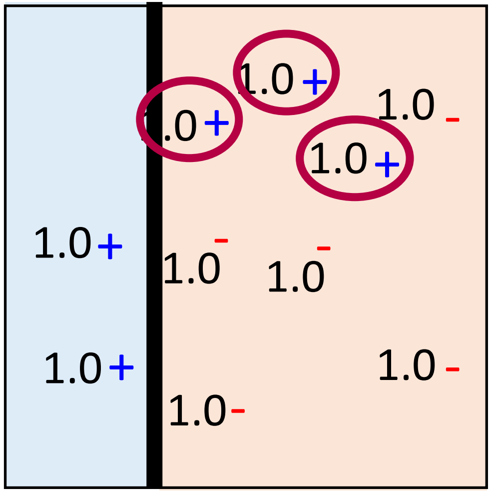
Ensemble: Boosting

Improving Weak Classifiers: Part II

Toy Example: Part I

t=1

T=3, weak classifier = decision stump

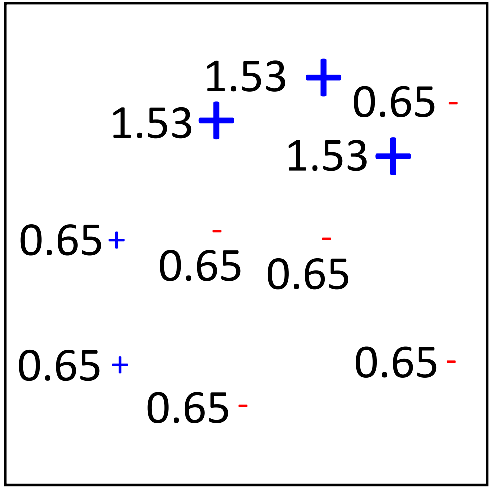


$f_1(x)$

$$\varepsilon_1 = 0.30$$

$$d_1 = 1.53$$

$$\alpha_1 = 0.42$$

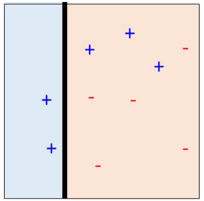


Toy Example: Part II

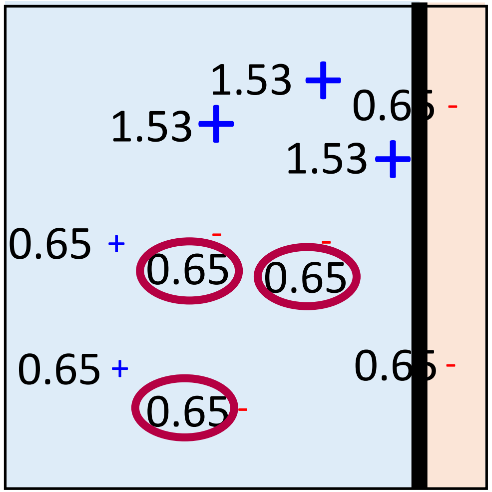
t=2

T=3, weak classifier = decision stump

$f_1(x)$:



$\alpha_1 = 0.42$

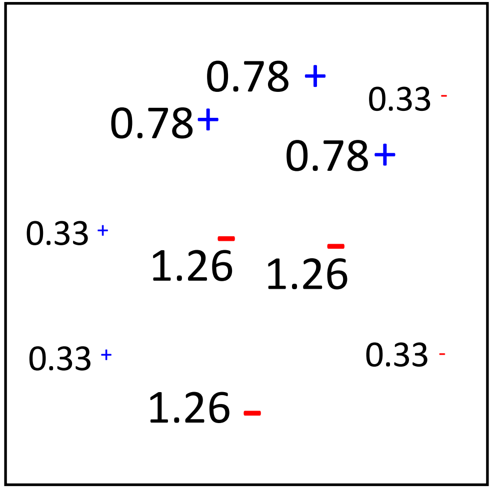


$f_2(x)$

$\varepsilon_2 = 0.21$

$d_2 = 1.94$

$\alpha_2 = 0.66$



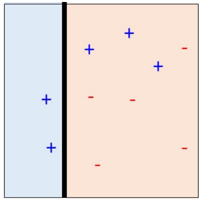
Toy Example: Part III

T=3, weak classifier = decision stump

t=3

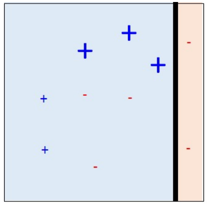
$f_1(x)$:

$\alpha_1 = 0.42$



$f_2(x)$:

$\alpha_2 = 0.66$



$f_3(x)$

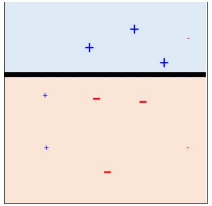
$\varepsilon_3 = 0.13$

$d_3 = 2.59$

$\alpha_3 = 0.95$

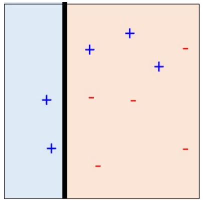
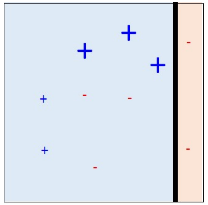
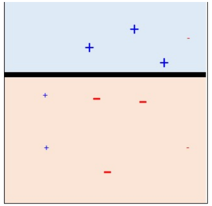
$f_3(x)$:

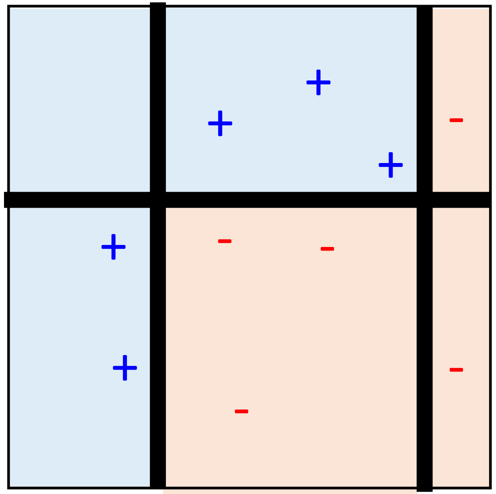
$\alpha_3 = 0.95$



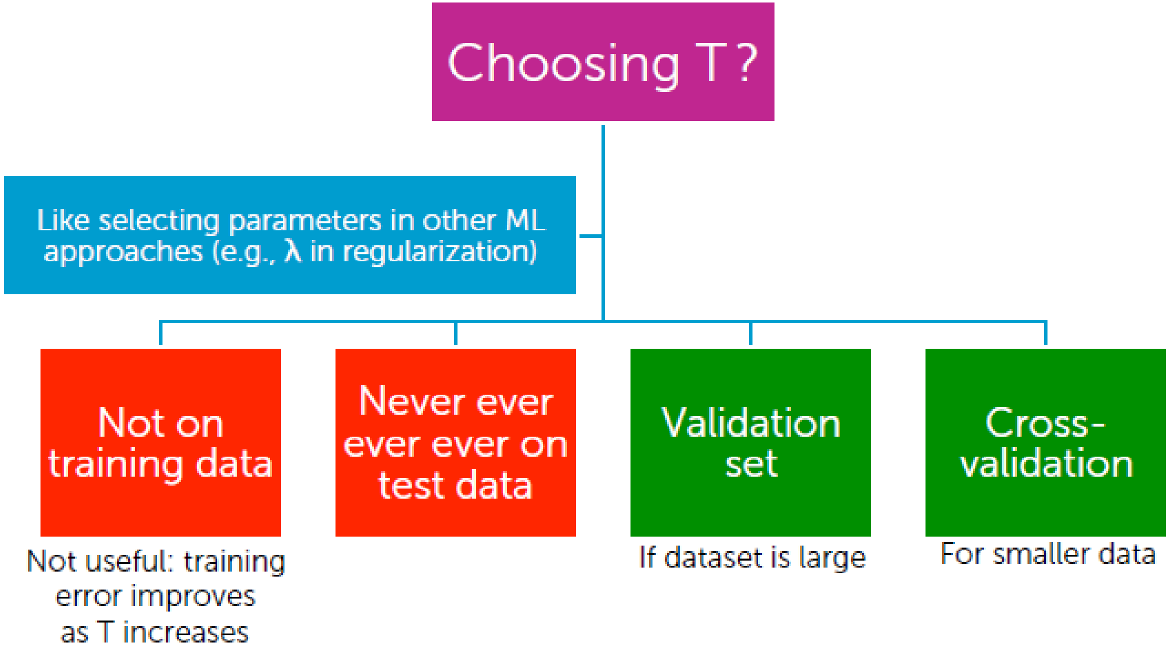
Toy Example: Part IV

Final Classifier: $H(x) = sign(\sum_{t=1}^{T} \alpha_t f_t(x))$

$sign($ 0.42  $+$ 0.66  $+$ 0.95  $)$



How to Choose T?



General Formulation of Boosting

- Initial function $g_0(x) = 0$
- For t = 1 to T:
  - Find a function $f_t(x)$ and α_t to improve $g_{t-1}(x)$
    - $g_{t-1}(x) = \sum_{i=1}^{t-1} \alpha_i f_i(x)$
  - $g_t(x) = g_{t-1}(x) + \alpha_t f_t(x)$
- Output: $H(x) = sign\big(g_T(x)\big)$

What is the learning target of $g(x)$?

Minimize $L(g) = \sum_n l\big(\hat{y}^n, g(x^n)\big)$

Gradient Boosting

- Find $\mathrm{g}(x)$ to minimize $L(g)$
  - If we already have $\mathrm{g}(x) = g_{t-1}(x)$, how to update $\mathrm{g}(x)$?

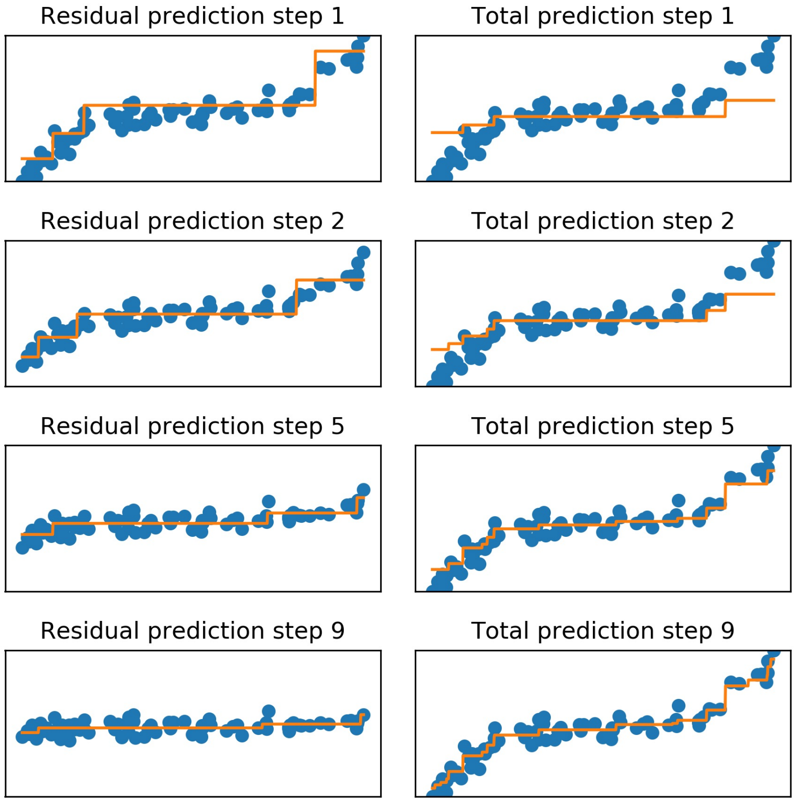
Gradient Descent:

$$g_t(x) = g_{t-1}(x) \boxed{- \eta \left. \frac{\partial L(g)}{\partial g(x)} \right|_{\mathrm{g}(x) = g_{t-1}(x)}}$$

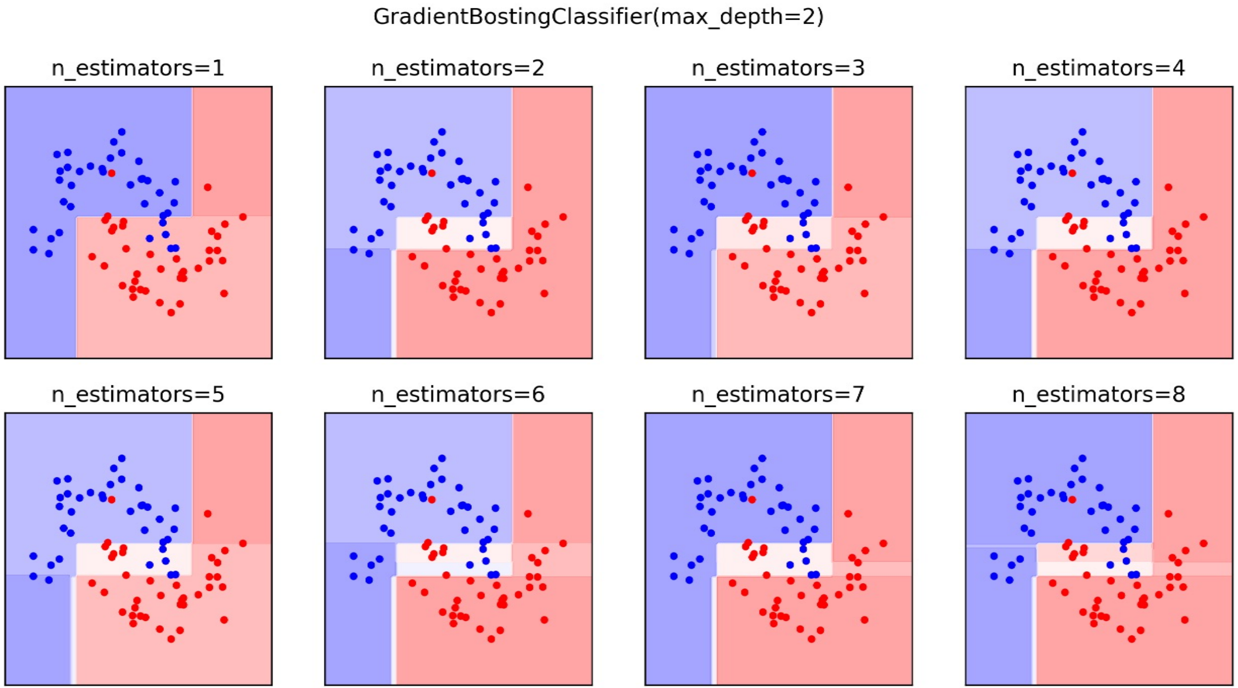
Same direction

$$g_t(x) = g_{t-1}(x) \boxed{+ \alpha_t f_t(x)}$$

GradientBoostingRegressor



GradientBoostingClassifier / HistGradientBoostingClassifier



Impact of Boosting

Amongst most useful
ML methods ever created

Extremely useful in
computer vision

- Standard approach for face detection, for example

Used by **most winners** of
ML competitions
(Kaggle, KDD Cup,...)

- Malware classification, credit fraud detection, ads click through rate estimation, sales forecasting, ranking webpages for search, Higgs boson detection,...

Most deployed ML systems
use model ensembles

- Coefficients chosen manually, with boosting, with bagging, or others

Gradient Boosted Decision Trees: Pros and Cons

Pros

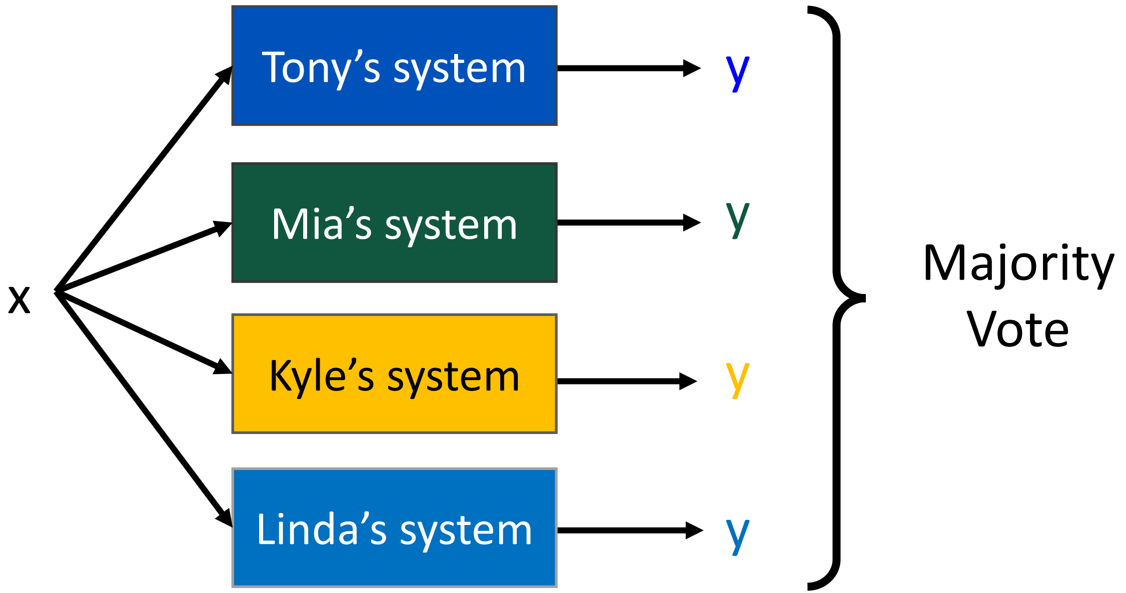
- Often best off-the-shelf accuracy on many problems.
- Using model for prediction requires only modest memory and is fast.
- Doesn't require careful normalization of features to perform well.
- Like decision trees, handles a mixture of feature types.

Cons

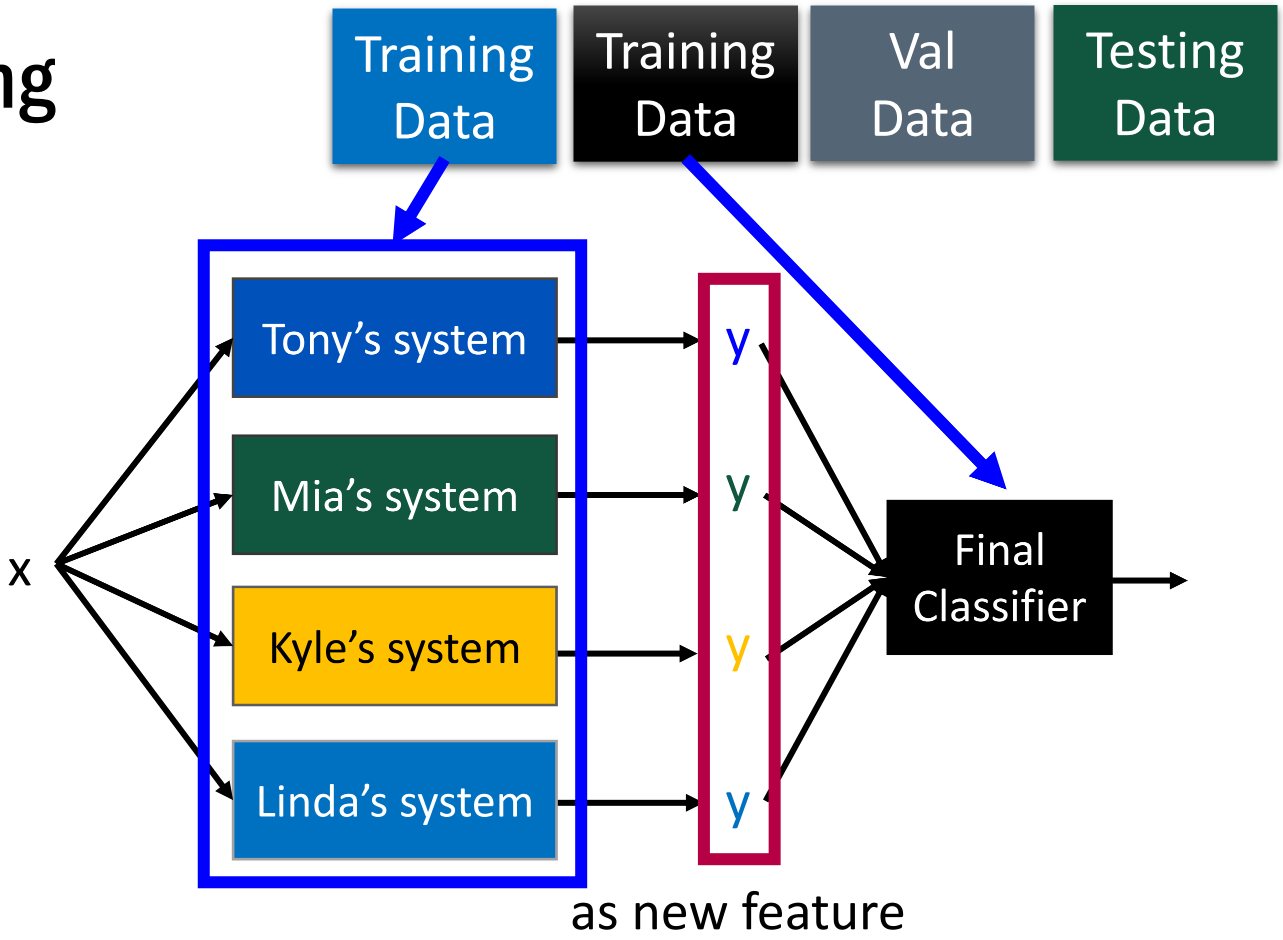
- Like random forests, the models are often difficult for humans to interpret.
- Requires careful tuning of the learning rate and other parameters.
- Training can require significant computation.
- Not recommended for text classification and other problems with very high dimensional sparse features, for accuracy and computational cost reasons.

Ensemble: Stacking

Voting


x
Tony’s system
Mia’s system
Kyle’s system
Linda’s system
y
y
y
y
Majority
Vote

Stacking



Training Data
Training Data
Val Data
Testing Data
x
Tony's system
Mia's system
Kyle's system
Linda's system
y
y
y
y
Final Classifier
as new feature