eCornell

CHEAT SHEET

Gradient Boosted Regression Trees (GBRT)

Algorithm Name	Gradient Boosted Regression Trees (GBRT)
Description	GBRT combines limited-depth CART classifiers into an ensemble. The classifiers are added sequentially to explicitly correct the error (residual) that the current ensemble is still making.
Applicability	Classification and regression problems.
Assumptions	Limited-depth regression trees perform better than random guessing (assuming local smoothness).
Underlying Mathematical Principles	Limited-depth trees have high bias. GBRT combines many high-bias classifiers to create a low-bias ensemble.
Hyperparameters	Number of trees to add, depth of the trees, and step size of the parameter (α).
Setting	Regression or classification ($y_i \in \{+1, -1\}$). While **regression** trees return continuous values, we can still use them to solve classification problems with discrete labels. For example, we can return the sign of the output of the tree.
Weak learners	$h \in \mathbb{H}$ are limited-depth regression trees.
Loss function	Gradient Boosting has two loss functions: 1. The global loss function — this is what you try to minimize with the ensemble (e.g., the squared loss, but it could also be a classification loss like the logistic loss or the exponential loss). The pseudocode implementation below assumes the squared loss: $\mathcal{L}(H) = \frac{1}{2}\sum_{i=1}^{n}\left(H\left(\mathbf{x}_i\right) - y_i\right)^2$ 2. The loss that the weak learners minimize — this loss is used to find the next weak learner h that is best aligned with the negative gradient. In GBRT, the weak loss is always the squared loss, as regression trees are the weak learners being used to fit the negative gradient t_i for each input $\mathbf{x}_i$. $\text{argmin}_h = \frac{1}{2}\sum_{i=1}^{n}(h(\mathbf{x}_i) - t_i)^2$

Residual	If the global loss $\mathcal{L}$ is the squared loss, then the negative gradient becomes the residual of the ensemble: $t_i = y_i - H(\mathbf{x}_i)$ The residual is the remaining difference between the actual label and the prediction of the current ensemble.
Ensemble	Combination of many classifiers (in this case, limited-depth regression trees): $H(\mathbf{x}) = \sum_{t=1}^{T} \alpha h_t(\mathbf{x})$

Pseudocode	GBRT$((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$, D, T, $\alpha)$
	H $\leftarrow 0$;
	for *t=1:T* **do**
	  for *i=1:n* **do**
	  $t_i \leftarrow y_i - H(\mathbf{x}_i)$;
	  end
	  $h_t \leftarrow CART((\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_n, t_n), maxdepth = D)$;
	  H $\leftarrow H + \alpha h_t$;
	end
	return H

Strengths	GBRT is particularly well suited for applications with heterogeneous features and fast testing times (as CART trees are blazingly fast to evaluate). It is a common choice for search engines.
Weakness	In contrast to Random Forests, GBRT is more prone to overfitting and the number of trees should be set carefully (ideally through early stopping on a holdout data set). The algorithm tends to excel in lower-dimensional feature spaces (<1000 dimensions) and is often not well suited for very high and sparse feature spaces (e.g., bag-of-words text data).
XGBoost	XGBoost is a popular implementation of GBRT and Random Forests. It is highly optimized and one of the most robust and widely used machine learning algorithms in practice.