Evaluation

The central question

Machine learning aims at good performance on *future unseen data*, not only on the observations used for training.

- Evaluation requires choosing performance measures that reflect the application objective and the costs of different errors.
- The generalization error cannot be computed over the entire data-generating distribution, so it must be estimated from finite data.
- Performance evaluation is needed for:
  - selecting hyperparameters and model variants;
  - estimating the quality of the final predictor;
  - comparing learning algorithms and quantifying uncertainty.

Evaluation is part of model design

The metric and validation protocol should mimic how the model will be used after deployment.

Why is evaluation difficult?

Consider classifiers for tumor detection, spam filtering, loan approval, and pedestrian recognition. They share the same prediction task, but not the same notion of a costly mistake.

Different applications, different objectives

- Missing a tumor and raising a false alarm have different costs.
- Class proportions may be highly unbalanced.
- Some applications require trustworthy probabilities, not only labels.
- Future data may differ from the data available during development.

Key idea

There is no universally best evaluation measure or data split.

Performance measures

Training Loss and performance measures

- A training loss function measures the cost paid for predicting $f(\mathbf{x})$ when the correct output is y
- It is designed to boost effectiveness and computational efficiency of the learning algorithm (e.g. hinge loss for SVM):
  - it is not necessarily the best measure of final performance
  - e.g. zero–one misclassification loss is rarely used directly for optimization as it is piecewise constant (not amenable to gradient descent)
- Multiple performance measures can be used to evaluate different aspects of a learner

Performance measures

Binary classification

True\ Pred	Positive	Negative
Positive	TP	FN
Negative	FP	TN

- The *confusion matrix* reports true (on rows) and predicted (on columns) labels
- Each entry contains the number of examples with the true label in its row and the predicted label in its column:

  tp True positives: positives predicted as positives
  tn True negatives: negatives predicted as negatives
  fp False positives: negatives predicted as positives
  fn False negatives: positives predicted as negatives

Binary classification

Accuracy

$$Acc = \frac{TP + TN}{TP + TN + FP + FN}$$

- *Accuracy* is the fraction of correctly labeled examples among all predictions
- It is one minus the misclassification cost

Problem

- For strongly unbalanced datasets (typically negatives much more than positives) it is not informative:
  - Predictions are dominated by the larger class
  - Predicting everything as negative often maximizes accuracy
- One possibility consists of *rebalancing* costs (e.g. a single positive counts as N/P where N=TN+FP and P=TP+FN)

Binary classification

Precision

$$Pre = \frac{TP}{TP + FP}$$

- It is the fraction of positives among examples predicted as positives
- It measures the *precision* of the learner when predicting positive

Recall or Sensitivity

$$Rec = \frac{TP}{TP + FN}$$

- It is the fraction of positive examples predicted as positives
- It measures the *coverage* of the learner in returning positive examples

Binary classification

Different errors can have different costs

- In tumor screening, false negatives may be especially harmful (focus on *recall*)
- In recommender systems, false positives may contain unwanted content (focus on *precision*)

Consequence
Accuracy treats every mistake equally. Precision, recall, and cost-sensitive measures reveal different aspects of performance.

Binary Classification

F-measure

$$F_\beta = \frac{(1+\beta^2)(Pre * Rec)}{\beta^2 Pre + Rec}$$

- Precision and recall are complementary: increasing precision typically reduces recall
- F-measure combines the two measures balancing the two aspects
- $\beta > 1$ emphasizes recall, while $\beta < 1$ emphasizes precision

F_1

$$F_1 = \frac{2(Pre * Rec)}{Pre + Rec}$$

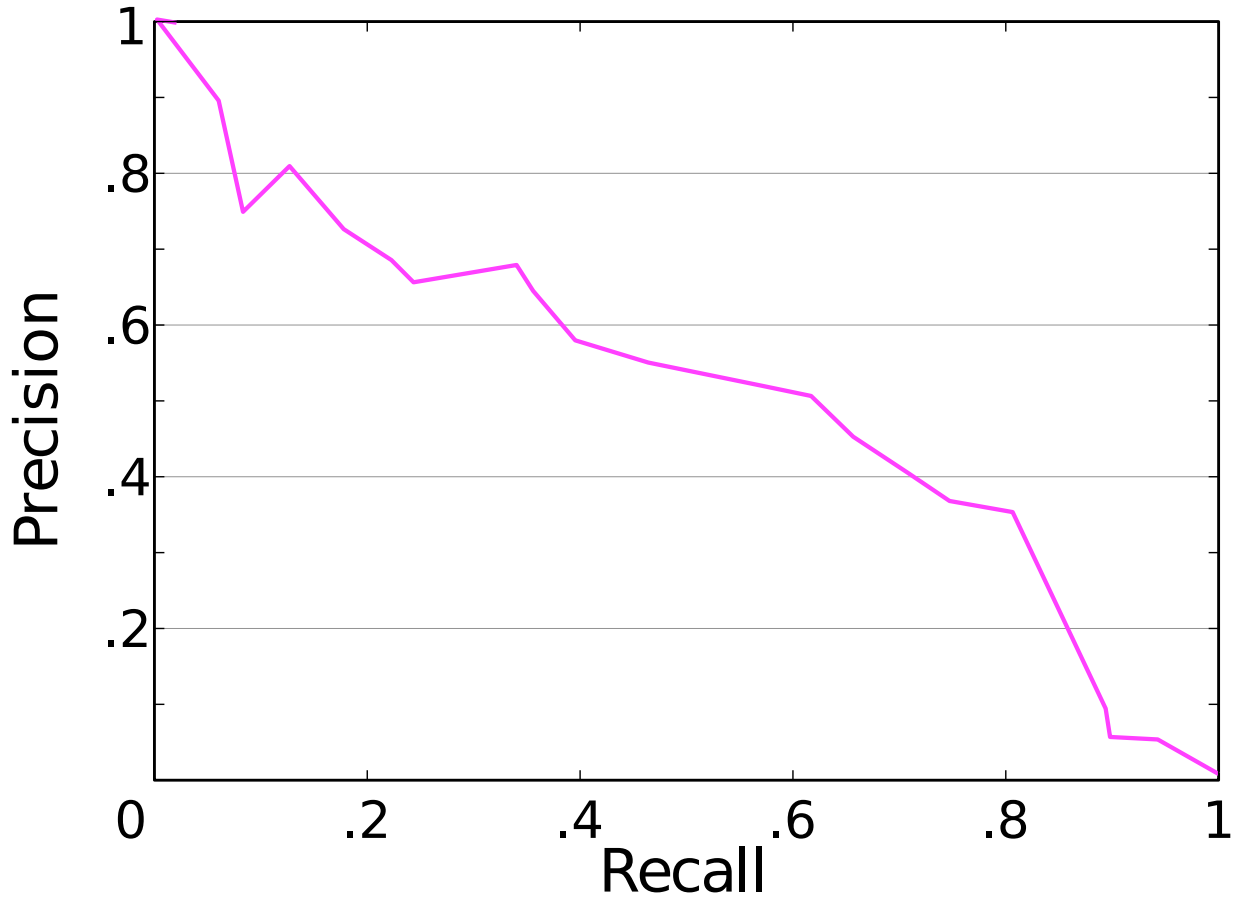
- It is the F-measure for $\beta = 1$
- It is the harmonic mean of precision and recall

Binary Classification

Precision-recall curve

- Classifiers often provide a confidence in the prediction (e.g. margin of SVM)
- A hard decision is made setting a threshold on the classifier (zero for SVM)
- Acc,Pre,Rec,F_1 all measure performance of a classifier for a specific threshold
- It is possible to change the threshold if interested in maximizing a specific performance (e.g. recall)

Binary Classification



Precision–Recall curve

- Vary the classification threshold from the most conservative to the most permissive prediction.
- Each threshold defines a different *precision–recall* operating point.
- The resulting curve summarizes the trade-off between precision and recall.
- It allows selecting an operating point that matches the application requirements (e.g., very high precision or high recall).

Binary Classification

Area Under the Precision–Recall Curve (AUPRC)

- A single numerical summary is obtained by computing the area under the precision–recall curve.
- It evaluates performance across all possible decision thresholds.
- Higher values indicate that both precision and recall remain high over a wide range of thresholds.
- Particularly informative when the positive class is rare (class imbalance).

Ranking quality versus probability quality

Threshold curves answer a ranking question
Precision–recall curves evaluate how well the score orders positive examples ahead of negative ones as the decision threshold changes.

They do not answer a probability question
A model can rank examples correctly and still be systematically overconfident or underconfident.

This motivates a different property: *calibration*.

Probability calibration

Suppose a classifier predicts $\hat{p}(Y = 1 \mid \mathbf{x}) = \hat{p}(\mathbf{x}) = 0.9$. Ideally, among many predictions made with confidence close to 0.9, approximately 90% should be correct.

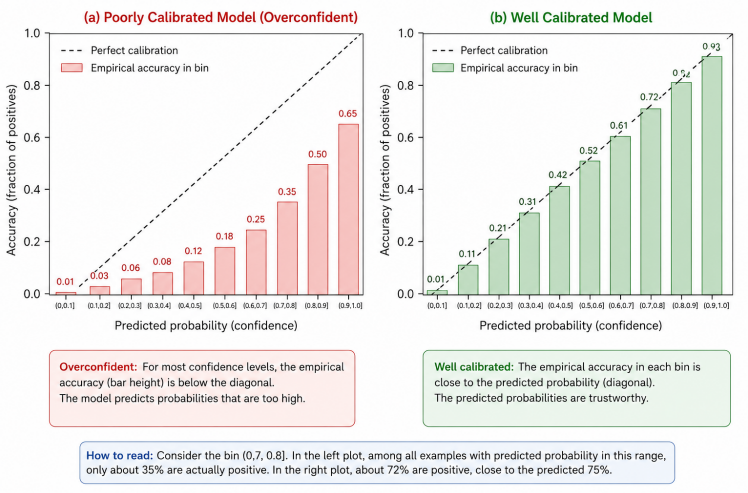
Perfect calibration

A probabilistic classifier is calibrated when

$$\Pr(Y = 1 \mid \hat{p}(\mathbf{x}) = p) = p.$$

- Calibration matters in risk-sensitive decisions and human–AI collaboration.
- Accuracy and ranking metrics do not measure calibration.
- Calibration should be evaluated on data not used to fit the model.

Reliability diagrams



Reliability diagram

Group predictions into confidence bins and compare, for each bin, average confidence with empirical accuracy. Perfect calibration lies on the diagonal.

Summarizing a Reliability Diagram

A reliability diagram provides a visual assessment of calibration.

Numerical metrics summarize the discrepancy between the bars and the diagonal.

Expected Calibration Error (ECE)

Measures the average distance between the empirical accuracy and the predicted confidence across a fixed set of confidence bins.

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} \left|\mathrm{acc}(B_m) - \mathrm{conf}(B_m)\right|.$$

Limitations

- Depends on the number of bins.
- Depends on the choice of bin boundaries.
- Sparse bins can produce unstable estimates.

Beyond Expected Calibration Error

Several alternatives have been proposed to reduce the sensitivity of ECE.

Adaptive (Equal-count) Calibration Error Constructs bins containing approximately the same number of predictions, instead of bins with equal confidence width.

- less sensitive to sparse regions;
- more stable estimates.

Brier Score

$$\mathrm{BS} = \frac{1}{n}\sum_{i=1}^{n}(p_i - y_i)^2.$$

Unlike ECE, the Brier score is a *proper scoring rule*: it evaluates the quality of the predicted probabilities directly, without relying on binning.

Evaluating Probabilistic Predictions

Take-home message

- Reliability diagrams provide the most informative picture.
- ECE summarizes the diagram with a single number, but depends on the chosen binning.
- Adaptive calibration errors reduce this dependence.
- The Brier score evaluates probabilistic predictions directly and avoids the need for binning.

Performance measures
Multiclass classification

T\P	y_1	y_2	y_3
y_1	n_{11}	n_{12}	n_{13}
y_2	n_{21}	n_{22}	n_{23}
y_3	n_{31}	n_{32}	n_{33}

- Confusion matrix is generalized version of binary one
- n_{ij} is the number of examples with class y_i predicted as y_j.
- The main diagonal contains true positives for each class
- The sum of off-diagonal elements along a column is the number of false positives for the column label
- The sum of off-diagonal elements along a row is the number of false negatives for the row label

$$FP_i = \sum_{j\neq i} n_{ji} \qquad FN_i = \sum_{j\neq i} n_{ij}$$

Performance measures

Multiclass classification

- ACC,Pre,Rec,F1 carry over to a per-class measure considering as negatives examples from other classes.
- E.g.:

$$Pre_i = \frac{n_{ii}}{n_{ii} + FP_i} \qquad Rec_i = \frac{n_{ii}}{n_{ii} + FN_i}$$

- *Multiclass accuracy* is the overall fraction of correctly classified examples:

$$MAcc = \frac{\sum_i n_{ii}}{\sum_i \sum_j n_{ij}}$$

Calibration in Multiclass Classification

Suppose a classifier predicts

$$\hat{y} = \arg\max_y p(y \mid x),$$

with confidence

$$\hat{p} = \max_y p(y \mid x).$$

The classifier is *confidence calibrated* if

$$P(Y = \hat{y} \mid \hat{p} = p) = p.$$

In other words,

Among all predictions made with confidence 80%, approximately 80% should be correct.

Calibration in Multiclass Classification

In practice

- Reliability diagrams group predictions according to their confidence $\hat{p}$.
- ECE and related metrics compare confidence with empirical accuracy within each bin.
- This is the most common notion of calibration for modern multiclass classifiers.

Regression Performance Measures

For a dataset $\mathcal{D}$ with $n = |\mathcal{D}|$:

- Mean absolute error:

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |f(\mathbf{x}_i) - y_i|$$

- Root mean squared error:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (f(\mathbf{x}_i) - y_i)^2}$$

- Coefficient of determination:

$$R^2 = 1 - \frac{\sum_i (y_i - f(\mathbf{x}_i))^2}{\sum_i (y_i - \bar{y})^2}$$

where $\bar{y} = 1/n \sum_i y_i$ is the mean output value

Regression Performance Measures

Different metrics emphasize different aspects of prediction quality.

Metric	Measures
MAE	Average prediction error. Easy to interpret and robust to outliers.
RMSE	Penalizes large errors more heavily than MAE. Useful when large errors are particularly undesirable.
R^2	Explained variance with respect to predicting the mean. Dimensionless, with $R^2 = 1$ for perfect prediction and $R^2 < 0$ indicating performance worse than the baseline.

Performance estimation
Hold-out procedure

- Computing performance measure on training set would be optimistically biased
- Need to retain an independent set on which to compute performance:

  validation set when used to estimate performance of different algorithmic settings (i.e. hyperparameters)

  test set when used to estimate final performance of selected model

- A possible split is 60%/20%/20% for training, validation, and testing; the appropriate proportions depend on dataset size

Problem

- Hold-out procedure depends on the specific test (and validation) set chosen (esp. for small datasets)

Data leakage

Evaluation is optimistically biased whenever information unavailable at prediction time influences training or model selection.

Common examples

- normalizing or imputing values before creating the split;
- selecting features using the complete dataset;
- placing records from the same patient or entity in different splits;
- using future information to predict the past;
- repeatedly tuning choices on the final test set.

Safe rule

Fit every data-dependent transformation using training data only, ideally inside a single learning pipeline.

The split must match deployment

Random splitting assumes examples are approximately independent and drawn from the same distribution.

When random folds are inappropriate

- **Temporal data:** train on the past and test on the future.
- **Grouped data:** keep all records from one patient, user, or device in the same fold.
- **Spatial or domain shift:** hold out locations, hospitals, sensors, or domains expected at deployment.

Guiding principle

The evaluation protocol should reproduce the information and distribution available when the deployed model makes a prediction.

Performance estimation

k-fold cross validation

- Split $\mathcal{D}$ in k equal sized disjoint subsets $\mathcal{D}_i$.
- For $i \in [1, k]$
  - train predictor on $\mathcal{T}_i = \mathcal{D} \setminus \mathcal{D}_i$
  - compute score S of predictor $L(\mathcal{T}_i)$ on test set $\mathcal{D}_i$: $S_i = S_{\mathcal{D}_i}[L(\mathcal{T}_i)]$
- return average score and sample standard deviation

$$\bar{S} = \frac{1}{k}\sum_{i=1}^{k} S_i \qquad \hat{\sigma} = \sqrt{\frac{1}{k-1}\sum_{i=1}^{k}(S_i - \bar{S})^2}.$$

Remark

The fold scores are not independent, since the training sets overlap. Therefore $\hat{\sigma}$ should be interpreted as the variability across folds, not as the uncertainty of the estimate.

Nested cross-validation

Using the same cross-validation results both to select hyperparameters and to report final performance introduces selection bias.

Two levels of resampling

- **Inner loop:** choose hyperparameters using only the outer training portion.
- **Outer loop:** evaluate the complete model-selection procedure on held-out data.

What is being evaluated?

Nested cross-validation estimates the performance of the entire learning procedure, including hyperparameter selection.

Take-home messages

1. Choose metrics according to the application and error costs.
2. Estimate generalization performance on data not used for fitting or model selection.
3. Evaluate ranking, decisions, and probability calibration separately.
4. Prevent leakage and choose splits that mimic deployment.
5. Use nested validation when tuning hyperparameters.

APPENDIX

Appendix
Additional reference material

Comparing learning algorithms

Hypothesis testing

- We want to compare generalization performance of two learning algorithms
- We want to know whether observed difference in performance is *statistically significant* (and not due to some noisy evaluation)
- Hypothesis testing allows to test the statistical significance of a hypothesis (e.g. the two predictors have different performance)

Hypothesis testing

Test statistic

null hypothesis H_0 default hypothesis, for rejecting which evidence should be provided

test statistic Given a sample of k realizations of random variables $X_1, \ldots, X_k$, a *test statistic* is a statistic $T = h(X_1, \ldots, X_k)$ whose value is used to decide whether to reject H_0 or not.

Example
Given a set of measurements $X_1, \ldots, X_k$, decide whether the actual value to be measured is zero.

null hypothesis the actual value is zero

test statistic sample mean:

$$T = h(X_1, \ldots, X_k) = \frac{1}{k} \sum_{i=1}^{k} X_i = \bar{X}$$

Hypothesis testing

Glossary

tail probability probability that T is at least as great (right tail) or at least as small (left tail) as the observed value t.

p-value assuming H_0 is true, the probability of observing a test statistic at least as extreme as the observed value.

Type I error reject the null hypothesis when it is true

Type II error fail to reject the null hypothesis when it is false

significance level the largest acceptable probability for committing a type I error

critical region set of values of T for which we reject the null hypothesis

critical values values on the boundary of the critical region

t-test

The test

- The test statistic is given by the standardized (also called *studentized*) mean:

$$T = \frac{\bar{X} - \mu_0}{\sqrt{\tilde{Var}[\bar{X}]}}$$

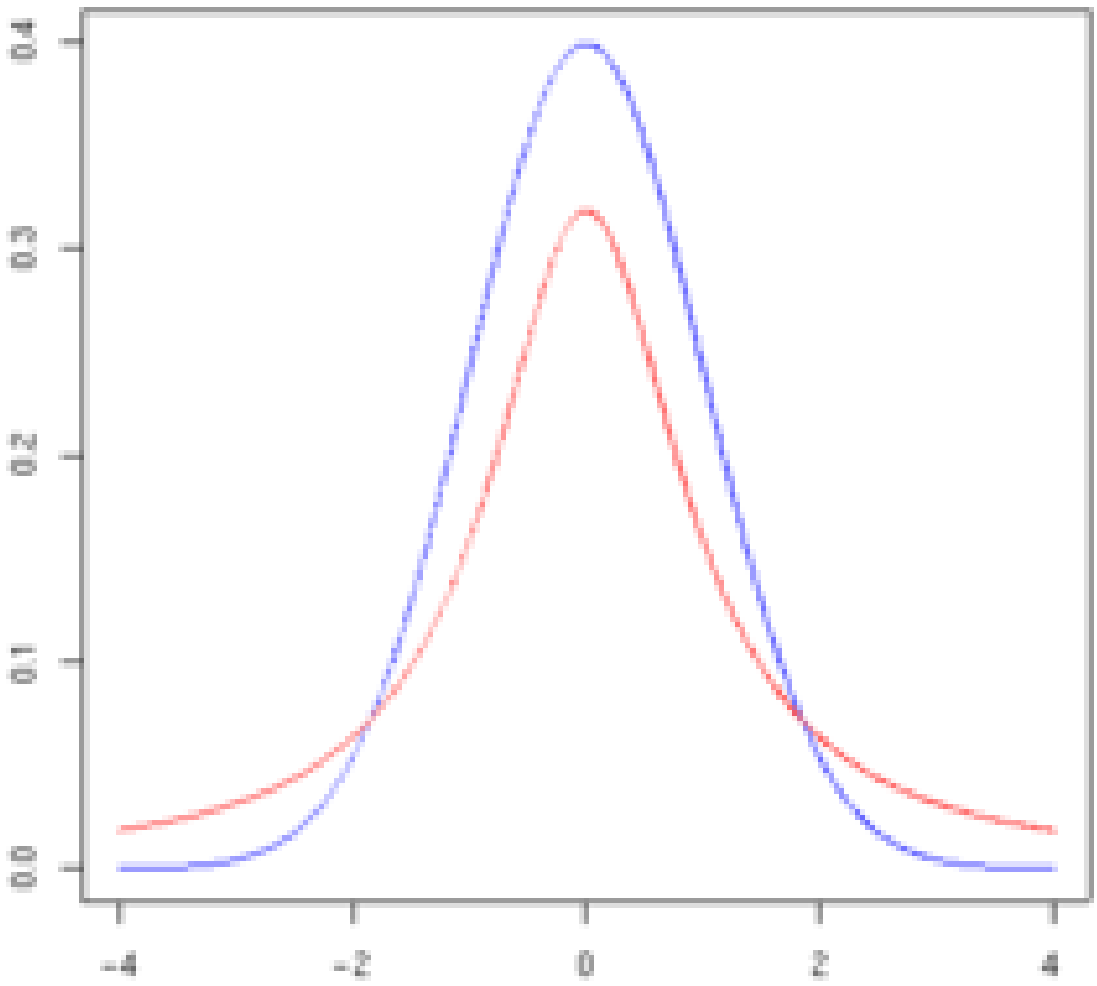
  where $\tilde{Var}[\bar{X}]$ is the approximated variance (using unbiased sample one)

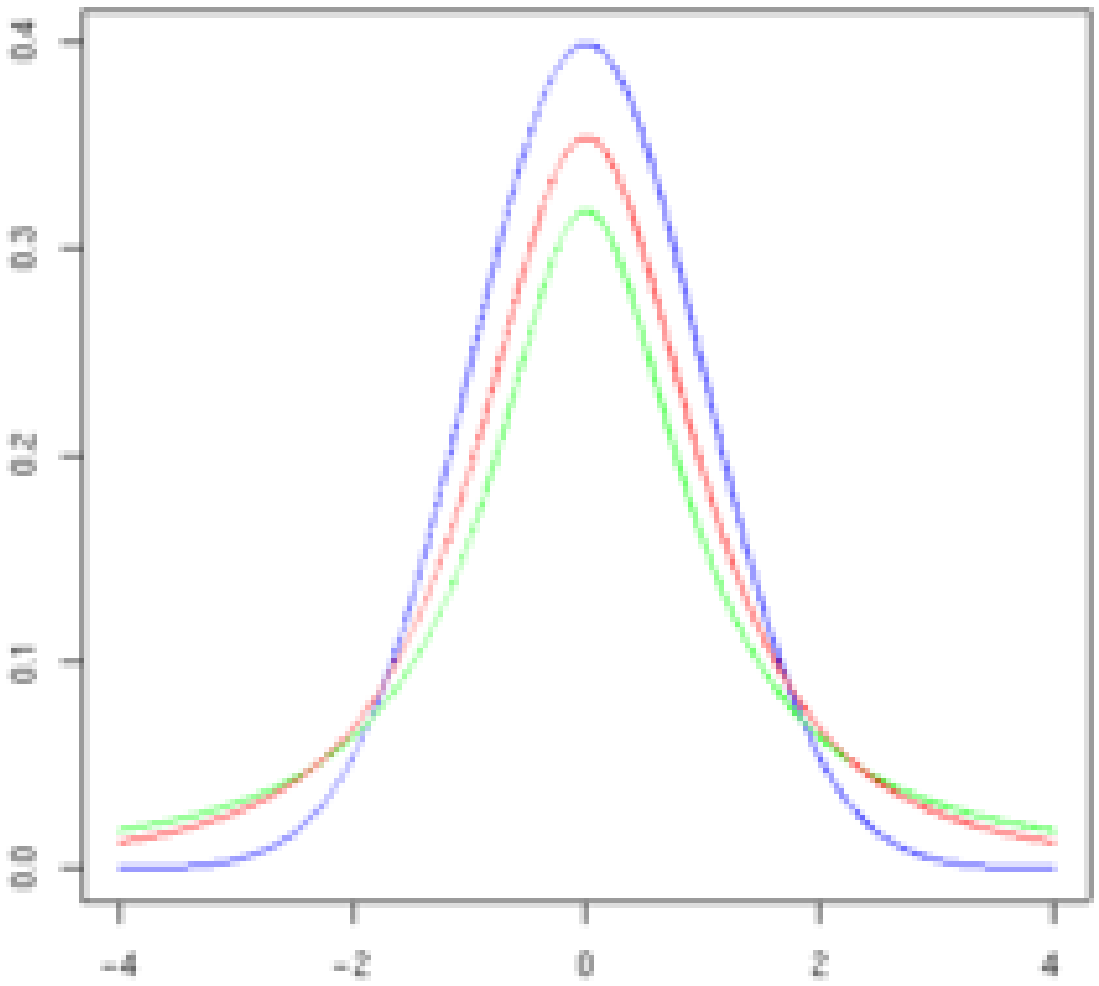
- Assuming the samples come from an unknown Normal distribution, the test statistic has a t_{k-1} distribution under the null hypothesis

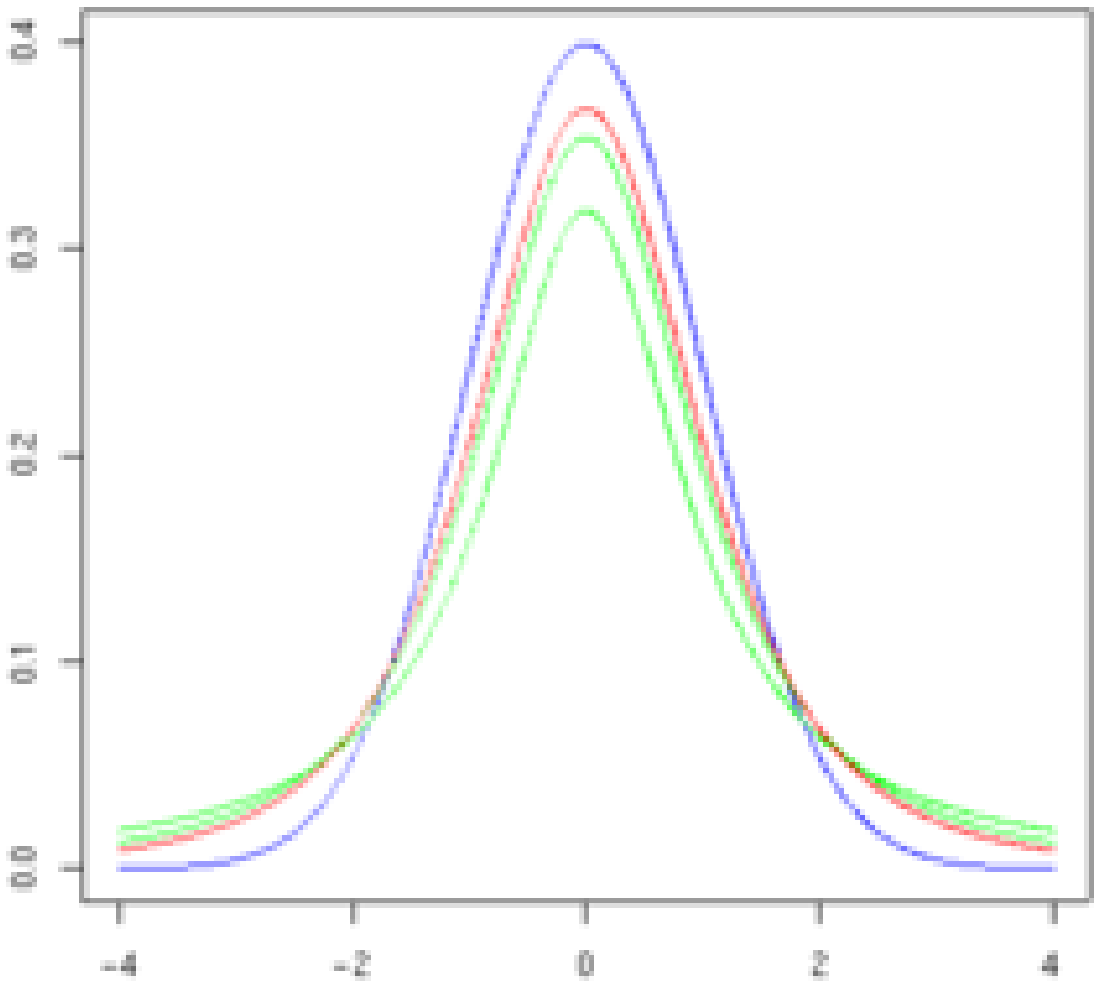
- The null hypothesis can be rejected at significance level α if:

$$T \leq -t_{k-1,\alpha/2} \qquad \text{or} \qquad T \geq t_{k-1,\alpha/2}$$

t-test


0.0
0.1
0.2
0.3
0.4
−4
−2
0
2
4


0.0
0.1
0.2
0.3
0.4
-4
-2
0
2
4



t_{k-1} *distribution*

- bell-shaped distribution similar to the Normal one
- wider and shorter: reflects greater variance due to using $\tilde{Var}[\bar{X}]$ instead of the true unknown variance of the distribution.
- $k-1$ is the number of degrees of freedom of the distribution (related to number of independent events observed)
- t_{k-1} tends to the standardized normal z for $k \to \infty$.

Comparing learning algorithms

Hypothesis testing

- Run k-fold cross validation procedure for algorithms A and B
- Compute mean performance difference for the two algorithms:

$$\hat{\delta} = \frac{1}{k}\sum_{i=1}^{k}\delta_i = \frac{1}{k}\sum_{i=1}^{k} S_{\mathcal{D}_i}[L_A(\mathcal{T}_i)] - S_{\mathcal{D}_i}[L_B(\mathcal{T}_i)]$$

- Null hypothesis is that mean difference is zero

Dependence between folds
The fold differences are correlated because the training sets overlap. The following paired t-test is a useful introduction, but its naive variance estimate can be optimistic for ordinary k-fold cross-validation.

Comparing learning algorithms: t-test

t-test

at significance level α:

$$\frac{\bar{\delta}}{\sqrt{\tilde{Var}[\bar{\delta}]}} \leq -t_{k-1,\alpha/2} \qquad \text{or} \qquad \frac{\bar{\delta}}{\sqrt{\tilde{Var}[\bar{\delta}]}} \geq t_{k-1,\alpha/2}$$

where:

$$\sqrt{\tilde{Var}[\bar{\delta}]} = \sqrt{\frac{1}{k(k-1)}\sum_{i=1}^{k}(\delta_i - \bar{\delta})^2}$$

Note

paired test the two hypotheses were evaluated over identical samples

two-tailed test if no prior knowledge can tell the direction of difference (otherwise use *one-tailed* test)

t-test example

10-fold cross validation

- Test errors:

$\mathcal{D}_i$	$S_{\mathcal{D}_i}[L_A(\mathcal{T}_i)]$	$S_{\mathcal{D}_i}[L_B(\mathcal{T}_i)]$	δ_i
$\mathcal{D}_1$	0.81	0.80	0.01
$\mathcal{D}_2$	0.82	0.77	0.05
$\mathcal{D}_3$	0.84	0.70	0.14
$\mathcal{D}_4$	0.78	0.83	-0.05
$\mathcal{D}_5$	0.85	0.80	0.05
$\mathcal{D}_6$	0.86	0.78	0.08
$\mathcal{D}_7$	0.82	0.75	0.07
$\mathcal{D}_8$	0.83	0.80	0.03
$\mathcal{D}_9$	0.82	0.78	0.04
$\mathcal{D}_{10}$	0.81	0.77	0.04

- Average error difference:

$$\bar{\delta} = \frac{1}{10}\sum_{i=1}^{10}\delta_i = 0.046$$

t-test example

10-fold cross validation

- Unbiased estimate of standard deviation:

$$\sqrt{\tilde{Var}[\bar{\delta}]} = \sqrt{\frac{1}{10 \cdot 9} \sum_{i=1}^{10} (\delta_i - \bar{\delta})^2} = 0.0154344$$

- Standardized mean error difference:

$$\frac{\bar{\delta}}{\sqrt{\tilde{Var}[\bar{\delta}]}} = \frac{0.046}{0.0154344} = 2.98$$

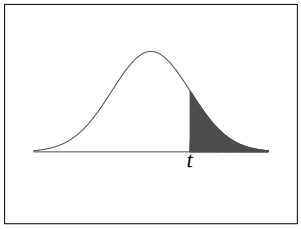
- t distribution for $\alpha = 0.05$ and $k = 10$:

$$t_{k-1,\alpha/2} = t_{9,0.025} = 2.262 < 2.98$$

- Under the assumptions of the test, reject H_0: the observed mean difference is statistically significant
- Statistical significance does not imply a practically important effect; report $\bar{\delta}$ and uncertainty together

t-test example

t-Distribution Table



The shaded area is equal to α for $t = t_\alpha$.

df	$t_{.100}$	$t_{.050}$	$t_{.025}$	$t_{.010}$	$t_{.005}$
1	3.078	6.314	12.706	31.821	63.657
2	1.886	2.920	4.303	6.965	9.925
3	1.638	2.353	3.182	4.541	5.841
4	1.533	2.132	2.776	3.747	4.604
5	1.476	2.015	2.571	3.365	4.032
6	1.440	1.943	2.447	3.143	3.707
7	1.415	1.895	2.365	2.998	3.499
8	1.397	1.860	2.306	2.896	3.355
9	1.383	1.833	2.262	2.821	3.250
10	1.372	1.812	2.228	2.764	3.169
11	1.363	1.796	2.201	2.718	3.106
12	1.356	1.782	2.179	2.681	3.055
13	1.350	1.771	2.160	2.650	3.012
14	1.345	1.761	2.145	2.624	2.977
15	1.341	1.753	2.131	2.602	2.947
16	1.337	1.746	2.120	2.583	2.921
17	1.333	1.740	2.110	2.567	2.898
18	1.330	1.734	2.101	2.552	2.878
19	1.328	1.729	2.093	2.539	2.861
20	1.325	1.725	2.086	2.528	2.845
21	1.323	1.721	2.080	2.518	2.831
22	1.321	1.717	2.074	2.508	2.819
23	1.319	1.714	2.069	2.500	2.807
24	1.318	1.711	2.064	2.492	2.797
25	1.316	1.708	2.060	2.485	2.787
26	1.315	1.706	2.056	2.479	2.779
27	1.314	1.703	2.052	2.473	2.771
28	1.313	1.701	2.048	2.467	2.763
29	1.311	1.699	2.045	2.462	2.756
30	1.310	1.697	2.042	2.457	2.750
32	1.309	1.694	2.037	2.449	2.738
34	1.307	1.691	2.032	2.441	2.728
36	1.306	1.688	2.028	2.434	2.719
38	1.304	1.686	2.024	2.429	2.712
∞	1.282	1.645	1.960	2.326	2.576

Gilles Cazelais. Typeset with LaTeX on April 20, 2006.

References

Evaluation T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning*, Springer, 2009.

Calibration A. Niculescu-Mizil and R. Caruana, "Predicting Good Probabilities with Supervised Learning", ICML, 2005.

Model comparison C. Nadeau and Y. Bengio, "Inference for the Generalization Error", Machine Learning, 2003.

Hypothesis testing T. Mitchell, *Machine Learning*, McGraw Hill, 1997 (chapter 5).