Machine Learning with H_2O

H_2O.ai

H2O Algorithms
Supervised

H2O.ai

H2O Algorithms

Supervised Learning

Statistical Analysis

- **Generalized Linear Models**: Binomial, Gaussian, Gamma, Poisson, and Tweedie
- **Naive Bayes**: Binary Text Classification

Ensembles

- **Distributed Random Forest**: Classification or Regression Models
- **Gradient Boosting Machine**: Ensembles of shallow decision trees with increasing refined approximations

Deep Neural Networks

- **Deep Learning**: Create multi-layer feed forward neural networks starting with an input layer followed by multiple layers of nonlinear transformations

H2O Algorithms

Unsupervised Learning

Clustering

- **K-means**: Partition observations into k clusters of the same spatial size. Categorical features are one hot encoded.
- **Archetypes [GLRM]**: Partition observations into k archetypes.

Dimensionality Reduction

- **Principal Component Analysis**: Linearly transforms correlated variables to independent components
- **Generalized Low Rank Model:** Approximates data set as a product of two low dimensional factors. Extends PCA to handle sparse data, categorical data, and adds regularization.

Anomaly Detection

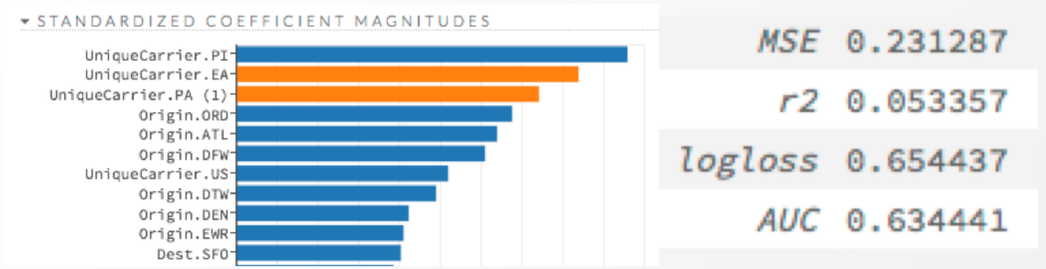
- **Autoencoders [Deep Learning]**: Create multi-layer feed forward neural networks starting with an input layer followed by multiple layers of nonlinear transformations

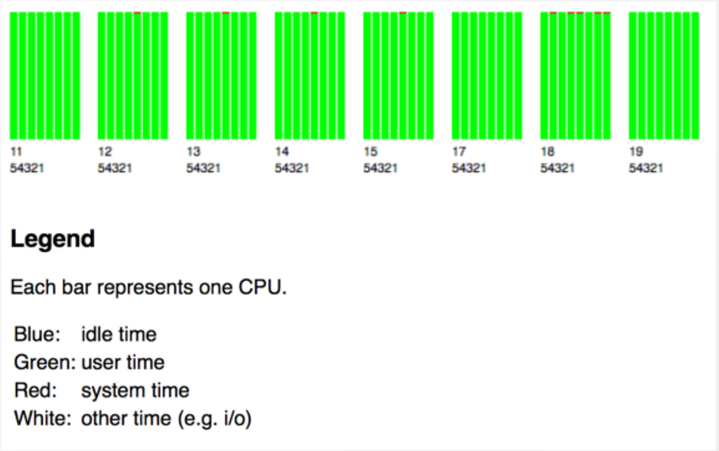
LINEAR MODELS

H2O GLM

- Full distributed & parallel
  - Scales ***linearly*** with number of rows!
- All standard GLM features
  - Standard distributions
  - Observation weights & offset
- Solvers
  - Iteratively reweighted least squares
  - L-BFGS for wide data sets
  - Coordinate Descent (exp)

Logistic Regression on 116milllion rows in~20 seconds
elastic net, alpha=0.5, lambda=1.379e-4 (auto)





H2O.ai

Generalized Linear Models in H2O

```
C$buildModel "glm"
```

16ms

Build a Model

Select an algorithm: Generalized Linear Modeling

PARAMETERS — *GRID?*

model_id	glm-58158ed9-11a1-4bee-8a5b-baddc1b8d6d6	Destination id for this model; auto-generated if not specified.
training_frame	(Choose...)	Id of the training data frame (Not required, to allow initial validation of model parameters).
validation_frame	(Choose...)	Id of the validation data frame.
nfolds	0	Number of folds for N-fold cross-validation (0 to disable or >= 2).
seed	-1	Seed for pseudo random number generator (if applicable)
response_column	(Choose...)	Response variable column.
ignored_columns	Search...	

All None Previous 100 Next 100

Only show columns with more than 0 % missing values.

ignore_const_cols		Ignore constant columns.
family	gaussian	Family. Use binomial for classification with logistic regression, others are for regression problems.
solver	AUTO	AUTO will set the solver based on given data and the other parameters. IRLSM is fast on on problems with small number of predictors and for lambda-search with L1 penalty, L_BFGS scales better for datasets with many columns. Coordinate descent is experimental (beta).
alpha		Distribution of regularization between the L1 (Lasso) and L2 (Ridge) penalties. A value of 1 for alpha represents Lasso regression, a value of 0 produces Ridge regression, and anything in between specifies the amount of mixing between the two. Default value of alpha is 0 when SOLVER = 'L-BFGS'; 0.5 otherwise.
lambda		Regularization strength
lambda_search		Use lambda search starting at lambda max, given lambda is then interpreted as lambda min
standardize		Standardize numeric columns to have zero mean and unit variance
non_negative		Restrict coefficients (not intercept) to be non-negative
beta_constraints	(Choose...)	Beta constraints

ADVANCED — *GRID?*

fold_assignment	AUTO	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.
fold_column	(Choose...)	Column with cross-validation fold index assignment per observation.
score_each_iteration		Whether to score during each iteration of model training.
offset_column	(Choose...)	Offset column. This will be added to the combination of columns before applying the link function.
weights_column	(Choose...)	Column with observation weights. Giving some observation a weight of zero is equivalent to excluding it from the dataset; giving an observation a relative weight of 2 is equivalent to repeating that row twice. Negative weights are not allowed.
compute_p_values		Request p-values computation, p-values work only with IRLSM solver and no regularization
remove_collinear_columns		In case of linearly dependent columns, remove some of the dependent columns
max_iterations	-1	Maximum number of iterations
link	family_default	
max_confusion_matrix_size	20	[Deprecated] Maximum size (# classes) for confusion matrices to be printed in the Logs
max_hit_ratio_k	0	Maximum number (top K) of predictions to use for hit ratio computation (for multi-class only, 0 to disable)
max_runtime_secs	0	Maximum allowed runtime in seconds for model training. Use 0 to disable.

EXPERT — *GRID?*

keep_cross_validation_predictions		Whether to keep the predictions of the cross-validation models.
keep_cross_validation_fold_assignment		Whether to keep the cross-validation fold assignment.
missing_values_handling	MeanImputation	Handling of missing values. Either MeanImputation or Skip.
intercept		Include constant term in the model
objective_epsilon	-1	Converge if objective value changes less than this. Default indicates: If lambda_search is set to True the value of objective_epsilon is set to .0001. If the lambda_search is set to False and lambda is equal to zero, the value of objective_epsilon is set to .000001, for any other value of lambda the default value of objective_epsilon is set to .0001.
beta_epsilon	0.0001	Converge if beta changes less (using L-infinity norm) than beta esilon, ONLY applies to IRLSM solver
gradient_epsilon	-1	Converge if objective changes less (using L-infinity norm) than this, ONLY applies to L-BFGS solver. Default indicates: If lambda_search is set to False and lambda is equal to zero, the default value of gradient_epsilon is equal to .000001, otherwise the default value is .0001. If lambda_search is set to True, the conditional values above are 1E-8 and 1E-6 respectively.
prior	-1	Prior probability for y==1. To be used only for logistic regression iff the data has been sampled and the mean of response does not reflect reality.
max_active_predictors	-1	Maximum number of active predictors during computation. Use as a stopping criterion to prevent expensive model building with many predictors. Default indicates: If the IRLSM solver is used, the value of max_active_predictors is set to 7000 otherwise it is set to 100000000.
interactions	Search...	

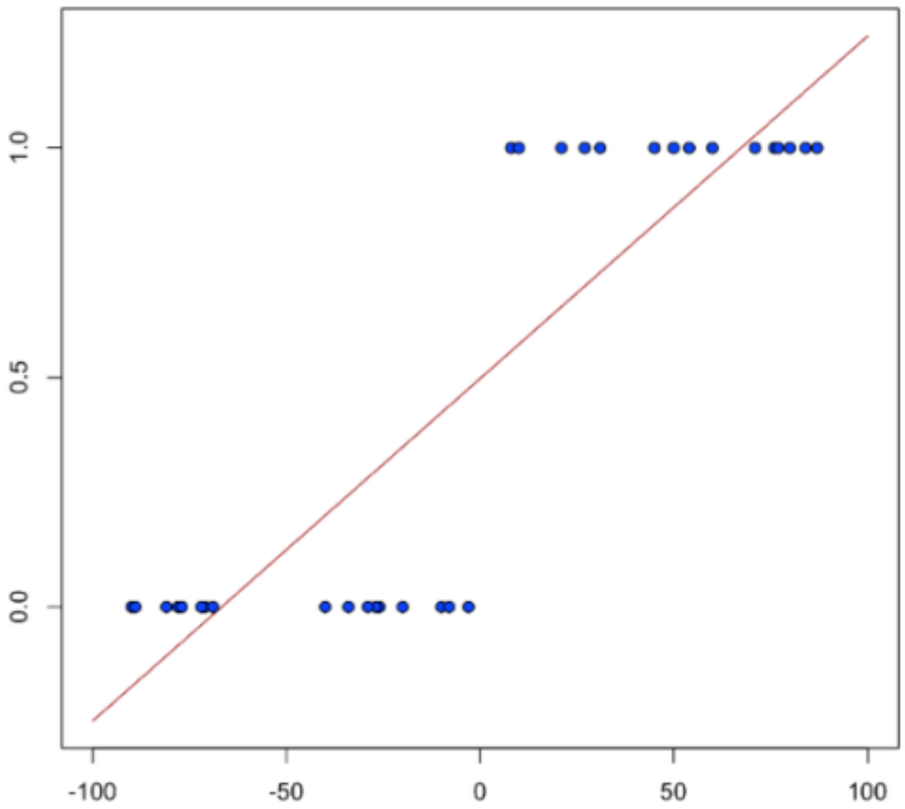
All None Previous 100 Next 100

Only show columns with more than 0 % missing values.

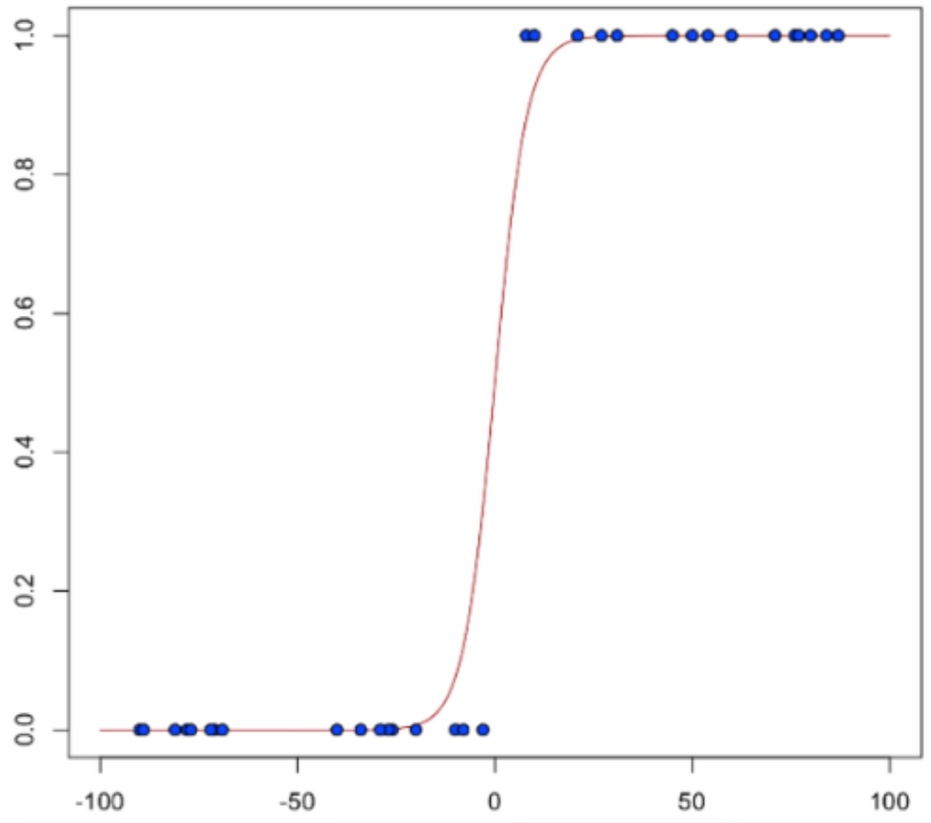
Build Model

Same predictors, different family and link functions

Linear Regression fit
(family=gaussian,link =identity)



Logistic Regression fit
(family=binomial,link = logit)



Pros and Cons of GLM's

Pros

- Fast & easy to fit
- Easy to interpret
- Well understood statistical properties
- Continuous, categorical, count and classification problems
- Transparent scoring functions

Cons

- Possibility of overfitting
- Difficult to account for non-linearities
- Issues of multicollinearity
  - Non-unique solutions
- Difficult to fully account for interaction effects
- Needs good features!
- Not as powerful OOB as RF/GBM/Deep Learning

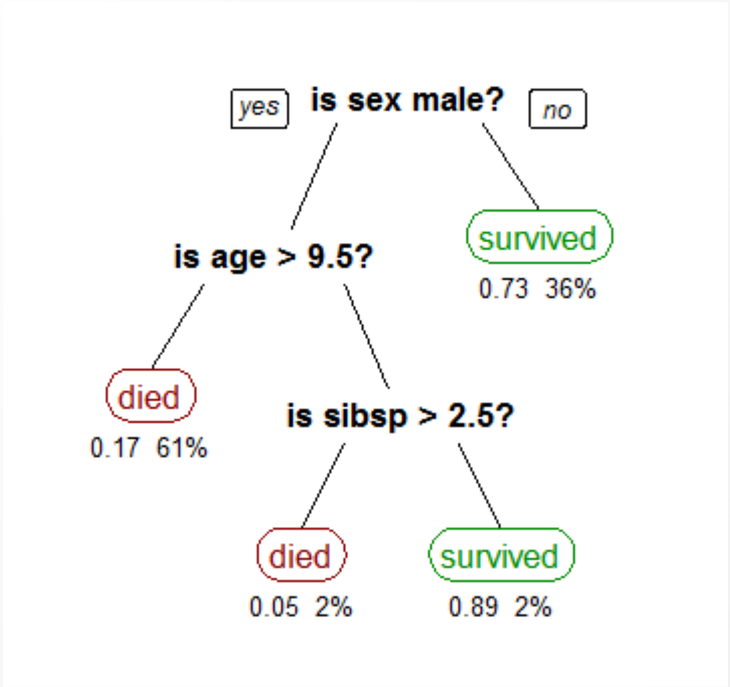
Best Practice Summary - GLM

- Which regularization to use?
  - Try both! Dense & Sparse solutions (L_1 & no L_1) ($\alpha = 0 \mid 1$)
- Should I grid search alpha exhaustively?
  - You can – but likely to get away [0, 0.5, 0.99]
  - Lambda search is a "built-in" grid search
- Wide datasets
  - If you have thousands of predictors – try L-BFGS solver
- Solvers
  - IRLSM (auto) + Lambda search works & is recommended ($\alpha > 0$)
  - L-BFGS + L_2 works, L_1 is slow!

DECISION TREES

Basics of Decision Trees

- Break sample data into homogenous pieces according to questions
  - Binary! Split into two at each point
- Can do both classification & regression
- Split points decided by min MSE



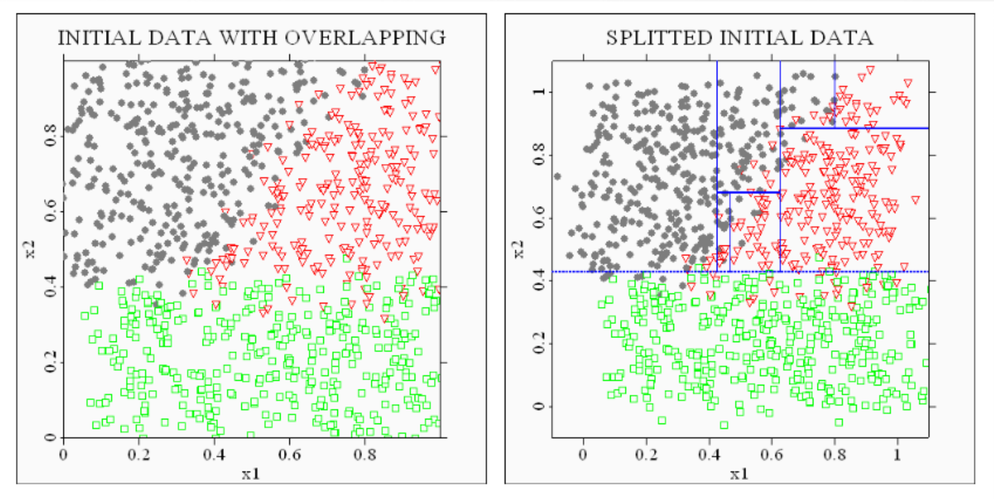
Wikipedia: Titanic Survivors

Pros and Cons of Decision Trees

Pros

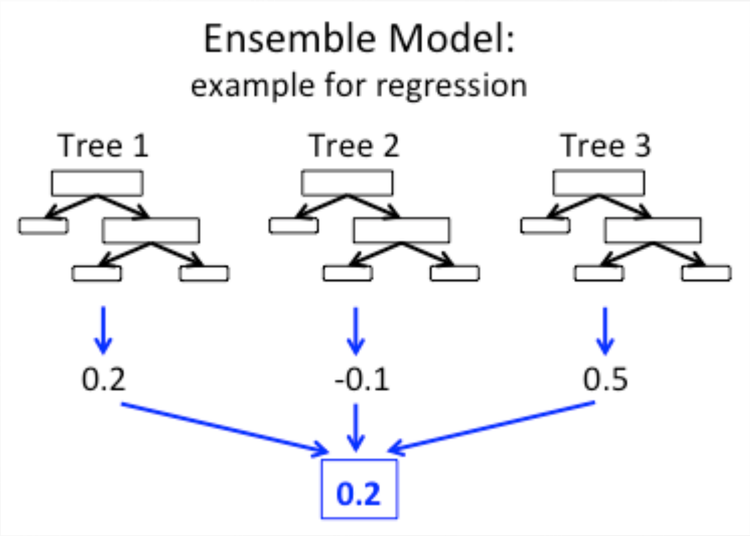
- Handles non-linearities in data
- Easy to compute
- Easy to interpret
- Robust to correlated features
- Robust to missing values
- Easy scoring functions

Cons

- Overfit data
- Will eliminate collinear variables
- 

What are Random Forests?

- Average many decision trees
  - Different trees on different samples and average
    - Random sample rows & columns
    - Reduce variance with minimal bias increase
  - "Bagging"



Distributed Random Forest in H2O

```
buildModel "drf"
```

9ms

Build a Model

Select an algorithm: Distributed Random Forest

PARAMETERS — GRID?

model_id	drf-fc859847-49cc-4907-9567-192156b81b06	Destination id for this model; auto-generated if not specified.
training_frame	(Choose...)	Id of the training data frame (Not required, to allow initial validation of model parameters).
validation_frame	(Choose...)	Id of the validation data frame.
nfolds	0	Number of folds for N-fold cross-validation (0 to disable or >= 2).
response_column	(Choose...)	Response variable column.
ignored_columns	Search...	
	All None	Previous 100 Next 100
	Only show columns with more than 0 % missing values.	
ignore_const_cols		Ignore constant columns.
ntrees	50	Number of trees.
max_depth	20	Maximum tree depth.
min_rows	1	Fewest allowed (weighted) observations in a leaf.
nbins	20	For numerical columns (real/int), build a histogram of (at least) this many bins, then split at the best point
seed	-1	Seed for pseudo random number generator (if applicable)
mtries	-1	Number of variables randomly sampled as candidates at each split. If set to -1, defaults to sqrt{p} for classification and p/3 for regression (where p is the # of predictors
sample_rate	0.6320000290870667	Row sample rate per tree (from 0.0 to 1.0)

ADVANCED — GRID?

score_each_iteration		Whether to score during each iteration of model training.
score_tree_interval	0	Score the model after every so many trees. Disabled if set to 0.
fold_assignment	AUTO	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.
fold_column	(Choose...)	Column with cross-validation fold index assignment per observation.
offset_column	(Choose...)	Offset column. This will be added to the combination of columns before applying the link function.
weights_column	(Choose...)	Column with observation weights. Giving some observation a weight of zero is equivalent to excluding it from the dataset; giving an observation a relative weight of 2 is equivalent to repeating that row twice. Negative weights are not allowed.
balance_classes		Balance training data class counts via over/under-sampling (for imbalanced data).
max_confusion_matrix_size	20	[Deprecated] Maximum size (# classes) for confusion matrices to be printed in the Logs
max_hit_ratio_k	0	Max. number (top K) of predictions to use for hit ratio computation (for multi-class only, 0 to disable)
nbins_top_level	1024	For numerical columns (real/int), build a histogram of (at most) this many bins at the root level, then decrease by factor of two per level
nbins_cats	1024	For categorical columns (factors), build a histogram of this many bins, then split at the best point. Higher values can lead to more overfitting.
r2_stopping	1.7976931348623157e+308	r2_stopping is no longer supported and will be ignored if set - please use stopping_rounds, stopping_metric and stopping_tolerance instead. Previous version of H2O would stop making trees when the R^2 metric equals or exceeds this
stopping_rounds	0	Early stopping based on convergence of stopping_metric. Stop if simple moving average of length k of the stopping_metric does not improve for k:=stopping_rounds scoring events (0 to disable)
stopping_metric	AUTO	Metric to use for early stopping (AUTO: logloss for classification, deviance for regression)
stopping_tolerance	0.001	Relative tolerance for metric-based stopping criterion (stop if relative improvement is not at least this much)
max_runtime_secs	0	Maximum allowed runtime in seconds for model training. Use 0 to disable.
checkpoint		Model checkpoint to resume training with.
col_sample_rate_per_tree	1	Column sample rate per tree (from 0.0 to 1.0)
min_split_improvement	0.00001	Minimum relative improvement in squared error reduction for a split to happen
histogram_type	AUTO	What type of histogram to use for finding optimal split points
categorical_encoding	AUTO	Encoding scheme for categorical features

EXPERT — GRID?

keep_cross_validation_predictions		Whether to keep the predictions of the cross-validation models.
keep_cross_validation_fold_assignment		Whether to keep the cross-validation fold assignment.
class_sampling_factors		Desired over/under-sampling ratios per class (in lexicographic order). If not specified, sampling factors will be automatically computed to obtain class balance during training. Requires balance_classes.
max_after_balance_size	5	Maximum relative size of the training data after balancing class counts (can be less than 1.0). Requires balance_classes.
build_tree_one_node		Run on one node only; no network overhead but fewer cpus used. Suitable for small datasets.
sample_rate_per_class		A list of row sample rates per class (relative fraction for each class, from 0.0 to 1.0), for each tree
binomial_double_trees		For binary classification: Build 2x as many trees (one per class) - can lead to higher accuracy.
col_sample_rate_change_per_level	1	Relative change of the column sampling rate for every level (from 0.0 to 2.0)
calibrate_model		Use Platt Scaling to calculate calibrated class probabilities. Calibration can provide more accurate estimates of class probabilities.
calibration_frame	(Choose...)	Calibration frame for Platt Scaling

Build Model

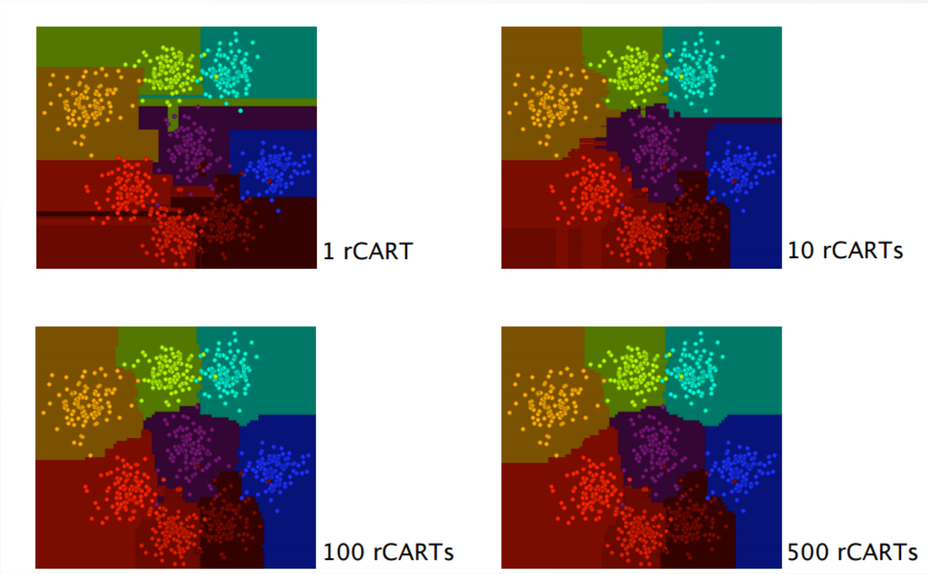
Strengths and Weaknesses of Random Forests

Strengths

- Easy to compute
  - Easy to parallelize
- Robust to outliers
- Variable importance
- Competitive accuracy on most data sets

Weaknesses

- Slow to score
- Not transparent



What is GBM?

- An ensemble of weak learners.
- The decision trees in GBM are built consecutively, with each new layer solving for the net loss of prior trees.
- This means the algorithm is inherits the pros of building decision trees while also making up for poor accuracy by creating an ensemble of trees that learn from prior trees.

Gradient Boosting Machine in H2O

```
buildModel "gbm"
```

12ms

Build a Model

Select an algorithm: Gradient Boosting Machine

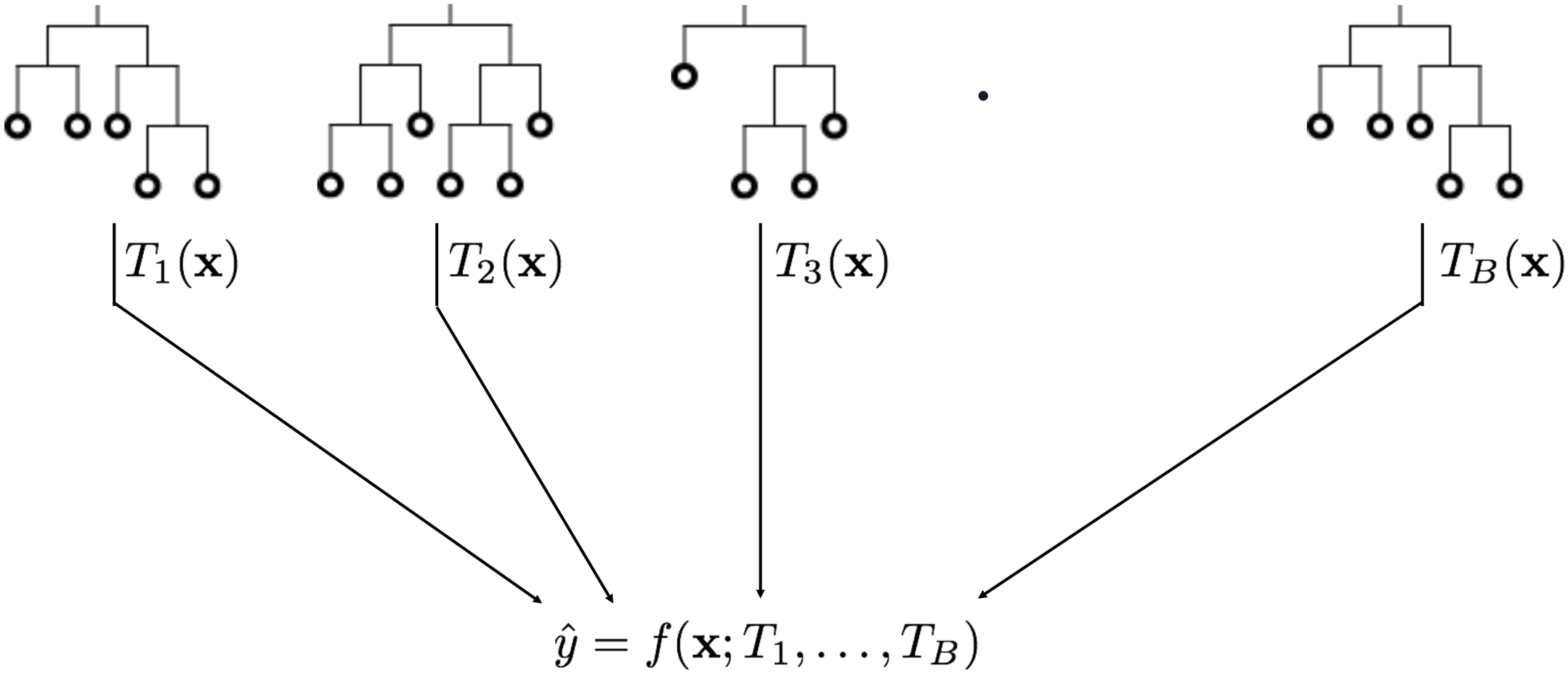
PARAMETERS		GRID?
model_id	gbm-d4cdac10-a7a6-4886-bc1c-271e31a56a7b	Destination id for this model; auto-generated if not specified.
training_frame	(Choose...)	Id of the training data frame (Not required, to allow initial validation of model parameters).
validation_frame	(Choose...)	Id of the validation data frame.
nfolds	0	Number of folds for N-fold cross-validation (0 to disable or >= 2).
response_column	(Choose...)	Response variable column.
ignored_columns	Search...	
	All None	Previous 100 Next 100
	Only show columns with more than 0 % missing values.	
ignore_const_cols		Ignore constant columns.
ntrees	50	Number of trees.
max_depth	5	Maximum tree depth.
min_rows	10	Fewest allowed (weighted) observations in a leaf.
nbins	20	For numerical columns (real/int), build a histogram of (at least) this many bins, then split at the best point
seed	-1	Seed for pseudo random number generator (if applicable)
learn_rate	0.1	Learning rate (from 0.0 to 1.0)
sample_rate	1	Row sample rate per tree (from 0.0 to 1.0)
col_sample_rate	1	Column sample rate (from 0.0 to 1.0)

ADVANCED		GRID?
score_each_iteration		Whether to score during each iteration of model training.
score_tree_interval	0	Score the model after every so many trees. Disabled if set to 0.
fold_assignment	AUTO	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.
fold_column	(Choose...)	Column with cross-validation fold index assignment per observation.
offset_column	(Choose...)	Offset column. This will be added to the combination of columns before applying the link function.
weights_column	(Choose...)	Column with observation weights. Giving some observation a weight of zero is equivalent to excluding it from the dataset; giving an observation a relative weight of 2 is equivalent to repeating that row twice. Negative weights are not allowed.
balance_classes		Balance training data class counts via over/under-sampling (for imbalanced data).
max_confusion_matrix_size	20	[Deprecated] Maximum size (# classes) for confusion matrices to be printed in the Logs
max_hit_ratio_k	0	Max. number (top K) of predictions to use for hit ratio computation (for multi-class only, 0 to disable)
nbins_top_level	1024	For numerical columns (real/int), build a histogram of (at most) this many bins at the root level, then decrease by factor of two per level
nbins_cats	1024	For categorical columns (factors), build a histogram of this many bins, then split at the best point. Higher values can lead to more overfitting.
r2_stopping	1.7976931348623157e+308	r2_stopping is no longer supported and will be ignored if set - please use stopping_rounds, stopping_metric and stopping_tolerance instead. Previous version of H2O would stop making trees when the R^2 metric equals or exceeds this
stopping_rounds	0	Early stopping based on convergence of stopping_metric. Stop if simple moving average of length k of the stopping_metric does not improve for k:=stopping_rounds scoring events (0 to disable)
stopping_metric	AUTO	Metric to use for early stopping (AUTO: logloss for classification, deviance for regression)
stopping_tolerance	0.001	Relative tolerance for metric-based stopping criterion (stop if relative improvement is not at least this much)
max_runtime_secs	0	Maximum allowed runtime in seconds for model training. Use 0 to disable.
learn_rate_annealing	1	Scale the learning rate by this factor after each tree (e.g., 0.99 or 0.999)
distribution	AUTO	Distribution function
quantile_alpha	0.5	Desired quantile for Quantile regression, must be between 0 and 1.
tweedie_power	1.5	Tweedie power for Tweedie regression, must be between 1 and 2.
huber_alpha	0.9	Desired quantile for Huber/M-regression (threshold between quadratic and linear loss, must be between 0 and 1).
checkpoint		Model checkpoint to resume training with.
col_sample_rate_per_tree	1	Column sample rate per tree (from 0.0 to 1.0)
min_split_improvement	0.00001	Minimum relative improvement in squared error reduction for a split to happen
histogram_type	AUTO	What type of histogram to use for finding optimal split points
categorical_encoding	AUTO	Encoding scheme for categorical features

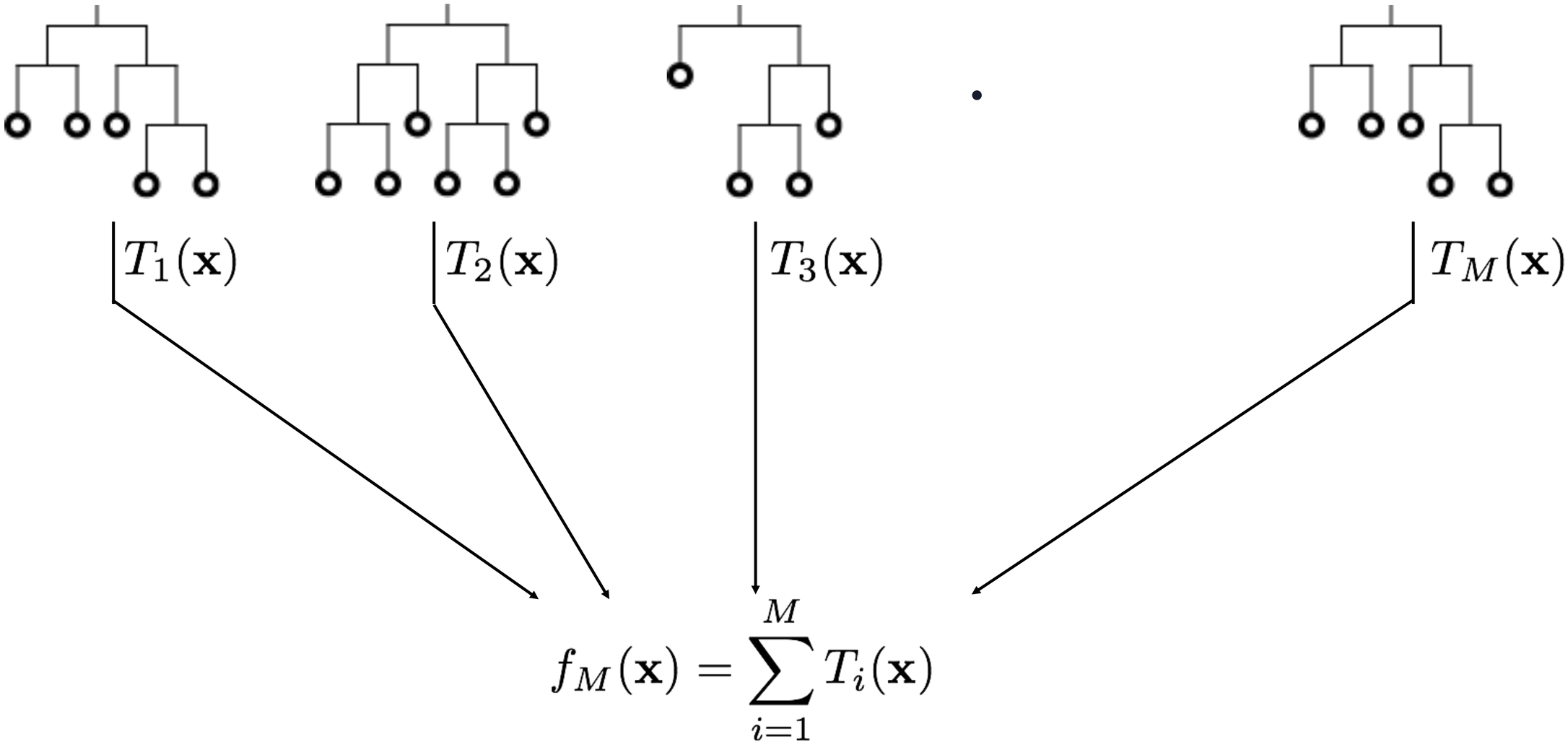
EXPERT		GRID?
keep_cross_validation_predictions		Whether to keep the predictions of the cross-validation models.
keep_cross_validation_fold_assignment		Whether to keep the cross-validation fold assignment.
class_sampling_factors		Desired over/under-sampling ratios per class (in lexicographic order). If not specified, sampling factors will be automatically computed to obtain class balance during training. Requires balance_classes.
max_after_balance_size	5	Maximum relative size of the training data after balancing class counts (can be less than 1.0). Requires balance_classes.
build_tree_one_node		Run on one node only; no network overhead but fewer cpus used. Suitable for small datasets.
sample_rate_per_class		A list of row sample rates per class (relative fraction for each class, from 0.0 to 1.0), for each tree
col_sample_rate_change_per_level	1	Relative change of the column sampling rate for every level (from 0.0 to 2.0)
max_abs_leafnode_pred	1.7976931348623157e+308	Maximum absolute value of a leaf node prediction
pred_noise_bandwidth	0	Bandwidth (sigma) of Gaussian multiplicative noise ~N(1,sigma) for tree node predictions
calibrate_model		Use Platt Scaling to calculate calibrated class probabilities. Calibration can provide more accurate estimates of class probabilities.
calibration_frame	(Choose...)	Calibration frame for Platt Scaling

Build Model

Ensemble of Trees



Gradient Boosting Machine (GBM)



GBM Functionalities in H2O

- Regression and Classification Models Exponential distributions including Poisson, Gamma, and Tweedie are available in addition to Bernoulli, multinomial, and gaussian distributions.
- Fast and CPU Efficient Parallel and distributed computation across multiple nodes and many cores.
- Grid Search Hyperparameter optimization allows the user to run through many parameters before selecting the best models.
- Early Stopping The user can specify the metric and the incremental change in the metric as convergence. If additional trees no longer provide an increase in AUC prune the tree early.
- Stochastic User can specify the sample rate that the algorithm will sample the column and row by for better generalization.
- Model Output The model is exportable as Java code and if you find the model overfitted after a certain number of trees, it is easy to reduce the number of trees in a POJO before putting it in production without rerunning model build.

When to use GBM?

- **Strengths**
  - Nonlinear models
  - Robust to correlated features
  - Robust to feature distributions
  - Robust to missing values
- **Weaknesses**
  - Tends to overfit on the training set
  - Sensitive to noise and outliers
  - More hyper parameters than Random Forest
  - Lack of transparency

Boosting

- Capacity
- Loss Function
- Regularization
- Scoring
- Early Stopping

Capacity

- `ntrees`: Number of trees. Defaults to 50.

Loss Function

- `distribution`: Distribution function Must be one of: AUTO, bernoulli, multinomial, gaussian, poisson, gamma, tweedie, laplace, quantile, huber. Defaults to AUTO.
  - `quantile_alpha`: Desired quantile for Quantile regression, must be between 0 and 1. Defaults to 0.5.
  - `tweedie_power`: Tweedie power for Tweedie regression, must be between 1 and 2. Defaults to 1.5.
  - `huber_alpha`: Desired quantile for Huber/M-regression (threshold between quadratic and lhnear loss, must be between 0 and 1). Defaults to 0.9.
  - `offset_column`: Offset column. This will be added to the combination of columns before applying the link function.

GBM – Learning Rate

- Learning Rate
  - In order for Gradient Descent to work we must set the λ (learning rate) to an appropriate value.
  - This parameter determines how fast or slow we will move towards the optimal weights.
  - If the λ is very large we will skip the optimal solution.
  - If it is too small we will need too many iterations to converge to the best values. So using a good λ is crucial.

GBM – Learning Rate Annealing

- Also called adaptive learning rates or learning rate schedule
- Adapting the learning rate for GBM optimization
  - Increase performance
  - Reduce training time.
- Use a constant learning rate to update network weights for each training epoch.

Regularization

- `learn_rate`: Learning rate (from 0.0 to 1.0) Defaults to 0.1.
- `learn_rate_annealing`: Scale the learning rate by this factor after each tree (e.g., 0.99 or 0.999) Defaults to 1.
- `sample_rate`: Row sample rate per tree (from 0.0 to 1.0) Defaults to 1.
- `sample_rate_per_class`: Row sample rate per tree per class (from 0.0 to 1.0)
- `col_sample_rate_per_tree`: Column sample rate per tree (from 0.0 to 1.0) Defaults to 1.

Scoring

- `score_each_iteration`: Logical. Whether to score during each iteration of model training. Defaults to FALSE.
- `score_tree_interval`: Score the model after every so many trees. Disabled if set to 0. Defaults to 0. Only comes into play is score each iteration is FALSE.

Early Stopping

- **`stopping_rounds`**: Early stopping based on convergence of stopping metric. Stop if simple moving average of length k of the stopping metric does not improve for k:=stopping rounds scoring events (0 to disable) Defaults to 0.
- **`stopping_metric`**: Metric to use for early stopping (AUTO: logloss for classification, deviance for regression) Must be one of: "AUTO", "deviance", "logloss", "MSE", "RMSE", "MAE", "RMSLE", "AUC", "lift_top_group", "misclassification", "mean_per_class_error". Defaults to AUTO.
- **`stopping_tolerance`**: Relative tolerance for metric-based stop- ping criterion (stop if relative improvement is not at least this much) Defaults to 0.001.

Tree Growth

- Size
- Splitting Criteria
- Leaf Node

Size

- `max_depth`: Maximum tree depth. Defaults to 5.

Splitting Criteria

- **`col_sample_rate`**: Column sample rate per split (from 0.0 to 1.0) Defaults to 1.
- **`col_sample_rate_change_per_level`**: Relative change of the column sampling rate for every level (from 0.0 to 2.0) Defaults to 1.
- **`nbins_cats`**: For categorical columns (factors), build a histogram of this many bins, then split at the best point. Higher values can lead to more overfitting. Defaults to 1024.
- **`nbins`**: For numerical columns (real/int), build a histogram of (at least) this many bins, then split at the best point Defaults to 20.
- **`nbins_top_level`**: For numerical columns (real/int), build a his- togram of (at most) this many bins at the root level, then decrease by factor of two per level Defaults to 1024.
- **`min_split_improvement`**: Minimum relative improvement in squared error reduction for a split to happen Defaults to 1e-05.
- **`histogram_type`**: What type of histogram to use for finding op- timal split points Must be one of: "AUTO", "UniformAdaptive", "Random", "QuantilesGlobal", "RoundRobin". Defaults to AUTO.

Leaf Node

- `min_rows`: Fewest allowed (weighted) observations in a leaf. De- faults to 10.

Others

- Cross Validation
- Predictions
- Data
- Performance

Cross Validation

- **`nfolds`**: Number of folds for N-fold cross-validation (0 to disable or != 2). Defaults to 0.
- **`keep_cross_validation_predictions`**: Logical. Whether to keep the predictions of the cross-validation models. Defaults to FALSE.
- **`keep_cross_validation_fold_assignment`**: Logical. Whether to keep the cross-validation fold assignment. Defaults to FALSE.
- **`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", "Strati- fied". Defaults to AUTO.
- **`fold_column`**: Column with cross-validation fold index assignment per observation.

Predictions

- `max_abs_leafnode_pred`: Maximum absolute value of a leaf node prediction Defaults to 1.797693135e+308
- `max_hit_ratio_k`: Max. number (top K) of predictions to use for hit ratio computation (for multiclass only, 0 to disable) Defaults to 0.

Data

- `categorical_encoding`: Encoding scheme for categorical features Must be one of: "AUTO", "Enum", "OneHotInternal", "OneHotExplicit", "Binary", "Eigen". Defaults to AUTO.
- `weights_column`: Column with observation weights. Giving some observation a weight of zero is equivalent to excluding it from the dataset; giving an observation a relative weight of 2 is equivalent to repeating that row twice. Negative weights are not allowed.
- `ignore_const_cols`: Logical. Ignore constant columns. Defaults to TRUE.

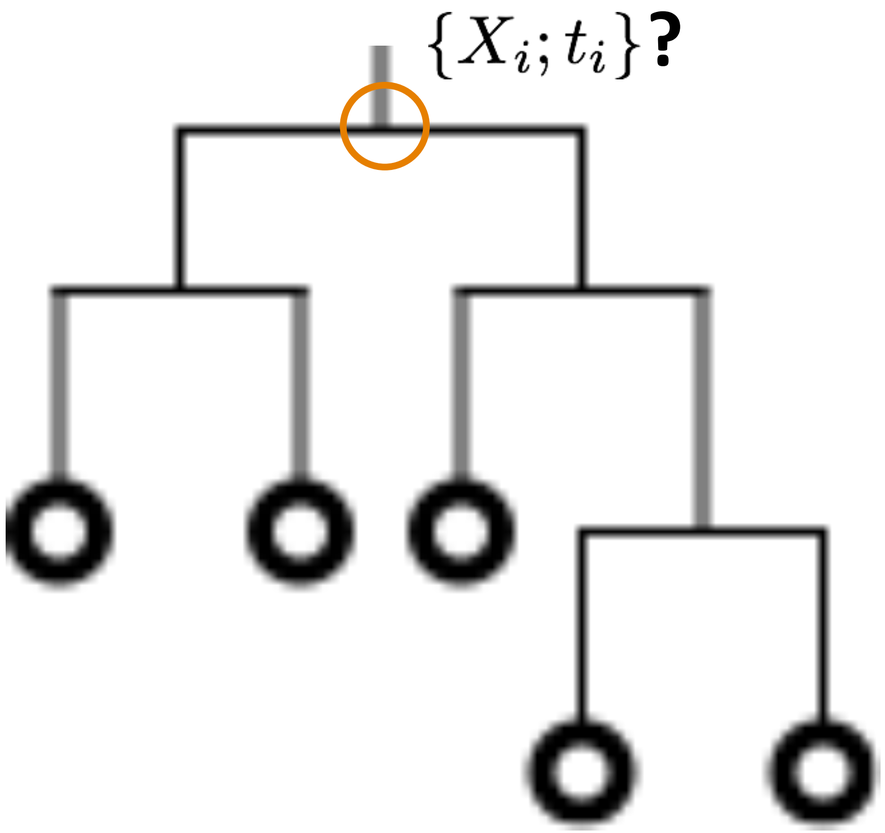
Class Imbalances

- `balance_classes`: Logical. Balance training data class counts via over/under-sampling (for imbalanced data). Defaults to FALSE.
- `class_sampling_factors`: Desired over/under-sampling ratios per class (in lexicographic order). If not specified, sampling fac- tors will be automatically
- `max_after_balance_size`: Maximum relative size of the training data after balancing class counts (can be less than 1.0). Requires balance classes. Defaults to 5.0.

Performance

- `max_runtime_secs`: Maximum allowed runtime in seconds for model training. Use 0 to disable. Defaults to 0.
- `build_tree_one_node`: Logical. Run on one node only; no net- work overhead but fewer cpus used. Suitable for small datasets. Defaults to FALSE.

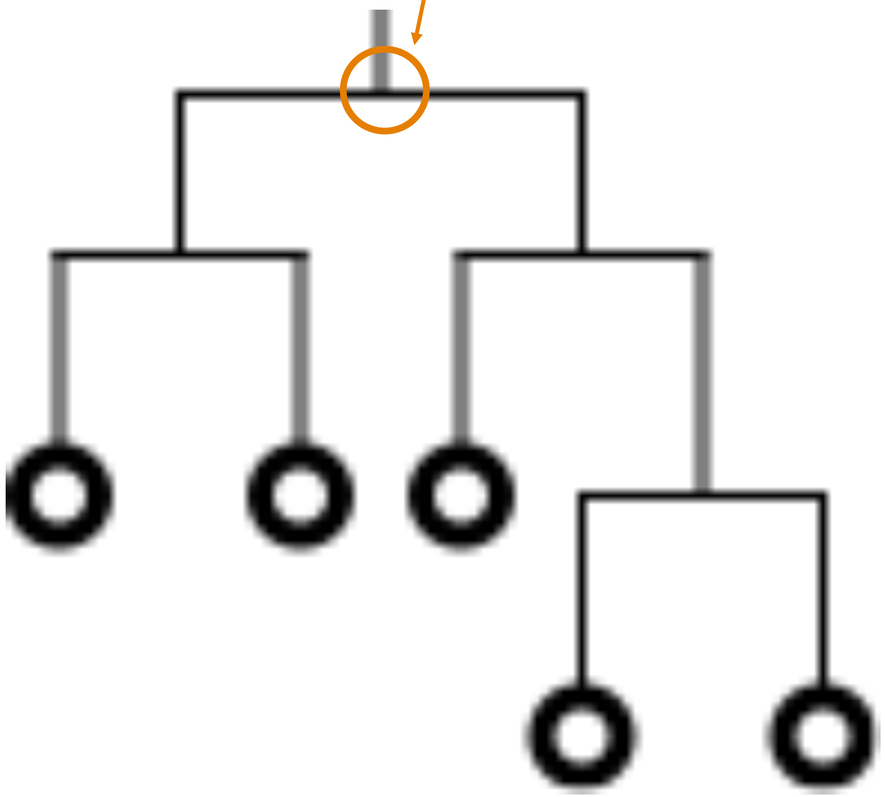
GBM Data Parallelism



$$\mathbf{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_K\}$$

GBM Data Parallelism

$$\{X_i; t_i\} = f(\mathbf{math}\,(\mathbf{X_1}), \ldots, \mathbf{math}\,(\mathbf{X_K}))$$

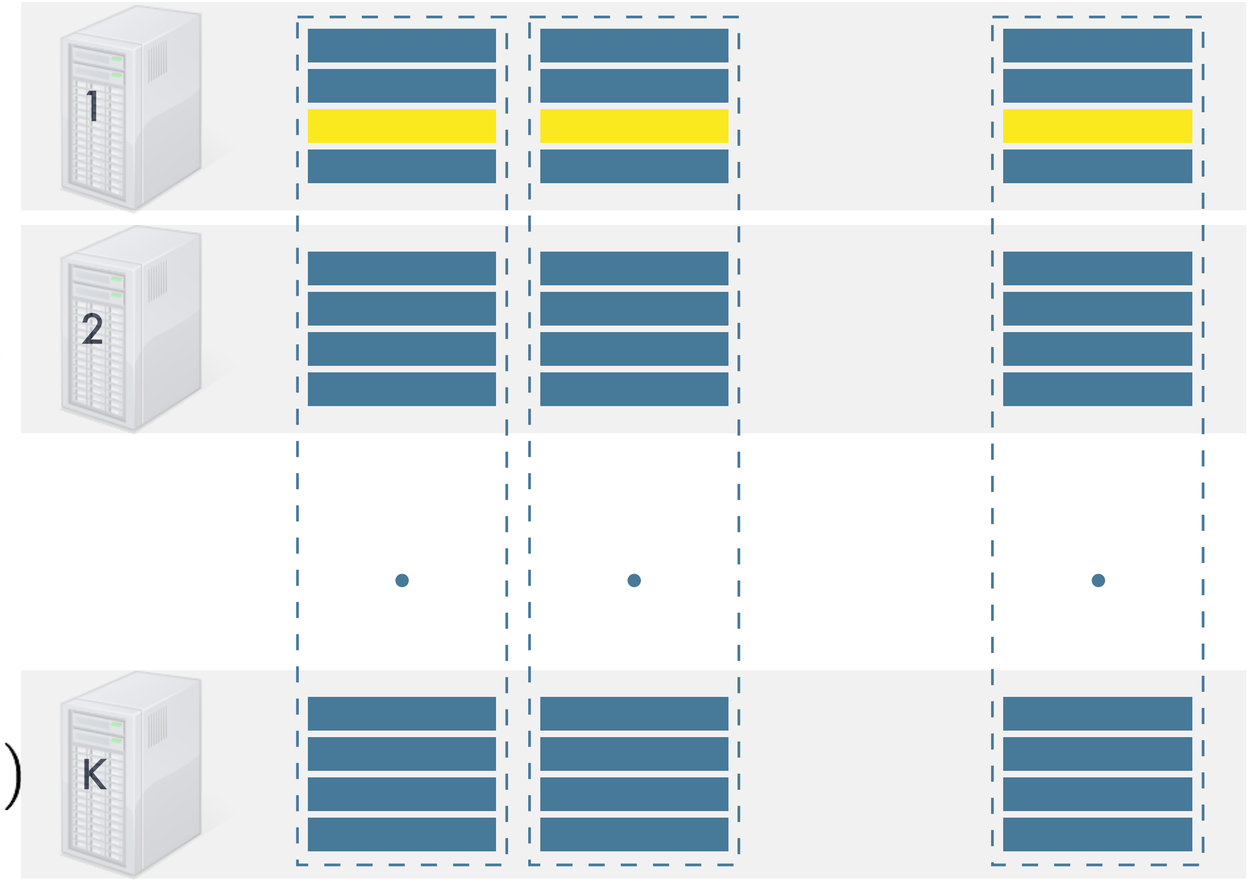


$$\mathbf{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_K\}$$

$\mathbf{math}\,(\mathbf{X_1})$

$\mathbf{math}\,(\mathbf{X_2})$

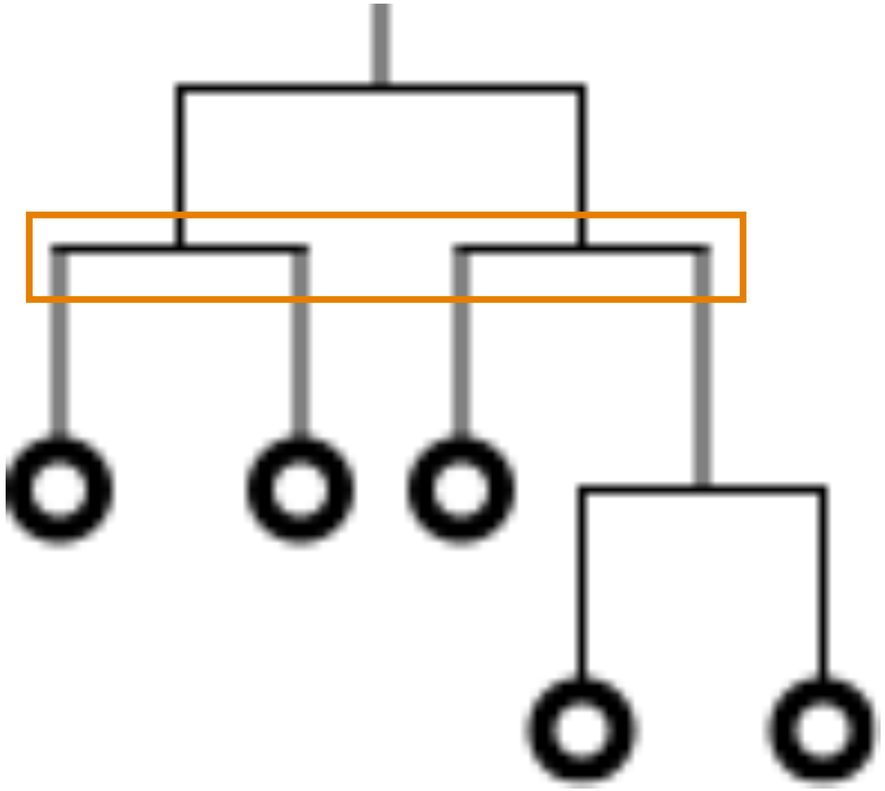
$\mathbf{math}\,(\mathbf{X_K})$



GBM Data Parallelism

$$\{X_i; t_i\} = f(\mathbf{math}\,(\mathbf{X_1}), \ldots, \mathbf{math}\,(\mathbf{X_K}))$$

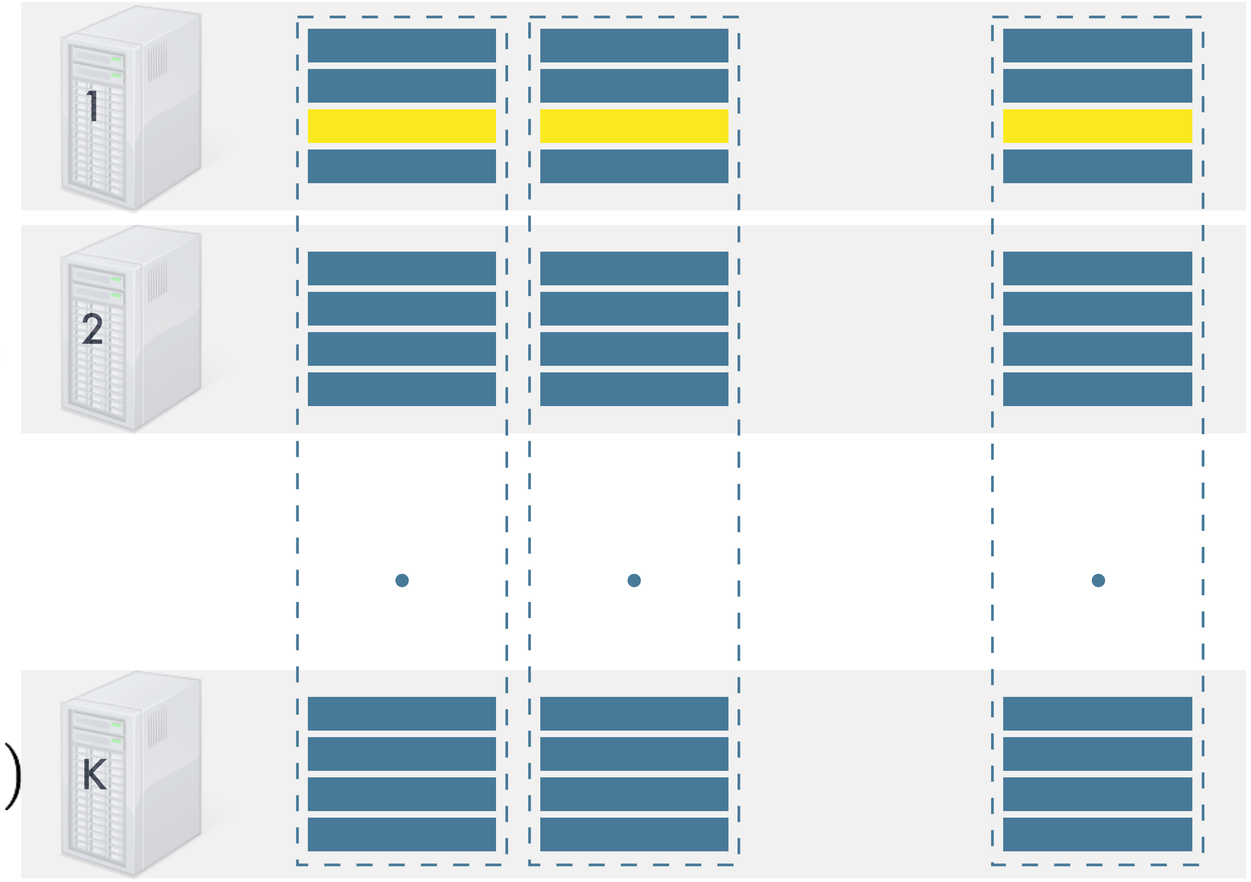
$$\mathbf{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_K\}$$



$\mathbf{math}\,(\mathbf{X_1})$

$\mathbf{math}\,(\mathbf{X_2})$

$\mathbf{math}\,(\mathbf{X_K})$



Full Data Parallelism for Each Level of Tree Growth!

avkash@h2o.ai