

HW 1

Perceptron classification of handwritten digits
Due Wednesday Jan 21 at 5pm.

CS 445/545
Machine Learning

Reading: T. M. Mitchell, Chapter 5 (will be on
electronic reserve at library)

Homework write-up and data on class web site

Data set

- From UC Irvine Machine Learning Repository:
<http://www.ics.uci.edu/~mllearn/MLRepository.html>

Data description: optdigits.names

Training data: optdigits.tra

Test data: optdigits.tes

Introduction to Matlab

- Tutorial and documentation linked from class web page
- Available on MCECS Linux, Unix, Windows, and Mac platforms
- Can run remotely (without GUI)
- Can use tools from Stats toolbox but write your own perceptron code (don't use Netlab for now) .

Training a perceptron

Start with random weights, $\mathbf{w} = (w_1, w_2, \dots, w_d)$.

Repeat for M epochs

Repeat for each training example (\mathbf{x}^n, t^n) .

Run the perceptron with input \mathbf{x}^n and weights \mathbf{w} to obtain o .

Update the weights:

$$w_i \leftarrow w_i + \Delta w_i$$

where

$$\Delta w_i = \eta (t^n - o^n) x_i^n$$

Some Issues

- Parameters to set: M, η

- How to initialize weights?

Some Issues

- Parameters to set: M, η

Let $\eta = 0.2$ (a guess)

For each perceptron, run learning algorithm for as many epochs it takes to get to $\leq 10\%$ error on training set.

- How to initialize weights?

From uniform random distribution between -1 and $+1$.

K-fold cross-validation algorithm (from reading)

1. Partition data D_0 into k disjoint test sets T_1, T_2, \dots, 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*
 - $S_i \leftarrow \{D_0 - T_i\}$
 - $h_A \leftarrow L_A(S_i)$
 - $h_B \leftarrow L_B(S_i)$
 - $\delta_i \leftarrow \text{error}_{T_i}(h_A) - \text{error}_{T_i}(h_B)$
3. Return the value $\bar{\delta}$, where

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

Presentation assignments

- In general, we'll cover the basic ideas/theory underlying a machine learning method, and then one or two recent papers on applications of that method and available tools for implementing that method.

Presentation assignments

- **Possible topics**

- **Matlab:** Loading data, using PCA to transform data , doing paired-t test, making plots (present on Wednesday, 1/14) (need 1)
 - **Kernel methods and support vector machines** (present on Wednesday, 1/14 or Wednesday, 1/21) (need approx 4)
 - **Bayesian networks** (need approx 5)
 - **Ensemble learning** (need approx 4)
 - **Unsupervised learning** (approx 4)
 - **Temporal learning** (need approx 4)
 - **Dimensionality reduction** (need approx 4)
 - **Other topics** (specify) (need approx 4)
- Put your name under three topics, with preference (1), (2), or (3)**

- For support vector machine presentations:

*Support Vector Machines and Kernels for
Computational Biology,*

pp. 1—5 (up to Kernels)

pp. 5—10 (Kernels to end)

Evaluating Hypotheses

- What is this topic about?
 - Mainly a review of some statistics and its application to machine learning
- What questions do we want to answer?
 - What is expected accuracy of a classification algorithm (i.e., over all instances in the instance space)?
 - Given two classification algorithms, which one has better accuracy for a given application, and how much better is it?

- Framework:

- An instance space X .
- A probability distribution D over instances in X .
- A hypothesis h that makes binary classifications on instances $\mathbf{x} \in X$.
- Sample $error_S(h)$ on a sample $S \subseteq X$, where

$$error_S(h) = \Pr_{\mathbf{x} \in S} [\text{incorrect}(h(\mathbf{x}))]$$

- True error $error_D(h)$ of h over x , given distribution D , where

$$error_D(h) = \Pr_{\mathbf{x} \in X} [\text{incorrect}(h(\mathbf{x}))]$$

Question to be addressed today

How well does $error_S(h)$ estimate $error_D(h)$? I.e., can we put a confidence interval around $error_S(h)$ that gives $N\%$ confidence that $error_D(h)$ is in that interval?

That is, we want to say something like:

“With 95% confidence, $error_D(h)$ lies in the interval $error_S(h) \pm M$.”

- Basics of sampling theory:

We collect a random sample S of n independently drawn examples from distribution D .

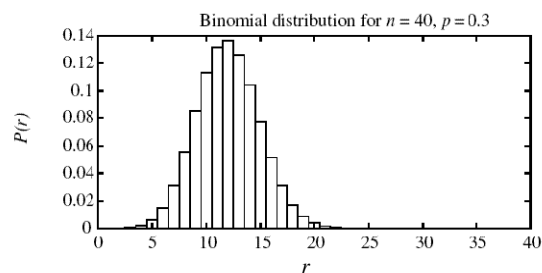
We measure sample error $error_S(h)$.

If we repeated this many times on different samples S_1, S_2, \dots, S_k , would get different values for $error_S(h)$. Thus $error_S(h)$ can be considered a random variable.

What is distribution of $error_S(h)$?

Binomial distribution.

When is a distribution Binomial?



- n identical trials
- Outcome of trial can be one of two possible values, say 0 and 1
- Probability of 1 on a single trial is given by constant p .
- Trials are independent of one another.

- Let p be the probability that $h(\mathbf{x})$ will be an error for a given $\mathbf{x} \in S$. Let r be the number of errors observed over sample S . Let R be a binomially distributed random variable corresponding to r .

$$E[R] = np$$

$$\text{Var}[R] = np(1-p).$$

$$\sigma_R = \sqrt{np(1-p)}$$

$$\sigma_{\text{error}_S(h)} = \frac{\sigma_R}{n} = \frac{\sqrt{np(1-p)}}{n} = \sqrt{\frac{p(1-p)}{n}}$$

- We can approximate:

$$p \approx \frac{r}{n} = \text{error}_S(h)$$

- So:

$$\sigma_{\text{error}_S(h)} \approx \sqrt{\frac{\text{error}_S(h)(1-\text{error}_S)}{n}}$$

Example (again)

- Suppose that after training, your classifier h misclassifies 12 out of 72 test examples.
- $error_S(h) = r / n = 12/72 = 0.17$

$$\begin{aligned}\sigma_{error_S(h)} &= \sqrt{\frac{p(1-p)}{n}} \approx \sqrt{\frac{(r/n)(1-r/n)}{n}} = \sqrt{\frac{error_S(h)(1-error_S(h))}{n}} \\ &= \sqrt{\frac{.17 \times .83}{72}} \approx .04\end{aligned}$$

Confidence intervals

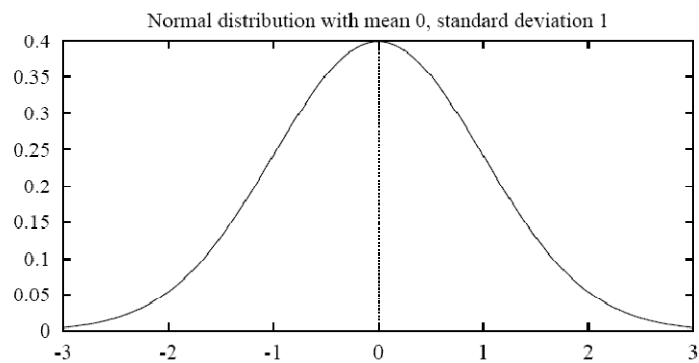
- We want to make a statement like:

“With 95% probability, the true error $error_D(h)$ lies in the interval

$$error_S(h) \pm f(error_S(h), n)$$

Confidence intervals

- Now, what is this function f ?
 - We know that $error_S(h)$ is binomially distributed, with mean $error_D(h)$ and standard deviation $\sqrt{\frac{error_S(h)(1-error_S(h))}{n}}$.
 - Need to find the interval centered around value $error_S(h)$ that contains $N\%$ of the total probability under this distribution. This interval will contain $error_D(h)$ $N\%$ of the time.
 - Hard to calculate exactly for Binomial distributions, but a good approximation (at large sample sizes) can be obtained by approximating the Binomial distribution with the Normal (Gaussian) distribution.



Central Limit Theorem: Sum of a large number of independent, identically distributed (iid) random variables follows a distribution that is approximately Normal.

For large n , any binomial distribution is closely approximated by a Normal distribution with the same mean and variance.

Example: Consider the interval centered around μ that contains 95% of the probability mass.

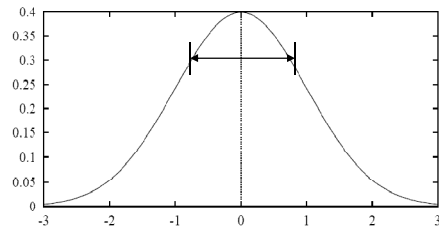
Statistics tables tell us that any new sample $error_S(h)$ has 95% probability of being at most 1.96 standard deviations from μ .

Equivalently, given $error_S(h)$, μ has 95% probability of being at most 1.96 standard deviations from $error_S(h)$.

Thus we can say that with 95% confidence,

$$error_S(h) \pm 1.96 \sigma$$

contains μ (= true error $error(h)$)

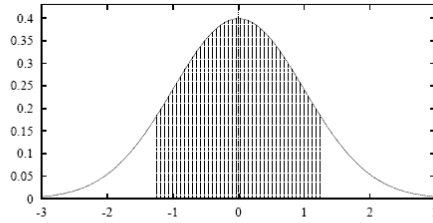


- Statistics tables give size of interval about the mean that contains $N\%$ of the probability mass under the Normal distribution.

- Example table:

Confidence Level $N\%$	50%	68%	80%	90%	95%	98%	99%
z_N	0.67	1.0	1.28	1.64	1.96	2.33	2.58

where z_N is half the width (measured in standard deviations) of the interval about the mean that contains $N\%$ of the total probability mass in the 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

Now, suppose random variable Y follows a Normal distribution, and we have a measured value y from Y . We can say:

y will fall into interval $\mu \pm z_N\sigma$ N% of the time, or

μ will fall into interval $y \pm z_N\sigma$ N% of the time

- Now we can derive general expression for N% confidence intervals for discrete-valued hypotheses.

$error_S(h)$ follows Binomial distribution with $\mu = error_D(h)$ and $\sigma = \sigma_R/n$.

- Approximate with Normal distribution with same mean and standard deviation.

N % confidence interval for discrete hypotheses:

With N % confidence, $error_D(h)$ is in the interval

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

Our two approximations ($error_D(h) \approx error_S(h)$ and Binomial distribution \approx Normal distribution) are good as long as $n \geq 30$.

Example (From reading)

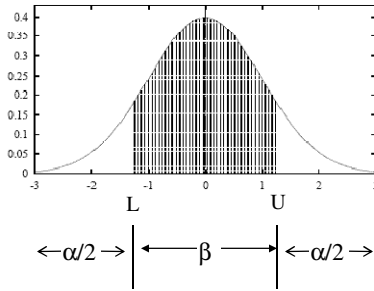
Suppose you test a hypothesis h and find that it commits $r=10$ errors on a sample S of $n=65$ randomly drawn test examples. What is the 90% two-sided confidence interval for the true error rate?

One-sided confidence bounds

In ML, usually care only about upper bound on error, not lower bound. E.g., we want to say,

“With N % confidence, $error_D(h) \leq U$ ”

- Note that Normal distribution is symmetric about mean, so any two-sided confidence interval based on a Normal distribution can be converted to corresponding one-sided interval with twice the confidence.



Let β be the probability that the error is in the interval $[L,U]$. I.e., $\beta = N/100$. Let $\alpha = 1 - \beta$. Then what is probability that the error is above U ? $\alpha/2$. So probability that error is has upper bound U and no lower bound is $(1 - \alpha/2)$.

In other words, a $100(1 - \alpha)\%$ confidence interval with lower bound L and upper bound U implies a $100(1 - \alpha/2)\%$ confidence interval with upper bound U and no lower bound.

Example

Suppose h 's error rate is measured as $12/40$.

Two-sided 95% confidence interval:

$$\begin{aligned} & error_s(h) \pm z_N \sqrt{\frac{error_s(h)(1 - error_s(h))}{n}} \\ & = .3 \pm 1.96 \sqrt{\frac{.3 \times .7}{40}} = .3 \pm .14 \end{aligned}$$

What is corresponding one-sided confidence interval?

$95\% = 100(1 - \alpha)$, so $\alpha = .05$.

$100(1 - \alpha/2) = 97.5\%$ confidence that $error(h) \leq .3 + 0.14 = .44$.

Exercise: Suppose h commits 10 errors over 50 examples.

(a) What is 90% confidence interval (two-sided) for the true error rate?

(b) What is the 95% one-sided interval (i.e., what is the upper bound U such that $error(h) \leq U$ with 95% confidence)?

(c) What is U for the 90% one-sided interval?

Difference in error of two hypotheses

- Given h_1 and h_2 for some discrete-valued target function, and S_1 and S_2 (with n_1 and n_2 examples, respectively) both drawn from distribution D .
- We have $error_{S_1}(h_1)$ and $error_{S_2}(h_2)$.
- We want to estimate

$$d = error_D(h_1) - error_D(h_2).$$

with a confidence interval.

Here's what we do:

1. Define estimator \hat{d} for d :

$$\hat{d} = error_{s_1}(h) - error_{s_2}(h)$$

2. Figure out what probability distribution governs \hat{d} :

For large n , both $error_{s_1}(h_1)$ and $error_{s_2}(h_2)$ approximately follow a Normal distribution. Thus \hat{d} approximately follows a Normal distribution with mean d .

3. What is the variance of \hat{d} ?

$$\begin{aligned} Var[\hat{d}] &= Var[error_{s_1}(h_1)] + Var[error_{s_2}(h_2)] \\ &\approx \frac{error_{s_1}(h_1)(1 - error_{s_1}(h_1))}{n_1} + \frac{error_{s_2}(h_2)(1 - error_{s_2}(h_2))}{n_2} \end{aligned}$$

4. Now give expression for approximate N % confidence intervals:

$$\hat{d} \pm z_N \sqrt{\frac{error_{s_1}(h_1)(1 - error_{s_1}(h_1))}{n_1} + \frac{error_{s_2}(h_2)(1 - error_{s_2}(h_2))}{n_2}}$$

Example: Let S_1 be our 72 test examples, and S_2 be an independent set of 45 examples. Suppose h_1 misclassifies 12 on S_1 and h_2 misclassifies 5 on S_2 .

Based on these observations, what is our estimate \hat{d} of the true difference d in error rate between h_1 and h_2 ?

With what level of confidence can we say that d is within two standard deviations of \hat{d} ?

Comparing Learning Algorithms

- Let L_1 and L_2 be two learning algorithms.
- How to determine if difference in performance of L_1 and L_2 is statistically significant?
- Many approaches to this. Reading describes one method: paired t test .

Defining “difference in performance”

- First, what does “difference in performance” mean?

Let

$$d = error_D(L_1(S)) - error_D(L_2(S))$$

where $L(S)$ is the hypothesis output by L on training set S .

We want expected difference in performance over all training sets S of size n drawn from distribution D :

$$E[d]$$

S drawn from D

Estimating expected difference in performance

- In practice, can't get all samples S of size n over D . So need an estimator.
- Suppose we have sample D_0 of data, and divide it into S_0 (training set) and T_0 (test set). Then our estimator for the difference in performance between L_1 and L_2 is:

$$\hat{d} = error_{T_0}(L_1(S_0)) - error_{T_0}(L_2(S_0))$$

Improving estimator using k -fold cross-validation

1. Partition data D_0 into k disjoint test sets T_1, T_2, \dots, 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

- $S_i \leftarrow \{D_0 - T_i\}$
- $h_A \leftarrow L_A(S_i)$
- $h_B \leftarrow L_B(S_i)$
- $\delta_i \leftarrow \text{error}_{T_i}(h_A) - \text{error}_{T_i}(h_B)$

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

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

Comparing Learning Algorithms (revisited)

We want to estimate

$$d = \text{error}_D(L_1(S)) - \text{error}_D(L_2(S))$$

I.e., we want to estimate

$$E[d]$$

S drawn from D

So we do k -fold cross-validation. For each iteration i , we get

$$\delta_i = \text{error}_{T_i}(h_1) - \text{error}_{T_i}(h_2)$$

and finally: $\bar{\delta} = \frac{1}{k} \sum_{i=1}^k \delta_i$

$\bar{\delta}$ is our estimate of $E[d]$
 S drawn from D

Note that $\bar{\delta}$ is the mean of samples from random variables δ_i .

$\bar{\delta}$ is called a “sample mean”. It is an unbiased estimator of the population mean $E[d]$
 S drawn from D

The sample standard deviation is:

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

But the true standard deviation of the sample mean, $\sigma_{\bar{\delta}}$, is unknown.

It is estimated by

$$S_{\bar{\delta}} = \frac{\sigma_{sample}}{\sqrt{n}}$$

This is also called “standard error”. It estimates the standard deviation of the distribution of samples. It reflects how much sampling fluctuation σ_{sample} has.

So we have:

$$S_{\bar{\delta}} = \frac{\sigma_{sample}}{\sqrt{k}} = \frac{\sqrt{\frac{1}{(k-1)} \sum_{i=1}^k (\delta_i - \bar{\delta})^2}}{\sqrt{k}} = \sqrt{\frac{1}{k(k-1)} \sum_{i=1}^k (\delta_i - \bar{\delta})^2}$$

Approximate N % confidence interval:

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

Here $t_{N,k-1}$ is a constant that corresponds to z_N .

When $k \rightarrow \infty$, $t_{N,k-1} \rightarrow z_N$.

Determining this confidence interval is called the “paired t test”.

Matlab has a function `ttest2`: more on that next time.

$t_{N,k}$ table

Confidence level N

	90%	95%	98%	99%
$k = 2$	2.92	4.30	6.96	9.92
$k = 5$	2.02	2.57	3.36	4.03
$k = 10$	1.81	2.23	2.76	3.17
$k = 20$	1.72	2.09	2.53	2.84
$k = 30$	1.70	2.04	2.46	2.75
$k = \infty$	1.64	1.96	2.33	2.58

Summary

1. Assessing the error of a single hypothesis h

A. Two-sided confidence interval:

$$\begin{aligned} & error_s(h) \pm z_N \sigma_s \\ & = error_s(h) \pm z_N \sqrt{\frac{error_s(h)(1 - error_s(h))}{n}} \end{aligned}$$

“With $N\%$ confidence, true error is in this interval”

B. One-sided confidence interval:

If the two-sided confidence interval is for $N = 100(1 - \alpha)\%$ with lower bound L and upper bound U , this implies a one-sided confidence interval of $N_U = 100(1 - \alpha/2)\%$.

“With $N_U\%$ confidence, the true error is smaller than U .”

2. Comparing the error of two hypothesis, h_1 and h_2

A. Confidence interval around observed difference in error rate:

N% confidence interval around

$$\hat{d} = error_{s_1}(h_1) - error_{s_2}(h_2)$$

is

$$\hat{d} \pm z_N \sqrt{\frac{error_{s_1}(h_1)(1 - error_{s_1}(h_1))}{n_1} + \frac{error_{s_2}(h_2)(1 - error_{s_2}(h_2))}{n_2}}$$

“With confidence N%, the true difference in error rate d is in this interval.

B. Hypothesis testing (corrected)

Given $\hat{d} = error_{s_1}(h_1) - error_{s_2}(h_2)$

what is the probability that $d > 0$?

$$\begin{aligned} P(d > 0) &= P(\hat{d} < d + \hat{d}) \\ &= P\left(\hat{d} < \mu_{\hat{d}} + \frac{\hat{d}}{\sigma_{\hat{d}}} \sigma_{\hat{d}}\right) \end{aligned}$$

Look up $\frac{\hat{d}}{\sigma_{\hat{d}}}$ in z_n table to get N .

Find associated one - sided confidence interval N_U .

“With confidence N_U %, we accept the hypothesis that $error(h_1) > error(h_2)$.”

3. Comparing two learning algorithms, L_A and L_B

k -fold cross-validation paired t -test:

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

where $\delta_i = \text{error}_{T_i}(h_A^i) - \text{error}_{T_i}(h_B^i)$.

We have confidence interval $\bar{d} \pm t_{N,k-1} s_{\bar{d}}$

where

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