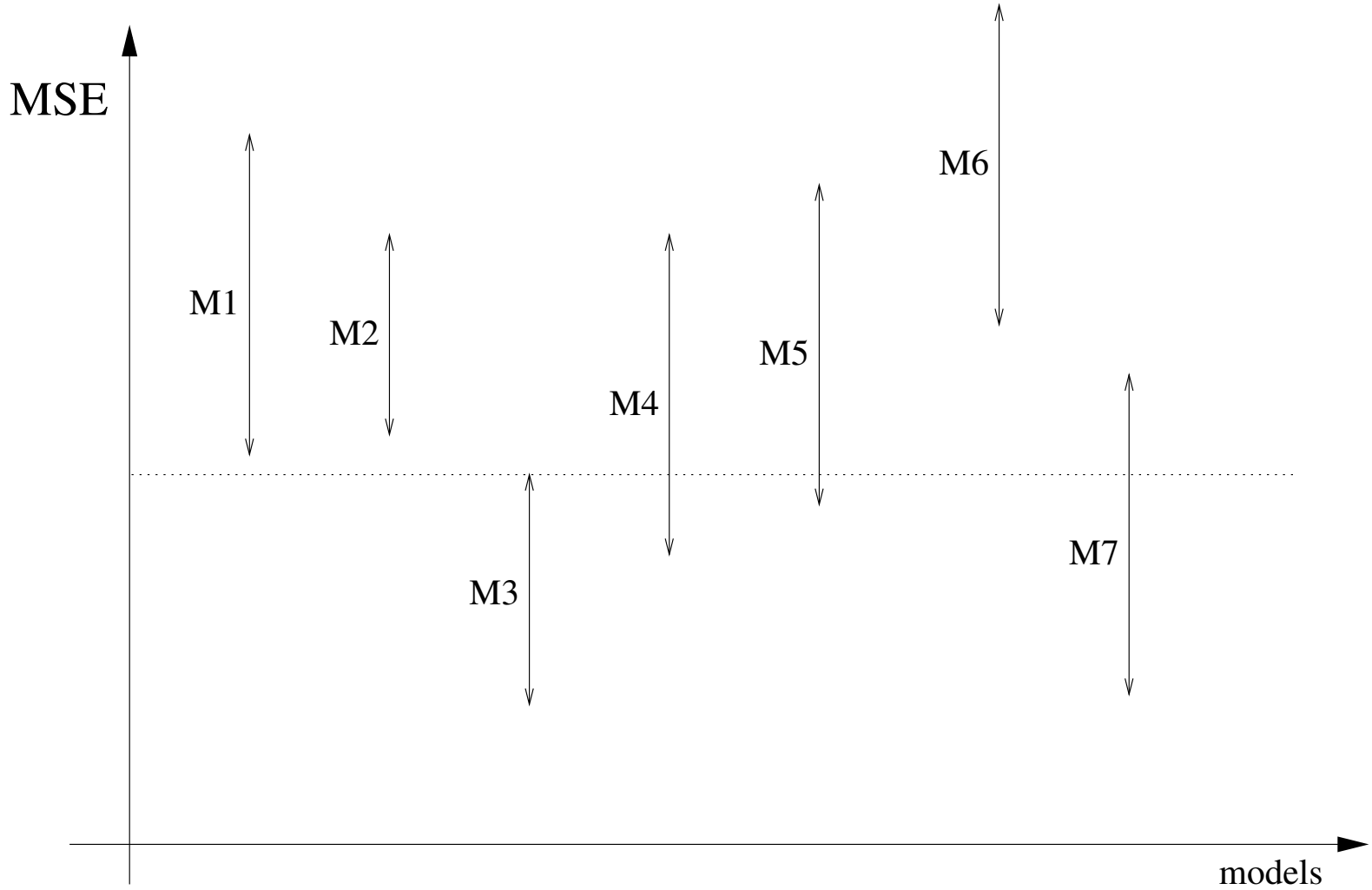
   where N_{ts} is the number of points used so far, B bounds the greatest possible error that a model can make, ϵ is the *accuracy* and δ is the *confidence*.

4. For a given δ, each model has a bound

$$\epsilon = \sqrt{\frac{B^2 \log(2/\delta)}{2N_{\text{ts}}}}$$

   within which its true average error lies. We eliminate those models whose best possible error is still greater than the worst error of the best candidate model.

The racing algorithm



The best upper bound of model M3 eliminates the models M1, M2 and M6.

The racing algorithm (III)

The algorithm iterates, repeatedly picking test points until one of three conditions occurs:

1. All but one of the models have been eliminated.
2. A sufficient number (e.g. N) of test points have been picked.
3. The accuracy ϵ has reached a certain threshold.

Note that in order to guarantee that the entire algorithm has some confidence $1 - \Delta$ of returning the best model, we have to make δ extremely low.
Example:

$$\text{Prob}\left\{\left|\text{MSE}^{(1)} - \widehat{\text{MSE}}_{\text{loo}}^{(1)}\right| < \epsilon\right\} > (1-\delta) \quad \& \quad \text{Prob}\left\{\left|\text{MSE}^{(2)} - \widehat{\text{MSE}}_{\text{loo}}^{(2)}\right| < \epsilon\right\} > (1-\delta) \Rightarrow$$

$$\Rightarrow \text{Prob}\left\{\left(\left|\text{MSE}^{(1)} - \widehat{\text{MSE}}_{\text{loo}}^{(1)}\right| < \epsilon\right) \& \left(\left|\text{MSE}^{(2)} - \widehat{\text{MSE}}_{\text{loo}}^{(2)}\right| < \epsilon\right)\right\} > (1 - 2\delta)$$

Alternative versions of racing

- Hoeffding's bounds do not make any assumption about the distribution of errors but typically their bounds are not very high.
- If we assume that errors are normally distributed, we can use parametric techniques to achieve tighter bounds.
- As an alternative, given the set of squared errors of two models

$$\{s_1^{(1)}, \ldots, s_{N_{\text{ts}}}^{(1)}\}, \quad \{s_1^{(2)}, \ldots, s_{N_{\text{ts}}}^{(2)}\}$$

  nonparametric resampling tests can also be used to test if $\text{MSE}^{(1)} = E[\mathbf{s}^{(1)}] < E[\mathbf{s}^{(2)}] = \text{MSE}^{(2)}$.
- Paired tests can help speeding up the racing when models have a large variance in error over the test set.

Alternative versions of racing

- Multiple tests can improve the confidence for large m.
- Suppose that H_1 : MSE$^{(1)} <$ MSE$^{(2)}$ and H_2 : MSE$^{(2)} <$ MSE$^{(3)}$. We have

$$\text{Prob}\,\{t(\mathbf{D}_N)|H_1 \quad \& \quad t(\mathbf{D}_N)|H_2\} \neq \text{Prob}\,\{t(\mathbf{D}_N)|H_1 \& H_2\}$$