Model Inference and Averaging

Thomas Lonon

Division of Financial Engineering
Stevens Institute of Technology

March 20, 2017

Denote training data $\mathbf{Z} = \{z_1, z_2, \ldots, z_N\}$, with $z_i = (x_i, y_i), i = 1, 2, \ldots, N$. Assume for now that the x's are one dimensional inputs and y_i is either categorial or continuous.

Lets fit a cubic spline with three knots placed at the quartiles. This will result in a seven-dimensional linear space

$$\mu(x) = \sum_{j=1}^{7} \beta_j h_j(x)$$

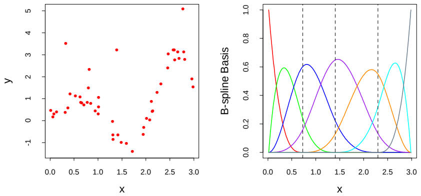

FIGURE 8.1. *(Left panel): Data for smoothing example. (Right panel:) Set of seven B-spline basis functions. The broken vertical lines indicate the placement of the three knots.*

Let **H** be the $N \times 7$ matrix with ij^{th} element $h_j(x_i)$. The estimate of β using squared error is given by:

$$\hat{\beta} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{y}$$

This is then used to determine:

$$\hat{\mu}(x) = \sum_{j=1}^{7} \hat{\beta}_j h_j(x)$$

with covariance matrix:

$$\hat{\mathbb{V}}(\hat{\beta}) = (\mathbf{H}^T\mathbf{H})^{-1}\hat{\sigma}^2$$

We are using here an estimate for the noise variance given by:

$$\hat{\sigma}^2 = \sum_{i=1}^{N} (y_i - \hat{\mu}(x_i))^2 / N$$

which is then used to determine the standard error of a prediction $\hat{\mu}(x) = h(x)^T \hat{\beta}$ as:

$$\hat{se}(\hat{\mu}(x)) = (h(x)^T (\mathbf{H}^T \mathbf{H})^{-1} h(x))^{\frac{1}{2}} \hat{\sigma}$$

To bootstrap, draw *B* datasets each of size *N* with replacement from the training data.

To each bootstrapped data set $\mathbf{Z}^*$, fit a cubic spline $\hat{\mu}^*(x)$. Using $B = 200$, we can determine a 95% confidence interval.

Nonparametric Bootstrap

Draw B datasets each of size N with replacement from our training data. To each dataset $\mathbf{Z}^*$ we fit a cubic spline $\hat{\mu}^*(x)$. Using these B bootstrap samples, we form a 95% confidence interval.

Parametric Bootstrap

Lets assume that the model errors are Gaussian, this leads us to:

$$Y = \mu(X) + \varepsilon; \varepsilon \sim N(0, \sigma^2),$$
$$\mu(x) = \sum_{j=1}^{7} \beta_j h_j(x)$$

We simulate new responses by adding Gaussian noise to the predicted values,

$$y_i^* = \hat{\mu}^*(x_i) + \varepsilon_i^*; \varepsilon_i^* \sim N(*0, \hat{\sigma}^2); i = 1, \ldots, N$$

The process is repeated B times and the the resulting bootstrap datasets, $(x_1, y_1^*), \ldots, (x_N, y_N^*)$ have the smoothing spline fit on each.

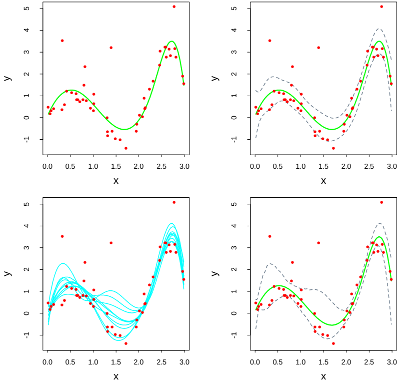

FIGURE 8.2. *(Top left:) B-spline smooth of data. (Top right:) B-spline smooth plus and minus* 1.96× *standard error bands. (Bottom left:) Ten bootstrap replicates of the B-spline smooth. (Bottom right:) B-spline smooth with* 95% *standard error bands computed from the bootstrap distribution.*

Begin by specifying a probability distribution for our observations

$$z_i \sim g_\theta(z)$$

This is called a *parametric model* for Z.

For example, if Z is normally distributed, then

$$\theta = (\mu, \sigma^2)$$

with

$$g_\theta(z) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(z-\mu)^2}{2\sigma^2}}$$

Maximum Likelihood Inference

This approach is based on the *likelihood function*

$$L(\theta; \mathbf{Z}) = \prod_{i=1}^{N} g_\theta(z_i)$$

Denote the logarithm of $L(\theta; \mathbf{Z})$:

$$\begin{aligned}\ell(\theta; \mathbf{Z}) &= \sum_{i=1}^{N} \ell(\theta; z_i) \\ &= \sum_{i=1}^{N} \log g_\theta(z_i)\end{aligned}$$

The *score function* is defined as:

$$\dot{\ell}(\theta, \mathbf{Z}) = \sum_{i=1}^{N} \frac{\partial \ell(\theta; z_i)}{\partial \theta}$$

The *information matrix* is defined as:

$$\mathbf{I}(\theta) = -\sum_{i=1}^{N} \frac{\partial^2 \ell(\theta; z_i)}{\partial \theta \partial \theta^T}$$

When $\mathbf{I}(\theta)$ is evaluated at $\theta = \hat{\theta}$, it is called *observed information*

The *Fisher information* is

$$\mathbf{i}(\theta) = \mathbb{E}_{\theta}[\mathbf{I}(\theta)]$$

and finally let θ_0 denote the true value of θ

The sampling distribution of the maximum likelihood estimator has limiting normal distribution

$$\hat{\theta} \to N(\theta_0, \mathbf{i}(\theta_0)^{-1})$$

as $N \to \infty$.

This done sampling independently from $g_{\theta_0}(z)$, which suggests that the sampling distribution can be approximated by

$$N(\hat{\theta}, \mathbf{i}(\hat{\theta})^{-1}) \text{ or } N(\hat{\theta}, \mathbf{I}(\hat{\theta})^{-1})$$

where $\hat{\theta}$ represents the maximum likelihood estimate from the observed data

Corresponding estimates for the standard errors are obtained from:

$$\sqrt{\mathbf{i}(\hat{\theta})^{-1}_{jj}} \text{ and } \sqrt{\mathbf{I}(\hat{\theta})^{-1}_{jj}}$$

The confidence points for θ_j can be constructed from these and have the form:

$$\hat{\theta}_j - z^{(1-\alpha)} * \sqrt{\mathbf{i}(\hat{\theta})^{-1}_{jj}} \text{ or } \hat{\theta}_j - z^{(1-\alpha)} * \sqrt{\mathbf{I}(\hat{\theta})^{-1}_{jj}}$$

where $z^{(1-\alpha)}$ is the $1-\alpha$ percentile of the standard normal distribution

Specify a sampling model $\mathbb{P}(\mathbf{Z}|\theta)$. Compute the posterior distribution:

$$\mathbb{P}(\theta|\mathbf{Z}) = \frac{\mathbb{P}(\mathbf{Z}|\theta) * \mathbb{P}(\theta)}{\int \mathbb{P}(\mathbf{Z}|\theta) * \mathbb{P}(\theta) d\theta}$$

The posterior distribution also provides the basis for predicting future observations z^{new}, via *predictive distribution*

$$\mathbb{P}(z^{\text{new}}|\mathbf{Z}) = \int \mathbb{P}(z^{\text{new}}|\theta) * \mathbb{P}(\theta|\mathbf{Z}) d\theta$$

Bayesian Approach to Smoothing Example

We start with the parametric model of the parametric bootstrap and assume that σ^2 is known and that the observed features $x_1, \ldots, x_N$ are fixed.

We will next need a prior distribution. By considering a basis for $\mu(x)$, we can instead provide a prior for the coefficients, β and implicitly define a prior for $\mu(x)$.

Choose a Gaussian prior centered at zero

$$\beta \sim N(0, \tau\Sigma)$$

This gives us the posterior distribution for β mean mean and covariance:

$$\mathbb{E}[\beta|\mathbf{Z}] = \left(\mathbf{H}^T\mathbf{H} + \frac{\sigma^2}{\tau}\Sigma^{-1}\right)^{-1}\mathbf{H}^T\mathbf{y}$$
$$\mathrm{cov}(\beta|\mathbf{Z}) = \left(\mathbf{H}^T\mathbf{H} + \frac{\sigma^2}{\tau}\Sigma^{-1}\right)^{-1}\sigma^2$$

This leads to the corresponding posterior values for $\mu(x)$ given by:

$$\mathbb{E}[\mu(x)|\mathbf{Z}] = h(x)^T \left(\mathbf{H}^T\mathbf{H} + \frac{\sigma^2}{\tau}\Sigma^{-1}\right)^{-1} \mathbf{H}^T\mathbf{y}$$

$$\text{cov}(\mu(x), \mu(x')|\mathbf{Z}) = h(x)^T \left(\mathbf{H}^T\mathbf{H} + \frac{\sigma^2}{\tau}\Sigma^{-1}\right)^{-1} h(x')\sigma^2$$

As $\tau \to \infty$, the distribution is called a *noninformative prior*

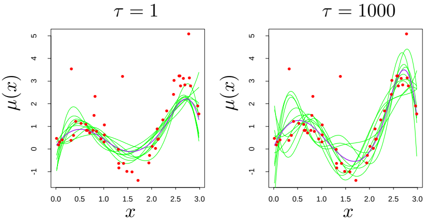

FIGURE 8.4. *Smoothing example: Ten draws from the posterior distribution for the function $\mu(x)$, for two different values of the prior variance τ. The purple curves are the posterior means.*

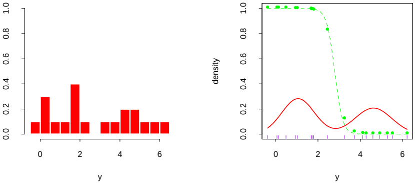

FIGURE 8.5. *Mixture example. (Left panel:) Histogram of data. (Right panel:) Maximum likelihood fit of Gaussian densities (solid red) and responsibility (dotted green) of the left component density for observation y, as a function of y.*

Simple Example

Because of the distribution, we can easily see that a normal distribution is a bad fit. Instead, we attempt to model this using a mixture of two normal distributions given by:

$$\begin{aligned} Y_1 &\sim N(\mu_1, \sigma_1^2) \\ Y_2 &\sim N(\mu_2, \sigma_2^2) \\ Y &= (1-\Delta)Y_1 + \Delta Y_2 \end{aligned}$$

where $\Delta \in \{0, 1\}$, with $\mathbb{P}(\Delta = 1) = \pi$. If we let $\phi_\theta(x)$ be the normal density with parameters θ, then the density of Y is:

$$g_Y(y) = (1-\pi)\phi_{\theta_1}(y) + \pi\phi_{\theta_2}(y)$$

If we want to fit this using MLE, we need to estimate the parameters:

$$\theta = (\pi, \theta_1, \theta_2) = (\pi, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2)$$

The log-likelihood based on the *N* cases is:

$$\ell(\theta; \mathbf{Z}) = \sum_{i=1}^{N} \log[(1-\pi)\phi_{\theta_1}(y_i) + \pi\phi_{\theta_2}(y_i)]$$

which is difficult to maximize

We can simplify it by introducing a new variable Δ_i (an unobserved latent variable) that takes values either 1 or 0. This signifies that if $\Delta_i = 1$, then Y_i comes from the second normal. With this, the log-likelihood function becomes:

$$\begin{aligned}\ell_0(\theta; \mathbf{Z}, \Delta) = &\sum_{i=1}^{N} [(1-\Delta_i)\log\phi_{\theta_1}(y_i) + \Delta_i \log\phi_{\theta_2}(y_i)] \\ &+ \sum_{i=1}^{N} [(1-\Delta_i)\log(1-\pi) + \Delta_i \log\pi]\end{aligned}$$

Since the values of the Δ_i's are unknown, we substitute for each Δ_i its expected value:

$$\gamma_i(\theta) = \mathbb{E}[\Delta_i|\theta, \mathbf{Z}] = \mathbb{P}(\Delta_i = 1|\theta, \mathbf{Z})$$

Algorithm 8.1: EM Algorithm for Two-component Gaussian Mixture

1. Take initial guesses for the parameters $\hat{\mu}_1$, $\hat{\sigma}_1^2$, $\hat{\mu}_2$, $\hat{\sigma}_2^2$, $\hat{\pi}$
2. *Expectation Step:* compute the responsibilites

$$\hat{\gamma}_i = \frac{\hat{\pi}\phi_{\hat{\theta}_2}(y_i)}{(1-\hat{\pi})\phi_{\hat{\theta}_1}(y_i) + \hat{\pi}\phi_{\hat{\theta}_2}(y_i)}, i = 1, \ldots, N$$

3. *Maximization Step:* compute the weighted means and variances.

$$\hat{\mu}_1 = \frac{\sum_{i=1}^{N}(1-\hat{\gamma}_i)y_i}{\sum_{i=1}^{N}(1-\hat{\gamma}_i)}, \qquad \hat{\sigma}_1^2 = \frac{\sum_{i=1}^{N}(1-\hat{\gamma}_i)(y_i-\hat{\mu}_1)^2}{\sum_{i=1}^{N}(1-\hat{\gamma}_i)}$$

$$\hat{\mu}_2 = \frac{\sum_{i=1}^{N}\hat{\gamma}_i y_i}{\sum_{i=1}^{N}\hat{\gamma}_i}, \qquad \hat{\sigma}_1^2 = \frac{\sum_{i=1}^{N}\hat{\gamma}_i(y_i-\hat{\mu}_2)^2}{\sum_{i=1}^{N}\hat{\gamma}_i}$$

   and the mixing probability $\hat{\pi} = \sum_{i=1}^{N}\hat{\gamma}_i/N$

4. Iterate steps 2 and 3 until convergence.[1]

Algorithm 8.2: The EM Algorithm

1. Start with initial guesses for the parameters $\hat{\theta}^{(0)}$.
2. *Expectation Step:* at the j^{th} step, compute

   $$Q(\theta', \hat{\theta}^{(j)}) = \mathbb{E}[\ell_0(\theta'; \mathbf{T})|\mathbf{Z}, \hat{\theta}^{(j)}]$$

   as a function of the dummy argument θ'
3. *Maximization Step:* determine the new estimate $\hat{\theta}^{(j+1)}$ as the maximizer of $Q(\theta', \hat{\theta}^{(j)})$ over θ'
4. Iterate steps 2 and 3 until convergence.

[1]

The Expectation-Maximization Algorithm

We introduce new unknown random variables (**Y**) and use them to create a simpler expression of the likelihood.

$$p(\mathbf{X}, \mathbf{Y}|\Theta) = p(\mathbf{Y}|\mathbf{X}, \Theta)\frac{p(\mathbf{X}, \mathbf{Y}|\Theta)}{p(\mathbf{Y}|\mathbf{X}, \Theta)} \tag{1}$$

- E-Step: $P^{(t)}(y) = P(y|x, \Theta^{(t)})$
- M-Step: $\Theta^{(t+1)} = \underset{\Theta}{\operatorname{argmax}} \left(\mathbb{E}_{P^{(t)}}[\ln P(y, x|\Theta)]\right)$

For a mixture of normals we have the lower bound:

$$\lambda(X,\Theta) \geq \sum_{i=1}^{N}\sum_{j=1}^{M} p^{(t)}(j|x_i,\Theta^{(t)}) \ln \frac{p_j g(x_i;\mu_j,\sigma_j^2)}{p^{(t)}(j|x_i,\Theta^{(t)})} = b_t$$

where $g\left(x_i;\mu_j^{(t)},\sigma_j^{2(t)}\right)$ denotes the Gaussian pdf.

Our Expectation Step is expressed as:

$$p^{(t)}(j|x_i,\Theta^{(t)}) = \frac{p_j^{(t)} g(x_i;\mu_j^{(t)},\sigma_j^{2(t)})}{\sum_{j=1}^{M} p_j^{(t)} g(x_i;\mu_j^{(t)},\sigma_j^{2(t)})}$$

Since b_t is a lower bound for the log-likelihood, if we maximize b_t we will improve the log-likelihood as well. Looking at b_t we can see:

$$b_t = \sum_{i=1}^{N}\sum_{j=1}^{M} p^{(t)}(j|x_i,\Theta^{(t)}) \ln p_j g(x_i;\mu_j,\sigma_j^2) - \sum_{i=1}^{N}\sum_{j=1}^{M} p^{(t)}(j|x_i,\Theta^{(t)}) \ln p^{(t)}(j|x_i,\Theta^{(t)})$$

$$\hat{\Theta} = \Theta^{(t+1)} = \underset{\Theta}{\operatorname{argmax}} \sum_{i=1}^{N}\sum_{j=1}^{M} p^{(t)}(j|x_i,\Theta^{(t)}) \ln p_j g(x_i;\mu_j,\sigma_j^2) \quad (2)$$

To ease writing out formulas we will define the function

$$q(j,i) = p_j g(x_i; \mu_j, \sigma_j^2)$$

This function has the following partial derivatives with respect to the parameters,

$$\frac{\partial q}{\partial \mu_j} = q(j,i)\left(\frac{x_i - \mu_j}{\sigma_j^2}\right)$$
$$\frac{\partial q}{\partial \sigma_j} = q(j,i)\left(\frac{(x_i - \mu_j)^2 - \sigma_j^2}{\sigma_j^3}\right)$$
$$\frac{\partial q}{\partial p_j} = g(x_i; \mu_j, \sigma_j^2)$$

As the term will appear often, we will substitute $p^{(t)}(j|x_i, \Theta^{(t)})$ with the simpler but less descriptive expression $p(j|i)$

$$\frac{\partial b_t}{\partial \mu_j} = \sum_{i=1}^{N} p(j|i) \frac{1}{q(j,i)} q(j,i) \left(\frac{x_i - \mu_j}{\sigma_j^2} \right)$$

$$0 = \sum_{i=1}^{N} p(j|i) \left(\frac{x_i - \mu_j}{\sigma_j^2} \right)$$

$$\mu_j^{(t+1)} = \frac{\sum_{i=1}^{N} p(j|i) x_i}{\sum_{i=1}^{N} p(j|i)}$$

Using this estimate for the value of μ_j for the next iteration we have:

$$\frac{\partial b_t}{\partial \sigma_j} = \sum_{i=1}^{N} p(j|i) \frac{1}{q(j,i)} q(j,i) \left(\frac{(x_i - \mu_j)^2 - \sigma_j^2}{\sigma_j^3} \right)$$

$$0 = \sum_{i=1}^{N} p(j|i) \left(\frac{(x_i - \mu_j)^2 - \sigma_j^2}{\sigma_j^3} \right)$$

$$\sigma_j^{(t+1)} = \sqrt{\frac{\sum_{i=1}^{N} p(j|i)(x_i - \mu_j^{(t+1)})^2}{\sum_{i=1}^{N} p(j|i)}}$$

The final step is to look at the partials with respect to the mixing probabilities:

$$\frac{\partial b_t}{\partial p_j} = \sum_{i=1}^{N} p(j|i) \frac{1}{q(j,i)} g(x_i; \mu_j, \sigma_j^2)$$

$$0 = \sum_{i=1}^{N} \frac{p(j|i)}{p_j}$$

This is a very problematic condition, as the only way this is equal to 0, is if each of the observed conditional probabilities are all equal to 0.

The reason for this difficulty lies in the fact that we have not enforced the constraint $\sum_{j=1}^{M} p_j = 1$. To this end we can express the probabilities through another set of variables $(\gamma_1, \ldots, \gamma_M)$ using a softmax function to ensure these conditions are met.

$$p_k = \frac{e^{\gamma_k}}{\sum_{j=1}^{M} e^{\gamma_j}} \tag{3}$$

We can now take the partial derivatives of our lower bound function with respect to these variables using

$$\frac{\partial p_j}{\partial \gamma_k} = \begin{cases} p_k - p_k^2 : k = j \\ -p_k p_j : k \neq j \end{cases}$$

Now, the partial derivatives with respect to γ_k are:

$$\begin{aligned}
\frac{\partial b_t}{\partial \gamma_k} &= \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq k}}^{M} p(j|i) \frac{1}{q(j,i)} g(x_i; \mu_j, \sigma_j^2)(-p_k p_j) + \sum_{i=1}^{N} p(k|i) \frac{1}{q(k,i)} g(x_i; \mu_k, \sigma_k^2)(p_k - p_k^2) \\
0 &= \sum_{i=1}^{N} p(k|i) \frac{1}{q(k,i)} g(x_i; \mu_k, \sigma_k^2)(p_k) - \sum_{i=1}^{N} \sum_{j=1}^{M} p(j|i) \frac{1}{q(j,i)} g(x_i; \mu_j, \sigma_j^2)(p_k p_j) \\
0 &= \sum_{i=1}^{N} p(k|i) - N p_k \\
p_k^{(t+1)} &= \frac{\sum_{i=1}^{N} p(k|i)}{N}
\end{aligned}$$

for $k \in \{1, \ldots, M\}$.

1. Determine initial estimates for the parameters $\Theta^{(0)}$.
2. E-Step: Calculate the membership probabilities based on current parameter estimates
$$p^{(t)}(j|x_i, \Theta^{(t)}) = \frac{p_j^{(t)} g(x_i; \mu_j^{(t)}, \sigma_j^{2(t)})}{\sum_{j=1}^{M} p_j^{(t)} g(x_i; \mu_j^{(t)}, \sigma_j^{2(t)})} = p(j|i) \tag{4}$$
3. M-Step: Calculate improved estimates for the parameters based on these membership probabilities
$$\mu_j^{(t+1)} = \frac{\sum_{i=1}^{N} p(j|i) x_i}{\sum_{i=1}^{N} p(j|i)} \tag{5}$$
$$\sigma_j^{(t+1)} = \sqrt{\frac{\sum_{i=1}^{N} p(j|i)(x_i - \mu_j^{(t+1)})^2}{\sum_{i=1}^{N} p(j|i)}} \tag{6}$$
$$p_j^{(t+1)} = \frac{\sum_{i=1}^{N} p(j|i)}{N} \tag{7}$$
4. Check if condition for stopping is satisfied. We use for our condition $(\hat{\Theta}^{(t+1)} - \hat{\Theta}^{(t)})^2 < \varepsilon$
5. If the condition is not met, repeat steps 2 and 3 with $\hat{\Theta}^{(t+1)} \rightarrow \hat{\Theta}^{(t)}$

MCMC for Sampling

The *Markov Chain Monte Carlo*(MCMC) approach to posterior sampling will next be discussed.

Algorithm 8.3: Gibbs Sampler

```
1. Take some initial values $U_k^{(0)}, k = 1, 2, \ldots, K$
2. Repeat for $t = 1, 2, \ldots$:
   For $k = 1, 2, \ldots, K$ generate $U_k^{(t)}$ from

   $$\mathbb{P}(U_k^{(t)} | U_1^{(t)}, U_2^{(t)}, \ldots, U_{k-1}^{(t)}, U_{k+1}^{(t-1)}, \ldots, U_K^{(t-1)}$$

3. Continue step 2 until the joint distribution of
   $(U_1^{(t)}, U_2^{(t)}, \ldots, U_K^{(t)})$ *does not change*
```

Algorithm 8.4: Gibbs sampling for mixtures

1. Take some initial values $\theta^{(0)} = (\mu_1^{(0)}, \mu_2^{(0)})$
2. Repeat for $t = 1, 2, \ldots$:
    - 2.1 For $i = 1, 2, \ldots, N$ generate $\Delta_i^{(t)} \in \{0, 1\}$ with $\mathbb{P}(\Delta_i^{(t)} = 1) = \hat{\gamma}_i(\theta^{(t)})$, from Algorithm 8.1
    - 2.2 Set

$$\hat{\mu}_1 = \frac{\sum_{i=1}^{N}(1 - \Delta_i^{(t)})y_i}{\sum_{i=1}^{N}(1 - \Delta_i^{(t)})}$$

$$\hat{\mu}_2 = \frac{\sum_{i=1}^{N} \Delta_i^{(t)} y_i}{\sum_{i=1}^{N} \Delta_i^{(t)}}$$

and generate $\mu_1^{(t)} \sim N(\hat{\mu}_1, \hat{\sigma}_1^2)$ and $\mu_2^{(t)} \sim N(\hat{\mu}_2, \hat{\sigma}_2^2)$

3. Continue step 2 until the joint distribution of $(\Delta^{(t)}, \mu_1^{(t)}, \mu_2^{(t)})$ doesn't change.

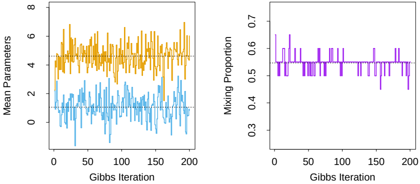

FIGURE 8.8. *Mixture example. (Left panel:)* 200 *values of the two mean parameters from Gibbs sampling; horizontal lines are drawn at the maximum likelihood estimates* $\hat{\mu}_1$, $\hat{\mu}_2$. *(Right panel:) Proportion of values with* $\Delta_i = 1$, *for each of the* 200 *Gibbs sampling iterations; a horizontal line is drawn at* $\sum_i \hat{\gamma}_i/N$.

Bagging

Suppose we fit a model to our training data $\mathbf{Z} = \{(x_1, y_x), (x_2, y_2), \ldots, (x_N, y_N)\}$, obtaining the prediction $\hat{f}(x)$ at input x.

Bagging or bootstrap aggregation averages this prediction over a collection of bootstrap samples.

For each bootstrap sample $\mathbf{Z}^{*b}, b = 1, 2, \ldots, B$, we fit our model giving prediction $\hat{f}^{*b}(x)$. The bagging estimate is defined by

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

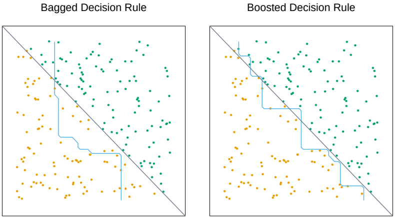

FIGURE 8.12. *Data with two features and two classes, separated by a linear boundary. (Left panel:) Decision boundary estimated from bagging the decision rule from a single split, axis-oriented classifier. (Right panel:) Decision boundary from boosting the decision rule of the same classifier. The test error rates are* 0.166*, and* 0.065*, respectively. Boosting is described in Chapter 10.*

Bumping

Similar to bagging, but instead of averaging the predictions, we choose the best one.

Draw bootstrap samples $\mathbf{Z}^{*1}, \ldots, \mathbf{Z}^{*B}$ and fit the model to each, giving predictions $\hat{f}^{*b}(x)$, then choose the model that produces the smallest prediction error over the original training set.

$$\hat{b} = \arg\min_{b} \sum_{i=1}^{N} [y_i - \hat{f}^{*b}(x_i)]^2$$

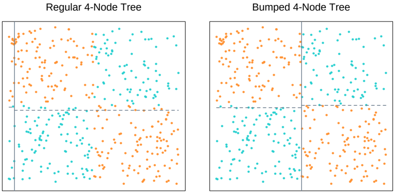

FIGURE 8.13. *Data with two features and two classes (blue and orange), displaying a pure interaction. The left panel shows the partition found by three splits of a standard, greedy, tree-growing algorithm. The vertical grey line near the left edge is the first split, and the broken lines are the two subsequent splits. The algorithm has no idea where to make a good initial split, and makes a poor choice. The right panel shows the near-optimal splits found by bumping the tree-growing algorithm* 20 *times.*

[1] Robert Tibshirani Trevor Hastie and Jerome Friedman. *The Elements of Stastical Learning: Data Mining, Inference, and Prediction*. Number v.2 in Springer Series in Statistics. Springer, 2009.