Boosting
Recap

STAT/CSE 416: Machine Learning

Emily Fox
University of Washington

May 1, 2018

Boosting question

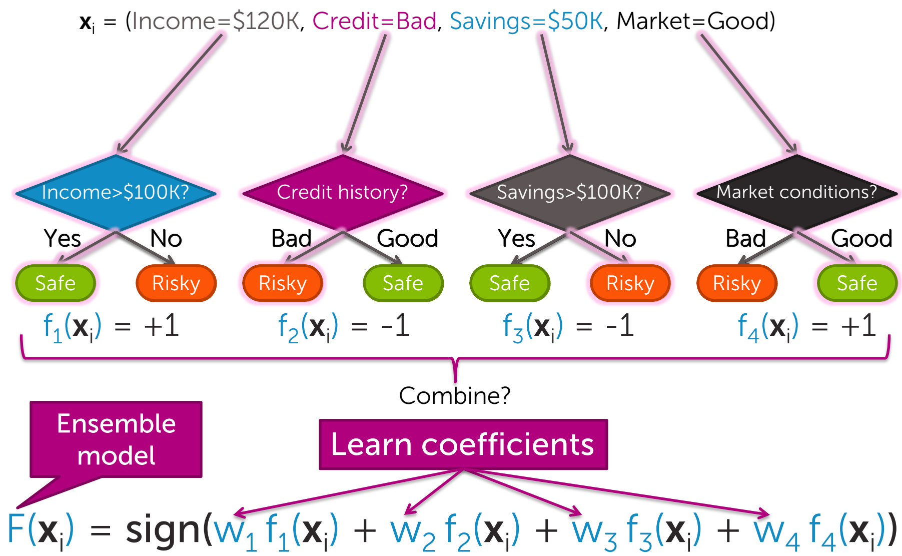
"Can a set of weak learners be combined to create a stronger learner?" *Kearns and Valiant (1988)*

Yes! *Schapire (1990)*

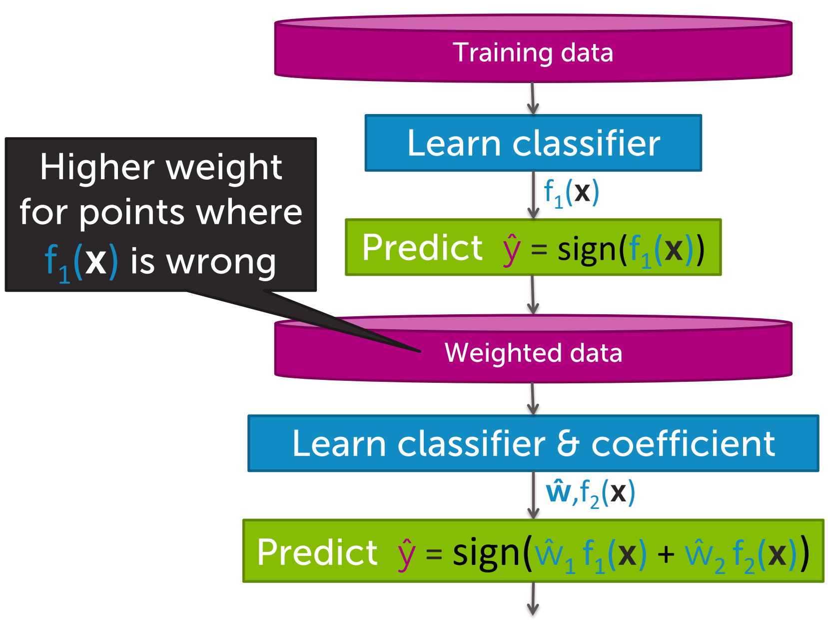
Boosting

Amazing impact: • simple approach • widely used in industry • wins most Kaggle competitions

Ensemble methods: Each classifier "votes" on prediction



Boosting = Greedy learning ensembles from data



AdaBoost: learning ensemble

[Freund & Schapire 1999]

- Start with same weight for all points: $\alpha_i = 1/N$
- For t = 1,…,T
  - Learn $f_t(\mathbf{x})$ with data weights α_i
  - Compute coefficient $\hat{w}_t$ — Problem 1: How much do I trust f_t?
  - Recompute weights α_i — Problem 2: Weigh mistakes more
- Final model predicts by:

$$\hat{y} = sign\left(\sum_{t=1}^{T} \hat{\mathbf{w}}_t f_t(\mathbf{x})\right)$$

AdaBoost: learning ensemble

- Start with same weight for all points: $\alpha_i = 1/N$
- For t = 1,…,T
  - Learn $f_t(\mathbf{x})$ with data weights α_i
  - Compute coefficient $\hat{w}_t$
  - Recompute weights α_i
  - Normalize weights α_i
- Final model predicts by:

$$\hat{y} = sign\left(\sum_{t=1}^{T} \hat{\mathbf{w}}_t f_t(\mathbf{x})\right)$$

$$\hat{\mathbf{w}}_t = \frac{1}{2} \ln\left(\frac{1 - weighted\_error(f_t)}{weighted\_error(f_t)}\right)$$

$$\alpha_i \leftarrow \begin{cases} \alpha_i e^{-\hat{w}_t}, & \text{if } f_t(\mathbf{x}_i)=y_i \\ \alpha_i e^{\hat{w}_t}, & \text{if } f_t(\mathbf{x}_i)\neq y_i \end{cases}$$

$$\alpha_i \leftarrow \frac{\alpha_i}{\sum_{j=1}^{N} \alpha_j}$$

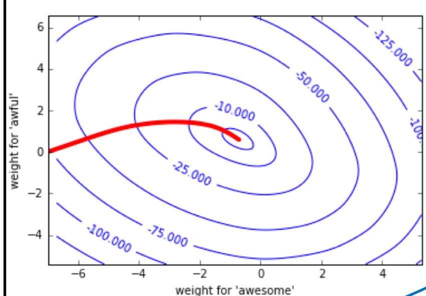
Learning from weighted data

Learning from weighted data in general

Often, learning from weighted data treats data point i as α_i replicates of that data point

Credit	Income	y	Weight α
A	$130K	Safe	0.5
B	$80K	Risky	1.5
C	$110K	Risky	1.2
A	$110K	Safe	0.8
A	$90K	Safe	0.6
B	$120K	Safe	0.7
C	$30K	Risky	3
C	$60K	Risky	2
B	$95K	Safe	0.8
A	$60K	Safe	0.7
A	$98K	Safe	0.9

Gradient ascent for logistic regression on *unweighted data*



init $\mathbf{w}^{(1)}=0$ (or randomly, or smartly), t=1

while $||\nabla \ell(\mathbf{w}^{(t)})|| > \varepsilon$ ← small threshold

for j=0,…,D

Difference between truth and prediction

partial[j] $= \sum_{i=1}^{N} h_j(\mathbf{x}_i)\left(\mathbb{1}[y_i=+1] - P(y=+1 \mid \mathbf{x}_i, \mathbf{w}^{(t)})\right)$

$w_j^{(t+1)} \leftarrow w_j^{(t)} + \eta$ partial[j]

t ← t + 1

$w_j^{(t+1)} \leftarrow w_j^{(t)} + \eta \sum_{i=1}^{N} h_j(x_i)\left(\mathbb{1}(\quad) - P(\quad)\right)$

$\mathbb{1}(y_i=+1) = \begin{cases} 1 & \text{if } y_i=+1 \\ 0 & \text{ow} \end{cases}$

Modify the logistic regression gradient update for *weighted data*

Sum over data points

$$w_j^{(t+1)} \leftarrow w_j^{(t)} + \eta \sum_{i=1}^{N} \alpha_i h_j(\mathbf{x}_i)\left(\mathbb{1}[y_i=+1] - P(y=+1 \mid \mathbf{x}_i, \mathbf{w}^{(t)})\right)$$

Weigh each point by α_i

Resulting w^* is "best" logistic classifier (with given features) on data with weights α_i

How to learn the "best" decision stump?

Credit	Income	y	Weight α
A	$130K	Safe	0.5
B	$80K	Risky	1.5
C	$110K	Risky	1.2
A	$110K	Safe	0.8
A	$90K	Safe	0.6
B	$120K	Safe	0.7
C	$30K	Risky	3
C	$60K	Risky	2
B	$95K	Safe	0.8
A	$60K	Safe	0.7
A	$98K	Safe	0.9

Goal:
Choose best feature (categorical input) or feature/threshold pair (real-valued input)

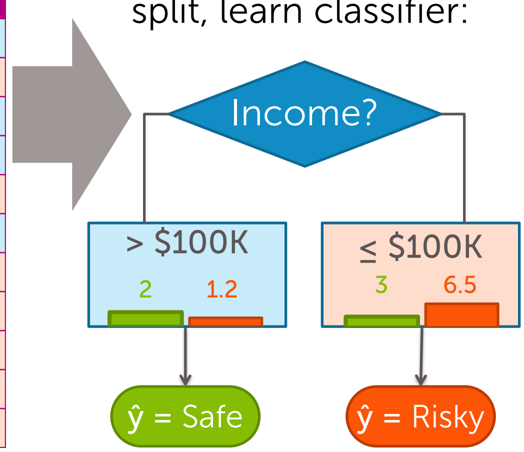
Questions:

1. For given feature or feature/threshold, how to determine classifier output using weighted data?
2. For that classifier, how to compute its accuracy on the training data?
3. For real-valued features, how to select the best threshold on weighted data?
4. Based on the above, select the best decision stump

Learning a decision stump on weighted data

Credit	Income	y	Weight α
A	$130K	Safe	0.5
B	$80K	Risky	1.5
C	$110K	Risky	1.2
A	$110K	Safe	0.8
A	$90K	Safe	0.6
B	$120K	Safe	0.7
C	$30K	Risky	3
C	$60K	Risky	2
B	$95K	Safe	0.8
A	$60K	Safe	0.7
A	$98K	Safe	0.9

For a given potential split, learn classifier:



How to learn the "best" decision stump?

Credit	Income	y	Weight α
A	$130K	Safe	0.5
B	$80K	Risky	1.5
C	$110K	Risky	1.2
A	$110K	Safe	0.8
A	$90K	Safe	0.6
B	$120K	Safe	0.7
C	$30K	Risky	3
C	$60K	Risky	2
B	$95K	Safe	0.8
A	$60K	Safe	0.7
A	$98K	Safe	0.9

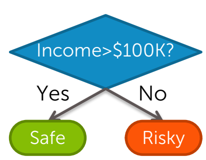
Goal:
Choose best feature (categorical input) or feature/threshold pair (real-valued input)

Questions:
1. For given feature or feature/threshold, how to determine classifier output using weighted data?
2. For that classifier, how to compute its accuracy on the training data?
3. For real-valued features, how to select the best threshold on weighted data?
4. Based on the above, select the best decision stump

Learning a decision stump on weighted data

Credit	Income	y	Weight α
A	$130K	Safe	0.5
B	$80K	Risky	1.5
C	$110K	Risky	**1.2**
A	$110K	Safe	0.8
A	$90K	Safe	**0.6**
B	$120K	Safe	0.7
C	$30K	Risky	3
C	$60K	Risky	2
B	$95K	Safe	**0.8**
A	$60K	Safe	**0.7**
A	$98K	Safe	**0.9**

Compute weighted error:

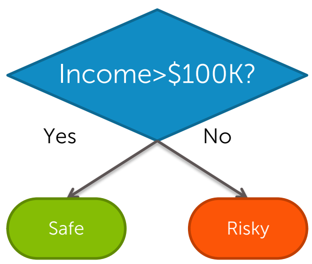


$$\text{Weighted error} = \frac{\text{total weight of mistakes}}{\text{total weight}}$$

$$= \frac{1.2 + 0.6 + 0.8 + 0.7 + 0.9}{\text{total weight}}$$

$$= \frac{4.2}{12.7} = 0.33$$

How to learn the "best" decision stump?

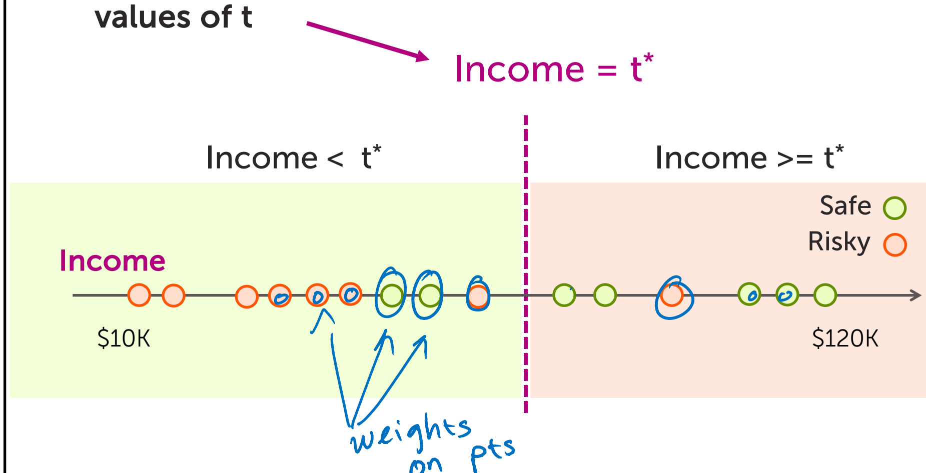


Goal:
Choose best feature (categorical input) or feature/threshold pair (real-valued input)

Questions:
1. For given feature or feature/threshold, how to determine classifier output using weighted data?
2. For that classifier, how to compute its accuracy on the training data?
3. For real-valued features, how to select the best threshold on weighted data?
4. Based on the above, select the best decision stump

Finding the best threshold split

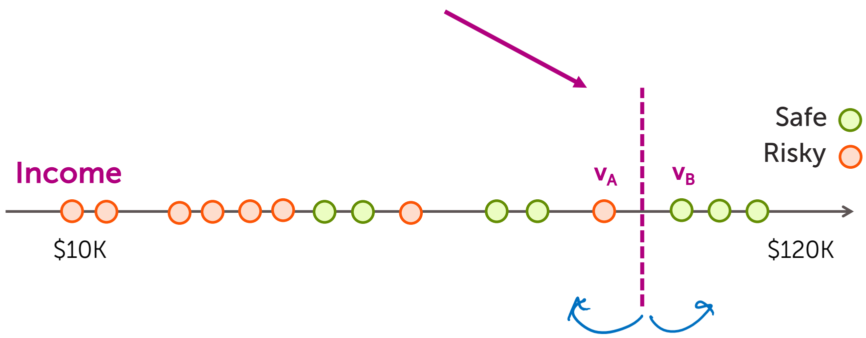
Infinite possible values of t



Consider a threshold between points

Same weighted error for any threshold split between v_A and v_B

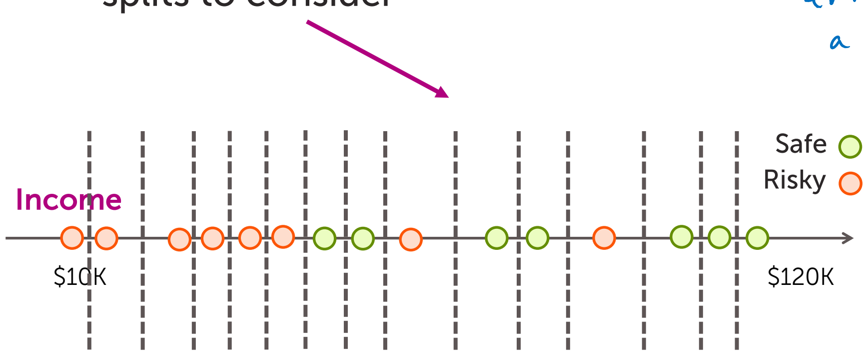
weighted error only changes when crossing over a (weighted) data pt



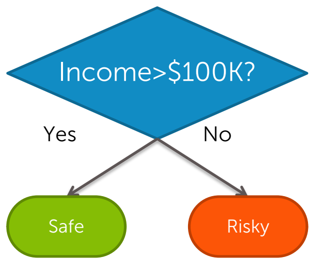
Only need to consider mid-points

Finite number of splits to consider

⇒ same alg. as before, but each considered threshold has a diff. weighted error.



How to learn the "best" decision stump?



Goal:
Choose best feature (categorical input) or feature/threshold pair (real-valued input)

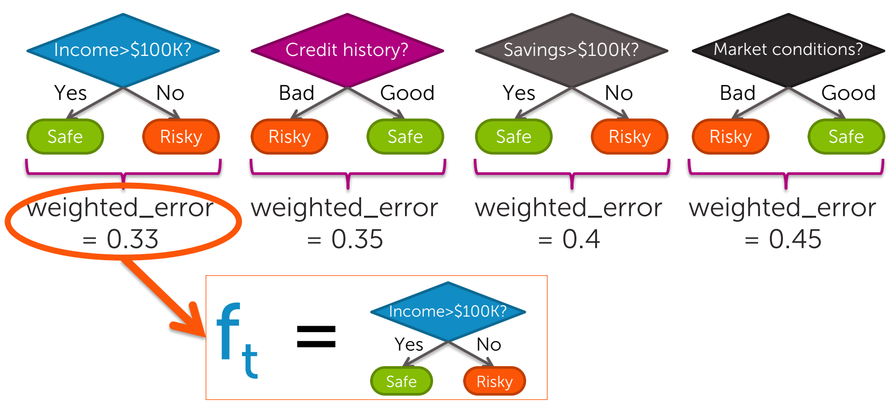
Questions:
1. For given feature or feature/threshold, how to determine classifier output using weighted data?
2. For that classifier, how to compute its accuracy on the training data?
3. For real-valued features, how to select the best threshold on weighted data?
4. Based on the above, select the best decision stump

19

Finding best next decision stump $f_t(\mathbf{x})$

Which weak learner to choose?

Consider splitting on each feature:



20

Revising AdaBoost example: **A visualization**

Boosted decision stumps

- Start same weight for all points: $\alpha_i = 1/N$
- For t = 1,…,T
  - Learn $f_t(\mathbf{x})$: pick decision stump with lowest weighted training error according to α_i
  - Compute coefficient $\hat{w}_t$
  - Recompute weights α_i
  - Normalize weights α_i
- Final model predicts by:

$$\hat{y} = sign\left(\sum_{t=1}^{T} \hat{\mathbf{w}}_t f_t(\mathbf{x})\right)$$

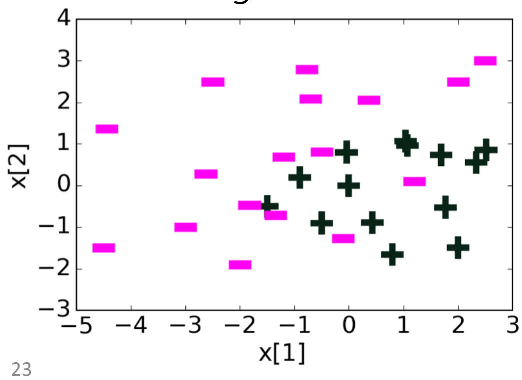
t=1: Just learn a classifier on original data

all pts have same weight

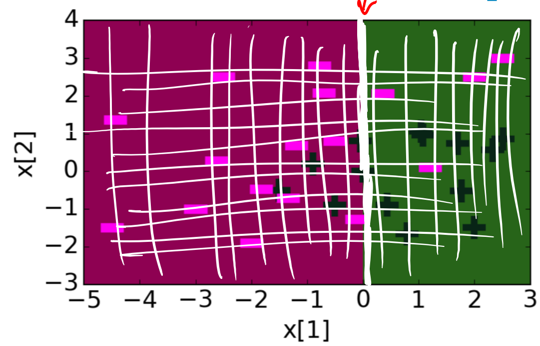
standard learning

lowest ~~weighted~~ error

Original data



Learned decision stump $f_1(\mathbf{x})$



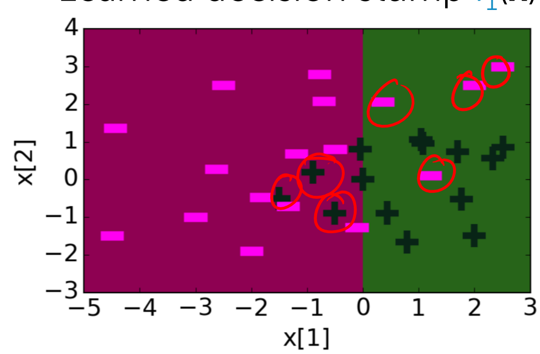
Determine weight (trust) of classifier

$$\hat{W}_t = \frac{1}{2}\ln\left(\frac{1-weighted\_error(f_t)}{weighted\_error(f_t)}\right)$$

$$= \frac{1}{2}\ln\left(\frac{1-\frac{7}{31}}{\frac{7}{31}}\right)$$

$$= 0.62$$

Learned decision stump $f_1(\mathbf{x})$



Updating weights α_i

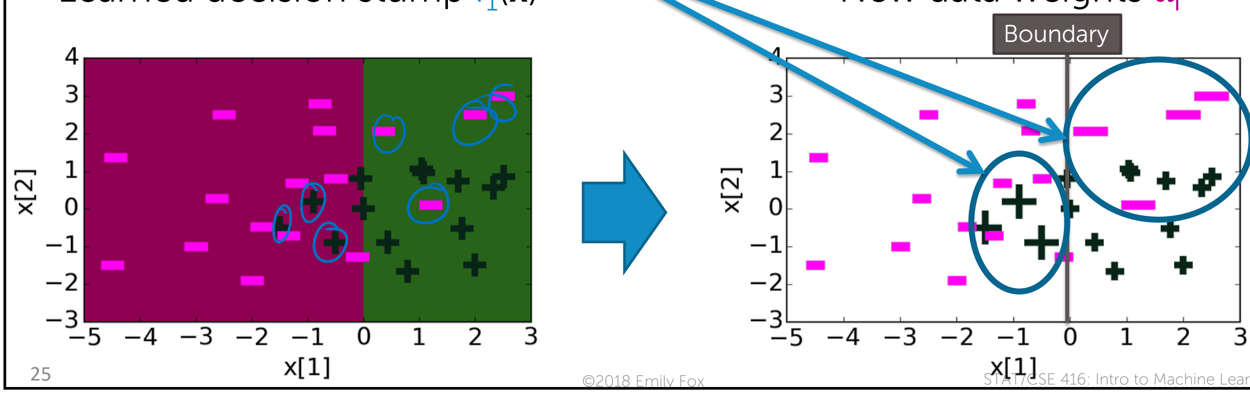
$$\alpha_i \leftarrow \begin{cases} \alpha_i e^{-\hat{w}_t} = 0.54\alpha_i, & \text{if } f_t(\mathbf{x}_i) = y_i \quad \text{correct} \\ \alpha_i e^{\hat{w}_t} = 1.85\alpha_i, & \text{if } f_t(\mathbf{x}_i) \neq y_i \quad \text{incorrect} \end{cases}$$

(handwritten: $\hat{w}_t = 0.62$)

Increase weight α_i of misclassified points

Learned decision stump $f_1(\mathbf{x})$

New data weights α_i



Normalize weights…

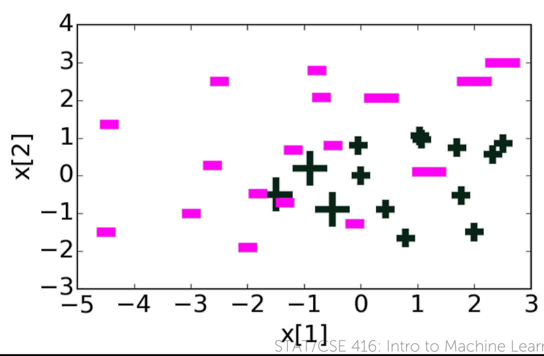
$$\alpha_i \leftarrow \frac{\alpha_i}{\sum_{j=1}^{N} \alpha_j}$$

$x_i[1]$	$x_i[2]$	y_i	α_i	α_i^{new} ← normalized
⋮	⋮	⋮	⋮	

divide by $\sum_{j=1}^{N} \alpha_i$ — total weights

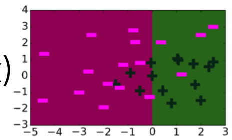
pass this column to next step

New data weights α_i



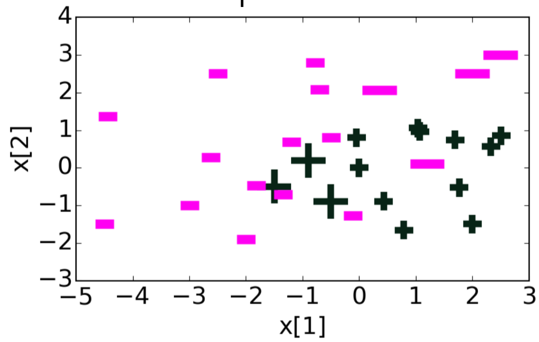
t=2: Learn classifier on weighted data

$\hat{w}_1 = 0.62$, $f_1(\mathbf{x})$

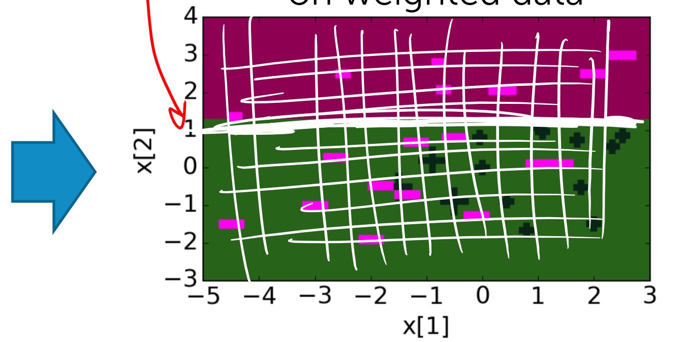


best for this weighted data

Weighted data: using α_i chosen in previous iteration



Learned decision stump $f_2(\mathbf{x})$ on weighted data



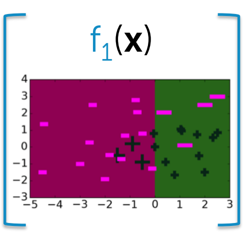
Ensemble becomes weighted sum of learned classifiers

$\hat{y} = \text{sign}(\hat{w}_1 f_1(x) + \hat{w}_2 f_2(x))$ — score(x)

Score(x) =

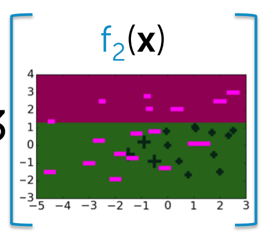
$\hat{w}_1$ $f_1(\mathbf{x})$

0.62

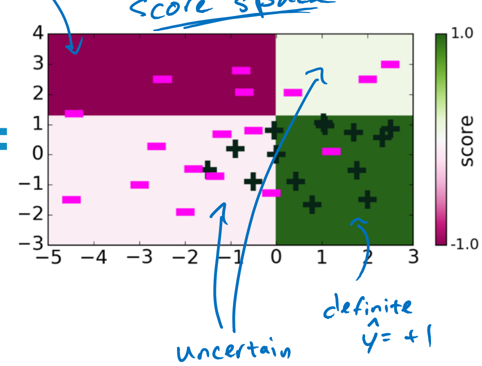


\+ $\hat{w}_2$ $f_2(\mathbf{x})$

0.53

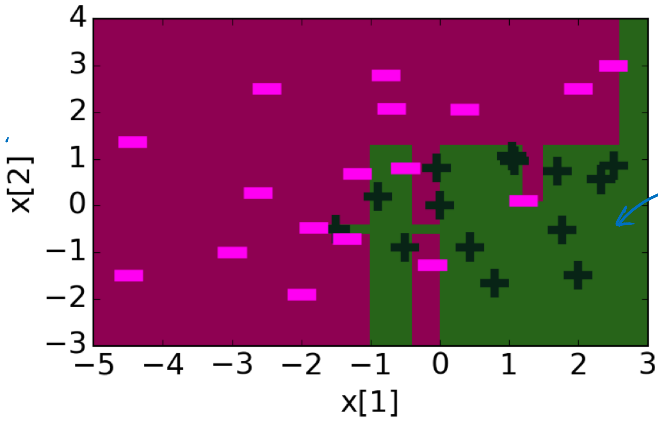


=



Decision boundary of ensemble classifier after 30 iterations

(30 classifiers, T=30)



decision boundary is crazy!
↓
probably overfitting

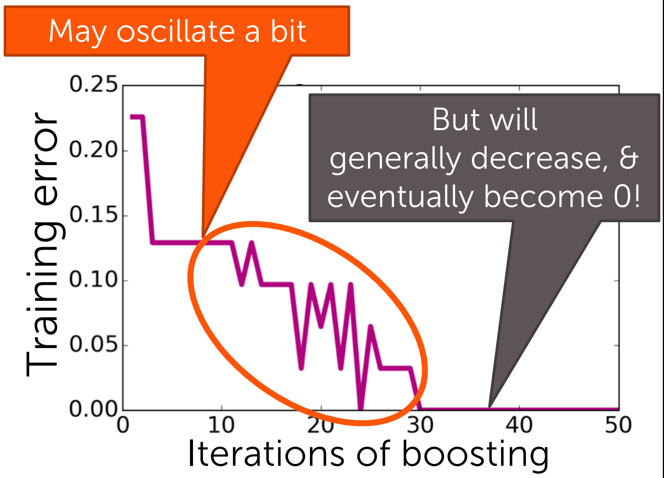
training_error = 0

AdaBoost: Revisiting convergence & overfitting

AdaBoost Theorem

Under some technical conditions…

Training error of boosted classifier → 0 as T→∞

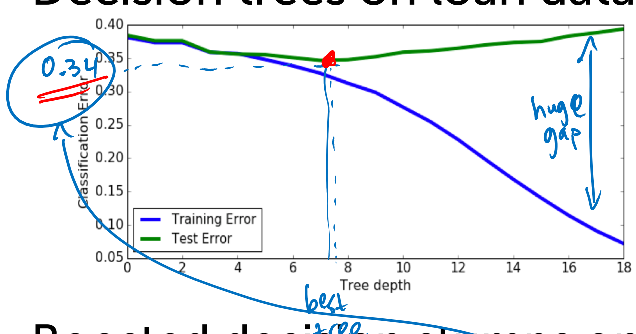


31

©2018 Emily Fox

STAT/CSE 416: Intro to Machine Learning

Decision trees on loan data

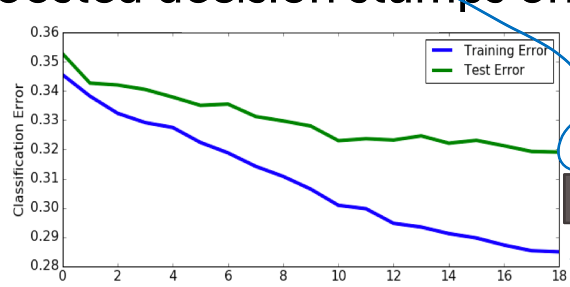


39% test error

Overfitting

8% training error

Boosted decision stumps on loan data



32% test error

Better fit & lower test error

28.5% training error

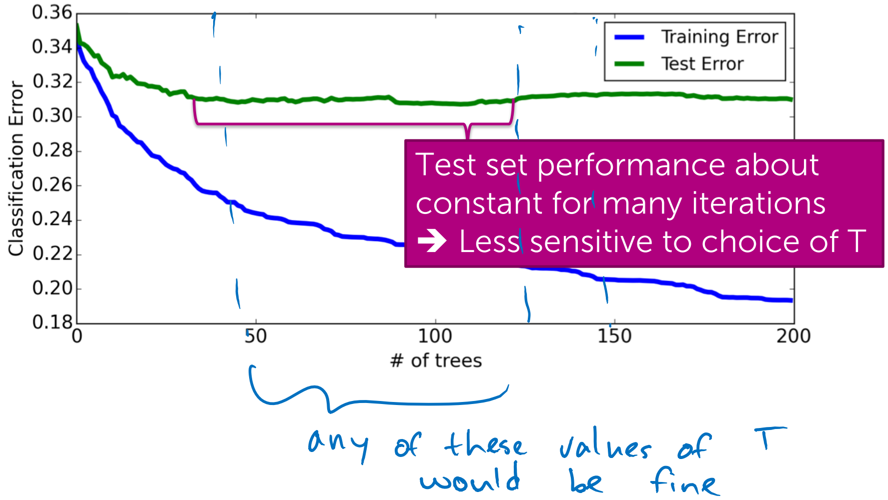
already, boosted decision stumps have better test error than best decision tree

32

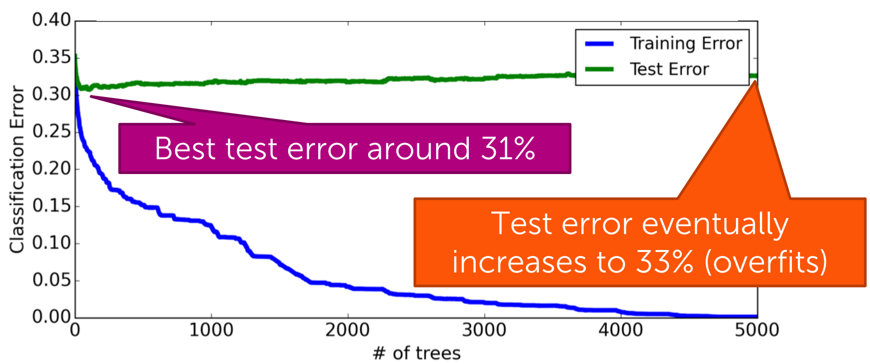
©2018 Emily Fox

STAT/CSE 416: Intro to Machine Learning

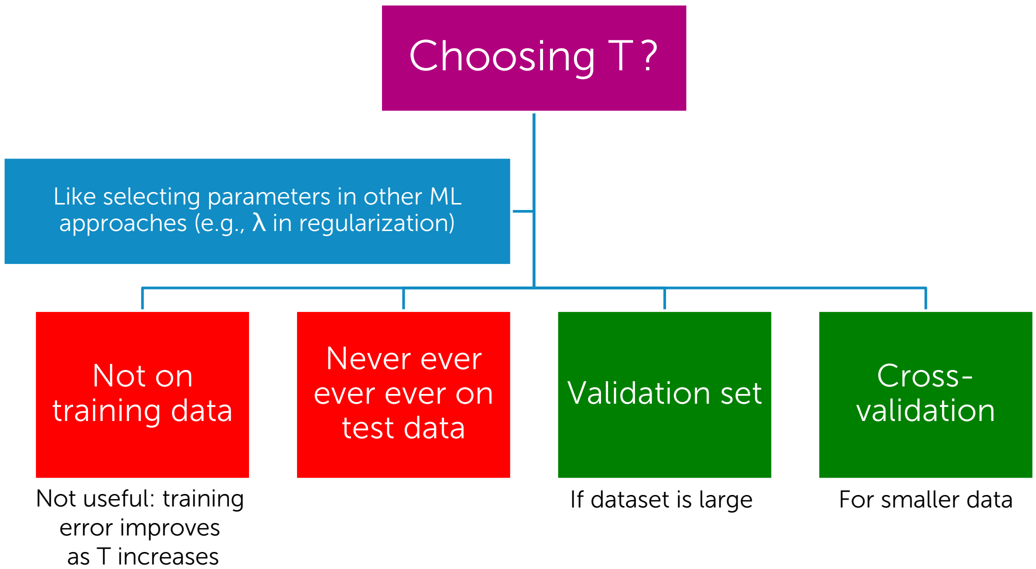
Boosting tends to be robust to overfitting



But boosting will eventually overfit, so must choose max number of components T



How do we decide when to stop boosting?



Summary of boosting

Variants of boosting and related algorithms

There are hundreds of variants of boosting, most important:

Gradient boosting
- Like AdaBoost, but useful beyond basic classification

Many other approaches to learn ensembles, most important:

Random forests
- **Bagging**: Pick random subsets of the data
  - Learn a tree in each subset
  - Average predictions
- Simpler than boosting & easier to parallelize
- Typically higher error than boosting for same # of trees (# iterations T)

Impact of boosting (*spoiler alert... HUGE IMPACT*)

Amongst most useful ML methods ever created

Extremely useful in computer vision
- Standard approach for face detection, for example (though deep learning is making serious inroads)

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

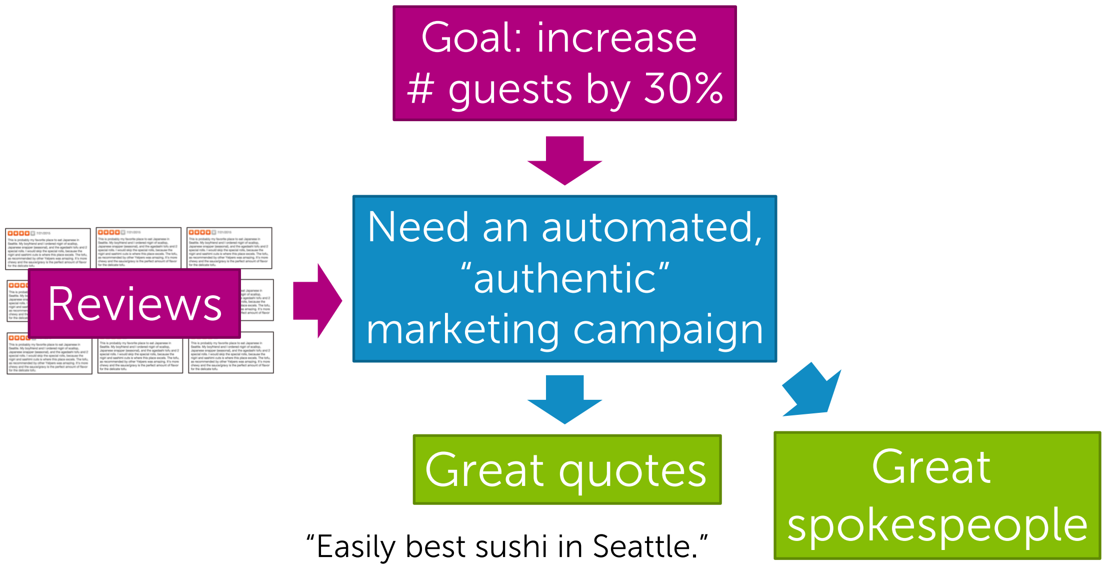
What you can do now…

- Identify notion ensemble classifiers
- Formalize ensembles as weighted combination of simpler classifiers
- Outline the boosting framework – sequentially learn classifiers on weighted data
- Describe the AdaBoost algorithm
  - Learn each classifier on weighted data
  - Compute coefficient of classifier
  - Recompute data weights
  - Normalize weights
- Implement AdaBoost to create an ensemble of decision stumps

Evaluating classifiers: Precision & Recall

STAT/CSE 416: Machine Learning
Emily Fox
University of Washington
May 1, 2018

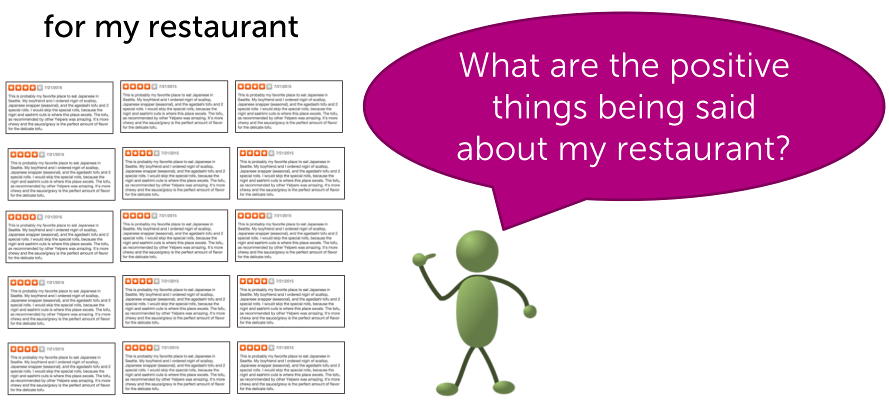
Using reviews to promote my restaurant



"Easily best sushi in Seattle."

How do I find sentences with positive sentiment?

All reviews
for my restaurant



Intelligent restaurant review system

All reviews for restaurant

Break all reviews into sentences

The seaweed salad was just OK, vegetable salad was just ordinary.

I like the interior decoration and the blackboard menu on the wall.

All the sushi was delicious.

My wife tried their ramen and it was pretty forgettable.

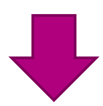
The sushi was amazing, and the rice is just outstanding.

The service is somewhat hectic.

Easily best sushi in Seattle.

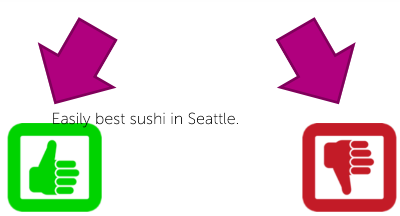
Sentiment classifier

Input $\mathbf{x}_i$: Easily best sushi in Seattle.



Sentence Sentiment Classifier

Output: $\hat{y}_i$
Predicted sentiment



Use the sentiment classifier model!

Sentences from all reviews for my restaurant

The seaweed salad was just OK, vegetable salad was just ordinary.

I like the interior decoration and the blackboard menu on the wall.

All the sushi was delicious.

My wife tried their ramen and it was pretty forgettable.

The sushi was amazing, and the rice is just outstanding.

The service is somewhat hectic.

Easily best sushi in Seattle.

→ Classifier MODEL

Sentences predicted to be positive
$\hat{y} = +1$

- Easily best sushi in Seattle.
- I like the interior decoration and the blackboard menu on the wall.
- All the sushi was delicious.
- The sushi was amazing, and the rice is just outstanding.

Show sentences with +1 prediction on website

Sentences predicted to be negative
$\hat{y} = -1$

- The seaweed salad was just OK, vegetable salad was just ordinary.
- My wife tried their ramen and it was pretty forgettable.
- The service is somewhat hectic.

56 ©2018 Emily Fox STAT/CSE 416: Intro to Machine Learning

What does it mean for a classifier to be good?

©2018 Emily Fox STAT/CSE 416: Intro to Machine Learning

Previously, we asked the question:
"What is good accuracy?"

We explored accuracy of random classifier as baseline

- For binary classification:
  - Half the time, you'll get it right! (on average)
    ➔ classification error = 0.5
- For k classes, error= 1 - 1/k
  - error = 0.666 for 3 classes, 0.75 for 4 classes,…

At the very, very, very least,
you should healthily beat random…
Otherwise, it's (usually) pointless…

We explored the pitfalls of imbalanced problems: *Is 90% accuracy good? Depends …*

90% of sentences are negative!

90% accuracy by predicting every sentence is negative!!!

Amazing "performance" but not useful for me right now!

Automated marketing campaign cares about something else…

Website shows 10 sentences from recent reviews

PRECISION

Did I (mistakenly) show a negative sentence???

RECALL

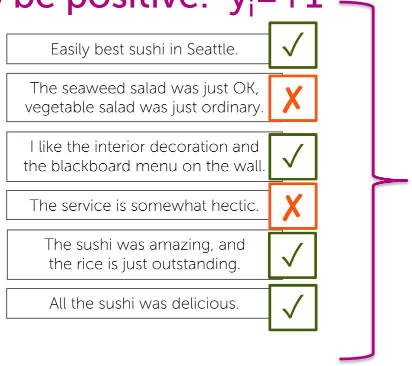
Did I not show a (great) positive sentence???

Accuracy doesn't capture these issues well…

Precision:
Fraction of positive predictions that are *actually positive*

What fraction of the positive predictions are correct?

Sentences predicted to be positive: $\hat{y}_i$=+1



Only 4 out of 6 sentences predicted to be positive are actually positive

Precision: Fraction of positive predictions that are actually positive

Subset of positive predictions that are actually positive

Positive sentences (correct predictions) $y_i=+1$

Negative sentences (incorrect predictions) $y_i=-1$

All sentences predicted to be positive $\hat{y}_i=+1$

Types of error: *Review*

Predicted label

True label	$\hat{y}_i=+1$	$\hat{y}_i=-1$
$y_i=+1$	True Positive	False Negative
$y_i=-1$	False Positive	True Negative

Confusion matrix for sentiment analysis

Predicted sentiment

True sentiment	$\hat{y}_i=+1$	$\hat{y}_i=-1$
$y_i=+1$	+1 sentence +1 prediction	+1 sentence -1 pred.
$y_i=-1$	-1 sentence +1 pred.	-1 sentence -1 pred.

missed a sentence

showed bad review on website :(

Precision - Formula

Fraction of positive predictions that are correct

$$\text{precision} = \frac{\#\text{ true positives}}{\#\text{ true positives} + \#\text{ false positives}}$$

– Best possible value : 1.0
– Worst possible value : 0.0

focus on positive predictions

Predicted label

True label	$\hat{y}_i=+1$	$\hat{y}_i=-1$
$y_i=+1$	True Positive	False Negative
$y_i=-1$	False Positive	True Negative

Example: Calculating precision

Sentences predicted to be positive: $\hat{y}_i=+1$

- Easily best sushi in Seattle. ✓
- The seaweed salad was just OK, vegetable salad was just ordinary. ✗
- I like the interior decoration and the blackboard menu on the wall. ✓
- The service is somewhat hectic. ✗
- The sushi was amazing, and the rice is just outstanding. ✓
- All the sushi was delicious. ✓

4 correct

2 mistakes

precision = $\frac{4}{6}$

= $2/3$

68

Why precision is important

Shown on website

Sentences predicted to be positive: $\hat{y}_i=+1$

- Easily best sushi in Seattle. ✓
- The seaweed salad was just OK, vegetable salad was just ordinary. ✗
- I like the interior decoration and the blackboard menu on the wall. ✓
- The service is somewhat hectic. ✗
- The sushi was amazing, and the rice is just outstanding. ✓
- All the sushi was delicious. ✓

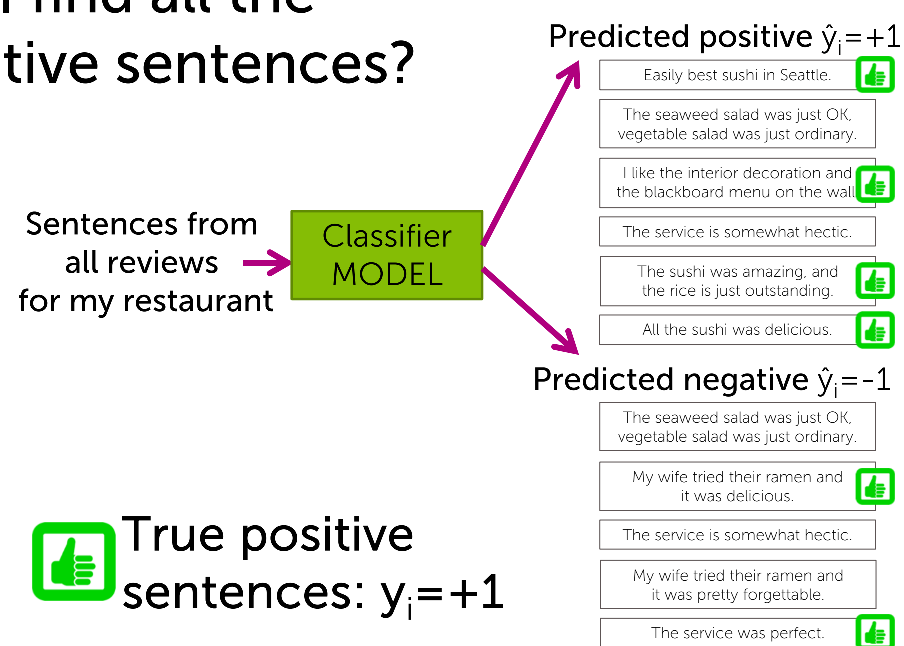
2 negative sentences shown to potential customers… ☹

High precision means positive predictions actually likely to be positive!

69

Recall:
Fraction of positive data
predicted to be positive

Did I find all the positive sentences?



What fraction of positive sentences were missed out?

Predicted positive $\hat{y}_i=+1$

- Easily best sushi in Seattle. 👍
- The seaweed salad was just OK, vegetable salad was just ordinary.
- I like the interior decoration and the blackboard menu on the wall 👍
- The service is somewhat hectic.
- The sushi was amazing, and the rice is just outstanding. 👍
- All the sushi was delicious. 👍

← Found 4 positive sentences

Predicted negative $\hat{y}_i=-1$

- The seaweed salad was just OK, vegetable salad was just ordinary.
- My wife tried their ramen and it was delicious. 👍
- The service is somewhat hectic.
- My wife tried their ramen and it was pretty forgettable.
- The service was perfect. 👍

← Missed 2 positive sentences

Model could not find 2 sentences that were actually positive

Recall: Fraction of positive data predicted to be positive

All positive data points $y=+1$

- Sentences predicted to be positive (correct predictions) $\hat{y}_i=+1$ — Subset of positive data points correctly identified
- Sentences predicted to be negative (incorrect predictions) $\hat{y}_i=-1$

Recall - Formula

Fraction of positive data points correctly classified

$$\text{Recall} = \frac{\text{\# true positives}}{\text{\# true positives} + \text{\# false negatives}}$$

– Best possible value : 1.0
– Worst possible value : 0.0

Predicted label

True label	$\hat{y}_i=+1$	$\hat{y}_i=-1$
$y_i=+1$	True Positive	False Negative
$y_i=-1$	False Positive	True Negative

focus on positive examples

Why is recall important?

Predicted positive $\hat{y}_i=+1$

- Easily best sushi in Seattle.
- The seaweed salad was just OK, vegetable salad was just ordinary.
- I like the interior decoration and the blackboard menu on the wall.
- The service is somewhat hectic.
- The sushi was amazing, and the rice is just outstanding.
- All the sushi was delicious.

Predicted negative $\hat{y}_i=-1$

- The seaweed salad was just OK, vegetable salad was just ordinary.
- My wife tried their ramen and it was delicious.
- The service is somewhat hectic.
- My wife tried their ramen and it was pretty forgettable.
- The service was perfect.

Want to show positive sentences on website

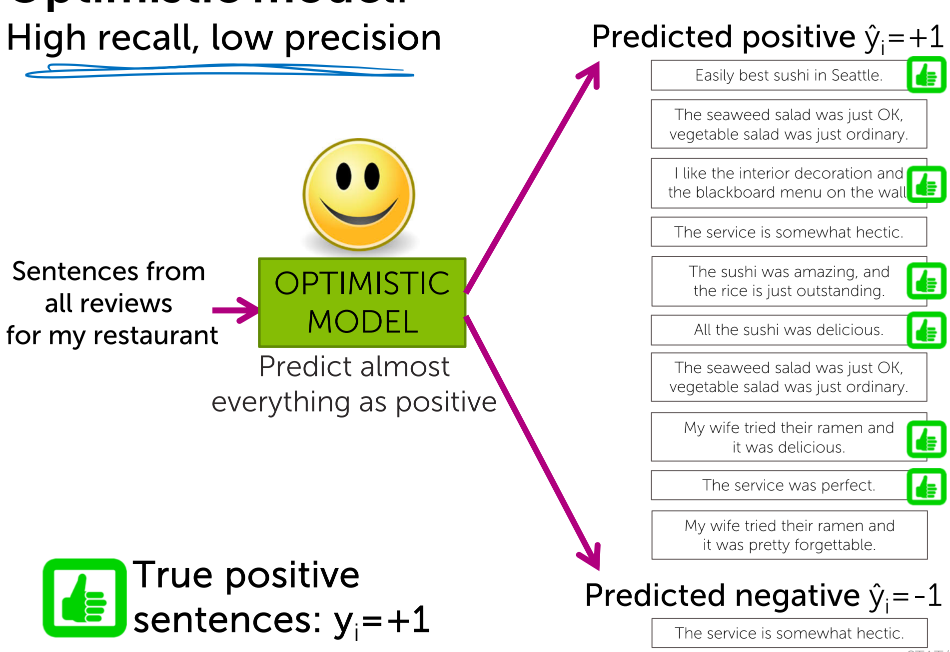
2 positive sentences not shown to potential customers… ☹

High recall means positive data points are very likely to be discovered!

Precision-recall extremes

Optimistic model:

High recall, low precision



Pessimistic model:

High precision, low recall

Sentences from all reviews for my restaurant → PESSIMISTIC MODEL

Predict positive only when very sure

Predicted positive $\hat{y}_i=+1$

- Easily best sushi in Seattle. (true positive)
- The sushi was amazing, and the rice is just outstanding. (true positive)

Predicted negative $\hat{y}_i=-1$

- I like the interior decoration and the blackboard menu on the wall. (true positive)
- The service is somewhat hectic.
- The seaweed salad was just OK, vegetable salad was just ordinary.
- All the sushi was delicious. (true positive)
- The seaweed salad was just OK, vegetable salad was just ordinary.
- My wife tried their ramen and it was delicious. (true positive)
- The service was perfect. (true positive)
- My wife tried their ramen and it was pretty forgettable.
- The service is somewhat hectic.

True positive sentences: $y_i=+1$

Balancing precision & recall

Want to find many positive sentences, but minimize risk of incorrect predictions!!

PESSIMISTIC MODEL ↔ OPTIMISTIC MODEL

PESSIMISTIC MODEL: Finds few positive sentences, but includes no false positives

OPTIMISTIC MODEL: Finds all positive sentences, but includes many false positives

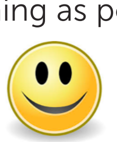
Tradeoff precision and recall

Can we tradeoff precision & recall?

Low precision, high recall

High precision, low recall

Optimistic Model
Predict almost everything as positive



Pessimistic Model
Predict positive only when very sure

How confident is your prediction?

"The sushi & everything else were awesome!"	"The sushi was good, the service was OK"
Definite +1	Not sure
$P(y=+1 \mid \mathbf{x}=$ "The sushi & everything else were awesome!" $) = 0.99$	$P(y=+1 \mid \mathbf{x}=$ "The sushi was good, the service was OK" $) = 0.55$

Can be used to tradeoff precision and recall

Basic classifier

Sentence from review

Input: $\mathbf{x}_i$

Predict most likely class

If $\hat{P}(y=+1|x_i) > 0.5$:

$\hat{y}_i = +1$

Else:

$\hat{y}_i = -1$

Pessimistic: High precision, low recall

Sentence from review

Input: $\mathbf{x}_i$

Pessimistic classifier

If $\hat{P}(y=+1|x_i) > 0.999$:

$\hat{y}_i = +1$ ← very few

Else:

$\hat{y}_i = -1$ ← many

Predict positive only when **very** sure

Optimistic: Low precision, high recall

Sentence from review

Input: $\mathbf{x}_i$

Optimistic classifier

If $\hat{P}(y=+1|x_i) > 0.001$:

$\hat{y}_i = +1$ ← many

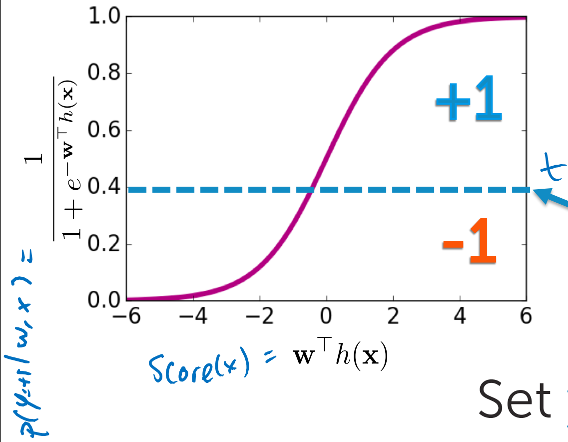
Else:

$\hat{y}_i = -1$

Predict positive almost always

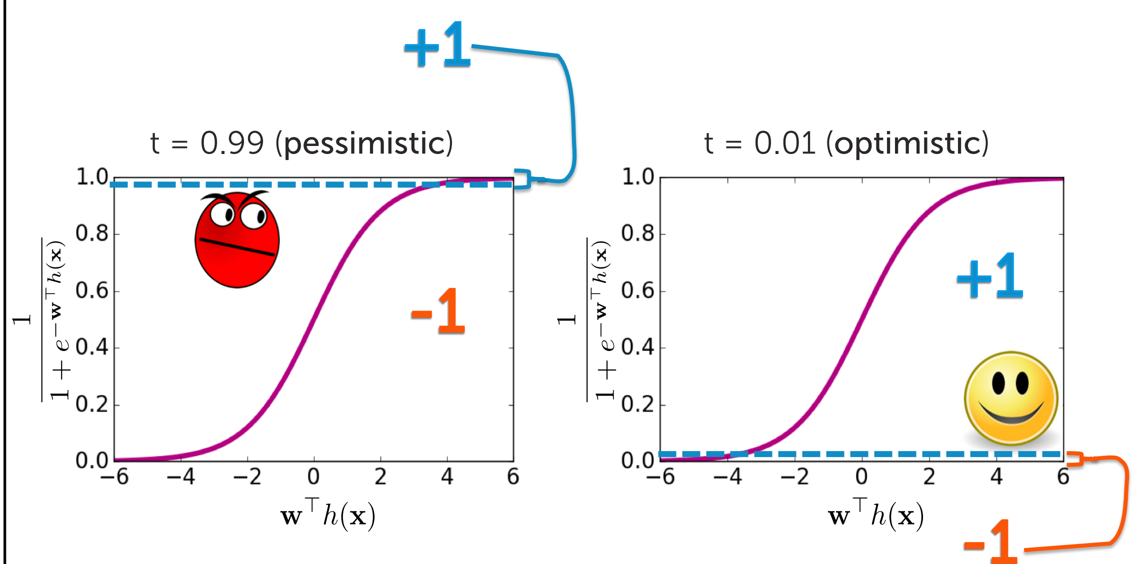
Prediction probability threshold

Probability t above which model predicts true



Set $\hat{y}$ = +1 if $\hat{P}(y|x)$ >= t

Example threshold values



Tradeoff precision & recall with threshold

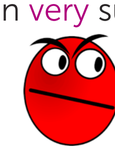
t = 0 ⟷ **t = 1**

Low precision,
high recall

High precision,
low recall

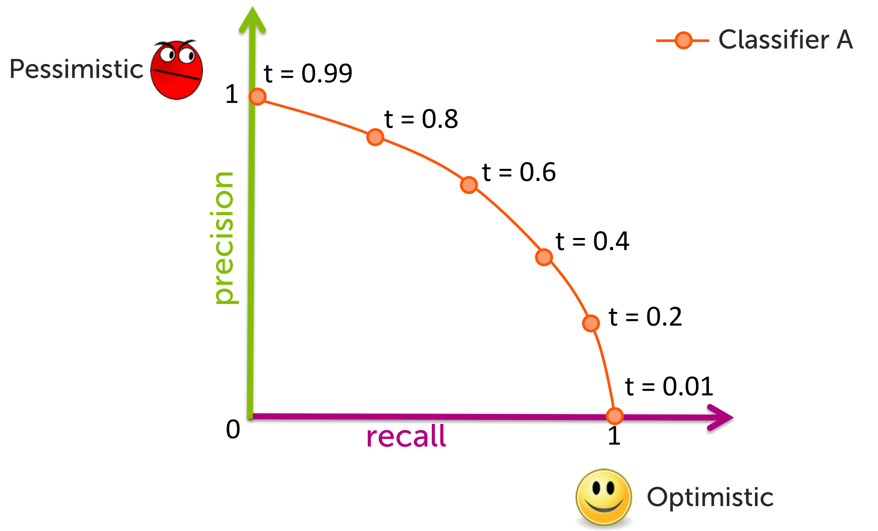
Optimistic Model
Predict almost everything as positive

Pessimistic Model
Predict positive only when *very* sure



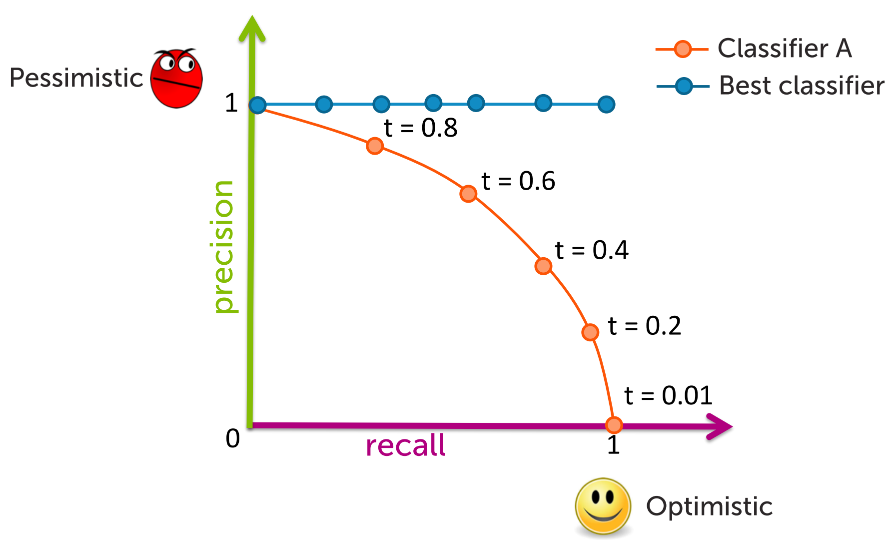
Precision-recall curve

The precision-recall curve



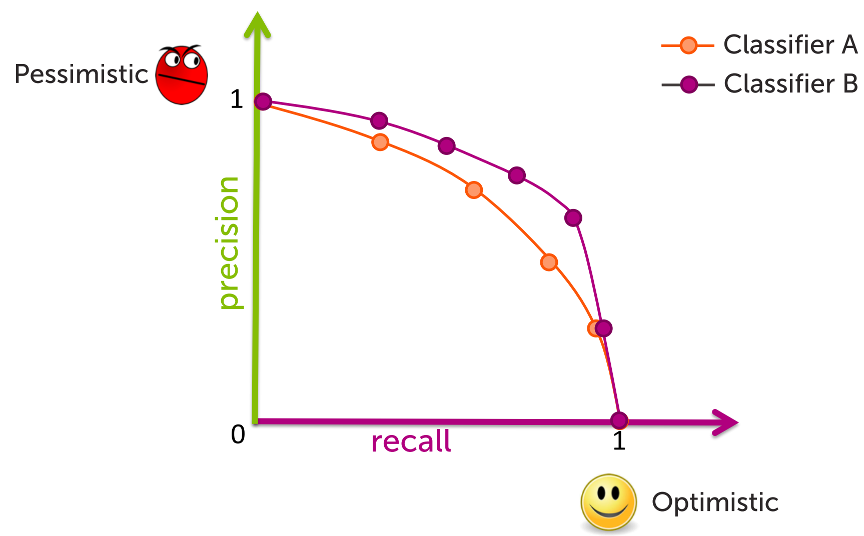
90 ©2018 Emily Fox STAT/CSE 416: Intro to Machine Learning

What does the perfect algorithm look like?

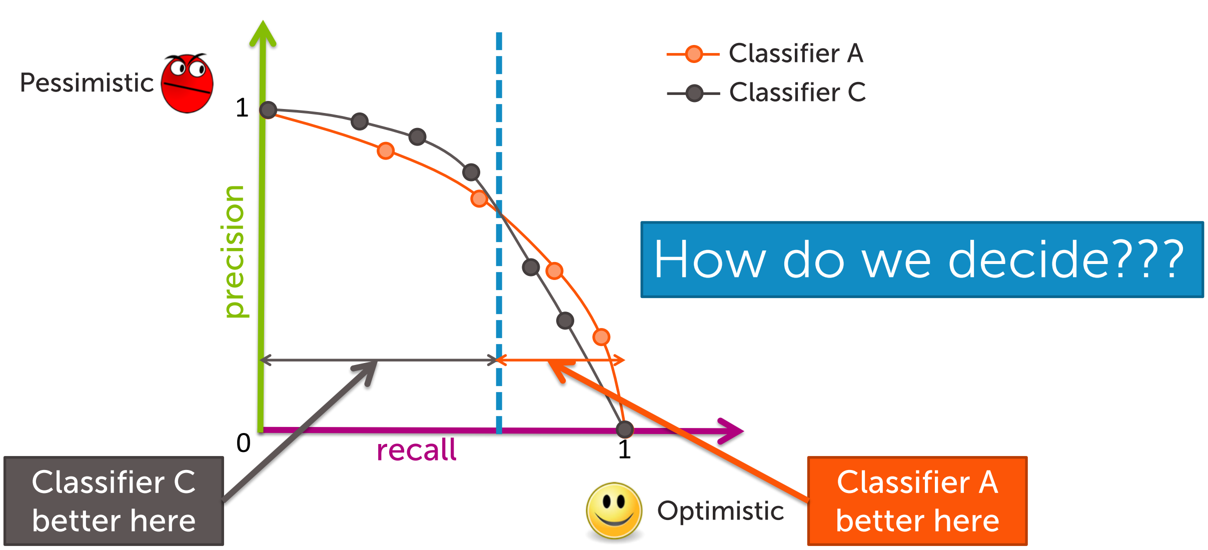


91 ©2018 Emily Fox STAT/CSE 416: Intro to Machine Learning

Which classifier is better? A or B?



Which classifier is better? A or C?



Compare algorithms

Often, reduce precision-recall to single number to compare algorithms

- F1 measure, area-under-the-curve (AUC),…

Precision at k

Showing k=5 sentences on website

Sentences model most sure are positive

- Easily best sushi in Seattle.
- My wife tried their ramen and it was pretty forgettable.
- The sushi was amazing, and the rice is just outstanding.
- All the sushi was delicious.
- The service was perfect.

precision at k = 0.8

Summary of precision-recall

What you can do now…

- Classification accuracy/error are not always right metrics
- Precision captures fraction of positive predictions that are correct
- **Recall** captures fraction of positive data correctly identified by the model
- Trade-off precision & recall by setting probability thresholds
- Plot precision-recall curves.
- Compare models by computing precision at k