+

Recommender Systems Ensemble Methods

Professor Robin Burke
Spring 2019

Thanks to Yisong Yue of Disney Research for some material in these slides

Ensemble Methods

- Standard idea in machine learning
  - Can be applied to recommendation
- Multiple weak learners can be combined
  - To improve performance
- Why does this work?
  - Bias / variance

Supervised Learning

- **Goal:** learn predictor h(x)
  - High accuracy (low error)
  - Using training data $\{(x_1,y_1),\ldots,(x_n,y_n)\}$

+ Generalization Error

- **"True" distribution:** $P(x,y)$
  - Unknown to us

- **Train:** $h(x) = y$
  - Using training data $S = \{(x_1,y_1),\ldots,(x_n,y_n)\}$
  - Sampled from $P(x,y)$

- **Generalization Error:**
  - $\mathcal{L}(h) = E_{(x,y)\sim P(x,y)}[\ f(h(x),y)\ ]$
  - E.g., $f(a,b) = (a-b)^2$

Bias/Variance Tradeoff

- Treat $h(x|S)$ has a random function
  - Depends on training data S
- $\mathcal{L} = E_S[\ E_{(x,y)\sim P(x,y)}[\ f(h(x|S),y)\ ]\ ]$
  - Expected generalization error
  - Over the randomness of S

Bias/Variance Tradeoff

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

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

Expected Error

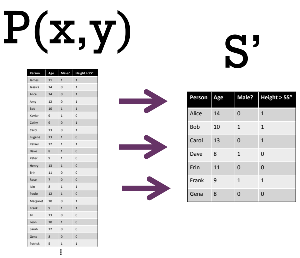
$$\begin{aligned} E_S[f(h(x|S),y)] &= E_S[Z^2] \\ &= \underbrace{E_S[(Z-\check{z})^2]}_{\textbf{Variance}} + \underbrace{\check{z}^2}_{\textbf{Bias}} \end{aligned}$$

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.)

+ Bagging



- **Goal:** reduce variance

- **Ideal setting:** many training sets S'
  - Train model using each S'
  - Average predictions

sampled independently

Variance reduces linearly
Bias unchanged

$$\underbrace{E_S[(h(x|S) - y)^2]}_{\text{Expected Error}} = \underbrace{E_S[(Z-\check{z})^2]}_{\textbf{Variance}} + \underbrace{\check{z}^2}_{\textbf{Bias}}$$

$$Z = h(x|S) - y$$
$$\check{z} = E_S[Z]$$

"Bagging Predictors" [Leo Breiman, 1994]
http://statistics.berkeley.edu/sites/default/files/tech-reports/421.pdf

+ Bagging

- **Goal:** reduce variance

from S

- **In practice:** resample S' with replacement
  - Train model using each S'
  - Average predictions

Variance reduces sub-linearly
(Because S' are correlated)
Bias often increases slightly

$$\underbrace{E_S[(h(x|S) - y)^2]}_{\text{Expected Error}} = \underbrace{E_S[(Z-\check{z})^2]}_{\textbf{Variance}} + \underbrace{\check{z}^2}_{\textbf{Bias}}$$

$Z = h(x|S) - y$

$\check{z} = E_S[Z]$

Bagging = Bootstrap Aggregation

"Bagging Predictors" [Leo Breiman, 1994]

http://statistics.berkeley.edu/sites/default/files/tech-reports/421.pdf

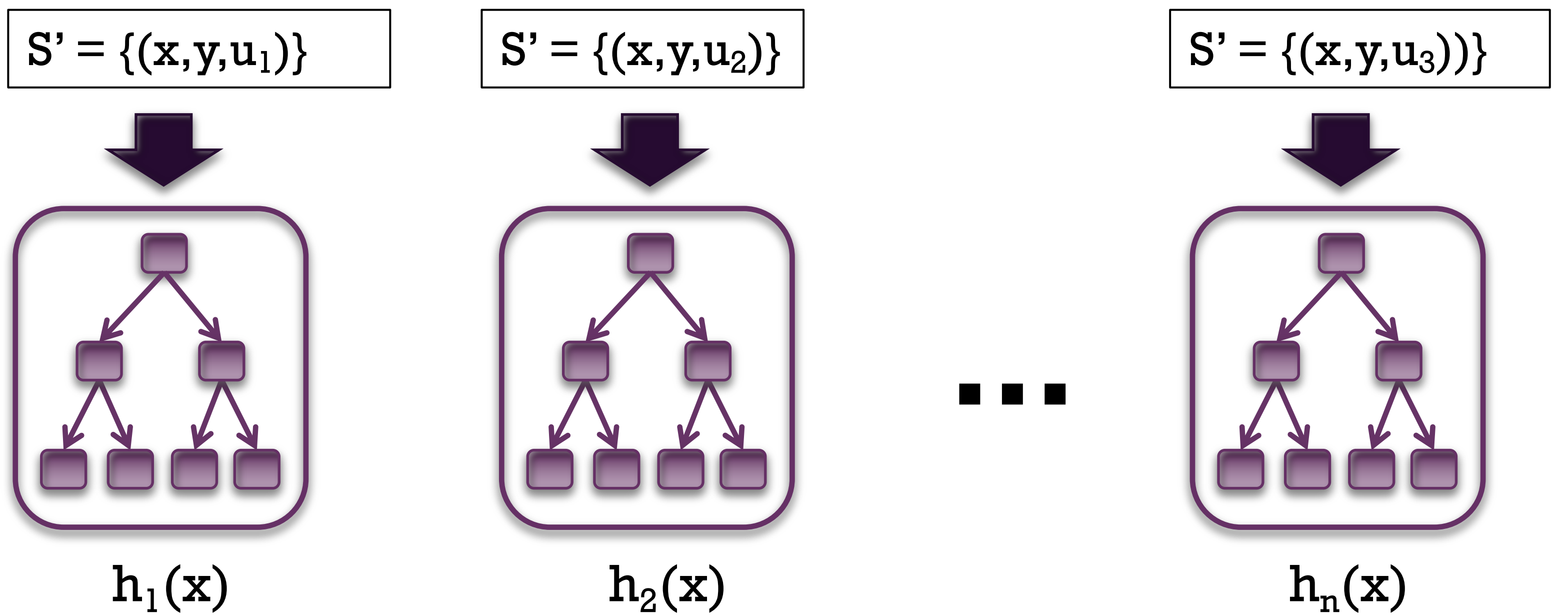
Application to recommendation

- Row-wise bootstrapping
  - Build a new ratings matrix
    - By sampling with replacement
  - Build a recommender for each one
  - Average the results
- Note that rows (users) may appear multiple times
  - Can treat these as "weighted" users
  - Recommendation algorithm must be able to accommodate
  - For example
    - Weighted errors in the loss function

$$\text{Minimize } J = \frac{1}{2} \sum_{(i,j) \in S} w_{ij} e_{ij}^2 + \frac{\lambda}{2} \sum_{i=1}^{m} \sum_{s=1}^{k} u_{is}^2 + \frac{\lambda}{2} \sum_{j=1}^{n} \sum_{s=1}^{k} v_{js}^2$$

+ Boosting (AdaBoost)

$$h(x) = a_1h_1(x) + a_2h_2(x) + \ldots + a_3h_n(x)$$



u – weighting on data points

a – weight of linear combination

Stop when validation performance plateaus

https://www.cs.princeton.edu/~schapire/papers/explaining-adaboost.pdf

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in \mathscr{X}$, $y_i \in \{-1, +1\}$.
Initialize $\boxed{D_1(i) = 1/m \text{ for } i = 1, \ldots, m.}$ ← Initial Distribution of Weights
For $t = 1, \ldots, T$:

- Train weak learner using distribution D_t.
- Get weak hypothesis $h_t : \mathscr{X} \to \{-1, +1\}$. ← Train model
- Aim: select h_t with low weighted error:

$$\boxed{\varepsilon_t = \Pr_{i \sim D_t}[h_t(x_i) \neq y_i].}$$ ← Error of model

- Choose $\boxed{\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \varepsilon_t}{\varepsilon_t} \right).}$ ← Coefficient of model
- Update, for $i = 1, \ldots, m$:

$$\boxed{D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}}$$ ← Update Distribution

where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

Output the final hypothesis:

$$\boxed{H(x) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x) \right).}$$ ← Final average

Theorem: training error drops exponentially fast

https://www.cs.princeton.edu/~schapire/papers/explaining-adaboost.pdf

Known problem

- Boosting works badly with noisy data
- Algorithm works very hard to classify the noisy parts
  - overfitting

Application in recommendation

- Assume weighted algorithm
- Weight the rows (users) and update the weights based on the errors

Ensemble Methods

- Have a theoretical basis
  - Probabilistic properties of the training data
- These only apply if there is just one set of training data
  - One knowledge source

More generally

- Any collection of recommendation algorithms can be put into a weighted ensemble

$$\hat{R} = \sum_{i=1}^{q} \alpha_i \hat{R}_i$$

- Note this is effectively combining across individual predictions

$$\hat{r}_{uj} = \sum_{i=1}^{q} \alpha_i \hat{r}^{i}_{uj}$$

Homework 3

- We can turn this into a regression model
  - Where the α values are to be learned
- But sparsity, etc.
  - May want to use gradient descent as with matrix factorization
  - This was the NetFlix winner with many different MF components

The difference between bagging and boosting is

- A. Bagging uses multiple trained predictors and boosting just uses one
- B. Bagging uses multiple samples of the training data and boosting just uses one
- C. Bagging builds components on concentrate on parts of the data where the prior components make more errors
- D. B and C