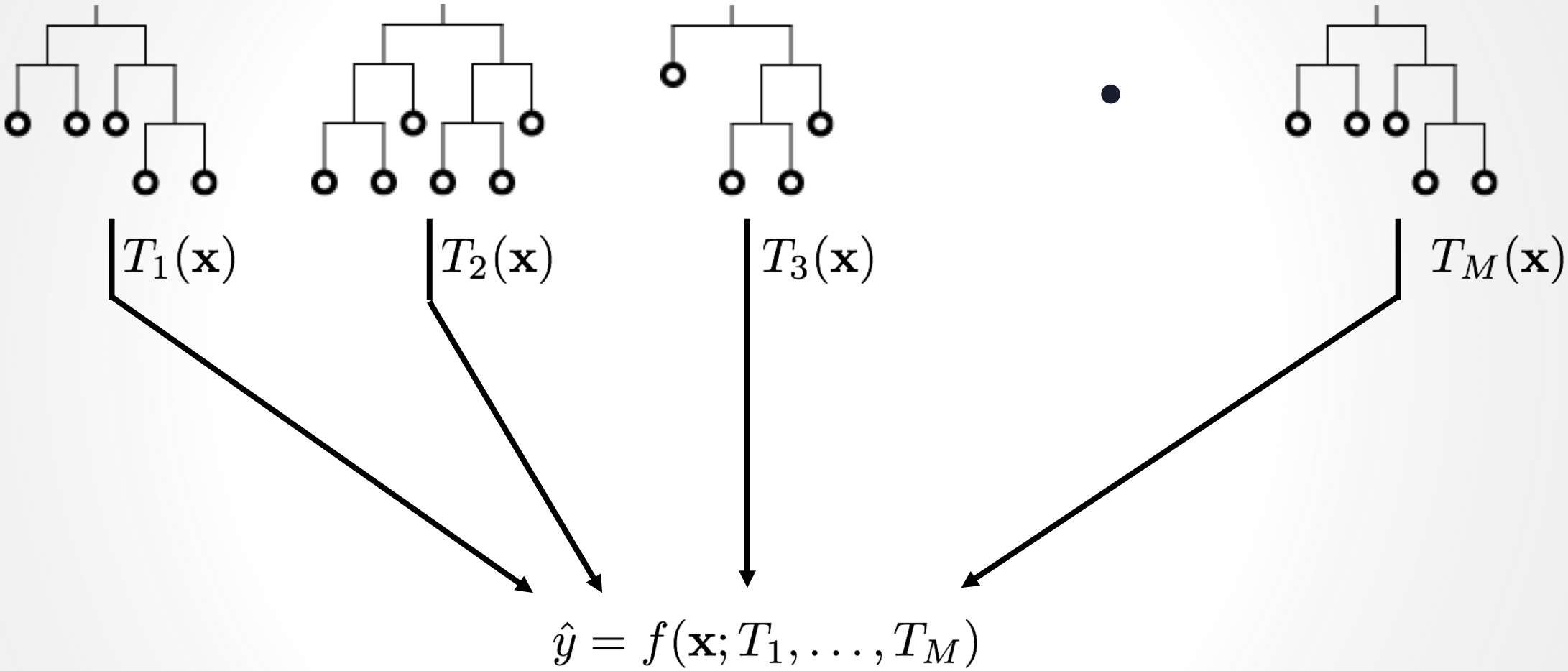
Supervised Learning:

TREE ENSEMBLES

Ensemble of Trees

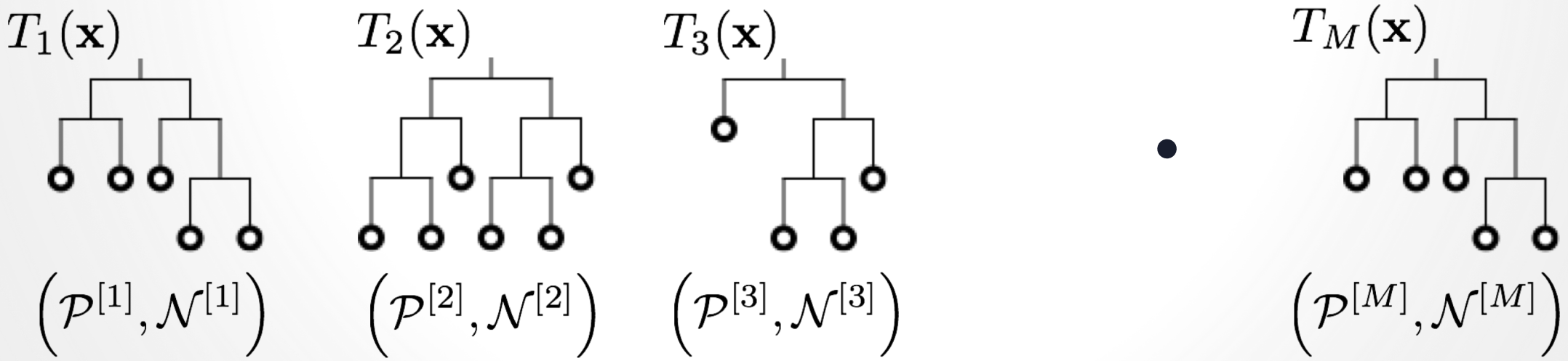


Regularization: Observation and Feature Sampling

- For each tree, select from *P* features (columns) and *N* rows

$$\mathcal{P} \in \{1, 2, \ldots, P\} \qquad \mathcal{N} \in \{1, 2, \ldots, N\}$$

$$\mathcal{P}^{[i]} \subseteq \mathcal{P},\ |\mathcal{P}^{[i]}| = P^{[i]} = \gamma_{\text{COL}} P \qquad \mathcal{N}^{[i]} \subseteq \mathcal{N},\ |\mathcal{N}^{[i]}| = N^{[i]} = \gamma_{\text{ROW}} N$$

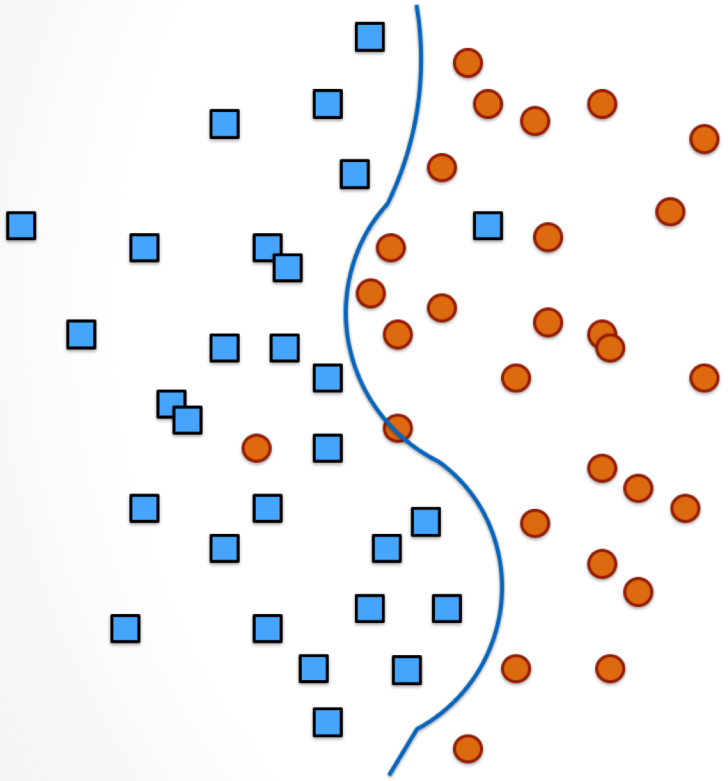


Scoring and Stopping

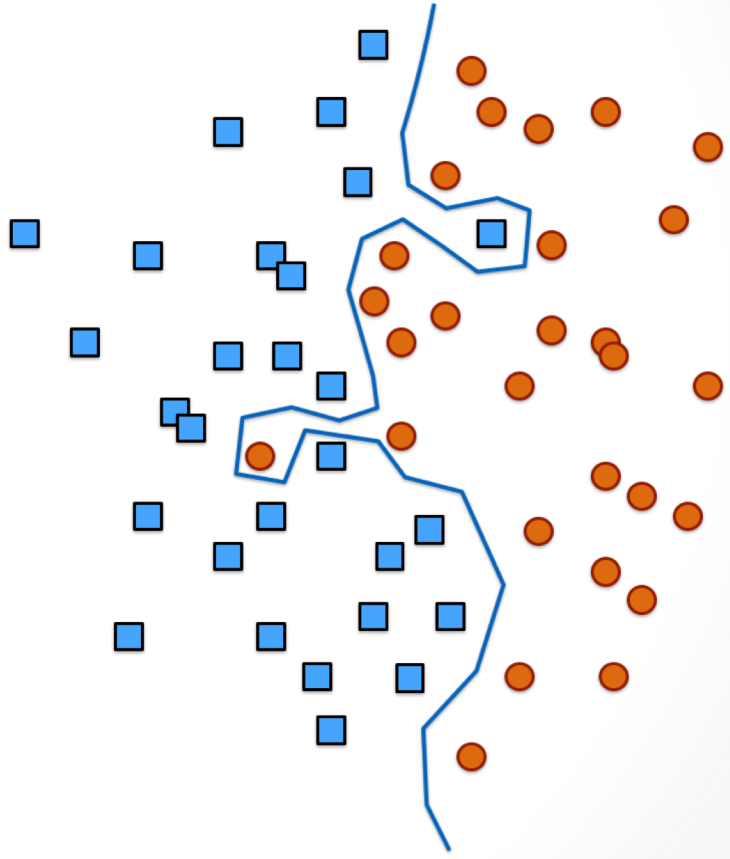
- How often should we check our validation error? (Computation time versus generalization)
  - **`score_each_iteration`**: score model after each tree
  - **`score_tree_interval`**: score model after n trees

- Setting criteria for stopping
  - **`stopping_rounds`**: early stop if stopping metric's moving average does not improve for this many rounds
  - **`stopping_metric`**: metric for early stopping
  - **`stopping_tolerance`**: Relative tolerance for metric-based stopping criterion (stop if relative improvement is not at least this much)
  - **`max_runtime_secs:`** maximum runtime to allow for model building

Finding the Signal in the Data

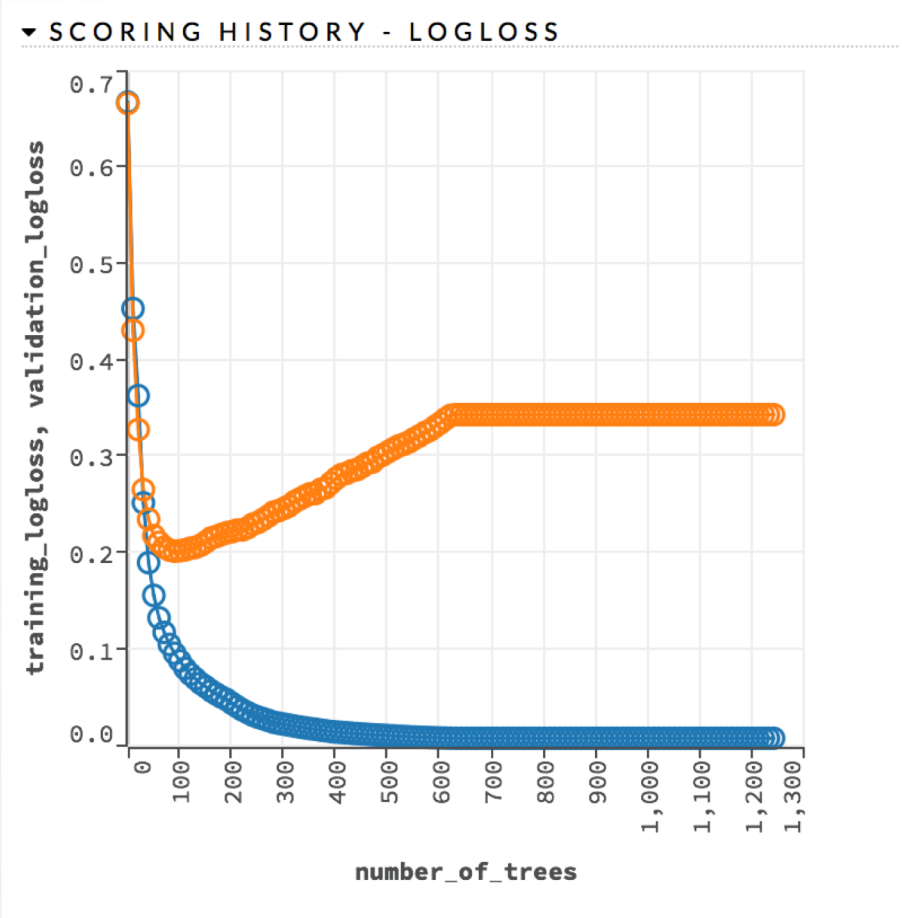


Memorizing the Data



Early Stopping

Overfitting



Early Stopping

Cross-Validation

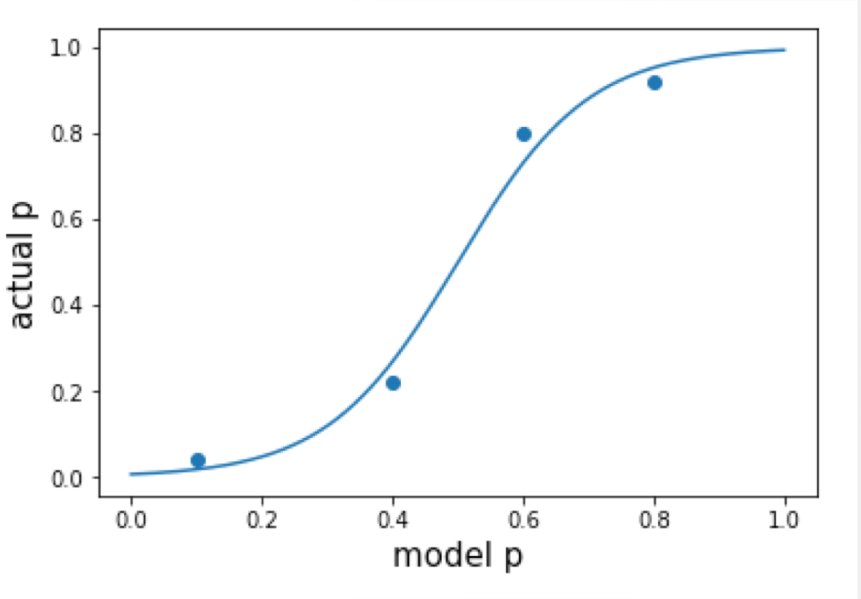
- **`nfolds`**: number of folds for N-fold cross-validation (default: 0, disabled)
- **`keep_cross_validation_predictions`**: keep the predictions of the cross-validation models
- **`keep_cross_validation_fold_assignment`**: keep the cross-validation fold assignment **`fold_assignment`**: cross-validation fold assignment scheme, if fold column is not specified. The ”Stratified” option will stratify the folds based on the response variable, for classification problems. Must be one of: ”AUTO”, ”Random”, ”Modulo”, ”Stratified”.
- **`fold_column`**: column with cross-validation fold index assignment per observation.

Platt Scaling

- Many ML algorithms introduce biases when it comes to class probability

SVMs, GBMs { underpredict high-prob classes; overpredict low-prob classes

Naïve Bayes { overpredict high-prob classes; underpredict low-prob classes



- Correct for this by fitting a sigmoid to the model output:

$$P(\mathbf{x}) = \frac{1}{1 + \exp(Af(\mathbf{x}) + B)}$$

- Split data into two frames: 1) Training dataframe: used to generate f(x)
  2) Platt calibration frame: used to fit A and B