

Machine Learning

Textbook: Machine Learning Refined, by Watt, Borhani, and Katsaggelos, 2nd edition:
MLRefined.com

Previous Lecture

- ◆ The perceptron loss penalizes misclassified observations and leads to the update $\mathbf{w} \leftarrow \mathbf{w} + \alpha y \tilde{\mathbf{x}}$.
- ◆ A maximum-margin classifier chooses the separating hyperplane that maximizes the minimum geometric distance to the training data.
- ◆ Under SVM canonical scaling, $y_p f_p \geq 1$ and maximizing margin is equivalent to minimizing $\frac{1}{2} \|\mathbf{v}\|^2$.
- ◆ Support vectors are the observations that lie on or constrain the margin.
- ◆ Hinge loss $\max(0, 1 - yf)$ penalizes both misclassifications and correctly classified points that lie inside the desired margin.
- ◆ A soft-margin SVM balances hinge loss against a coefficient-norm penalty.



§6.8 - Classification Performance

We now know several ways to train a binary classifier:

logistic regression,
perceptron,
support vector machines.

But:

a low training loss is not the same thing
as good classification performance.

Today:

how do we evaluate predictions?

Positive and Negative Classes

For binary classification,

designate one class as **positive**
and the other as **negative**.

Examples:

disease / no disease;

fraud / legitimate;

spam / not spam.

The choice of which class is called positive
affects the interpretation of many metrics.

Usually the positive class is the event we especially want to detect.

Four Possible Outcomes

For each observation:

	Actual Positive	Actual Negative
Predicted Positive	True Positive	False Positive
Predicted Negative	False Negative	True Negative

Abbreviations:

$TP, FP, FN, TN.$

These four counts form the **confusion matrix**.

Every binary classification metric begins here.

A Running Example

Suppose we classify 100 observations.

The confusion matrix is

	Actual +	Actual -	Total
Predicted +	16	8	24
Predicted -	4	72	76
Total	20	80	100

Thus

$TP = 16,$
 $FP = 8,$
 $FN = 4,$
 $TN = 72.$

We will use these same numbers repeatedly.

Accuracy

Accuracy asks:

what fraction of all predictions were correct?

Correct predictions:

$TP + TN.$

Therefore

$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}.$

For our example:

$\frac{16+72}{100}$

= 0.88.

Accuracy:

88%.

Classification Error Rate

Error rate asks the complementary question:
what fraction were wrong?

Wrong predictions:

$$FP + FN.$$

Thus

$$\text{Error Rate} = \frac{FP + FN}{P}.$$

For our example:

$$\frac{8+4}{100} = 0.12.$$

And

$$\text{Accuracy} = 1 - \text{Error Rate}.$$

Why Accuracy Can Mislead

Suppose we have

1000

observations.

Only

20

are positive.

The other

980

are negative.

A classifier predicts

negative for everyone.

Then

$$TN = 980,$$

$$FN = 20.$$

Accuracy:

$$\frac{980}{1000} = 98\%.$$

But the classifier detects

none

of the positive observations.

High accuracy can be nearly useless under class imbalance.

Recall, Among the Actual Positives

Recall asks:
of the observations that were actually positive,
how many did we detect?

Actual positives:
 $TP + FN$.

Therefore
$$\text{Recall} = \frac{TP}{TP+FN}.$$

For our example:
$$\frac{16}{16+4}$$
$$= 0.80.$$

We detected
80%
of the positive class.

Sensitivity and True Positive Rate

Recall is also called
sensitivity
or
true positive rate.

Thus
$$\text{TPR} = \text{Sensitivity} = \text{Recall} = \frac{TP}{TP+FN}.$$

Different fields prefer different names.

Same quantity.

False Negative Rate

Among actual positives,
the ones we miss are false negatives.

Define
$$\text{FNR} = \frac{FN}{TP+FN}.$$

For our example:
$$\frac{4}{20} = 0.20.$$

Since every actual positive is either
true positive or false negative:
$$\text{TPR} + \text{FNR} = 1.$$

Specificity, Among the Actual Negatives

Specificity asks:
of the observations that were actually negative,
how many did we correctly reject?

Actual negatives:
 $TN + FP$.

Therefore
$$\text{Specificity} = \frac{TN}{TN+FP}.$$

For our example:
$$\frac{72}{72+8}$$
$$= 0.90.$$

We correctly reject
90%
of the negative class.

False Positive Rate

Among actual negatives,
false positives are the negatives
incorrectly labeled positive.

Define
$$\text{FPR} = \frac{FP}{FP+TN}.$$

For our example:
$$\frac{8}{80} = 0.10.$$

Therefore
$$\text{Specificity} = 1 - \text{FPR}.$$

Precision, Among the Predicted Positives

Precision asks a different question.

Of the observations we **predicted positive**,
how many were actually positive?

Predicted positives:
 $TP + FP$.

Thus
$$\text{Precision} = \frac{TP}{TP+FP}.$$

For our example:
$$\frac{16}{16+8}$$

$$= \frac{2}{3} \approx 0.667.$$

About two-thirds of our positive predictions were correct.

Precision and Recall Ask Different Conditional Questions

Recall:

$$\frac{TP}{TP+FN} \cdot$$

Denominator:

actual positives.

Question:

did we find the positives?

Precision:

$$\frac{TP}{TP+FP} \cdot$$

Denominator:

predicted positives.

Question:

can we trust a positive prediction?

Do not confuse the denominators.

When Recall Matters More

High recall matters when

missing a positive case is especially costly.

Examples:

screening for a serious disease;

detecting dangerous equipment failure;

initial fraud screening.

A false negative may be much more costly

than a false positive.

Then we may accept more false alarms

to catch more true positives.

When Precision Matters More

High precision matters when

a positive prediction triggers an expensive action.

Examples:

automatically blocking a legitimate transaction;

sending a costly intervention;

flagging content for severe enforcement.

A false positive may be especially costly.

Then we may prefer fewer positive predictions
with greater reliability.

F_1 Score

Sometimes we want one number
that combines precision and recall.

The F_1 score is their harmonic mean:

$$F_1 = 2 \frac{(\text{Precision})(\text{Recall})}{\text{Precision} + \text{Recall}}.$$

Equivalent count formula:

$$F_1 = \frac{2TP}{2TP + FP + FN}.$$

For our example:

$$\begin{aligned} F_1 &= \frac{32}{32+8+4} \\ &= \frac{32}{44} \\ &\approx 0.727. \end{aligned}$$

Why the Harmonic Mean?

The ordinary arithmetic mean of

1
and
0.1
is
0.55.

The harmonic mean is

$$\begin{aligned} &\frac{2(1)(0.1)}{1+0.1} \\ &\approx 0.182. \end{aligned}$$

Thus F_1 becomes small
when either precision or recall is small.

A model cannot obtain a high F_1
by excelling at one while failing badly at the other.

Balanced Accuracy

When classes are imbalanced,
ordinary accuracy can be dominated by the majority class.

One alternative:

Balanced Accuracy = $\frac{\text{Recall}+\text{Specificity}}{2}$.

For our example:

$\frac{0.80+0.90}{2}$
= 0.85.

The positive and negative classes receive equal weight regardless of their frequencies.

Return to Probabilities

Logistic regression gives

$p_p = P(Y = 1 \mid \mathbf{x}_p)$.

A hard class prediction requires a threshold:

$\hat{y}_p = \begin{cases} 1, & p_p \geq t, \\ 0, & p_p < t. \end{cases}$

The familiar choice is

$t = 0.5$.

But:

0.5 is not mandatory.

Changing the threshold changes the confusion matrix.

A Threshold Example

Suppose

Observation	1	2	3	4	5	6
True y	1	1	1	0	0	0
Probability p	.95	.70	.40	.60	.30	.10

At threshold

$t = 0.5$:

predictions are

1, 1, 0, 1, 0, 0.

Thus

$TP = 2, \quad FN = 1, \quad FP = 1, \quad TN = 2$.

Metrics at Threshold 0.5

Accuracy:

$\frac{2+2}{6} = \frac{2}{3}$.

Precision:

$$\frac{2}{2+1} = \frac{2}{3}.$$

Recall:

$$\frac{2}{2+1} = \frac{2}{3}.$$

Specificity:

$$\frac{2}{2+1} = \frac{2}{3}.$$

Here all four happen to agree.

That is not true in general.

Lower the Threshold

Use

$$t = 0.3.$$

Predictions:

$$1, 1, 1, 1, 1, 0.$$

Then

$$\begin{aligned} TP &= 3, \\ FN &= 0, \\ FP &= 2, \\ TN &= 1. \end{aligned}$$

Recall:

$$\frac{3}{3} = 1.$$

Specificity:

$$\frac{1}{3} \approx 0.333.$$

Precision:

$$\frac{3}{5} = 0.60.$$

Lower threshold:
more positives detected,
but more false positives.

Raise the Threshold

Use

$$t = 0.7.$$

Predictions:

$$1, 1, 0, 0, 0, 0.$$

Then

$$\begin{aligned} TP &= 2, \\ FN &= 1, \end{aligned}$$

$$FP = 0,$$
$$TN = 3.$$

Precision:

$$\frac{2}{2} = 1.$$

Recall:

$$\frac{2}{3} \approx 0.667.$$

Specificity:

$$\frac{3}{3} = 1.$$

Higher threshold:

fewer false positives,
but potentially more missed positives.

Threshold Is a Decision Choice

The model produces probabilities or scores.

The threshold converts them into actions.

Therefore:

model fitting
and
decision policy
are related but distinct.

A sensible threshold depends on:

relative costs of false positives and false negatives;

class frequencies;

operational constraints;

desired recall / precision tradeoff.

There is no universally optimal threshold.

ROC Curve

Instead of choosing one threshold immediately,
vary the threshold across many values.

For each threshold compute:

$$TPR = \frac{TP}{TP+FN},$$

and

$$FPR = \frac{FP}{FP+TN}.$$

Plot

TPR versus FPR.

This is the **receiver operating characteristic**
or **ROC curve**.

Each point corresponds to a threshold.

Reading an ROC Curve

Ideal classifier:

high TPR,
low FPR.

So desirable points lie near
the upper-left corner:
(0, 1).

A random ranking tends toward the diagonal
 $\text{TPR} \approx \text{FPR}$.

ROC curves describe
threshold tradeoffs.

They do not select the operating threshold for us.

Area Under the ROC Curve

The area under the ROC curve is
AUC.

Typical range:
 $0 \leq \text{AUC} \leq 1$.

AUC near
1:
strong ranking ability.

AUC near
0.5:
roughly random ranking.

One interpretation:
 $\text{AUC} = P(\text{a random positive receives a higher score than a random negative})$

when ties are handled appropriately.

AUC evaluates ranking across thresholds.

It does not tell us whether probabilities are well calibrated.

Precision–Recall Tradeoff

For rare positive classes,
precision can change dramatically
even when the false-positive rate looks small.

So another useful curve plots
Precision versus Recall
as the threshold varies.

Precision–recall analysis focuses directly on
performance for the positive class.

This can be especially informative
when the positive class is rare.

Training Loss Versus Evaluation Metric

A logistic classifier may be trained by minimizing
cross-entropy.

An SVM may be trained by minimizing
hinge loss plus regularization.

But after training we may evaluate:

accuracy;

recall;

specificity;

precision;

F_1 ;

ROC AUC.

The training objective and evaluation metric
need not be the same formula.

Evaluate on Unseen Data

As with regression:
metrics computed on training data
can be optimistic.

The classifier was fitted using those observations.

So classification performance should ultimately be measured
on data not used to fit the parameters.

Training set:

learn parameters.

Validation data:

choose settings such as threshold or model complexity.

Test data:

final estimate of performance on unseen cases.

We will formalize this later with cross-validation.

Confusion Matrix in Python

Suppose

```
y = np.array([1, 1, 1, 0, 0, 0])
p = np.array([.95, .70, .40, .60, .30, .10])

threshold = 0.5
y_pred = (p >= threshold)
```

Compute the four counts:

```
TP = np.sum((y_pred == 1) & (y == 1))
FP = np.sum((y_pred == 1) & (y == 0))
FN = np.sum((y_pred == 0) & (y == 1))
TN = np.sum((y_pred == 0) & (y == 0))
```

New Python idea:

&

means elementwise logical “and”

for Boolean arrays.

Compute the Metrics in Python

```
accuracy = (TP + TN) / len(y)

precision = TP / (TP + FP)
recall = TP / (TP + FN)
specificity = TN / (TN + FP)

f1 = 2*precision*recall / (precision + recall)
```

At threshold

0.5:

```
accuracy    = 0.6667
precision   = 0.6667
recall      = 0.6667
specificity = 0.6667
f1          = 0.6667
```

The formulas match the hand calculation.

Sweep Over Thresholds

```
for threshold in [0.3, 0.5, 0.7]:
    y_pred = (p >= threshold)
```

```

TP = np.sum((y_pred == 1) & (y == 1))
FP = np.sum((y_pred == 1) & (y == 0))
FN = np.sum((y_pred == 0) & (y == 1))
TN = np.sum((y_pred == 0) & (y == 0))

recall = TP / (TP + FN)
specificity = TN / (TN + FP)

print(threshold, recall, specificity)

```

This lets us see directly
 how one trained model
 can produce different operating behavior
 under different thresholds.

A Practical Classification Checklist

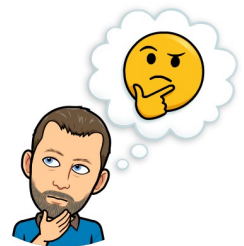
Before reporting “the classifier is accurate”:

- identify which class is positive;
- inspect the confusion matrix;
- check for class imbalance;
- report metrics connected to the real costs of errors;
- examine threshold tradeoffs;
- evaluate on unseen data.

A classifier is not adequately described
 by one percentage.

What did we learn?

- ♦ A binary confusion matrix records true positives, false positives, false negatives, and true negatives.
- ♦ Accuracy can be misleading when the classes are strongly imbalanced.
- ♦ Recall measures the fraction of actual positives detected;
 specificity measures the fraction of actual negatives correctly rejected.
- ♦ Precision measures the fraction of predicted positives that are actually positive.
- ♦ F_1 combines precision and recall, while balanced accuracy combines recall and specificity.
- ♦ Changing the classification threshold trades false positives against false negatives;
 ROC and precision–recall curves summarize this behavior across thresholds.



Next

- Binary classification uses two classes.
- Many real problems have three or more possible labels.
- Next we extend linear classification beyond a single decision boundary.

● 1.19 - Multiclass Classification

Prepared by Dr. Jodin Morey.

Materials for Other Courses Found at [MathTalker.org](https://math.talker.org)