Statistical Learning– MATH 6333
Set 7 (Ensemble Methods)

Tamer Oraby

UTRGV
tamer.oraby@utrgv.edu

* Last updated November 10, 2021

A preamble to Bootstrap

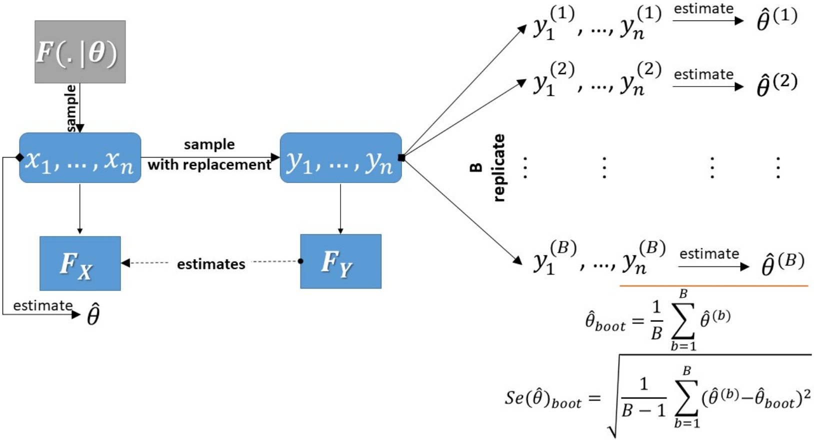
Bootstrap

By Efron (1979, 1981), to estimate

- Bias
- Standard error
- Confidence interval (5 different ways)
- correlation, regression parameters, prediction

Bootstrap

Estimation


$F(.|\theta)$
sample
$x_1, \dots, x_n$
sample
with replacement
$y_1, \dots, y_n$
F_X
estimates
F_Y
estimate
$\hat{\theta}$
B replicate
$y_1^{(1)}, \dots, y_n^{(1)}$ estimate $\hat{\theta}^{(1)}$
$y_1^{(2)}, \dots, y_n^{(2)}$ estimate $\hat{\theta}^{(2)}$
$y_1^{(B)}, \dots, y_n^{(B)}$ estimate $\hat{\theta}^{(B)}$

$$\hat{\theta}_{boot} = \frac{1}{B}\sum_{b=1}^{B}\hat{\theta}^{(b)}$$

$$Se(\hat{\theta})_{boot} = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}(\hat{\theta}^{(b)} - \hat{\theta}_{boot})^2}$$

Bootstrap

Estimation

- $B = 50$ is good enough.
- The probability that any item is selected in any one of the bootstrap samples is given by

$$1 - (1 - \frac{1}{n})^n \approx 1 - e^{-1} = .632$$

- Expected number of distinct points in a sample is $.632n$.

Bootstrap

Confidence Interval

$100(1-\alpha)\%$ Bootstrap Confidence Intervals (BCI):

1. Standard Normal BCI

$$\hat{\theta}_{boot} \pm z_{\alpha/2} se(\hat{\theta})_{boot}$$

2. Basic BCI

$$\left(2\hat{\theta}_{boot} - \hat{\theta}_{1-\alpha/2}, 2\hat{\theta}_{boot} - \hat{\theta}_{\alpha/2}\right)$$

3. Percentile BCI

$$\left(\hat{\theta}_{\alpha/2}, \hat{\theta}_{1-\alpha/2}\right)$$

4. t-type – BCI

$$\left(\hat{\theta}_{boot} - t^*_{1-\alpha/2} se(\hat{\theta}), \hat{\theta}_{boot} + t^*_{\alpha/2} se(\hat{\theta})\right)$$

Bootstrap

Confidence Interval

with t_α^* is the α quantile of $\{t^{(1)}, \ldots, t^{(B)}\}$ where

$$t^{(i)} = \frac{\hat{\theta}^{(i)} - \hat{\theta}_{boot}}{se(\hat{\theta}^{(i)})}$$

and estimation of $se(\hat{\theta}^{(i)})$ requires a further bootstrap from the bootstrapped sample $y_1^{(i)}, \ldots, y_n^{(i)}$

100$(1-\alpha)$% Bootstrap Confidence Intervals (BCI):

1. Bias Corrected accelerated BCI or BCa – BCI

$$\left(\hat{\theta}^*_{\alpha_1}, \hat{\theta}^*_{\alpha_2}\right)$$

are the α_1 and α_2 quantiles of $\{\hat{\theta}^{(1)}, \ldots, \hat{\theta}^{(B)}\}$ and

$$\alpha_1 = \Phi(\hat{z}_0 + \frac{\hat{z}_0 + z_{\alpha/2}}{1 - \hat{a}\,(\hat{z}_0 + z_{\alpha/2})})$$

and

$$\alpha_2 = \Phi(\hat{z}_0 + \frac{\hat{z}_0 + z_{1-\alpha/2}}{1 - \hat{a}\,(\hat{z}_0 + z_{1-\alpha/2})})$$

Bootstrap

Confidence Interval

where Φ is the cdf of the standard normal, $z_{\alpha/2}$ is the standard normal quantile, and the bias corrector

$$\hat{z}_0 = \Phi^{-1}\left(\frac{1}{B}\sum_{i=1}^{B} I(\hat{\theta}^{(i)} \leq \hat{\theta})\right)$$

where I is the indicator function, and the acceleration factor

$$\hat{a} = \frac{\sum_{i=1}^{B}(\hat{\theta}^{(i)} - \hat{\theta})^3}{6\left(\sum_{i=1}^{B}(\hat{\theta}^{(i)} - \hat{\theta})^2\right)^{3/2}}$$

which measures skewness.

Bootstrap

Prediction

- For each b ($b = 1, 2, \ldots, B$), bootstrap sample $y_1^{(b)}, \ldots, y_n^{(b)}$ could be use to make a prediction function $\hat{f}^{(b)}(x)$ and the prediction error for that bootstrap training is

  $$\widehat{Err}^{(b)} = \frac{1}{n} \sum_{i=1}^{n} L(y_i, \hat{f}^{(b)}(x_i))$$

  and then the bootstrap error is estimated by

  $$\widehat{Err}_{boot} = \frac{1}{B} \sum_{i=1}^{B} \widehat{Err}^{(b)}$$

  but it can be far below the correct error.

Bootstrap

Prediction

- So another suggested error is the Leave-one-out bootstrap estimate of prediction error is

$$\widehat{Err}^{(1)}_{boot} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|C_{-i}|}\sum_{b\in C_{-i}} L(y_i, \hat{f}^{(b)}(x_i))$$

  where C_{-i} is the set of all indices of bootstrap samples that don't contain observation i. (Only if they they are non-empty.)

  But it is upwardly biased.

Bootstrap

Prediction

- It is suggested to use the ".632 estimator" that corrects for that bias
$$\widehat{Err}^{(632)}_{boot} = .368\bar{err} + .632\widehat{Err}^{(1)}_{boot}$$
where $\bar{err}$ is the training error rate.

  Still not the best. Look for $\widehat{Err}^{(632+)}_{boot}$.

A preamble to Jackknife

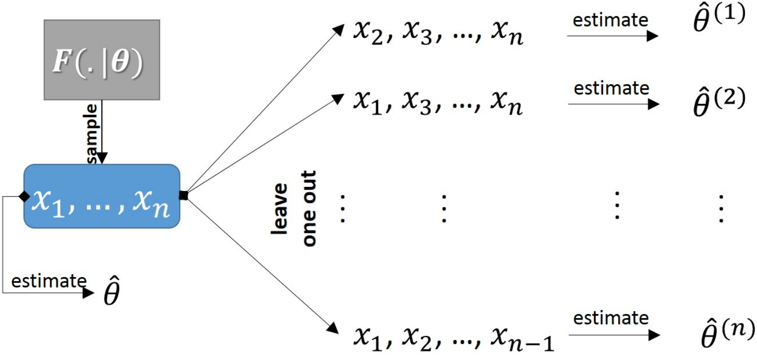
Jackknife

By Quenouille and Tukey, to estimate

- Bias
- Standard error

The estimate $\hat{\theta}$ must be a smooth plug-in estimator: small changes in the data results in small changes in the value of the estimate. The sample mean is a smooth plug-in for the population mean while the sample median is not.

Jackknife



$F(.|\theta)$

sample

$x_1, \dots, x_n$

estimate $\hat{\theta}$

leave one out

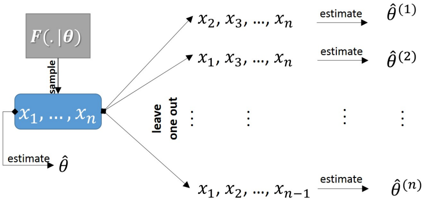
$x_2, x_3, \dots, x_n$ estimate $\hat{\theta}^{(1)}$

$x_1, x_3, \dots, x_n$ estimate $\hat{\theta}^{(2)}$

$\vdots \quad \vdots \quad \vdots \quad \vdots$

$x_1, x_2, \dots, x_{n-1}$ estimate $\hat{\theta}^{(n)}$

Jackknife



Let $\bar{\hat{\theta}}_J = \frac{1}{n}\sum_{i=1}^{n} \hat{\theta}^{(i)}$

- $\hat{Bias} = (n-1)(\bar{\hat{\theta}}_J - \hat{\theta})$ where $\hat{\theta}$ is the estimate of θ using the original sample $x_1, \ldots, x_n$

- Standard error $se(\hat{\theta}_J)$ is $\sqrt{n-1}$ times the standard deviation of the jackknife estimates $\hat{\theta}^{(1)}, \ldots, \hat{\theta}^{(n)}$

Ensemble Methods

Ensemble Methods

The idea is to combine predictions from many "building blocks." They are usually weak learners that wouldn't stand on their own. It is usually based on weighted output of their predictions and performance-based evaluation.

- Bayesian model averaging
- Bagging (Bootstrap aggregating)
- Stacking
- Boosting
- Random forests

Bayesian Model Averaging

Ensemble Methods

Bayesian model averaging

- Recall that in Bayesian inference, the posterior distribution is given by

$$P(\theta|X) = \frac{P(X|\theta)\,P(\theta)}{\int_\Theta P(X|\theta')\,P(\theta')d\theta'}$$

- The maximum posterior distribution (MAP) is commonly used as a point estimate.

- To make predictions we use the predictive posterior distribution

$$P(x_*|X) = \int_\Theta P(x_*|\theta')\,P(\theta'|X)d\theta'$$

Ensemble Methods

Bayesian model averaging

- Recall that in Bayesian inference, the posterior distribution is given by

$$P(\theta|X) = \frac{P(X|\theta)\,P(\theta)}{\int_{\Theta} P(X|\theta')\,P(\theta')d\theta'}$$

- The maximum posterior distribution (MAP) is commonly used as a point estimate.

- To make predictions we use the predictive posterior distribution

$$P(x_*|X) = \int_{\Theta} P(x_*|\theta')\,P(\theta'|X)d\theta'$$

Ensemble Methods

Bayesian model averaging

- Recall that in Bayesian inference, the posterior distribution is given by

$$P(\theta|X) = \frac{P(X|\theta)\, P(\theta)}{\int_{\Theta} P(X|\theta')\, P(\theta') d\theta'}$$

- The maximum posterior distribution (MAP) is commonly used as a point estimate.

- To make predictions we use the predictive posterior distribution

$$P(x_*|X) = \int_{\Theta} P(x_*|\theta')\, P(\theta'|X) d\theta'$$

Ensemble Methods

Bayesian model averaging

Similarly, for L number of models $\mathcal{M}_k$, $k = 1, 2, \ldots, L$

- $P(X|\mathcal{M}_k) = \int P(X|\theta_k, \mathcal{M}_k) P(\theta_k|\mathcal{M}_k) d\theta_k$
- $P(\mathcal{M}_k|X) = \dfrac{P(X|\mathcal{M}_k)\, P(\mathcal{M}_k)}{\sum_{\ell=1}^{L} P(X|\mathcal{M}_\ell)\, P(\mathcal{M}_\ell)}$
- and so the posterior distribution of prediction $f(x_*)$ is given by
$$P(f(x_*)|X) = \sum_{\ell=1}^{L} P(f(x_*)|\mathcal{M}_\ell, X)\, P(\mathcal{M}_\ell|X)$$
- and the mean (as a weighted average) is
$$E(f(x_*)|X) = \sum_{\ell=1}^{L} E(f(x_*)|\mathcal{M}_\ell, X)\, P(\mathcal{M}_\ell|X)$$

Ensemble Methods

Bayesian model averaging

Similarly, for L number of models $\mathcal{M}_k$, $k = 1, 2, \ldots, L$

- $P(X|\mathcal{M}_k) = \int P(X|\theta_k, \mathcal{M}_k) P(\theta_k|\mathcal{M}_k) d\theta_k$
- $P(\mathcal{M}_k|X) = \dfrac{P(X|\mathcal{M}_k)\, P(\mathcal{M}_k)}{\sum_{\ell=1}^{L} P(X|\mathcal{M}_\ell)\, P(\mathcal{M}_\ell)}$
- and so the posterior distribution of prediction $f(x_*)$ is given by

$$P(f(x_*)|X) = \sum_{\ell=1}^{L} P(f(x_*)|\mathcal{M}_\ell, X)\, P(\mathcal{M}_\ell|X)$$

- and the mean (as a weighted average) is

$$E(f(x_*)|X) = \sum_{\ell=1}^{L} E(f(x_*)|\mathcal{M}_\ell, X)\, P(\mathcal{M}_\ell|X)$$

Ensemble Methods

Bayesian model averaging

Similarly, for L number of models $\mathcal{M}_k$, $k = 1, 2, \ldots, L$

- $P(X|\mathcal{M}_k) = \int P(X|\theta_k, \mathcal{M}_k) P(\theta_k|\mathcal{M}_k) d\theta_k$
- $P(\mathcal{M}_k|X) = \dfrac{P(X|\mathcal{M}_k)\, P(\mathcal{M}_k)}{\sum_{\ell=1}^{L} P(X|\mathcal{M}_\ell)\, P(\mathcal{M}_\ell)}$
- and so the posterior distribution of prediction $f(x_*)$ is given by
$$P(f(x_*)|X) = \sum_{\ell=1}^{L} P(f(x_*)|\mathcal{M}_\ell, X)\, P(\mathcal{M}_\ell|X)$$
- and the mean (as a weighted average) is
$$E(f(x_*)|X) = \sum_{\ell=1}^{L} E(f(x_*)|\mathcal{M}_\ell, X)\, P(\mathcal{M}_\ell|X)$$

Ensemble Methods

Bayesian model averaging

Similarly, for L number of models $\mathcal{M}_k$, $k = 1, 2, \ldots, L$

- $P(X|\mathcal{M}_k) = \int P(X|\theta_k, \mathcal{M}_k) P(\theta_k|\mathcal{M}_k) d\theta_k$
- $P(\mathcal{M}_k|X) = \dfrac{P(X|\mathcal{M}_k)\, P(\mathcal{M}_k)}{\sum_{\ell=1}^{L} P(X|\mathcal{M}_\ell)\, P(\mathcal{M}_\ell)}$
- and so the posterior distribution of prediction $f(x_*)$ is given by

$$P(f(x_*)|X) = \sum_{\ell=1}^{L} P(f(x_*)|\mathcal{M}_\ell, X)\, P(\mathcal{M}_\ell|X)$$

- and the mean (as a weighted average) is

$$E(f(x_*)|X) = \sum_{\ell=1}^{L} E(f(x_*)|\mathcal{M}_\ell, X)\, P(\mathcal{M}_\ell|X)$$

Ensemble Methods

Bayesian model averaging

Other types...

- Committee method that the prediction is a simple average

$$\frac{1}{L}\sum_{\ell=1}^{L} E(f(x_*)|\mathcal{M}_\ell, X)$$

- Using the Bayesian Information Criterion (BIC) for the same model type with the same number of parameters

$$\sum_{\ell=1}^{L} E(f(x_*)|\mathcal{M}_\ell, X)\,\frac{e^{-BIC_\ell}}{\sum_{k=1}^{L} e^{-BIC_k}}$$

Ensemble Methods

Bayesian model averaging

Other types...

- Committee method that the prediction is a simple average

$$\frac{1}{L}\sum_{\ell=1}^{L} E(f(x_*)|\mathcal{M}_\ell, X)$$

- Using the Bayesian Information Criterion (BIC) for the same model type with the same number of parameters

$$\sum_{\ell=1}^{L} E(f(x_*)|\mathcal{M}_\ell, X)\,\frac{e^{-BIC_\ell}}{\sum_{k=1}^{L} e^{-BIC_k}}$$

Bagging (Bootstrap aggregating)

Ensemble Methods

Bagging

- Helps to reduce high variance learning methods (especially decision trees)
- Recall that, if $\{X_i; i = 1, \ldots, n\}$ are independent with mean μ and variance σ^2, then $V(\bar{X}) = V(\frac{1}{n}\sum_{i=1}^{n} X_i) = \frac{\sigma^2}{n} < \sigma^2$ (decreased), while $E(\bar{X}) = \mu$ (remains)
- If they were only identical with correlation ρ then $V(\bar{X}) = V(\frac{1}{n}\sum_{i=1}^{n} X_i) = \frac{\sigma^2}{n}(1 + (n-1)\rho) \leq \sigma^2$
- which could be done for prediction over n training data sets, but we can not have that many training data sets ...

Ensemble Methods

Bagging

- Helps to reduce high variance learning methods (especially decision trees)

- Recall that, if $\{X_i; i = 1, \ldots, n\}$ are independent with mean μ and variance σ^2, then $V(\bar{X}) = V(\frac{1}{n}\sum_{i=1}^{n} X_i) = \frac{\sigma^2}{n} < \sigma^2$ (decreased), while $E(\bar{X}) = \mu$ (remains)

- If they were only identical with correlation ρ then $V(\bar{X}) = V(\frac{1}{n}\sum_{i=1}^{n} X_i) = \frac{\sigma^2}{n}(1 + (n-1)\rho) \leq \sigma^2$

- which could be done for prediction over n training data sets, but we can not have that many training data sets ...

Ensemble Methods

Bagging

- Helps to reduce high variance learning methods (especially decision trees)

- Recall that, if $\{X_i; i = 1, \ldots, n\}$ are independent with mean μ and variance σ^2, then $V(\bar{X}) = V(\frac{1}{n}\sum_{i=1}^{n} X_i) = \frac{\sigma^2}{n} < \sigma^2$ (decreased), while $E(\bar{X}) = \mu$ (remains)

- If they were only identical with correlation ρ then $V(\bar{X}) = V(\frac{1}{n}\sum_{i=1}^{n} X_i) = \frac{\sigma^2}{n}(1 + (n-1)\rho) \leq \sigma^2$

- which could be done for prediction over n training data sets, but we can not have that many training data sets ...

Ensemble Methods

Bagging

- Helps to reduce high variance learning methods (especially decision trees)

- Recall that, if $\{X_i; i = 1, \ldots, n\}$ are independent with mean μ and variance σ^2, then $V(\bar{X}) = V(\frac{1}{n}\sum_{i=1}^{n} X_i) = \frac{\sigma^2}{n} < \sigma^2$ (decreased), while $E(\bar{X}) = \mu$ (remains)

- If they were only identical with correlation ρ then $V(\bar{X}) = V(\frac{1}{n}\sum_{i=1}^{n} X_i) = \frac{\sigma^2}{n}(1 + (n-1)\rho) \leq \sigma^2$

- which could be done for prediction over n training data sets, but we can not have that many training data sets ...

Ensemble Methods

Bagging

- Helps to reduce high variance learning methods (especially decision trees)

- Recall that, if $\{X_i; i = 1, \ldots, n\}$ are independent with mean μ and variance σ^2, then $V(\bar{X}) = V(\frac{1}{n}\sum_{i=1}^{n} X_i) = \frac{\sigma^2}{n} < \sigma^2$ (decreased), while $E(\bar{X}) = \mu$ (remains)

- If they were only identical with correlation ρ then $V(\bar{X}) = V(\frac{1}{n}\sum_{i=1}^{n} X_i) = \frac{\sigma^2}{n}(1 + (n-1)\rho) \leq \sigma^2$

- which could be done for prediction over n training data sets, but we can not have that many training data sets ...

Ensemble Methods

Bagging

- Helps to reduce high variance learning methods (especially decision trees)
- Recall that, if $\{X_i; i = 1, \dots, n\}$ are independent with mean μ and variance σ^2, then $V(\bar{X}) = V(\frac{1}{n}\sum_{i=1}^{n} X_i) = \frac{\sigma^2}{n} < \sigma^2$ (decreased), while $E(\bar{X}) = \mu$ (remains)
- If they were only identical with correlation ρ then $V(\bar{X}) = V(\frac{1}{n}\sum_{i=1}^{n} X_i) = \frac{\sigma^2}{n}(1 + (n-1)\rho) \leq \sigma^2$
- which could be done for prediction over n training data sets, but we can not have that many training data sets ...

Ensemble Methods

Bagging

- Use bootstraping to produce B bootstrap samples, train the statistical learning method (like trees) on each one of them and make a prediction $\hat{f}^{(b)}(x)$ for regression or a class $\hat{C}^{(b)}(x)$ for classification

- from which we get the bagging prediction

$$\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^{(b)}(x) \text{ for regression}$$

and

$$\hat{C}_{bag}(x) = \text{ majority vote } \{\hat{C}^{(1)}(x), \hat{C}^{(2)}(x), \ldots, \hat{C}^{(B)}(x)\}$$

for classification.

Ensemble Methods

Bagging

- Use bootstraping to produce B bootstrap samples, train the statistical learning method (like trees) on each one of them and make a prediction $\hat{f}^{(b)}(x)$ for regression or a class $\hat{C}^{(b)}(x)$ for classification

- from which we get the bagging prediction

$$\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{(b)}(x) \text{ for regression}$$

and

$$\hat{C}_{bag}(x) = \text{ majority vote } \{\hat{C}^{(1)}(x), \hat{C}^{(2)}(x), \ldots, \hat{C}^{(B)}(x)\}$$

for classification.

Ensemble Methods

Bagging

- Out-of-Bag (OOB) error estimation: OOB are the approximately .386n points not selected on a bootstrap sample and can be used for estimating testing error

$$\widehat{Err}^{(1)}_{boot} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|C_{-i}|}\sum_{b\in C_{-i}} L(y_i, \hat{f}^{(b)}(x_i))$$

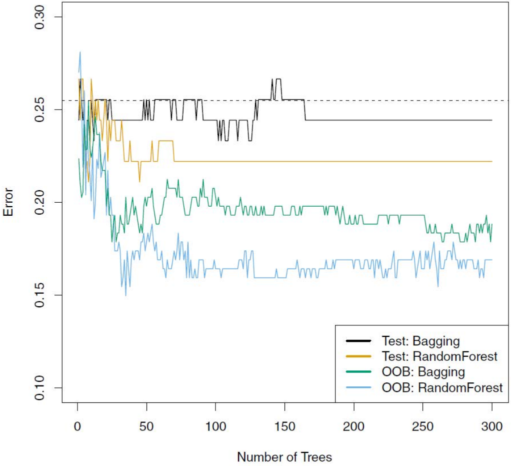
where C_{-i} is the set of all indices of bootstrap samples that don't contain observation i.

Ensemble Methods

Bagging

Example (Another Heart data)

Test error vs OOB



Stacking

Ensemble Methods

Stacking

- Uses LOOCV for model averaging with normalized weights $\hat{w}_k$ that are relatively lower for complex model and not best-fit models
- If there are K number of models $\mathcal{M}_k$ with vector parameter θ_k, $k = 1, 2, \ldots, K$, which is to be estimated using the training data then

$$\hat{f}_{stack}(x) = \sum_{k=1}^{K} \hat{w}_k f_k(x|\hat{\theta}_k)$$

- Where

$$(\hat{w}_1, \hat{w}_2, \ldots, \hat{w}_K) = argmin_w \sum_{i=1}^{n} \left(y_i - \sum_{k=1}^{K} w_k f_k^{(-i)}(x_i|\hat{\theta}_k) \right)^2$$

- LOOCV selection of the best model happens if we require one $w_k = 1$ and the rest are zeros.

Ensemble Methods

Stacking

- Uses LOOCV for model averaging with normalized weights $\hat{w}_k$ that are relatively lower for complex model and not best-fit models

- If there are K number of models $\mathcal{M}_k$ with vector parameter θ_k, $k = 1, 2, \ldots, K$, which is to be estimated using the training data then

$$\hat{f}_{stack}(x) = \sum_{k=1}^{K} \hat{w}_k f_k(x|\hat{\theta}_k)$$

- Where

$$(\hat{w}_1, \hat{w}_2, \ldots, \hat{w}_K) = argmin_w \sum_{i=1}^{n} \left(y_i - \sum_{k=1}^{K} w_k f_k^{(-i)}(x_i|\hat{\theta}_k) \right)^2$$

- LOOCV selection of the best model happens if we require one $w_k = 1$ and the rest are zeros.

Ensemble Methods

Stacking

- Uses LOOCV for model averaging with normalized weights $\hat{w}_k$ that are relatively lower for complex model and not best-fit models

- If there are K number of models $\mathcal{M}_k$ with vector parameter θ_k, $k = 1, 2, \ldots, K$, which is to be estimated using the training data then

$$\hat{f}_{stack}(x) = \sum_{k=1}^{K} \hat{w}_k f_k(x|\hat{\theta}_k)$$

- Where

$$(\hat{w}_1, \hat{w}_2, \ldots, \hat{w}_K) = argmin_w \sum_{i=1}^{n} \left(y_i - \sum_{k=1}^{K} w_k f_k^{(-i)}(x_i|\hat{\theta}_k) \right)^2$$

- LOOCV selection of the best model happens if we require one $w_k = 1$ and the rest are zeros.

Ensemble Methods

Stacking

- Uses LOOCV for model averaging with normalized weights $\hat{w}_k$ that are relatively lower for complex model and not best-fit models

- If there are K number of models $\mathcal{M}_k$ with vector parameter θ_k, $k = 1, 2, \ldots, K$, which is to be estimated using the training data then

$$\hat{f}_{stack}(x) = \sum_{k=1}^{K} \hat{w}_k f_k(x|\hat{\theta}_k)$$

- Where

$$(\hat{w}_1, \hat{w}_2, \ldots, \hat{w}_K) = argmin_w \sum_{i=1}^{n} \left(y_i - \sum_{k=1}^{K} w_k f_k^{(-i)}(x_i|\hat{\theta}_k) \right)^2$$

- LOOCV selection of the best model happens if we require one $w_k = 1$ and the rest are zeros.

Ensemble Methods

Stacking

- What can go wrong if we rather do find

$$(\hat{w}_1, \hat{w}_2, \ldots, \hat{w}_K) = argmin_w \sum_{i=1}^{n} \left(y_i - \sum_{k=1}^{K} w_k f_k(x_i|\hat{\theta}_k) \right)^2$$

- If it is a linear regression problem with subsets as the the K models then the full model with the full model will have the weight $= 1$ and the rest $w_k = 0$ since

Ensemble Methods

Stacking

- What can go wrong if we rather do find

$$(\hat{w}_1, \hat{w}_2, \ldots, \hat{w}_K) = argmin_w \sum_{i=1}^{n} \left(y_i - \sum_{k=1}^{K} w_k f_k(x_i|\hat{\theta}_k) \right)^2$$

- If it is a linear regression problem with subsets as the the K models then the full model with the full model will have the weight $= 1$ and the rest $w_k = 0$ since

Boosting

Ensemble Methods

Boosting

- Used mostly for classification problems and could be extended for regression
- It is a committee of weak learners whose error rate are a little bit better than random guessing $(e\bar{r}r = \frac{1}{n}\sum_{i=1}^{n} I(y_i \neq G(x_i)) < .5)$
- There are several boosting algorithms, like the Adaptive Boosting algorithm Adaboost.M1, which are very powerful than other classification methods

Ensemble Methods

Boosting

- Used mostly for classification problems and could be extended for regression
- It is a committee of weak learners whose error rate are a little bit better than random guessing $(\bar{err} = \frac{1}{n}\sum_{i=1}^{n} I(y_i \neq G(x_i)) < .5)$
- There are several boosting algorithms, like the Adaptive Boosting algorithm Adaboost.M1, which are very powerful than other classification methods

Ensemble Methods

Boosting

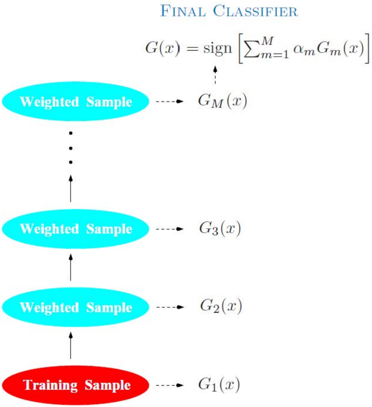
- Used mostly for classification problems and could be extended for regression
- It is a committee of weak learners whose error rate are a little bit better than random guessing ($\bar{err} = \frac{1}{n}\sum_{i=1}^{n} I(y_i \neq G(x_i)) < .5$)
- There are several boosting algorithms, like the Adaptive Boosting algorithm Adaboost.M1, which are very powerful than other classification methods

Ensemble Methods

Boosting - AdaBoost.M1 Algorithm *aka* Discrete AdaBoost

Consider a two class classification problem $Y \in \{-1, 1\}$

- Give weights to each item (x_i, y_i) of w_i^m such that $w_i^1 = \frac{1}{n}$
- The algorithm keep modifying the data through re-weighting and train the learning method on the new weighted data to produce weak learners that form a committee at the end
- The sequence of weights $\{\alpha_m; m = 1, 2, \ldots, M\}$ are produced by the algorithm

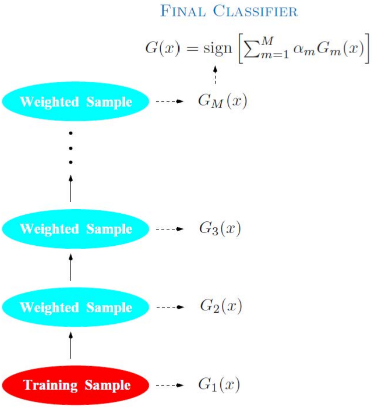


Ensemble Methods

Boosting - AdaBoost.M1 Algorithm *aka* Discrete AdaBoost

Consider a two class classification problem $Y \in \{-1, 1\}$

- Give weights to each item (x_i, y_i) of w_i^m such that $w_i^1 = \frac{1}{n}$
- The algorithm keep modifying the data through re-weighting and train the learning method on the new weighted data to produce weak learners that form a committee at the end
- The sequence of weights $\{\alpha_m; m = 1, 2, \ldots, M\}$ are produced by the algorithm

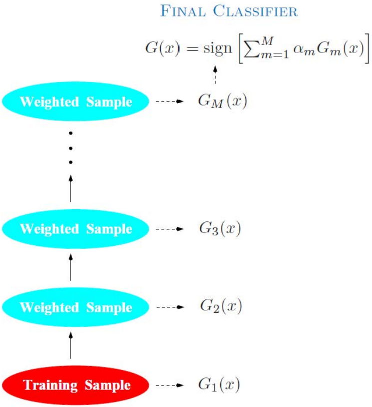


Ensemble Methods

Boosting - AdaBoost.M1 Algorithm *aka* Discrete AdaBoost

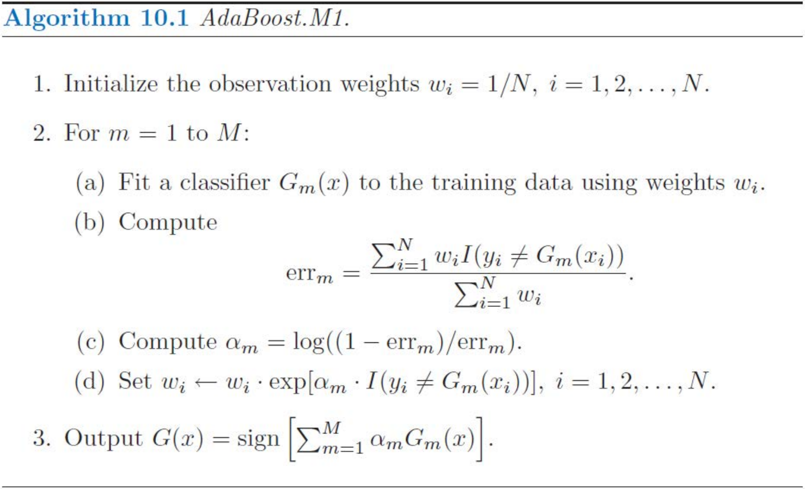
Consider a two class classification problem $Y \in \{-1, 1\}$

- Give weights to each item (x_i, y_i) of w_i^m such that $w_i^1 = \frac{1}{n}$
- The algorithm keep modifying the data through re-weighting and train the learning method on the new weighted data to produce weak learners that form a committee at the end
- The sequence of weights $\{\alpha_m; m = 1, 2, \ldots, M\}$ are produced by the algorithm



Ensemble Methods

Boosting - AdaBoost.M1 Algorithm *aka* Discrete AdaBoost



Algorithm 10.1 *AdaBoost.M1.*

```
1. Initialize the observation weights $w_i = 1/N,\ i = 1, 2, \ldots, N$.
2. For $m = 1$ to $M$:
    (a) Fit a classifier $G_m(x)$ to the training data using weights $w_i$.
    (b) Compute
            $\text{err}_m = \frac{\sum_{i=1}^{N} w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^{N} w_i}$.
    (c) Compute $\alpha_m = \log((1 - \text{err}_m)/\text{err}_m)$.
    (d) Set $w_i \leftarrow w_i \cdot \exp[\alpha_m \cdot I(y_i \neq G_m(x_i))],\ i = 1, 2, \ldots, N$.
3. Output $G(x) = \text{sign}\left[\sum_{m=1}^{M} \alpha_m G_m(x)\right]$.
```

Random Forests

Ensemble Methods

Random Forests

- It is bagging of a decision tree. It decreases variance and keep the low bias.

- Again, bagging does use $V(\bar{X}) = V(\frac{1}{n}\sum_{i=1}^{n} X_i) = \frac{\sigma^2}{n}(1 + (n-1)\rho) = \rho\sigma^2 + \frac{1-\rho}{n}\sigma^2 \leq \sigma^2$, while random forest tries to diminish the second term by using many bootstrap samples (large B) and break down the correlation (making $\rho \sim 0$) to decrease the first term and still not upsetting σ^2.

- The latter is achieved by random selection and usage of $m < p$ inputs at each split for the trees used on every bootstrap sample.

Ensemble Methods

Random Forests

- It is bagging of a decision tree. It decreases variance and keep the low bias.
- Again, bagging does use $V(\bar{X}) = V(\frac{1}{n}\sum_{i=1}^{n} X_i) = \frac{\sigma^2}{n}(1+(n-1)\rho) = \rho\sigma^2 + \frac{1-\rho}{n}\sigma^2 \leq \sigma^2$, while random forest tries to diminish the second term by using many bootstrap samples (large B) and break down the correlation (making $\rho \sim 0$) to decrease the first term and still not upsetting σ^2.
- The latter is achieved by random selection and usage of $m < p$ inputs at each split for the trees used on every bootstrap sample.

Ensemble Methods

Random Forests

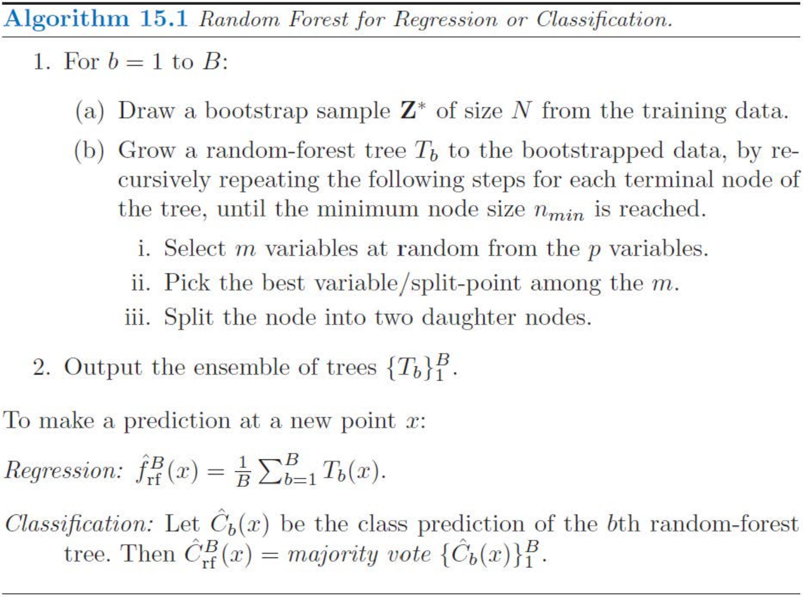
- It is bagging of a decision tree. It decreases variance and keep the low bias.

- Again, bagging does use $V(\bar{X}) = V(\frac{1}{n}\sum_{i=1}^{n} X_i) = \frac{\sigma^2}{n}(1+(n-1)\rho) = \rho\sigma^2 + \frac{1-\rho}{n}\sigma^2 \leq \sigma^2$, while random forest tries to diminish the second term by using many bootstrap samples (large B) and break down the correlation (making $\rho \sim 0$) to decrease the first term and still not upsetting σ^2.

- The latter is achieved by random selection and usage of $m < p$ inputs at each split for the trees used on every bootstrap sample.

Ensemble Methods

Random Forests



Algorithm 15.1 *Random Forest for Regression or Classification.*

```
1. For $b = 1$ to $B$:
    (a) Draw a bootstrap sample $\mathbf{Z}^*$ of size $N$ from the training data.
    (b) Grow a random-forest tree $T_b$ to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size $n_{min}$ is reached.
        i. Select $m$ variables at random from the $p$ variables.
        ii. Pick the best variable/split-point among the $m$.
        iii. Split the node into two daughter nodes.
2. Output the ensemble of trees $\{T_b\}_1^B$.
```

To make a prediction at a new point x:

Regression: $\hat{f}_{\text{rf}}^B(x) = \frac{1}{B}\sum_{b=1}^B T_b(x)$.

Classification: Let $\hat{C}_b(x)$ be the class prediction of the bth random-forest tree. Then $\hat{C}_{\text{rf}}^B(x) = \textit{majority vote}\ \{\hat{C}_b(x)\}_1^B$.

Ensemble Methods

Random Forests

- Tuning parameters:
  - For regression problems: $m = \lfloor \frac{p}{3} \rfloor$ and $n_{min} = 5$
  - For classification problems: $m = \lfloor \sqrt{p} \rfloor$ and $n_{min} = 1$
  - $m = p$ is just bagging

Ensemble Methods

Random Forests

- Tuning parameters:
  - For regression problems: $m = \lfloor \frac{p}{3} \rfloor$ and $n_{min} = 5$
  - For classification problems: $m = \lfloor \sqrt{p} \rfloor$ and $n_{min} = 1$
  - $m = p$ is just bagging

Ensemble Methods

Random Forests

- Tuning parameters:
  - For regression problems: $m = \lfloor \frac{p}{3} \rfloor$ and $n_{min} = 5$
  - For classification problems: $m = \lfloor \sqrt{p} \rfloor$ and $n_{min} = 1$
  - $m = p$ is just bagging

Ensemble Methods

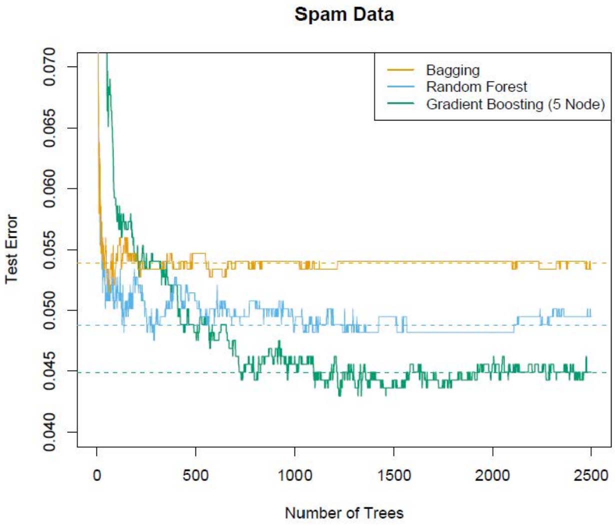
Random Forests

- Tuning parameters:
  - For regression problems: $m = \lfloor \frac{p}{3} \rfloor$ and $n_{min} = 5$
  - For classification problems: $m = \lfloor \sqrt{p} \rfloor$ and $n_{min} = 1$
  - $m = p$ is just bagging

Ensemble Methods

Random Forests

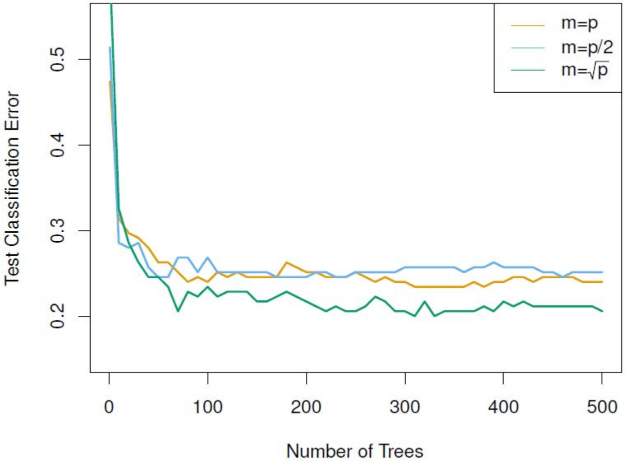
Example (Spam data)



Ensemble Methods

Random Forests

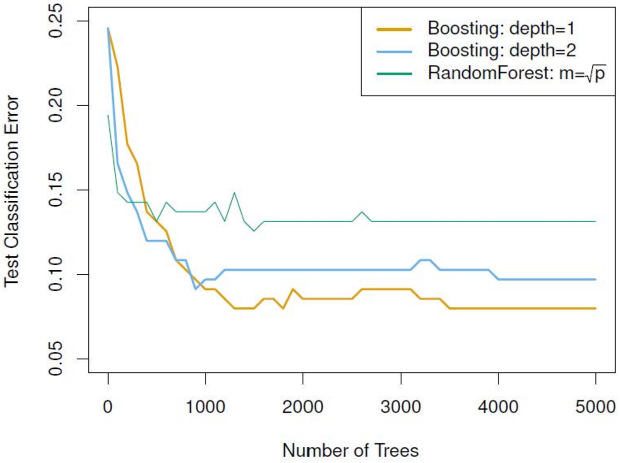
Example (Gene data)



Ensemble Methods

Random Forests

Example (Gene data)



Ensemble Methods

DIY in R

1. Carry out a boosting regression tree for the prostate cancer data using library(gbm)
2. Carry out a random forest for the SA hearth disease data using library(randomForest)

Please study the different methods in the ISL book.

Ensemble Methods

DIY in R

1. Carry out a boosting regression tree for the prostate cancer data using library(gbm)
2. Carry out a random forest for the SA hearth disease data using library(randomForest)

Please study the different methods in the ISL book.

Ensemble Methods

DIY in R

1. Carry out a boosting regression tree for the prostate cancer data using library(gbm)
2. Carry out a random forest for the SA hearth disease data using library(randomForest)

Please study the different methods in the ISL book.

Ensemble Methods

DIY in R

1. Carry out a boosting regression tree for the prostate cancer data using library(gbm)
2. Carry out a random forest for the SA hearth disease data using library(randomForest)

Please study the different methods in the ISL book.

End of Set 7

EM algorithm

Expectation – Maximization (EM) algorithm

- EM is used for incomplete data, e.g. missing data, censored data or latent variables.

- If the complete data $X = (O, M)$ where O is the observed data and M is the missing data.

- Note that
$$f(X|\theta) = f(M|\theta, O) \cdot f(O|\theta)$$
That is
$$L(\theta|X) = f(M|\theta, O) \cdot L(\theta|O)$$

- $L(\theta|X)$ is the complete likelihood function and $L(\theta|O)$ is the incomplete likelihood function

Expectation – Maximization (EM) algorithm

and

$$logL(\theta|O) = logL(\theta|X) - log(f(M|\theta, O))$$

so

$$\int logL(\theta|O)f(M|\theta', O)dM = \int logL(\theta|X)f(M|\theta', O)dM$$

$$- \int log(f(M|\theta, O))f(M|\theta', O)dM$$

or

$$logL(\theta|O) = \mathbf{E}_{M|\theta',O}(logL(\theta|X)) - \mathbf{E}_{M|\theta',O}(log(f(M|\theta, O)))$$

Expectation – **M**aximization (EM) algorithm

EM Algorithm:

1. Start from initial point $\theta^{(0)}$, then for each $k \geq 1$

2. **E** step: Find $Q_k(\theta|\theta^{(k)}) := \mathbf{E}_{M|\theta^{(k)},O}(logL(\theta|X))$

3. **M** step: Find $\theta^{(k+1)} = \text{argmax}_{\theta\in\Theta} Q_k(\theta|\theta^{(k)})$

4. Stop when $\left|\theta^{(k+1)} - \theta^{(k)}\right| / \theta^{(k)} < TOL$

Remark: Convergence is theoretically guaranteed.

Expectation – Maximization (EM) algorithm

Example: Let $x_1, x_2, \ldots, x_n$ be an observed data of completion time at checking out at a grocery store with two cashiers and no waiting lines. They are modeled by a mixture of two exponential distributions with rates λ_1 and λ_2 with probability of selection (mixture weights) p and $1-p$.
The parameter vector is $\theta = (p, \lambda_1, \lambda_2)$ and $f(x_i|\lambda) = \lambda e^{-\lambda x_i}$.
The incomplete likelihood function

$$L(\theta|x) = \prod_{i=1}^{n} (p \cdot f(x_i|\lambda_1) + (1-p) \cdot f(x_i|\lambda_2) \cdot)$$

But what we didn't observe is from where each data point is coming from. That corresponds to latent variable $z_1, \ldots, z_n$ for which cashier was selected, encoded as $z_i = 1$ if cashier 1 is selected and $z_i = 0$ if cashier 2 is selected.

Expectation – Maximization (EM) algorithm

By Bayes' theorem

$$p_i := P(Z_i = 1|X = x_i, \theta) = \frac{p \cdot f(x_i|\lambda_1)}{p \cdot f(x_i|\lambda_1) + (1-p) \cdot f(x_i|\lambda_2)}$$

and the complete likelihood function

$$L(\theta|x, z) = \prod_{i=1}^{n} (z_i p \cdot f(x_i|\lambda_1) + (1 - z_i)(1 - p) \cdot f(x_i|\lambda_2))$$

and

$$\mathbf{E}_{Z|X,\theta}(logL(\theta|x, z)) = \sum_{j=0}^{1} \sum_{i=1}^{n} \log\left(z_i p \cdot f(x_i|\lambda_1) + (1 - z_i)(1 - p)\right.$$

$$\left. \cdot f(x_i|\lambda_2)\right) \cdot P(Z_i = j|X = x_i, \theta)$$

Expectation – Maximization (EM) algorithm

Thus, E step:

$$Q_k(\theta|\theta^{(k)}) = \mathbf{E}_{Z|X,\theta^{(k)}}(logL(\theta|x,z)) =$$

$$\sum_{i=1}^{n} p_i^{(k)}(\log(p)+\log(f(x_i|\lambda_1)))+(1-p_i^{(k)})(\log(1-p)+\log(f(x_i|\lambda_2))) =$$

$$\sum_{i=1}^{n} \left(p_i^{(k)} \log(p) + (1-p_i^{(k)}) \log(1-p)\right) +$$

$$\sum_{i=1}^{n} \left(p_i^{(k)} \log(f(x_i|\lambda_1)) + (1-p_i^{(k)}) \log(f(x_i|\lambda_2))\right)$$

where

$$p_i^{(k)} := P(Z_i = 1|X = x_i, \theta^{(k)}) = \frac{p^{(k)} \cdot f(x_i|\lambda_1^{(k)})}{p^{(k)} \cdot f(x_i|\lambda_1^{(k)}) + (1-p^{(k)}) \cdot f(x_i|\lambda_2^{(k)})}$$

Expectation – Maximization (EM) algorithm

And, M step can be split into
M sub-step 1: Find

$$p^{(k+1)} = \text{argmax}_{p\in(0,1)} \sum_{i=1}^{n} \left(p_i^{(k)} \log(p) + (1 - p_i^{(k)}) \log(1 - p) \right)$$

M sub-step 2: Find

$$\lambda_1^{(k+1)} = \text{argmax}_{\lambda_1\in(0,\infty)} \sum_{i=1}^{n} \left(p_i^{(k)} \log(f(x_i|\lambda_1)) \right)$$

M sub-step 3: Find

$$\lambda_2^{(k+1)} = \text{argmax}_{\lambda_2\in(0,\infty)} \sum_{i=1}^{n} \left((1 - p_i^{(k)}) \log(f(x_i|\lambda_2)) \right)$$

The last two are weighted MLE's.

Expectation – Maximization (EM) algorithm

M sub-step 1: Gives

$$p^{(k+1)} = \frac{\sum_{i=1}^{n} p_i^{(k)}}{n}$$

M sub-step 2: Gives

$$\lambda_1^{(k+1)} = \frac{\sum_{i=1}^{n} p_i^{(k)}}{\sum_{i=1}^{n} p_i^{(k)} x_i}$$

M sub-step 3: Gives

$$\lambda_2^{(k+1)} = \frac{\sum_{i=1}^{n} (1 - p_i^{(k)})}{\sum_{i=1}^{n} (1 - p_i^{(k)}) x_i}$$

where

$$p_i^{(k)} := P(Z_i = 1 | X = x_i, \theta^{(k)}) = \frac{p^{(k)} \cdot f(x_i|\lambda_1^{(k)})}{p^{(k)} \cdot f(x_i|\lambda_1^{(k)}) + (1 - p^{(k)}) \cdot f(x_i|\lambda_2^{(k)})}$$

Expectation – Maximization (EM) algorithm

Example: Use a sample of $n = 1000$ generated from a mixture of $exp(\lambda_1 = .3)$ and $exp(\lambda_2 = .5)$ with probabilities $p = .2$ and $1 - p = .8$, respectively, to estimate p, λ_1 and λ_2.
You consider them as 1000 finishing time of 1000 transactions through two different cashiers that you have collected.
First, generate the 1000 points

```
n<-1000;p<-.2;lambda1<-.3;lambda2<-.5
lambda<-c(lambda1,lambda2)
K<-sample(1:2,n,prob=c(p,1-p),rep=T)
x<-rexp(n,rate=lambda[K])
```

Expectation – Maximization (EM) algorithm

```
TOL<-1e-8;j<-0
pold<-0.9;lambda1old<-0.1;lambda2old<-0.9
pnew<-.1;lambda1new<-1;lambda2new<-.1
vpnew<-pnew*dexp(x,lambda1new)/
(pnew*dexp(x,lambda1new)+(1-pnew)*dexp(x,lambda2new
while(max(abs(pnew-pold)/pold,
abs(lambda1new-lambda1old)/lambda1old,
abs(lambda2new-lambda2old)/lambda2old)>TOL){
 j<-j+1
 pold<-pnew
 lambda1old<-lambda1new
 lambda2old<-lambda2new
 vpold<-vpnew
 pnew<-mean(vpold)
 lambda1new<-1/weighted.mean(x, vpold)
 lambda2new<-1/weighted.mean(x, 1-vpold)
```

Expectation – Maximization (EM) algorithm

```
 vpnew<-(pnew*dexp(x,lambda1new))/
(pnew*dexp(x,lambda1new)+(1-pnew)*dexp(x,lambda2new
}
j
[1] 6074
pnew
[1] 0.8089904
lambda1new
[1] 0.512815
lambda2new
[1] 0.2730581
```

Why the switch? Look at the initial values of the parameters. Practical advice: Use different initial values of parameters λ_1 and λ_2 or p will not get updated ($p^{(k)} = p^{(0)}$ for all k).