Machine Learning: Chenhao Tan

University of Colorado Boulder

LECTURE 11

Slides adapted from Jordan Boyd-Graber, Chris Ketelsen

Logistics

- HW2 available on Github, due in 2 days
- Prelim format

Learning objectives

- Understand train-val-test and cross validation
- Understand precision, recall, F1
- Understand the ROC curve and AUC

Outline

Train-val-test

K-fold cross validation

Precision, recall, F1

ROC, AUC

The story so far

We've seen several machine learning models now (decision tree, KNN, perceptron, Logistic Regression, etc)
You've done your own experiments where you've selected hyperparameters:

- K in K-nearest neighbors
- number of epochs in perceptron

We've talked about the importance of evaluating a learning model on unseen validation data
You've been introduced to the confusion matrix and what it can tell you about how your learning algorithm makes mistakes

The story so far

We've seen several machine learning models now (decision tree, KNN, perceptron, Logistic Regression, etc)

You've done your own experiments where you've selected hyperparameters:

- K in K-nearest neighbors
- number of epochs in perceptron

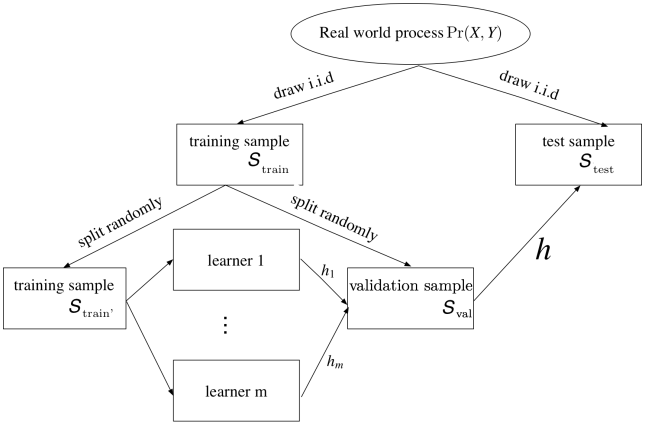
We've talked about the importance of evaluating a learning model on unseen validation data

You've been introduced to the confusion matrix and what it can tell you about how your learning algorithm makes mistakes

Next:

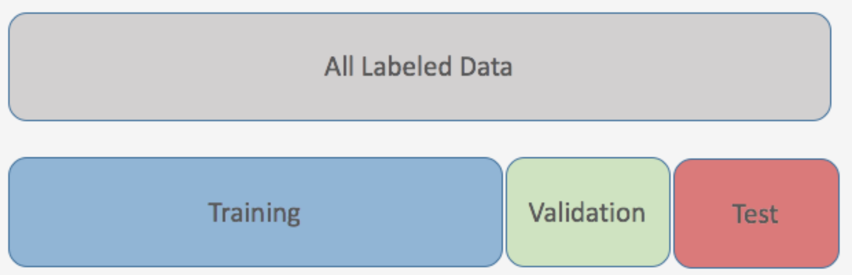
- Validation
- Evaluation metrics

Train-val-test



- training: run machine learning algorithm m times (e.g., parameter search).
- validation error: Errors $\mathrm{Err}_{S_{\text{val}}}(\hat{h}_i)$ is an estimate of $\mathrm{Err}_P(h_i)$.
- selection: Use h_i with $\min \mathrm{Err}_{S_{\text{val}}}(\hat{h}_i)$ for prediction on test examples.

Train-val-test



Typical ratio:

- 70%/10%/20%
- 80%/10%/10%

Outline

Train-val-test

K-fold cross validation

Precision, recall, F1

ROC, AUC

K-fold cross validation

When the number of training instances is small, it seems wasteful to have a separate validation set. What can we do?

K-fold cross validation

When the number of training instances is small, it seems wasteful to have a separate validation set. What can we do?

Using all training data:

- Input: a sample S and a learning algorithm A.
- Procedure: Randomly split S into K equally-sized folds

$$S_1, \ldots, S_K$$

  For each S_i, apply A to S_{-i}, get $\hat{h}_i$, and compute $\mathrm{Err}_{S_i}(\hat{h}_i)$
- Training performance estimates: $\frac{1}{K}\sum_{i=1}^{K} \mathrm{Err}_{S_i}(\hat{h}_i)$

K-fold Cross Validation

Example use:
5-fold CV: Randomly split $N = 25$ examples into five folds $F_i, i = 1, 2, 3, 4, 5$,

train on	test on	error rate
F_1, F_2, F_3, F_4	F_5	1/5
F_1, F_2, F_3, F_5	F_4	0/5
F_1, F_2, F_4, F_5	F_3	0/5
F_1, F_3, F_4, F_5	F_2	2/5
F_2, F_3, F_4, F_5	F_1	0/5

Average error rate: $\frac{1}{5}\sum_{i=1}^{5} \mathrm{Err}_{F_i} = 12\%$

K-fold Cross Validation

Example use:
5-fold CV: Randomly split $N = 25$ examples into five folds $F_i, i = 1, 2, 3, 4, 5$,

train on	test on	error rate
F_1, F_2, F_3, F_4	F_5	1/5
F_1, F_2, F_3, F_5	F_4	0/5
F_1, F_2, F_4, F_5	F_3	0/5
F_1, F_3, F_4, F_5	F_2	2/5
F_2, F_3, F_4, F_5	F_1	0/5

Average error rate: $\frac{1}{5}\sum_{i=1}^{5} \mathrm{Err}_{F_i} = 12\%$
Repeat this process for different hyperparameters and find the hyperparameter with the lowest error rate.

K-fold Cross Validation

Another example:

- Find good features F using S_{train}
- Split S_{train} into K folds
- For each fold, use the rest training data and features F to build a classifier and estimate *prediction error* using average error rates on each fold

K-fold Cross Validation

Another example (This is wrong!):

- Find good features F using S_{train}
- Split S_{train} into K folds
- For each fold, use the rest training data and features F to build a classifier and estimate *prediction error* using average error rates on each fold

Note: the feature selection step actually has information about the supposedly heldout set.
Never *ever* touch your test data in any way!

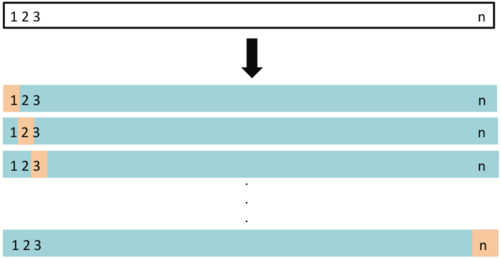
K-fold cross validation

K-fold cross validation can be used for

- selecting best models from training data
- nested cross-validation for performance estimation

Leave-one out cross validation

A special case where $k = N$



LOOCV error rate

$$\frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i^{h_{-i}}),$$

where h_{-i} represents the model trained using all the instances other than i.

Outline

Train-val-test

K-fold cross validation

Precision, recall, F1

ROC, AUC

Accuracy

Thus far, for classification problems we've been primarily concerned with the misclassification error rate and the standard definition of accuracy:

$$\text{error rate} = \frac{1}{n}\sum_{i=1}^{n} I(\hat{y}_i \neq y_i)$$

$$\text{accuracy} = \frac{1}{n}\sum_{i=1}^{n} I(\hat{y}_i = y_i) = 1 - \text{error rate}$$

And for many classification tasks, this makes perfect sense. In fact, many classification techniques are designed specifically to minimize this error rate.

But the misclassification error rate can be misleading in many scenarios.

Accuracy

Consider the case when your training set is heavily skewed towards a particular class.
If 98% of training data is from the negative class, should you feel good about a model with a 98.5% classification accuracy?

Accuracy

Consider the case when your training set is heavily skewed towards a particular class.
If 98% of training data is from the negative class, should you feel good about a model with a 98.5% classification accuracy?

What about when there are different consequences for false positives vs. false negatives?

Can you think of specific examples of this case?

Precision

Confusion matrix:

		predicted labels	
		positive (1)	negative (0)
true labels	positive (1)	true positive (TP)	false negative (FN)
	negative (0)	false positive (FP)	true negative (TN)

Precision

Confusion matrix:

		predicted labels 1	predicted labels 0
true labels	1	TP	FN
	0	FP	TN

Precision measures how accurate the predicted positive class are (exactness).

$$\text{precision} = \frac{TP}{TP + FP}$$

Recall

		predicted labels 1	predicted labels 0
true labels	1	TP	FN
	0	FP	TN

Recall measures the fraction of positives that are correctly identified (completeness).

Recall

		predicted labels 1	predicted labels 0
true labels	1	TP	FN
	0	FP	TN

Recall measures the fraction of positives that are correctly identified (completeness).

$$\text{recall} = \frac{TP}{TP + FN}$$

F1

F1 score strikes a balance between precision and recall.

$$F1 = 2\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

F1 score of the minority class is usually used when evaluating classifiers on imbalanced datasets.
$F1$ is a special case of $F_\beta = (1+\beta^2)\frac{\text{precision}\cdot\text{recall}}{\beta^2\cdot\text{precision}+\text{recall}}$.

Example

		predicted labels 1	predicted labels 0
true labels	1	80	20
	0	10	90

- Baseline (majority accuracy)
- Accuracy
- Precision
- Recall
- F1

Example

		predicted labels 1	predicted labels 0
true labels	1	80	20
	0	10	90

- Baseline (majority accuracy): 50%
- Accuracy: 85%
- Precision: 0.889
- Recall: 0.8
- F1: 0.842

Example

		predicted labels 1	predicted labels 0
true labels	1	10	10
	0	20	160

- Baseline (majority accuracy)
- Accuracy
- Precision
- Recall
- F1

Example

		predicted labels 1	predicted labels 0
true labels	1	10	10
	0	20	160

- Baseline (majority accuracy): 90%
- Accuracy: 85%
- Precision: 0.333
- Recall: 0.5
- F1: 0.4

Outline

Train-val-test

K-fold cross validation

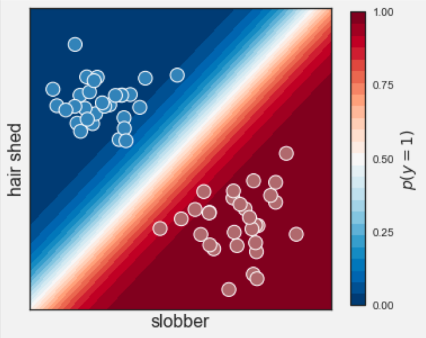
Precision, recall, F1

ROC, AUC

Prediction score

We have so far assumed all predictions are binary.
We can differentiate the "confidence" of a prediction with its predicted score.
For example, in logistic regression,

$$P(y = 1 \mid \boldsymbol{x}) = \sigma(\beta^T \boldsymbol{x})$$

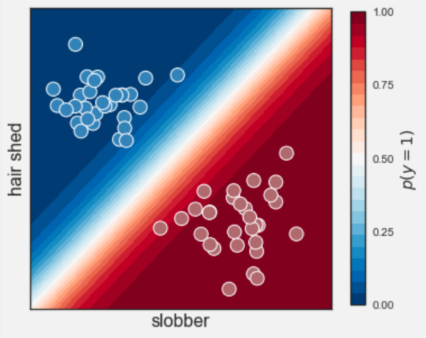


Prediction score

We have so far assumed all predictions are binary.
We can differentiate the "confidence" of a prediction with its predicted score.
For example, in logistic regression,

$$P(y = 1 \mid \boldsymbol{x}) = \sigma(\beta^T \boldsymbol{x})$$

We have always used 0.5 as a threshold to generate a binary prediction, but choosing the threshold can be tricky for imbalanced classes.



TPR and FPR

		predicted labels	
		positive (1)	negative (0)
true labels	positive (1)	true positive (TP)	false negative (FN)
	negative (0)	false positive (FP)	true negative (TN)

- True positive rate, $TPR = \frac{TP}{TP+FN}$
- False positive rate, $FPR = \frac{FP}{FP+TN}$

TPR and FPR

Example: Suppose you build a logistic regression classifier to predict credit card fraud from recent transactions. Customers would rather be warned even when things are OK than let actual fraud be missed.
This means we're willing to accept a high _____ in order to secure a high _____ by choosing a _____ threshold.

A. TPR, FPR, high
B. FPR, TPR, low

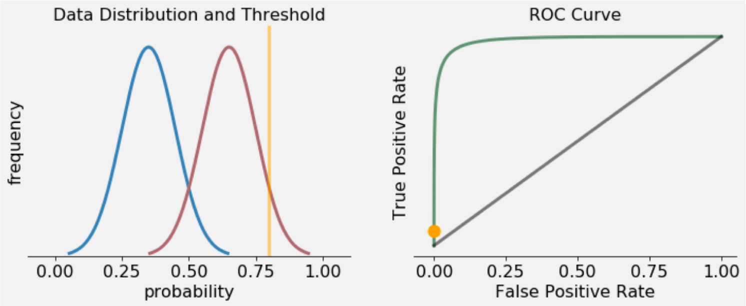
TPR and FPR

Example: Suppose you build a logistic regression classifier to predict credit card fraud from recent transactions. Customers would rather be warned even when things are OK than let actual fraud be missed.
This means we're willing to accept a high _____ in order to secure a high _____ by choosing a _____ threshold.

A. TPR, FPR, high
B. FPR, TPR, low

The answer is B. A ROC Curve gives us a visual way to evaluate suitable thresholds to fit our needs.

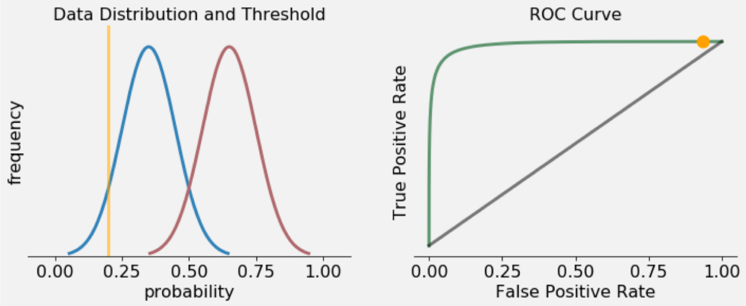
The ROC curve



A ROC Curve is a plot of FPR (horizontal) vs. TPR (vertical) for all possible threshold values.

Convenient to see how a model would perform at all thresholds simultaneously, rather than looking at misclassification rate for each threshold individually.

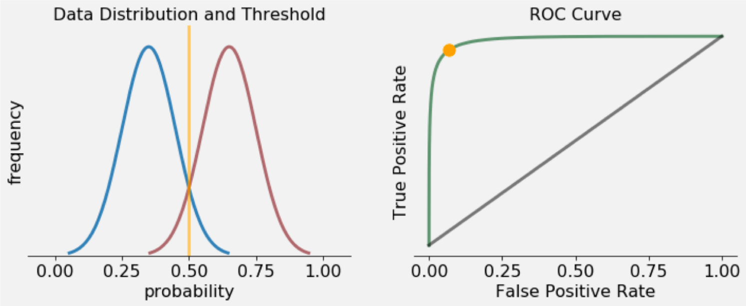
The ROC curve



A ROC Curve is a plot of FPR (horizontal) vs. TPR (vertical) for all possible threshold values.

Convenient to see how a model would perform at all thresholds simultaneously, rather than looking at misclassification rate for each threshold individually.

The ROC curve

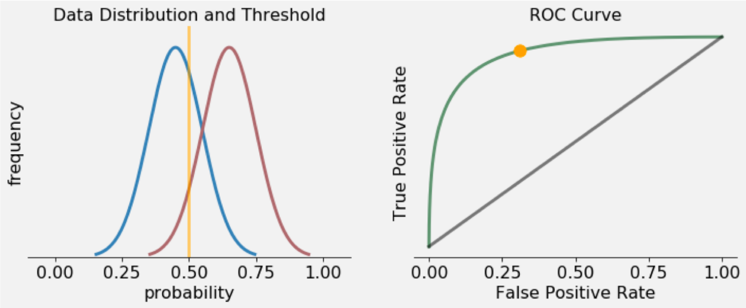


The threshold gives the parameterization of the ROC curve (i.e., it moves the dot).

When the threshold separates the two classes fairly well, the curve is far away from the diagonal.

What happens if we can't separate the classes very well?

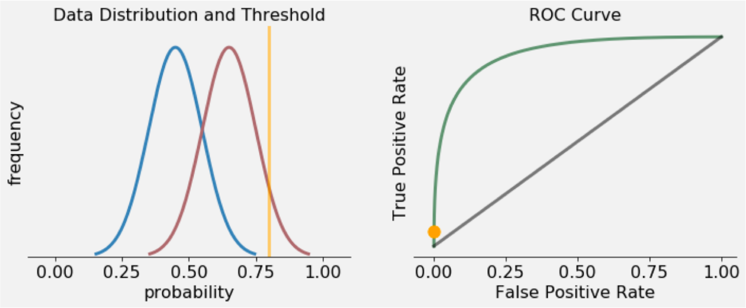
The ROC curve



Now we're not doing so well at separating the classes.

The ROC curve starts bending towards the center.

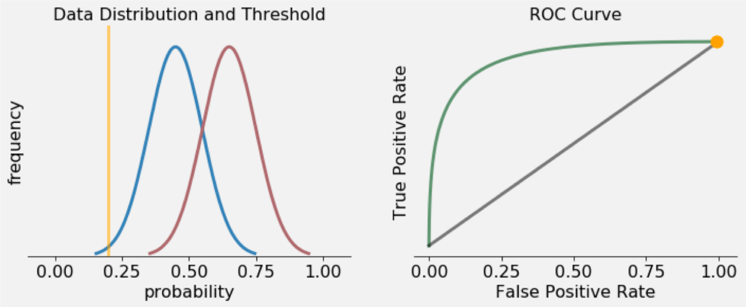
The ROC curve



Now we're not doing so well at separating the classes.

The ROC curve starts bending towards the center.

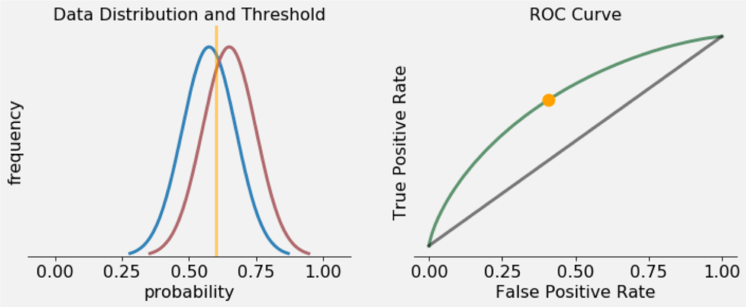
The ROC curve



Now we're not doing so well at separating the classes.

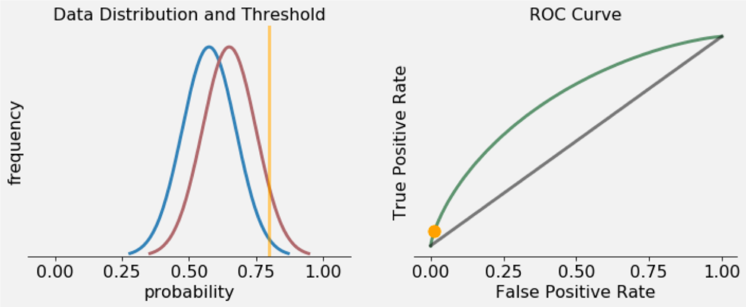
The ROC curve starts bending towards the center.

The ROC curve



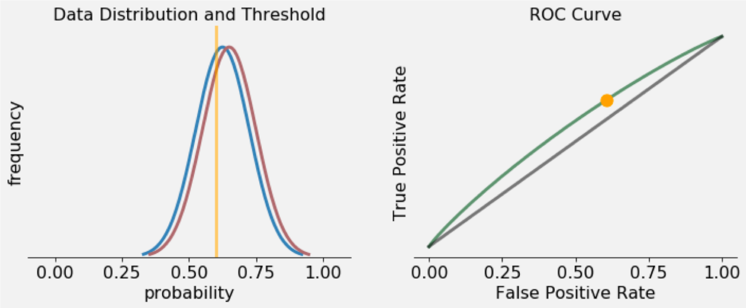
And as we do a poorer job of separating the classes, the curve continues to bend.

The ROC curve



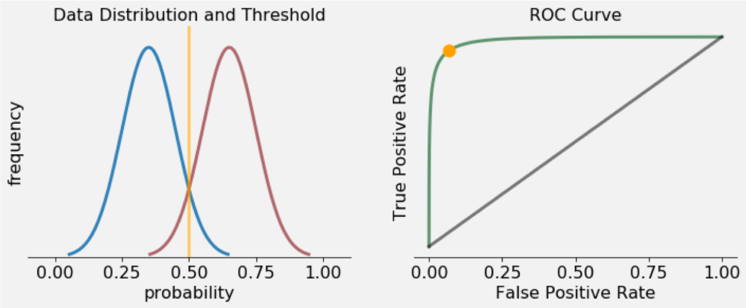
And as we do a poorer job of separating the classes, the curve continues to bend.

The ROC curve



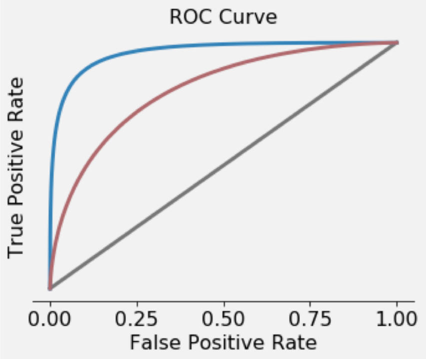
And if we do a terrible job, the curve approaches the random chance line, indicating that our classifier is not much better than a random guess.

The ROC curve



The ROC curve addresses the cases when we're worried about FPs and TPs simultaneously.
But, if you want a single number, evaluating how the model will do in all cases
You can compute the AUC (Area under the ROC curve).

ROC-AUC comparisons



To compare two models, plot their ROC curves on the same axes.
If one encloses the other, then it's better on both ends of the spectrum, and has higher AUC.

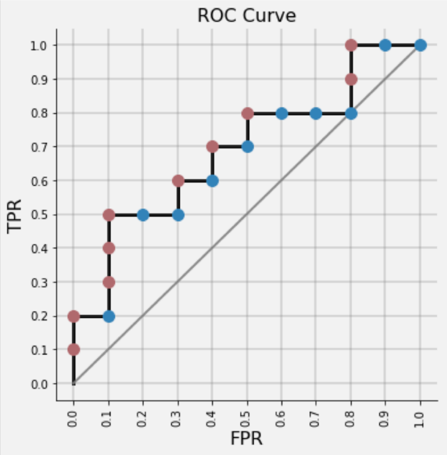
Constructing a ROC curve

You need a classifier that is able to rank examples by predicted score.

- Order all examples by prediction confidence
- Move threshold to each point, one at a time
- If point is true positive, move vertically (1/NP)
- If point is true negative, move horizontally (1/NN)

#	c	$\hat{p}$	#	c	$\hat{p}$
1	P	0.90	11	P	0.40
2	P	0.80	12	N	0.39
3	N	0.70	13	P	0.38
4	P	0.60	14	N	0.37
5	P	0.55	15	N	0.36
6	P	0.54	16	N	0.35
7	N	0.53	17	P	0.34
8	N	0.52	18	P	0.33
9	P	0.51	19	N	0.30
10	N	0.50	20	N	0.10

Constructing a ROC curve



#	c	$\hat{p}$	#	c	$\hat{p}$
1	P	0.90	11	P	0.40
2	P	0.80	12	N	0.39
3	N	0.70	13	P	0.38
4	P	0.60	14	N	0.37
5	P	0.55	15	N	0.36
6	P	0.54	16	N	0.35
7	N	0.53	17	P	0.34
8	N	0.52	18	P	0.33
9	P	0.51	19	N	0.30
10	N	0.50	20	N	0.10

ROC curve

ROC cares both about TPR and FPR, so it values both positive examples and negative examples.
If only positive examples are important, one can plot precision and recall curve.

Wrap up

- Misclassification error / accuracy is unsatisfactory if you have imbalanced classes or care more about false positives or false negatives.
- Accuracy is only one view of confusion matrix.
- Precision, recall, F1, ROC, AUC are your friends during evaluation.