09s1: COMP9417 Machine Learning and Data Mining

Evaluating Hypotheses

May 6, 2009

Acknowledgement: Material derived from slides for the book Machine Learning, Tom M. Mitchell, McGraw-Hill, 1997 `http://www-2.cs.cmu.edu/~tom/mlbook.html`

and the book Data Mining, Ian H. Witten and Eibe Frank, Morgan Kaufmann, 2000. `http://www.cs.waikato.ac.nz/ml/weka`

Aims

This lecture will enable you to apply statistical and graphical methods to the evaluation of hypotheses in machine learning. Following it you should be able to:

- describe the problem of estimating hypothesis accuracy (error)
- define sample error and true error
- derive confidence intervals for observed hypothesis error
- compare learning algorithms using paired t-test
- define and use common evaluation measures
- generate lift charts and ROC curves

[Recommended reading: Mitchell, Chapter 5]
[Recommended exercises: 5.2 – 5.4]

Relevant WEKA programs:
weka.gui.experiment.Experimenter

Estimating Hypothesis Accuracy

- how well does a hypothesis generalize *beyond* the training set ?
  - need to estimate off-training-set error
- what is the probable error in this estimate ?
- if one hypothesis is more accurate than another on a data set, how probable is this difference in general ?

Two Definitions of Error

The **sample error** of h with respect to target function f and data sample S is the proportion of examples h misclassifies

$$error_S(h) \equiv \frac{1}{n} \sum_{x \in S} \delta(f(x) \neq h(x))$$

Where $\delta(f(x) \neq h(x))$ is 1 if $f(x) \neq h(x)$, and 0 otherwise ($cf.$ $0-1$ loss).

The **true error** of hypothesis h with respect to target function f and distribution $\mathcal{D}$ is the probability that h will misclassify an instance drawn at random according to $\mathcal{D}$.

$$error_{\mathcal{D}}(h) \equiv \Pr_{x \in \mathcal{D}}[f(x) \neq h(x)]$$

Question: How well does $error_S(h)$ estimate $error_{\mathcal{D}}(h)$?

Estimators

Experiment:

1. choose sample S of size n according to distribution $\mathcal{D}$
2. measure $error_S(h)$

$error_S(h)$ is a random variable (i.e., result of an experiment)

$error_S(h)$ is an unbiased *estimator* for $error_{\mathcal{D}}(h)$

Given observed $error_S(h)$ what can we conclude about $error_{\mathcal{D}}(h)$?

Problems Estimating Error

1. *Bias:* If S is training set, $error_S(h)$ is optimistically biased

$$bias \equiv E[error_S(h)] - error_{\mathcal{D}}(h)$$

   For unbiased estimate, h and S must be chosen independently

2. *Variance:* Even with unbiased S, $error_S(h)$ may still *vary* from $error_{\mathcal{D}}(h)$

Problems Estimating Error

Note: *Estimation bias* not to be confused with *Inductive bias* – former is a numerical quantity [comes from statistics], latter is a set of assertions [comes from concept learning].

More on this in the lecture on $ensemble$ methods.

Example

Hypothesis h misclassifies 12 of the 40 examples in S

$$error_S(h) = \frac{12}{40} = .30$$

What is $error_{\mathcal{D}}(h)$?

Confidence Intervals

If

- S contains n examples, drawn independently of h and each other
- $n \geq 30$

Then

- With approximately 95% probability, $error_{\mathcal{D}}(h)$ lies in interval

$$error_S(h) \pm 1.96\sqrt{\frac{error_S(h)(1 - error_S(h))}{n}}$$

Confidence Intervals

If

- S contains n examples, drawn independently of h and each other
- $n \geq 30$

Then

- With approximately N% probability, $error_{\mathcal{D}}(h)$ lies in interval

$$error_S(h) \pm z_N\sqrt{\frac{error_S(h)(1 - error_S(h))}{n}}$$

Confidence Intervals

Where do the z_N values come from ? Statistical tables, e.g.

$N\%$:	50%	68%	80%	90%	95%	98%	99%
z_N:	0.67	1.00	1.28	1.64	1.96	2.33	2.58

Confidence Intervals

Example:

Hypothesis h misclassifies 12 of the 40 examples in S

$$error_S(h) = \frac{12}{40} = .30$$

What is $error_{\mathcal{D}}(h)$?

Given no other information, our best estimate is $.30$

...

Confidence Intervals

Example (continued):

..., but for repeated samples of 40 examples, expect some variation in the sample error. With approximately 95% probability, $error_{\mathcal{D}}(h)$ lies in interval

$$\begin{aligned} error_S(h) \pm 1.96\sqrt{\frac{error_S(h)(1-error_S(h))}{n}} \\ = .30 \pm 1.96\sqrt{\frac{.30 \times .70}{40}} \\ = .30 \pm 1.96 \times .072 \\ = .30 \pm .14 \end{aligned}$$

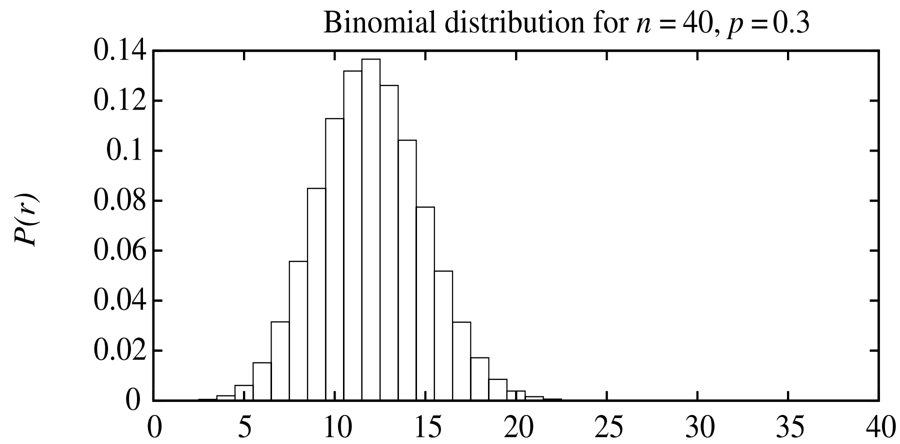
$error_S(h)$ is a Random Variable

Rerun the experiment with different randomly drawn S (of size n)

Probability of observing r misclassified examples:

$$P(r) = \frac{n!}{r!(n-r)!}\, error_{\mathcal{D}}(h)^r (1 - error_{\mathcal{D}}(h))^{n-r}$$

Binomial Probability Distribution



Binomial Probability Distribution

Probability $P(r)$ of r heads in n coin flips, if $p = \Pr(heads)$

$$P(r) = \frac{n!}{r!(n-r)!} p^r (1-p)^{n-r}$$

Binomial Probability Distribution

- Expected, or mean value of X, $E[X]$, is

$$E[X] \equiv \sum_{i=0}^{n} iP(i) = np$$

- Variance of X is

$$Var(X) \equiv E[(X - E[X])^2] = np(1-p)$$

- Standard deviation of X, σ_X, is

$$\sigma_X \equiv \sqrt{E[(X - E[X])^2]} = \sqrt{np(1-p)}$$

Examples

Suppose you test a hypothesis h and find that it commits $r = 12$ errors on a sample S of $n = 40$ randomly drawn test examples. An unbiased etimate for $error_{\mathcal{D}}(h)$ is given by $error_{\mathcal{S}}(h) = r/n = 0.3$.
The variance in this estimate arises from r alone (n is a constant).

From the Binomial distribution, this variance is $np(1-p)$.

We can substitute r/n as an estimate for p. Then the variance for r is estimated to be $40 \times 0.3(1-0.3) = 8.4$ and the standard deviation is $\sqrt{8.4} \approx 2.9$.

Therefore the standard deviation in $error_{\mathcal{S}}(h) = r/n$ is approximately $2.9/40 = 0.07$.

$error_{\mathcal{S}}(h)$ is observed to be 0.30 with standard deviation of approximately 0.07.

Examples

Suppose you test a hypothesis h and find that it commits $r = 300$ errors on a sample S of $n = 1000$ randomly drawn test examples. What is the standard deviation in $error_S(h)$?

The standard deviation for r is estimated to be $\sqrt{1000 \times 0.3(1 - 0.3)} \approx 14.5$.

Therefore the standard deviation in $error_S(h) = r/n$ is approximately $14.5/1000 = .0145$.

$error_S(h)$ is observed to be 0.30 with standard deviation of approximately $.0145$.

Normal Distribution Approximates Binomial

$error_S(h)$ follows a *Binomial* distribution, with

- mean $\mu_{error_S(h)} = error_{\mathcal{D}}(h)$
- standard deviation $\sigma_{error_S(h)}$

$$\sigma_{error_S(h)} = \sqrt{\frac{error_{\mathcal{D}}(h)(1 - error_{\mathcal{D}}(h))}{n}}$$

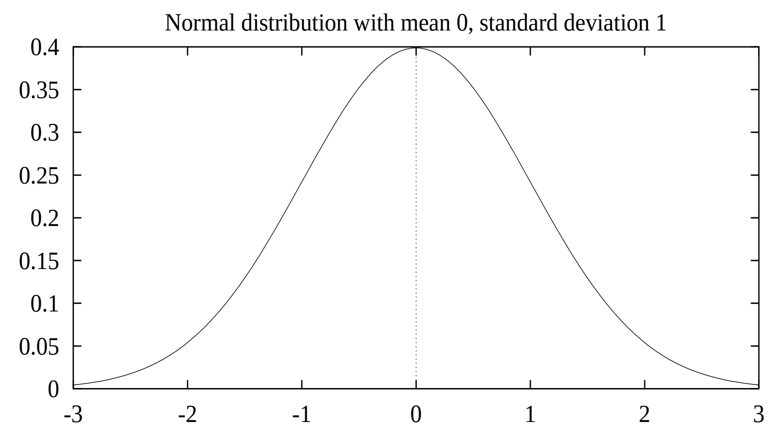
Normal Distribution Approximates Binomial

Approximate this by a *Normal* distribution with

- mean $\mu_{error_S(h)} = error_{\mathcal{D}}(h)$
- standard deviation $\sigma_{error_S(h)}$

$$\sigma_{error_S(h)} \approx \sqrt{\frac{error_S(h)(1 - error_S(h))}{n}}$$

Normal Probability Distribution



$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}$$

Normal Probability Distribution

The probability that X will fall into the interval (a, b) is given by

$$\int_a^b p(x)dx$$

- Expected, or mean value of X, $E[X]$, is

$$E[X] = \mu$$

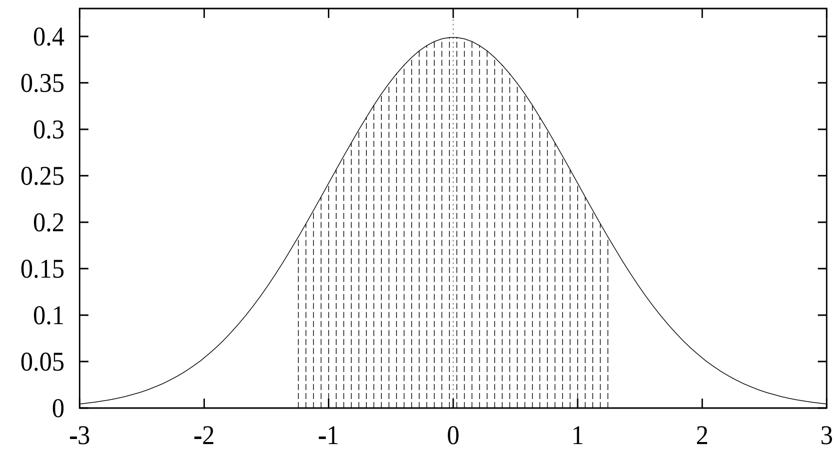
- Variance of X is

$$Var(X) = \sigma^2$$

- Standard deviation of X, σ_X, is

$$\sigma_X = \sigma$$

Normal Probability Distribution



Normal Probability Distribution

80% of area (probability) lies in $\mu \pm 1.28\sigma$

N% of area (probability) lies in $\mu \pm z_N\sigma$

$N\%$:	50%	68%	80%	90%	95%	98%	99%
z_N:	0.67	1.00	1.28	1.64	1.96	2.33	2.58

Note: with 80% confidence the value of the random variable will lie in the two-sided interval $[-1.28, 1.28]$.

With 10% confidence it will lie to the right of this interval (resp. left).

With 90% confidence it will lie in the one-sided interval $[-\infty, 1.28]$

Let α be the probability that the value lies *outside* the interval.

Then a $100(1-\alpha)\%$ two-sided confidence interval with lower-bound L and upper-bound U can be converted into a $100(1-(\alpha/2))\%$ one-sided confidence interval with lower bound L and no upper bound (resp. upper bound U and no lower bound).

Confidence Intervals, More Correctly

If

- S contains n examples, drawn independently of h and each other
- $n \geq 30$

Then

- With approximately 95% probability, $error_S(h)$ lies in interval

$$error_{\mathcal{D}}(h) \pm 1.96\sqrt{\frac{error_{\mathcal{D}}(h)(1 - error_{\mathcal{D}}(h))}{n}}$$

Confidence Intervals, More Correctly

equivalently, $error_{\mathcal{D}}(h)$ lies in interval

$$error_S(h) \pm 1.96\sqrt{\frac{error_{\mathcal{D}}(h)(1 - error_{\mathcal{D}}(h))}{n}}$$

which is approximately

$$error_S(h) \pm 1.96\sqrt{\frac{error_S(h)(1 - error_S(h))}{n}}$$

Central Limit Theorem

Consider a set of independent, identically distributed random variables $Y_1 \ldots Y_n$, all governed by an arbitrary probability distribution with mean μ and finite variance σ^2. Define the sample mean,

$$\bar{Y} \equiv \frac{1}{n}\sum_{i=1}^{n} Y_i$$

Central Limit Theorem. As $n \rightarrow \infty$, the distribution governing $\bar{Y}$ approaches a Normal distribution, with mean μ and variance $\frac{\sigma^2}{n}$.

the sum of a large number of independent, identically distributed (i.i.d) random variables follows a distribution that is approximately Normal.

Calculating Confidence Intervals

1. Pick parameter p to estimate

- $error_{\mathcal{D}}(h)$

2. Choose an estimator

- $error_S(h)$

3. Determine probability distribution that governs estimator

- $error_S(h)$ governed by Binomial distribution, approximated by Normal when $n \geq 30$

4. Find interval (L, U) such that N% of probability mass falls in the interval

- Use table of z_N values

Difference Between Hypotheses

Two classifiers h_1, h_2. Test h_1 on sample S_1, test h_2 on S_2.

Apply the four-step procedure:

1. Pick parameter to estimate

$$d \equiv error_{\mathcal{D}}(h_1) - error_{\mathcal{D}}(h_2)$$

2. Choose an estimator

$$\hat{d} \equiv error_{S_1}(h_1) - error_{S_2}(h_2)$$

Difference Between Hypotheses

3. Determine probability distribution that governs estimator

$$\sigma_{\hat{d}} \approx \sqrt{\frac{error_{S_1}(h_1)(1 - error_{S_1}(h_1))}{n_1} + \frac{error_{S_2}(h_2)(1 - error_{S_2}(h_2))}{n_2}}$$

4. Find interval (L, U) such that N% of probability mass falls in the interval

$$\hat{d} \pm z_N \sqrt{\frac{error_{S_1}(h_1)(1 - error_{S_1}(h_1))}{n_1} + \frac{error_{S_2}(h_2)(1 - error_{S_2}(h_2))}{n_2}}$$

Paired t test to compare h_A,h_B

1. Partition data into k disjoint test sets $T_1, T_2, \ldots, T_k$ of equal size, where this size is at least 30.
2. For i from 1 to k, do

   $$\delta_i \leftarrow error_{T_i}(h_A) - error_{T_i}(h_B)$$

3. Return the value $\bar{\delta}$, where

$$\bar{\delta} \equiv \frac{1}{k} \sum_{i=1}^{k} \delta_i$$

sample mean of the difference in error between the 2 learning methods.

Paired t test to compare h_A,h_B

N% confidence interval estimate for d (difference between the true errors of the hypotheses):

$$\bar{\delta} \pm t_{N,k-1}\ s_{\bar{\delta}}$$

$$s_{\bar{\delta}} \equiv \sqrt{\frac{1}{k(k-1)} \sum_{i=1}^{k} (\delta_i - \bar{\delta})^2}$$

where $s_{\bar{\delta}}$ is the estimated standard deviation.

Note δ_i approximately Normally distributed

Comparing learning algorithms L_A and L_B

What we'd like to estimate:

$$E_{S \subset \mathcal{D}}[error_{\mathcal{D}}(L_A(S)) - error_{\mathcal{D}}(L_B(S))]$$

where $L(S)$ is the hypothesis output by learner L using training set S

i.e., the expected difference in true error between hypotheses output by learners L_A and L_B, when trained using randomly selected training sets S drawn according to distribution $\mathcal{D}$.

Comparing learning algorithms L_A and L_B

But, given limited data D_0, what is a good estimator?

- could partition D_0 into training set S and training set T_0, and measure

$$error_{T_0}(L_A(S_0)) - error_{T_0}(L_B(S_0))$$

- even better, repeat this many times and average the results (next slide)

Comparing learning algorithms L_A and L_B

1. Partition data D_0 into k disjoint test sets $T_1, T_2, \ldots, T_k$ of equal size, where this size is at least 30.
2. For i from 1 to k, do

   use T_i for the test set, and the remaining data for training set S_i
   - $S_i \leftarrow \{D_0 - T_i\}$
   - $h_A \leftarrow L_A(S_i)$
   - $h_B \leftarrow L_B(S_i)$
   - $\delta_i \leftarrow error_{T_i}(h_A) - error_{T_i}(h_B)$
3. Return the value $\bar{\delta}$, where

$$\bar{\delta} \equiv \frac{1}{k} \sum_{i=1}^{k} \delta_i$$

Comparing learning algorithms L_A and L_B

Notice we'd like to use the paired t test on $\bar{\delta}$ to obtain a confidence interval

but not really correct, because the training sets in this algorithm are not independent (they overlap!)

more correct to view algorithm as producing an estimate of

$$E_{S \subset D_0}[error_{\mathcal{D}}(L_A(S)) - error_{\mathcal{D}}(L_B(S))]$$

instead of

$$E_{S \subset \mathcal{D}}[error_{\mathcal{D}}(L_A(S)) - error_{\mathcal{D}}(L_B(S))]$$

but even this approximation is better than no comparison

Parameter tuning

- It is important that the test data is not used *in any way* to create the classifier
- Some learning schemes operate in two stages:
  - Stage 1: builds the basic structure
  - Stage 2: optimizes parameter settings
- The test data can't be used for parameter tuning!
- Proper procedure uses *three* sets: $training\ data$, $validation\ data$, and $test\ data$
- Validation data is used to optimize parameters

Making the most of the data

- Once evaluation is complete, *all the data* can be used to build the final classifier
- Generally, the larger the training data the better the classifier (but returns diminish)
- The larger the test data the more accurate the error estimate
- *Holdout* procedure: method of splitting original data into training and test set
  - Dilemma: ideally we want both, a large training and a large test set

Loss functions

- Most common performance measure: predictive accuracy ($cf.$ sample error)
- Also called $0-1$ $loss\ function$:

$$\sum_i \begin{cases} 0 & \text{if prediction is correct} \\ 1 & \text{if prediction is incorrect} \end{cases}$$

- Classifiers can produce *class probabilities*
- What is the accuracy of the probability estimates ?
- 0-1 loss is not appropriate

Quadratic loss function

- $p_1, \ldots, p_k$ are probability estimates of all possible outcomes for an instance
- c is the index of the instance's actual class
- i.e. $a_1, \ldots, a_k$ are zero, except for a_c which is 1
- the *quadratic loss* is:

$$E\left[\sum_j (p_j - a_j)^2\right] = \left(\sum_{j \neq c} p_j^2\right) + (1 - p_c)^2$$

- leads to preference for predictors giving best guess at true probabilities

Informational loss function

- the informational loss function is $-\log(p_c)$, where c is the index of the actual class of an instance
- number of bits required to communicate the actual class
- if $p_1^*, \ldots, p_k^*$ are the true class probabilities
- then the expected value of the informational loss function is:

$$-p_1^* \log_2(p_1) - \ldots - p_k^* \log_2(p_k)$$

- which is minimized for $p_j = p_j^*$
- giving the *entropy* of the true distribution

$$-p_1^* \log_2(p_1^*) - \ldots - p_k^* \log_2(p_k^*)$$

Which loss function ?

- quadratic loss functions takes into account all the class probability estimates for an instance
- informational loss focuses only on the probability estimate for the actual class
- quadratic loss is bounded by $1 + \sum_j p_j^2$, can never exceed 2
- informational loss can be infinite
- informational loss related to MDL principle (can use bits for complexity as well as accuracy)

Costs of predictions

- In practice, different types of classification errors often incur different costs
- Examples:
  - Medical diagnosis (has cancer vs. not)
  - Loan decisions
  - Fault diagnosis
  - Promotional mailing

Confusion matrix

Two-class prediction case:

Actual Class	Predicted Class: Yes	Predicted Class: No
Yes	True Positive (TP)	False Negative (FN)
No	False Positive (FP)	True Negative (TN)

Two kinds of error:
False Positive and False Negative may have different costs.

Two kinds of correct prediction:
True Positive and True Negative may have different "benefits".

Note: total number of test set examples $N = TP + FN + FP + TN$

Common evaluation measures

Accuracy

$$\frac{TP + TN}{N}$$

Error rate

equivalent to 1 - Accuracy, i.e.,

$$\frac{FP + FN}{N}$$

Precision

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

(also called: **Correctness**, **Positive Predictive Value**)

Recall

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

(also called: TP **rate**, **Hit rate**, **Sensitivity**, **Completeness**)

Sensitivity

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

Specificity

equivalent to 1 - FP rate

$$\frac{TN}{TN+FP}$$

(also called: TN **rate**)

True Positive (TP) Rate

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

False Positive (FP) Rate

equivalent to 1 - Specificity, i.e.,

$$\frac{FP}{FP+TN}$$

(also called: **False alarm rate**)

Negative Predictive Value

$$\frac{TN}{TN+FN}$$

Coverage

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

Note:

- this is not an exhaustive list ...
- same measures used under different names in different disciplines

Common evaluation measures

Actual Class	Predicted Class: Yes	Predicted Class: No
Yes	TP	FN
No	FP	TN

E.g., in concept learning, the number of instances in a sample predicted to be in (resp. not in) the concept is the sum of the first (resp. second) column.

The number of positive (resp. negative) examples of the concept in a sample is the sum of the first (resp. second) row.

$$N_{\text{pred}} = TP + FP \qquad N_{\text{not\_pred}} = FN + TN$$
$$N_{\text{pos}} = TP + FN \qquad N_{\text{neg}} = FP + TN$$

Common evaluation measures

We can treat the evaluation measures as conditional probabilities:

$$
\begin{aligned}
P(\text{pred} \mid \text{pos}) &= \tfrac{TP}{TP+FN} && \text{(Sensitivity)} \\
P(\text{pred} \mid \text{neg}) &= \tfrac{FP}{FP+TN} && \text{(FP rate)} \\
P(\text{not\_pred} \mid \text{pos}) &= \tfrac{FN}{TP+FN} && \text{(FN rate)} \\
P(\text{not\_pred} \mid \text{neg}) &= \tfrac{TN}{FP+TN} && \text{(Specificity)} \\
P(\text{pos} \mid \text{pred}) &= \tfrac{TP}{TP+FP} && \text{(Pos. Pred. Value)} \\
P(\text{neg} \mid \text{pred}) &= \tfrac{FP}{TP+FP} && \\
P(\text{pos} \mid \text{not\_pred}) &= \tfrac{FN}{FN+TN} && \text{(FN rate)} \\
P(\text{neg} \mid \text{not\_pred}) &= \tfrac{TN}{FN+TN} && \text{(Neg. Pred. Value)}
\end{aligned}
$$

Trade-off

Trade-off

good coverage of positive examples: increase TP at risk of increasing FP i.e. increase generality

good proportion of positive examples: decrease FP at risk of decreasing TP i.e. decrease generality, i.e. increase specificity

Different techniques give different trade-offs and can be plotted as *two different lines* on any of the graphical charts: Lift, ROC or recall-precision curves.

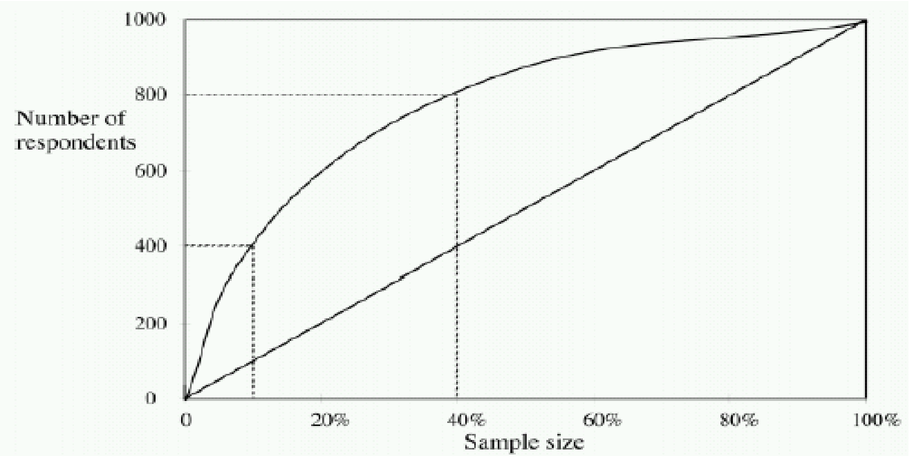
Lift charts

- In practice, costs are rarely known precisely
- Instead decisions often made by comparing possible scenarios
- Lift comes from market research, where a typical goal is to identify a "profitable" target sub-group out of the total population
- Example: promotional mailout to population of 1,000,000 potential respondents
  - Baseline is that 0.1% of all households in total population will respond (1000)
  - Situation 1: classifier 1 identifies target sub-group of 100,000 most promising households of which 0.4% will respond (400)
  - Situation 2: classifier 2 identifies target sub-group of 400,000 most promising households of which 0.2% will respond (800)

Lift charts

- Lift $= \frac{\text{response rate of target sub-group}}{\text{response rate of total population}}$
- Situation 1 gives lift of $\frac{0.4}{0.1} = 4$
- Situation 2 gives lift of $\frac{0.2}{0.1} = 2$
- Note that which situation is more profitable depends on cost estimates
- A lift chart allows for a visual comparison

Hypothetical Lift Chart



Generating a lift chart

Instances are sorted according to their predicted probability of being a true positive:

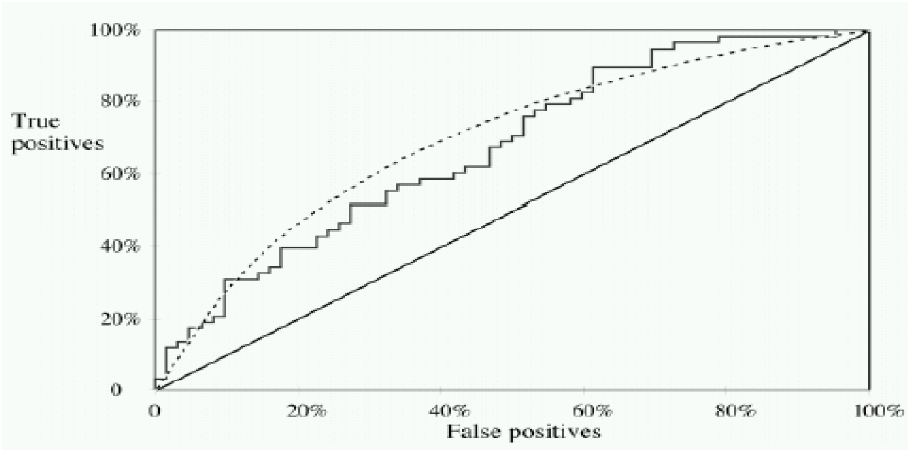
Rank	Predicted probability	Actual class
1	0.95	Yes
2	0.93	Yes
3	0.93	No
4	0.88	Yes
...	...	...

In lift chart, x axis is sample size and y axis is number of true positives

ROC curves

- ROC curves are similar to lift charts
  - ROC stands for receiver operating characteristic
  - Used in signal detection to show tradeoff between hit rate and false alarm rate over noisy channel
- Differences to lift chart:
  - y axis shows percentage of true positives in sample (rather than absolute number)
  - x axis shows percentage of false positives in sample (rather than sample size)

A sample ROC curve



Numeric prediction evaluation measures

Based on differences between predicted (p_i) and actual (a_i) values on a test set of n examples:

Mean squared error

$$\frac{(p_1 - a_1)^2 + \ldots + (p_n - a_n)^2}{n}$$

Root mean squared error

$$\sqrt{\frac{(p_1 - a_1)^2 + \ldots + (p_n - a_n)^2}{n}}$$

Numeric prediction evaluation measures

Mean absolute error

$$\frac{|p_1 - a_1| + \ldots + |p_n - a_n|}{n}$$

Relative absolute error

$$\frac{|p_1 - a_1| + \ldots + |p_n - a_n|}{|a_1 - \bar{a}| + \ldots + |a_n - \bar{a}|}, \text{ where } \bar{a} = \frac{1}{n}\sum_i a_i$$

plus others, see, e.g., Weka

Summary

- Evaluation for machine learning and data mining is a complex issue
- Many accepted methods not theoretically well-founded ...
- ... but have been found to work well in practice, e.g.,
  - 10 $\times$ 10-fold cross-validation
  - corrected resampled t-test (Weka)