Looking for Lumps:

boosting and bagging for density estimation

Greg Ridgeway

RAND Statistics Group

Density estimation

- We observe a dataset of n iid d-dimensional observations, $x_1, \ldots, x_n$ drawn from an unknown density $f(x)$.

- The problem is to produce an estimate, $\widehat{f}(x)$, of $f(x)$ from the dataset alone.

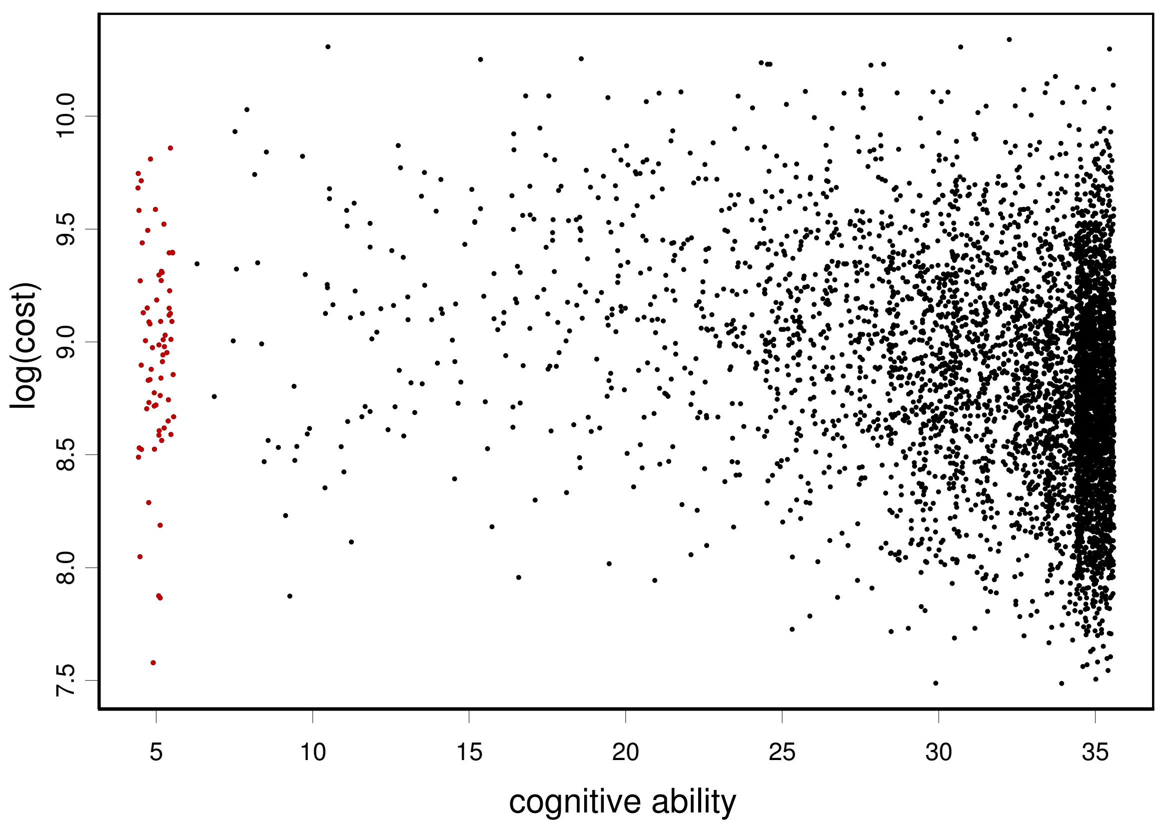
- We can assess the quality of the estimate using the expected log-likelihood

$$J(\widehat{f}\,) = \mathsf{E}_x \log \widehat{f}(x).$$

- Indeed the true density, $f(x)$, maximizes $J(\widehat{f}\,)$.

Data mining and density estimation

- Density estimation is the first step in detecting lumps (as well as holes) in the data.



Data mining and density estimation

- Clustering or segmentation often involves interpreting a mixture density.
- There are currently few reported high-dimensional density estimators.
- This presentation will also demonstrate the utility and flexibility of bagging and boosting variations for creating practical algorithms for modeling massive datasets.

With training data

We do not know $f(x)$ but we can approximate.

- Generalization error

$$J(\widehat{f}\,) \;=\; \mathsf{E}_x \log \widehat{f}(x) \approx$$

- Training error

$$\widehat{J}(\widehat{f}\,) \;=\; \frac{1}{n}\sum_{i=1}^{n} \log \widehat{f}(x_i)$$

Without constraints, the $\widehat{f}(x)$ that maximizes the training error puts point mass on the observed x_i.

Boosting for classification and regression

The general boosting strategy says

1. Initialize $\widehat{f}(x) = c$ where c minimizes $\widehat{J}(c)$.

2. Find an improvement, $g(x)$, to $\widehat{f}(x)$ such that
$$\widehat{J}(\widehat{f} + g) < \widehat{J}(\widehat{f}\,).$$

3. Adjust the predictor using a line search in the direction of $g(x)$.
$$\widehat{f}(x) \leftarrow \widehat{f}(x) + \lambda g(x)$$

Relationship to boosting

Although we are dealing with density estimation, the problem is akin to previous applications of boosting.

- There is a functional measuring generalization error or model fit, J.
- We want to find a function that maximizes (or minimizes) the objective function, J.
- We cannot compute J explicitly but can only approximate it with our sample.

An algorithm for boosted density estimation

1. Let $\widehat{f}(x)$ be an initial, naïve guess for the density. For example

$$\widehat{f}(x) = \varphi(x; \bar{x}, \mathbf{S})$$

where $\bar{x}$ is the sample mean and $\mathbf{S}$ is an estimate of the covariance.

2. Mix $\widehat{f}(x)$ with another density, $g(x)$, so that

$$\widehat{J}((1-\alpha)\widehat{f} + \alpha g) > \widehat{J}(\widehat{f})$$

The multivariate uniform and normal might be good candidates for $g(x)$.

3. Update the density estimate.

$$\widehat{f}(x) \leftarrow (1-\alpha)\widehat{f}(x) + \alpha g(x)$$

Selecting a normal proposal

Find a normal density $\varphi(x; \mu, \Sigma)$, so that

$$\hat{J}((1-\alpha)\hat{f} + \alpha\varphi) > \hat{J}(\hat{f})$$

The normal requires $O(d^2)$ parameters but finding them can be fast and easy to program.

At each iteration assume that each x_i either comes from

- $\hat{f}(x)$ with probability $(1-\alpha)$ or from
- $\varphi(x; \mu, \Sigma)$ with probability α.

This is the setting for a two component mixture model (except one component is fixed). The EM algorithm can find a good choice for (μ, Σ, α).

Likelihood optimization using EM

- Initialize $(\mu, \boldsymbol{\Sigma}, \alpha)$.
  - $\mu =$ randomly sampled x_i
  - $\Sigma = \mathrm{S}$, the sample covariance
  - α = 0.2
- E-step

$$p_i = \frac{\alpha\varphi(x_i; \mu, \boldsymbol{\Sigma})}{(1-\alpha)\widehat{f}(x_i) + \alpha\varphi(x_i; \mu, \boldsymbol{\Sigma})}$$

- M-step: Find $(\mu, \boldsymbol{\Sigma}, \alpha)$ that maximize the expected complete data log-likelihood.

$$\begin{aligned} \mu &\leftarrow \frac{\sum p_i x_i}{\sum p_i} \\ \boldsymbol{\Sigma} &\leftarrow \frac{\sum p_i (x_i - \mu)(x_i - \mu)^T}{\sum p_i} \\ \alpha &\leftarrow \frac{\sum p_i}{N} \end{aligned}$$

The algorithm

Set $\widehat{f}(x) = \varphi(x; \bar{x}, \mathbf{S})$
While stopping criterion is not satisfied
$\{$

1. Initialize the EM algorithm
   $\alpha = 0.2$,
   $\mu =$ randomly selected x_i,
   $\mathbf{\Sigma} =$ sample covariance of the x_i's
2. Iterate the EM algorithm to convergence
   (a) $p_i = \frac{\alpha\varphi(x_i;\mu,\mathbf{\Sigma})}{(1-\alpha)f(x_i)+\alpha\varphi(x_i;\mu,\mathbf{\Sigma})}$
   (b) $\mu =$ weighted mean of the x_i's
   (c) $\mathbf{\Sigma} =$ weighted covariance of the x_i's
   (d) $\alpha =$ mean of the p_i's
3. Update the density estimate as
   $\widehat{f}(x) \leftarrow (1-\alpha)\widehat{f}(x) + \alpha\varphi(x_i; \mu, \mathbf{\Sigma})$.

$\}$

Stochastic boosting

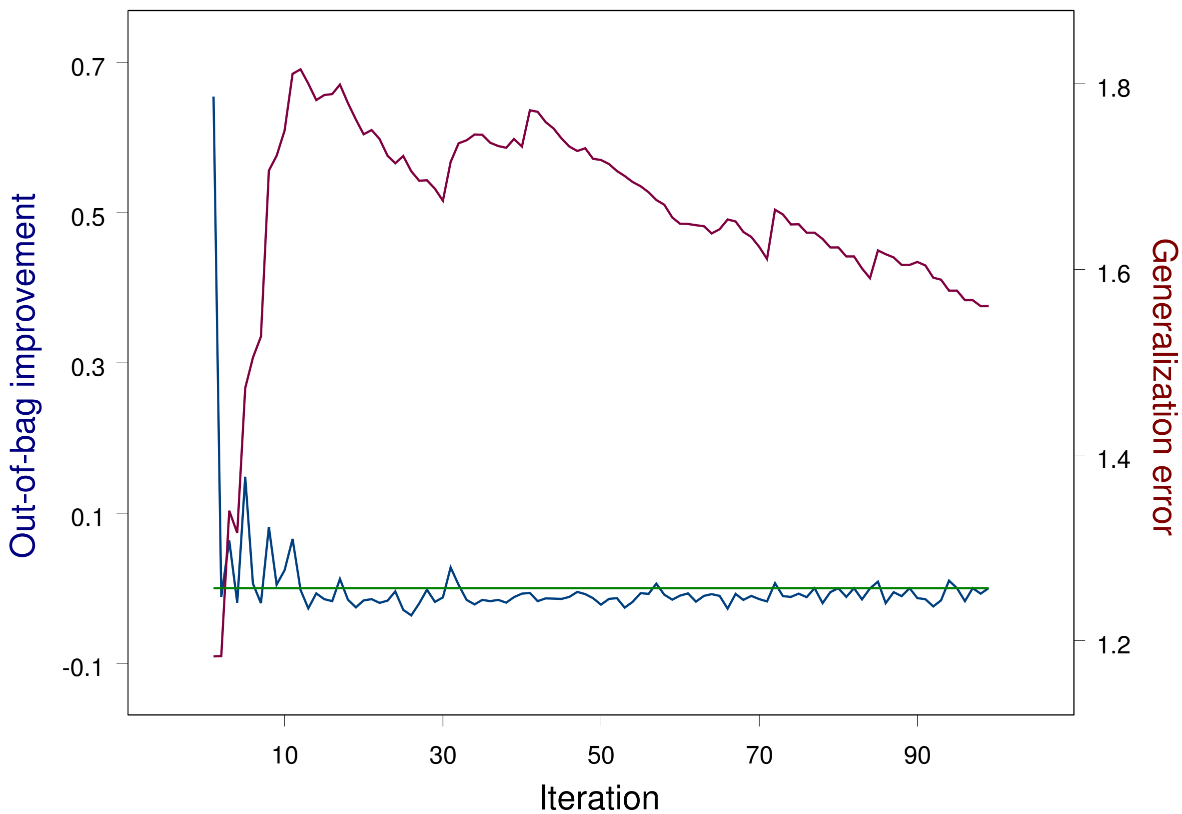
Variance reduction and automatic stopping

- Sending 50% of the observations to the EM algorithm helps to reduce overfitting.
- It also preserves a set of observations useful in determining whether the proposed addition actually offers an improvement.

With the data not used by the EM algorithm compute

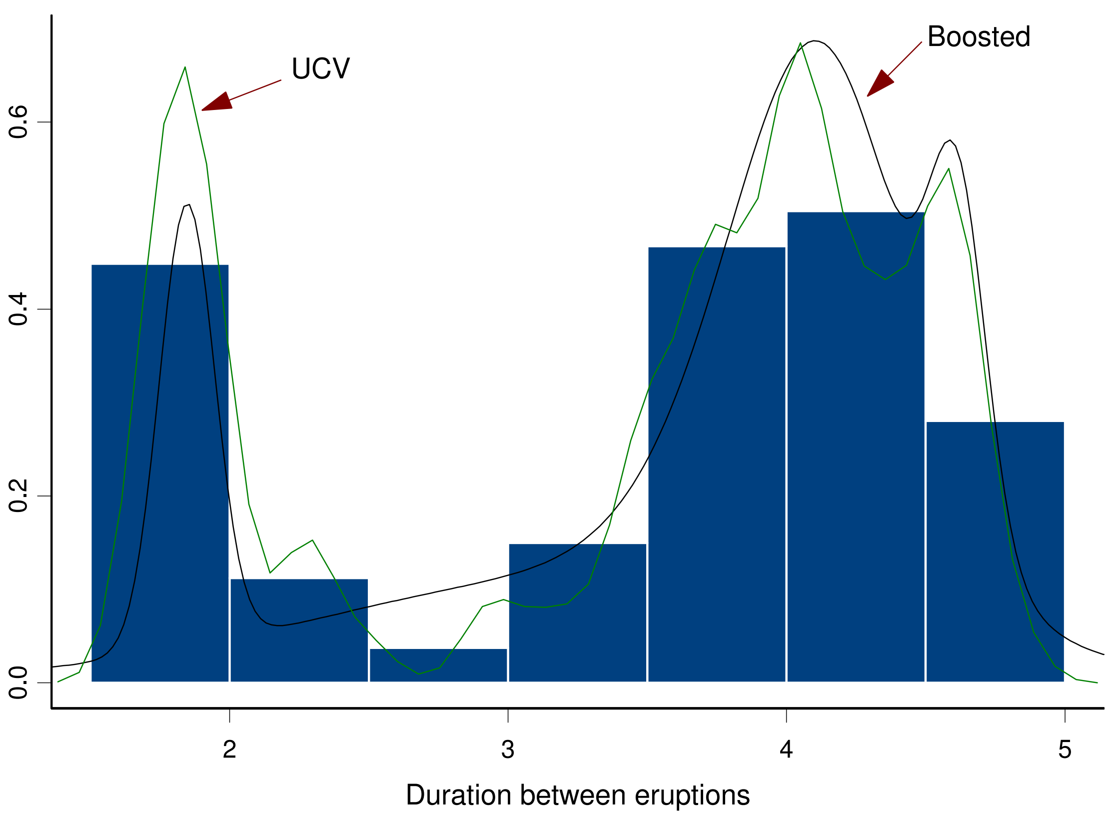
$$\begin{aligned}\Delta J &= \sum_{i \in \text{out-of-bag}} \log\big((1-\alpha)\hat{f}(x_i) + \alpha\varphi(x_i;\mu,\Sigma)\big) - \log \hat{f}(x_i) \\ &= \sum_{i \in \text{out-of-bag}} \log\left((1-\alpha) + \alpha\frac{\varphi(x_i;\mu,\Sigma)}{\hat{f}(x_i)}\right)\end{aligned}$$

Out-of-bag gradient and generalization error



Example 1: Old faithful data

Scott (1992) presents data on the duration of 107 eruptions of the Old Faithful geyser.



Density	$J(\widehat{f})$
Unbiased cross-validation	-1.01
Boosted density estimate	-1.07

Example 2: Simulated mixture distributions

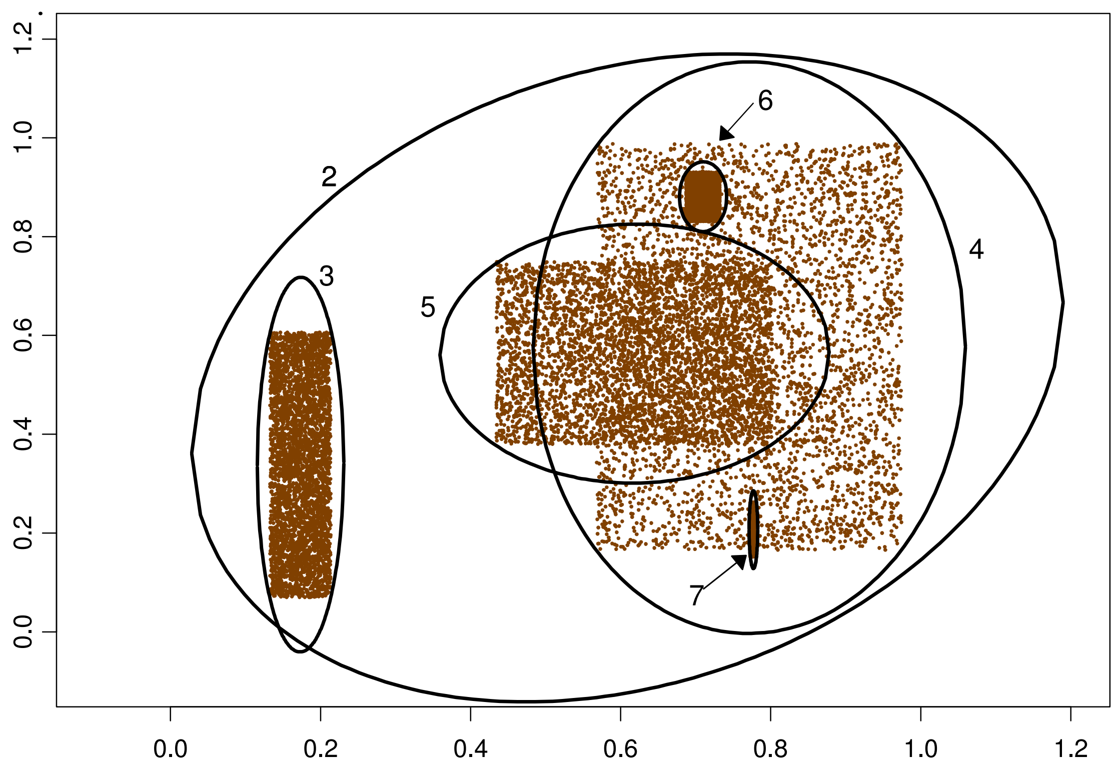
- Multivariate normal, $N = 100,000$, $d = 20$, mixture of 5 equally weighted normal components with similar location but random covariance.

Density	$J(\widehat{f}\,)$
Knowing true structure	-88.40
Boosted density estimate	-89.01

- Multivariate uniform, $N = 100,000$, $d = 20$, mixture of 5 equally weighted components. Each component was a random box in $[0,1]^{20}$.

Density	$J(\widehat{f}\,)$
True density	28.5
Boosted density estimate	23.9

Projection of the multivariate uniform



95% contours ordered by when they entered the mixture. The first component's contour is outside of the picture.

Some interesting algorithmic features

- Point mass proposals
  - The first several iterations capture the largest lumps.
  - At some point the EM algorithm starts proposing point masses.
  - I rejected the proposal when the smallest eigenvalue became too close to machine precision.
- Slowing the learning rate
  - Surprisingly, shrinking α toward 0 seemed to decrease the algorithm's performance.

Data mining applications

- Density estimates reveal lumps in the dataset. Few tools exist for density estimation in dimensions greater than 4.

- Sending a subsample to the EM algorithm eases computation.
  - It reduces the size of the dataset for the especially intensive calculations.
  - We could use an even smaller, stratified sample.

- Substitute the multivariate normal with a multivariate uniform to get more interpretable, box shaped regions.