Bagging

Bias/Variance Tradeoff

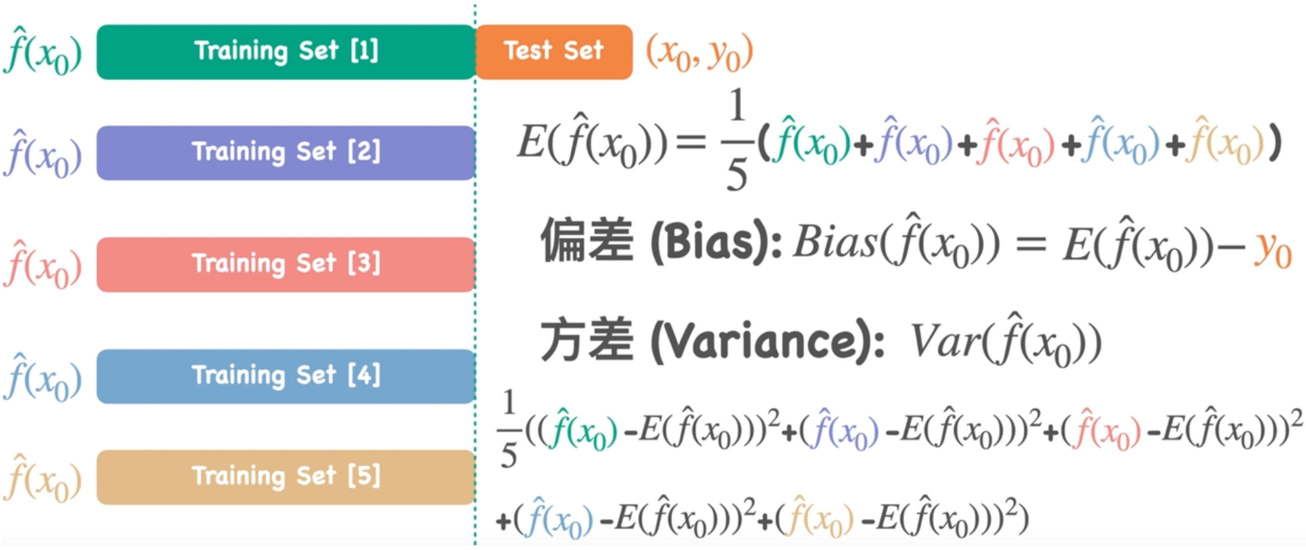
Bias and Variance

bias 和 variance 被用来评估模型的性能。

假如我们有 5 个不同的训练集（符合同一分布），和一个测试集，我们在这 5 个训练集上训模型（可以使用相同或不同的算法），最后得到 5 个模型。

之后，我们取测试集里的一组数据 (x_0, y_0)，将 x_0 输入 5 个模型，得到5 个预测结果。

现在，我们可以得到 5 个模型的预测的期望值 $E\left(\hat{f}\left(x_0\right)\right)$。



那么，现在我们可以得到 5 个模型的 bias 和 variance。

bias（偏差）：期望和实际结果的差距。高偏差意味着模型的准确率很差，即欠拟合。

variance（方差）：模型和期望的方差。高方差意味着模型的泛化能力不好，即过拟合。

bias low, variance low

bias high, variance low:

bias low, variance high:

bias high, variance high:

注：上图的每个点代表每个模型的预测结果（也可以看作对单个模型使用多个点来测试），原点代表真实结果。点离原点越近，表示 bias 越小（准确率高）；点越密集，代表方差越小（模型稳定，泛化性好）。

GENERALIZATION ERROR

损失函数如下：其中 P(x,y) 代表整个数据集，(x,y) 代表其中一个子集，(x,y)~P(x,y) 代表 P(x,y) 的子集 (x,y)，h(x) 是对 x 的预测，y 是 ground truth。

$$\mathcal{L}(h) = E_{(x,y)\sim P(x,y)}[\ f(h(x),y)\ ]$$

E.g., $f(a,b) = (a-b)^2$

之后可以得到：

- Squared loss: $f(a,b) = (a-b)^2$
- Consider one data point (x,y)
- Notation:
  - $Z = h(x|S) - y$
  - $\check{z} = E[Z]$
  - $Z - \check{z} = h(x|S) - E[h(x|S)]$

$$\boxed{\begin{aligned} E_S[(Z-\check{z})^2] &= E[Z^2 - 2Z\check{z} + \check{z}^2] \\ &= E[Z^2] - 2E[Z]\check{z} + \check{z}^2 \\ &= E[Z^2] - \check{z}^2 \end{aligned}}$$

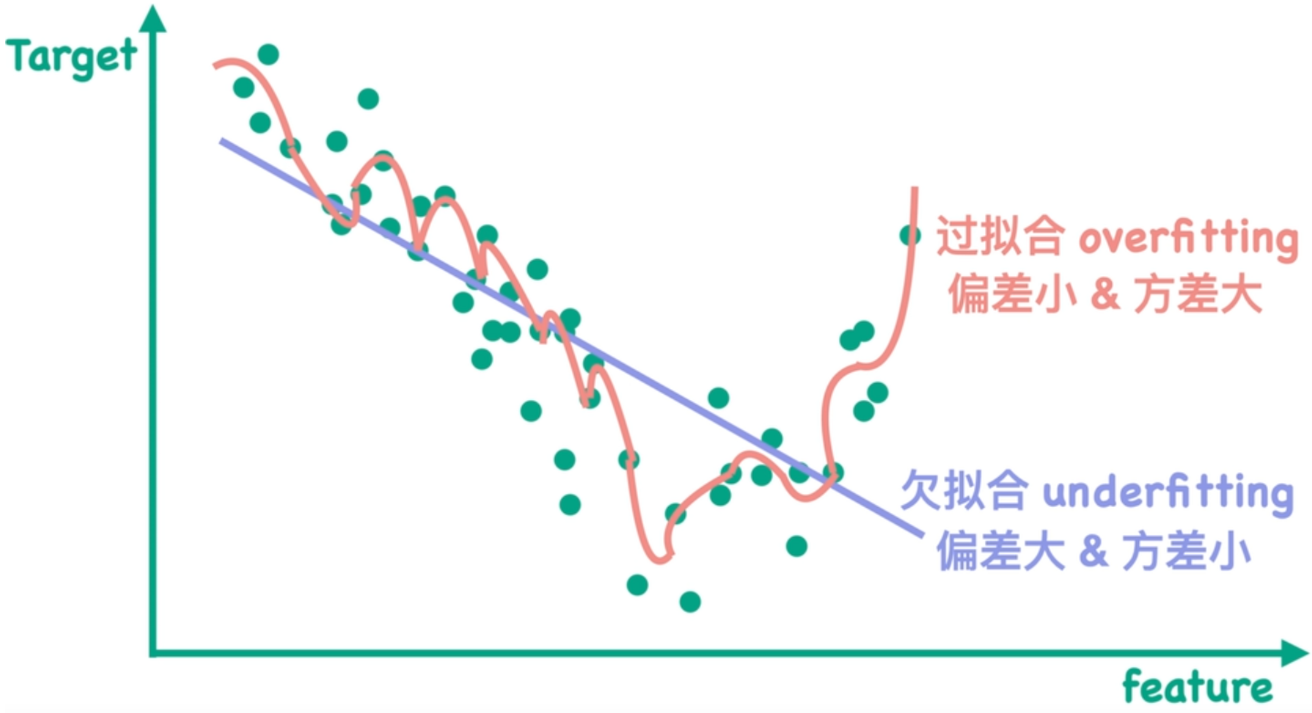
$$\overbrace{\boxed{\begin{aligned} E[f(h(x|S),y)] &= E[Z^2] \\ &= \underbrace{E[(Z-\check{z})^2]}_{\text{Variance}} + \underbrace{\check{z}^2}_{\text{Bias}} \end{aligned}}}^{\text{Expected Error}}$$

Bias/Variance for all (x,y) is expectation over P(x,y).
Can also incorporate measurement noise.
(Similar flavor of analysis for other loss functions.)

注：其中 h(x|S) 代表根据数据集 S 中的数据 x 进行预测。

上面主要得到 expected error，而它是由 variance 和 bias 组成的。和其他的监督学习一样，我们希望 error 最小。因此，我们希望 variance 和 bias 都最小。

然而，这不容易做到。比如下图，蓝色的模型预测的准确率很差，因此偏差大，但把它用在其他数据上，它总的结果不会有太大变化，因此它的方差小；而红色的模型在这个数据集上准确率很高，因此它偏差小，但换个数据集，它的误差一定很大，因此它方差大。



我们希望模型在所有情况下都有较好的表现，因此我们需要对偏差和误差做出平衡。

Ensemble learning

为了获得更好的性能，我们可以训练多个模型，然后平均它们的结果 (Ensemble learning，集成学习)。

两种类型的方法：

- 不使用随机性 (randomness) 的模型
- 包含随机性的模型

分类器集成 (Ensembles of Classifiers)：

- 组合来自不同分类器的分类结果以生成最终输出：
    - 未加权投票 (Unweighted voting)
    - 加权投票

集成的方法：

- Bagging：Bootstrap aggregating
- Boosting
- Random Forests：Bagging reborn

Ensemble methods that minimize variance

这里介绍两种方法：Bagging，Random Forests，它们的目的都是最小化方差。

Bagging

Bagging = Bootstrap Aggregation

大概思路是：从数据集 S 里随机采样生成 M 个子训练集 (理想情况下，每个子训练集都不一样，但实际上训练集没有那么大，所以子训练集里可以有重复的元素，并且子训练集要和原始训练集一样大。理想情况下就叫 bagging，实际情况下就是 Bootstrap Aggregation，Bootstrap 的意思是'自助的'，就是有放回的抽样)，根据训练集生成 M 个模型，最后让这 M 个模型投票，票数最多的就是最终的预测结果。

进行 bagging 后，方差在亚线性下减少（因为 S' 是相关的，即有相同元素），偏差通常略有增加 (假如极端情况下，所以模型都过拟合，因为大家抱团投票，所以结果比较集中，所以方差小；但因为都过拟合，可能得到稀奇古怪的结果，所以偏差大)。

THE BAGGING ALGORITHM

生成 M 个数据集 D_m，并建立 M 个模型 $G_m(x)$，这一步是 bootstrap。

Given data: $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$

For $m = 1 : M$

- Obtain bootstrap sample D_m from the training data D
- Build a model $G_m(\mathbf{x})$ from bootstrap data D_m

接下来是 aggregation，进行投票。对于回归和分类两种任务，有以下两种投票方式：回归就求平均，分类就找众数。

- Regression

$$\hat{y} = \frac{1}{M}\sum_{m=1}^{M} G_m(\mathbf{x})$$

- Classification:
  - Vote over classifier outputs $G_1(\mathbf{x}), \ldots, G_M(\mathbf{x})$

理想情况下，每个子训练集都不相关，因此训练出来的 predictor 可能有不相关的错误。而这正是我们需要的，也是该方法的工作原理。通过处理这些不相关的错误，我们可以最小化方差。

Shortages

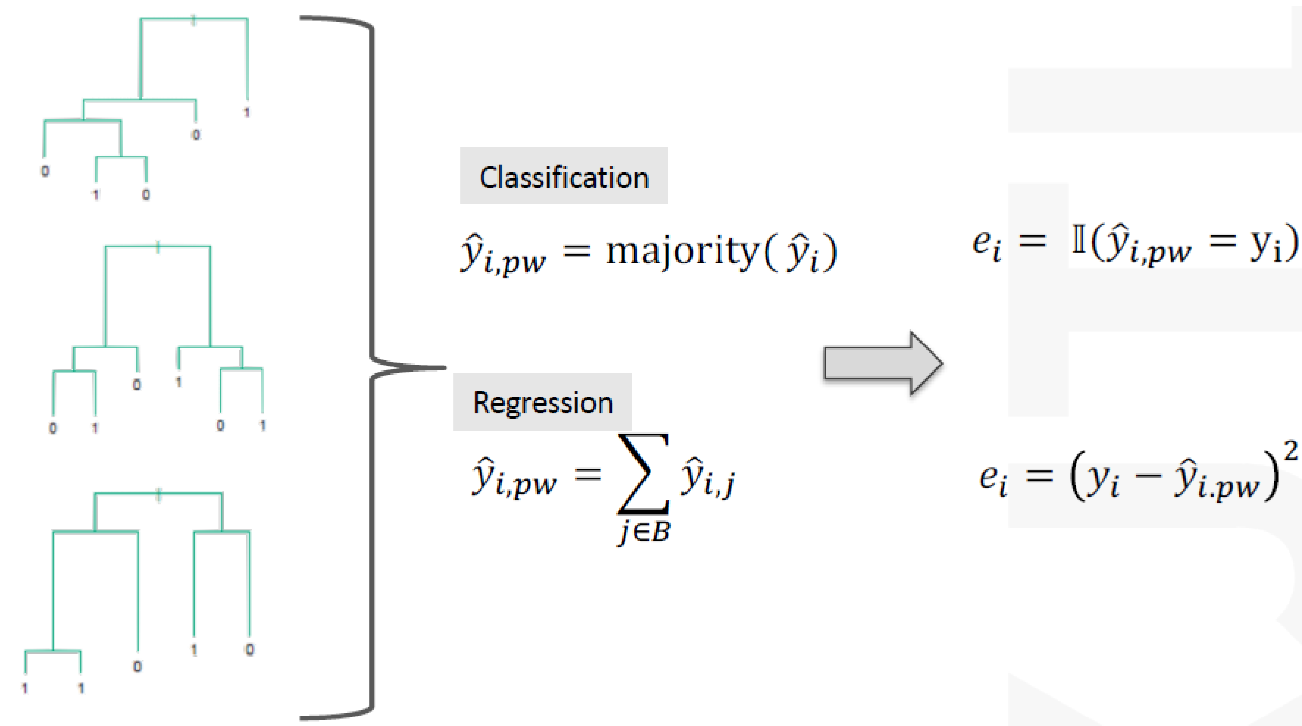
如果模型太多，可能仍会出现过拟合；而如果模型太少，可能出现欠拟合。

Bagging（以及我们将要研究的其他集成方法）的主要缺点是：平均模型不再易于解释。即人们无法再通过基于预测变量值的一系列决策来跟踪输出的"逻辑"。

Random Forests

Bagging 只能 resampling training data (生成子数据集)，而 random forests 可以 sample data and features。

大概思路：从数据集中随机采样 N 个数据 (这里和 bagging 采样采的都是 observations)，这样采样 M 次，生成 M 个子数据集。然后对每个子数据集建立决策树，并对这些决策树进行训练，最后就得到 M 个决策树组成的随机森林。此外，在随机森林中，单个树进行节点分裂 (split) 时，我们从所有特征中随机选取 K 个 feature，再根据某种策略从 K 个已选特征中确定分裂特征。



最后，我们进行预测。对于回归和分类两种任务，有以下两种投票方式：回归就求平均 (上图有误)，分类就找众数。

OOB ERROR

随机森林里的每一棵树是怎么训练和测试的？这里要用到 oob error，即 out-of-bag error (袋外错误率)。

我们把训练当前树的数据集里的数据叫做‘袋内’，那么剩余的数据就是‘袋外’。因此，在训练单个树时，我们将袋内的数据做训练集，讲袋外的数据做测试集，并求出 oob error。

Classification

$$Error_{OOB} = \sum_{i}^{n} e_i = \sum_{i}^{n} \mathbb{I}(\hat{y}_{i,pw} = \mathrm{y_i})$$

Regression

$$Error_{OOB} = \sum_{i}^{n} e_i = \sum_{i}^{n} (y_i - \hat{y}_{i,pw})^2$$

注：$\mathbb{I}\left(\hat{y}_{i,pw} = \mathrm{y_i}\right)$ 意思是：如果预测值和真实值一样，返回 0；否则，返回 1。

TUNING RANDOM FORESTS

随机林模型具有多个要优化的超参数：

- 单个树每次分裂时要随机选择的 feature 数
- 树的总数
- 最小叶节点大小（或数的深度）

FINAL THOUGHTS ON RANDOM FORESTS

- 当 predictors 的数量很大（feature 很多），但 relevant predictors 的数量较少时（分裂时选取的 feature 少），随机森林的性能可能会很差
  - 在每次分裂中，选择 relevant predictors 的几率将很低，因此森林中的大多数树将是弱模型（weak models）。就是只考虑了一小部分，还有好多因素没考虑到，因此性能差。
- 增加森林中的树的数量通常不会增加过拟合的风险
- 同样，by decomposing the generalization error in terms of bias and variance，我们看到增加树的数量会产生一个至少与单个树一样健壮的模型
- 但是，如果树的数量太大，则森林中的树可能会变得更加相关，从而增加方差