CSC 580 Principles of Machine Learning

05 Practical Considerations

Chicheng Zhang

Department of Computer Science

THE UNIVERSITY OF ARIZONA

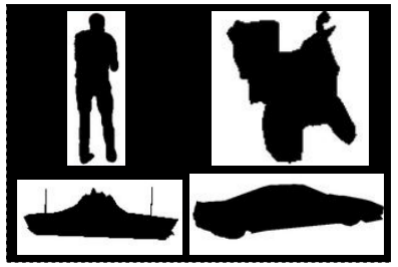
*slides credit: built upon CSC 580 Fall 2021 lecture slides by Kwang-Sung Jun

The role of feature in supervised learning

The importance of good feature representation

- Pixel representation:
  - represent an image as a w*h*3 dimensional vector
  - treat all coordinates in the same role
  - throw away all locality information in the image

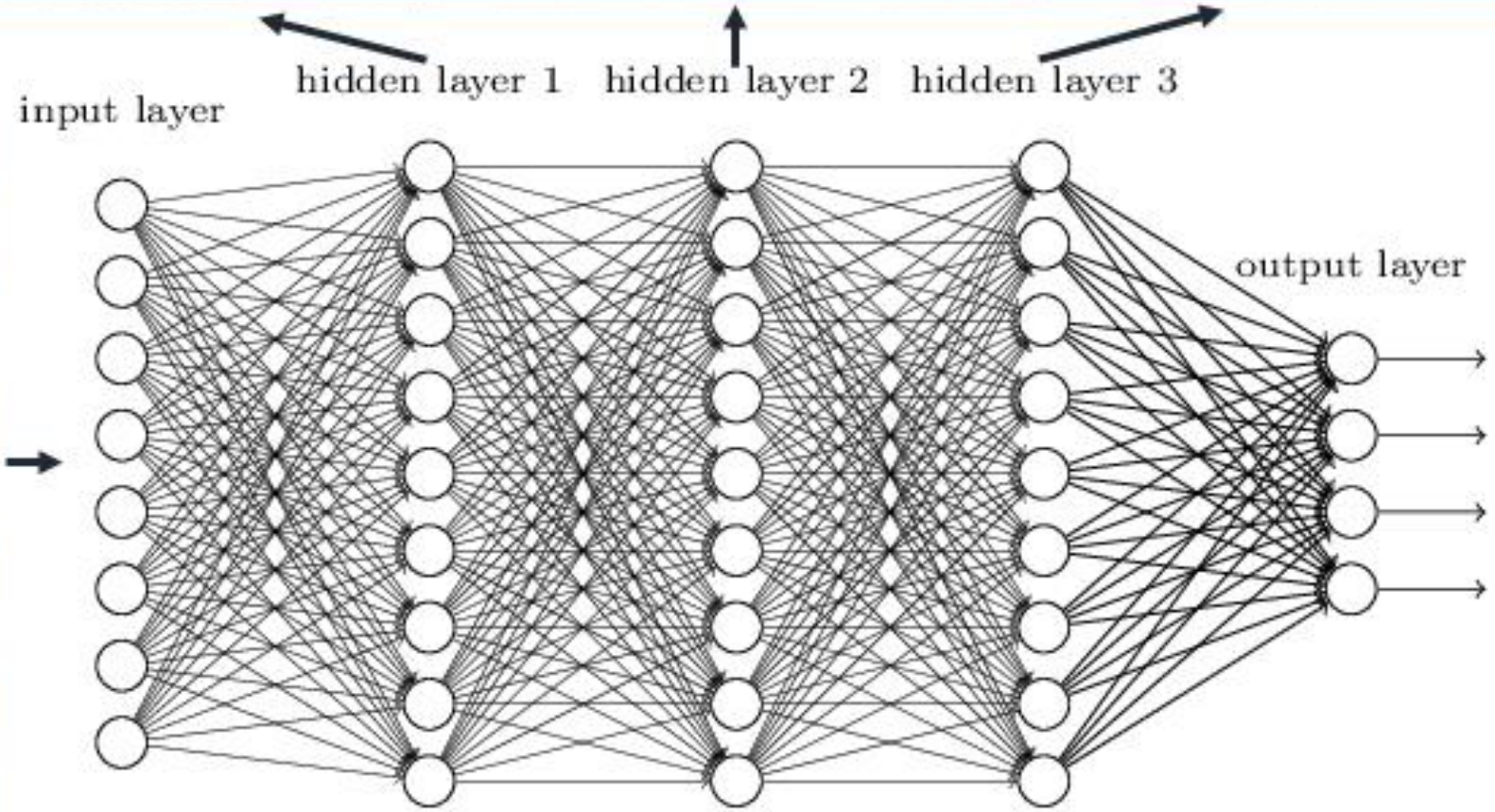
- Shape representation:
  - represent a colored image with a w*h black-white image



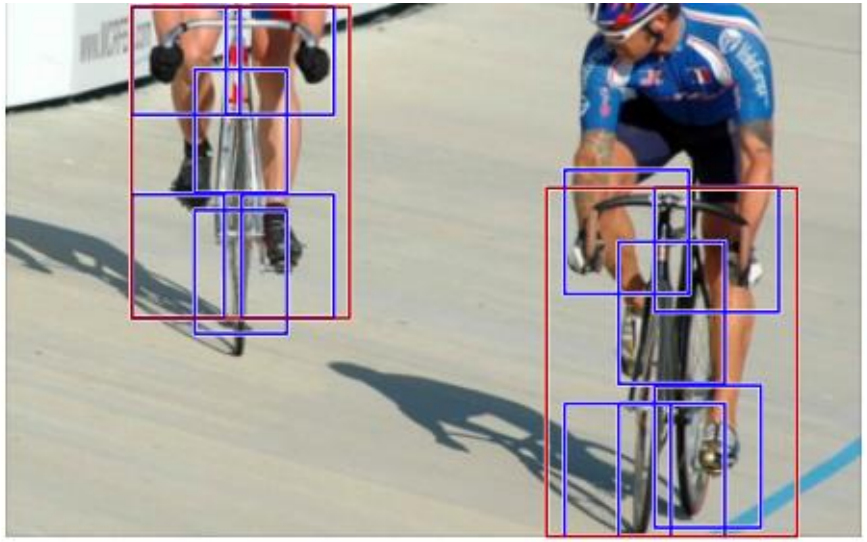
- Bag-of-words representation:

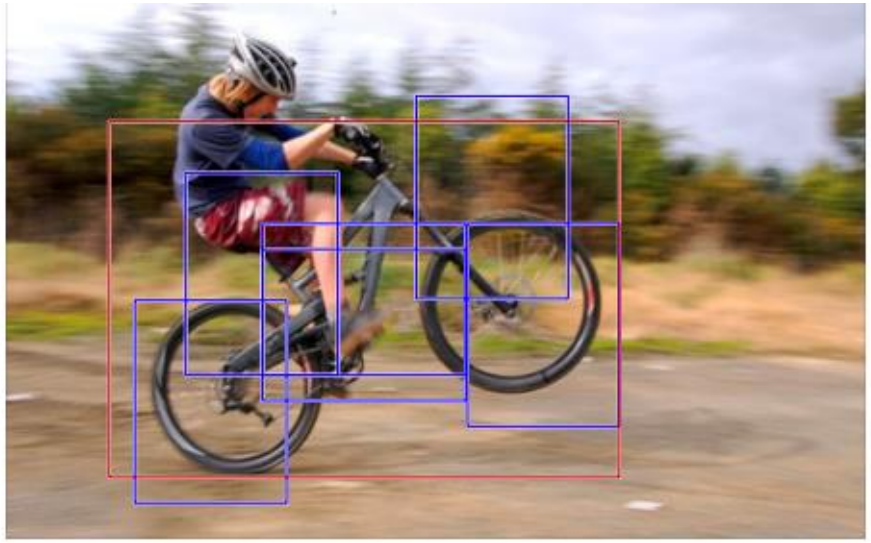
	free	offer	lecture	cs	Spam?
Email 1	2	1	0	0	+1
Email 2	0	1	3	1	-1

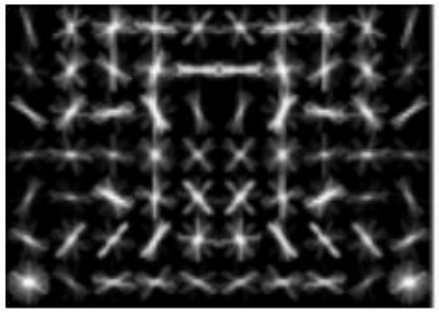
Deep neural networks learn hierarchical feature representations

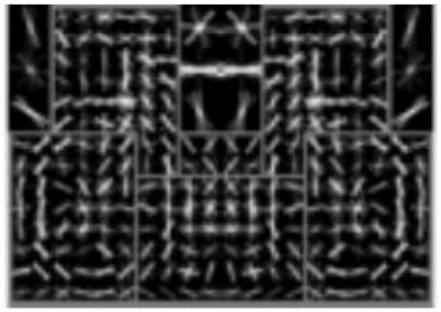


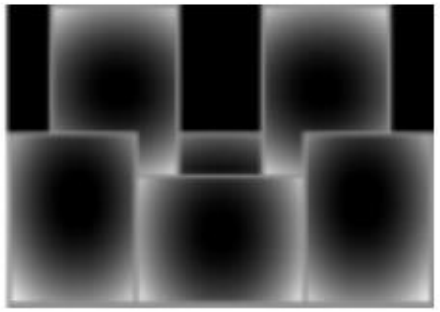
https://www.rsipvision.com/exploring-deep-learning/

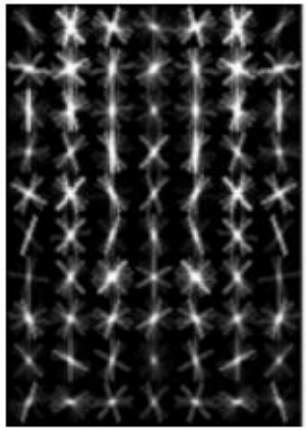


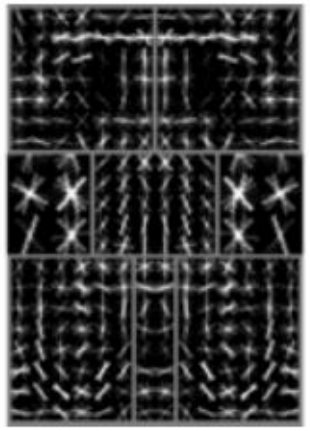


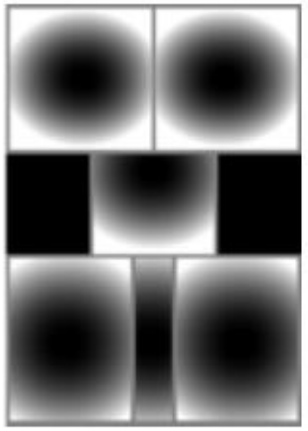












https://computervisionblog.wordpress.com/2016/01/23/distributed-code-or-grandmother-cells-insights-from-convolutional-neural-networks/

Irrelevant and redundant features

- Irrelevant features
  - y is independent of f
  - y = Road walkability, f = duck activities in the pond

  - If #features is large and #examples is small ⇒ spurious correlation between some feature & label

- Redundant features
  - Given f_1, y is (nearly) independent of f_2

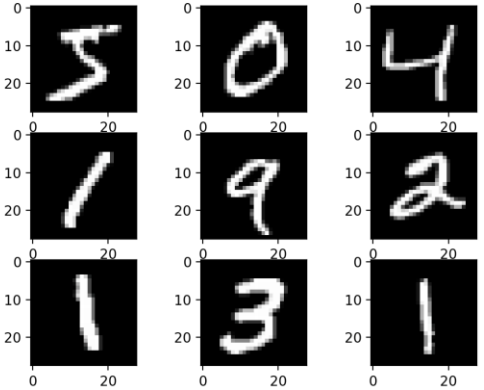
- Learning decision trees implicitly handles these two issues
- How about nearest neighbors / Perceptron?

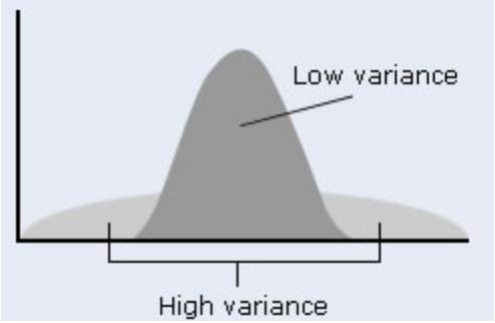
https://www.microsoft.com/en-us/research/blog/ppe-a-fast-and-provably-efficient-rl-algorithm-for-exogenous-noise/

Feature pruning

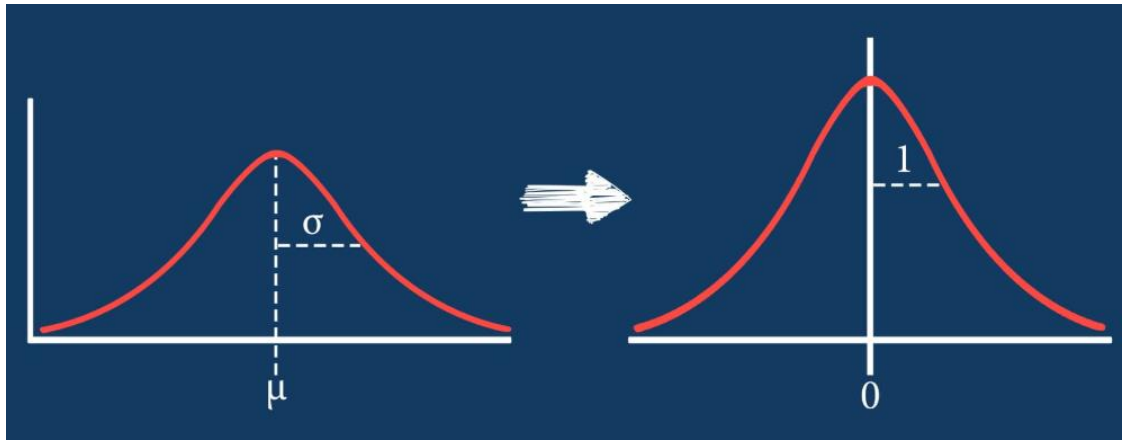
- Removing features that are not very useful for prediction
  - E.g. text classification with bag-of-word representation, remove words that appear <= K docs
  - E.g. digit classification, remove pixels with low variance

$$\mu_f = \frac{1}{N}\sum_{i=1}^{N} x_{i,f} \qquad \sigma_f^2 = \frac{1}{N}\sum_{i=1}^{N}(x_{i,f} - \mu_f)^2$$





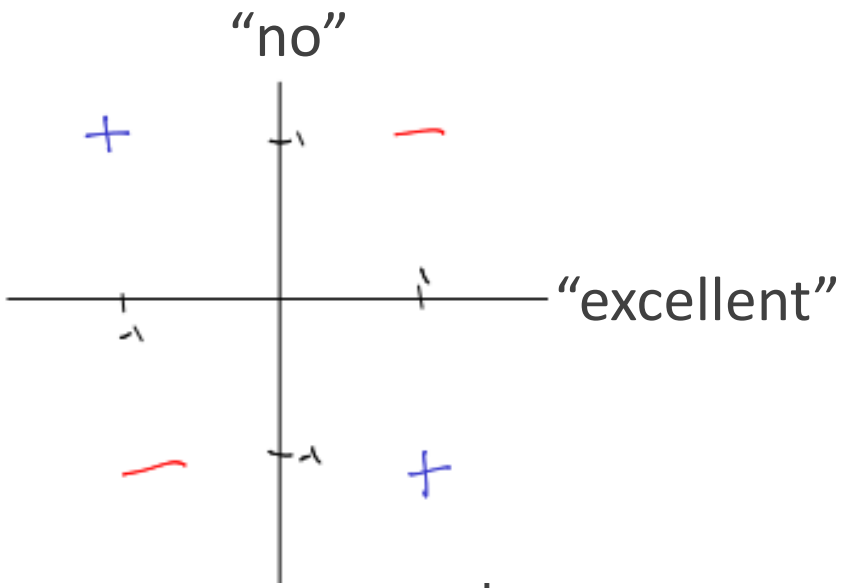
Feature normalization

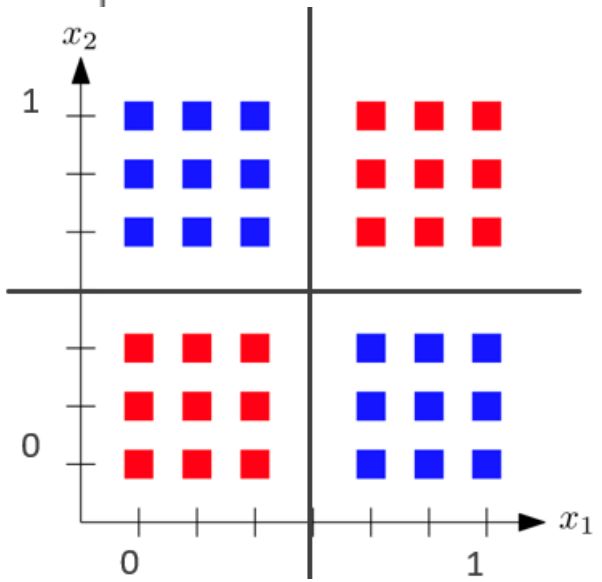


- Centering:
  - $x'_{i,f} = x_{i,f} - \mu_f \Rightarrow \mu'_f = 0$
- Variance scaling:
  - $x'_{i,f} = x_{i,f}/\sigma_f \Rightarrow (\sigma'_f)^2 = 1$
- Absolute scaling
  - $x'_{i,f} = x_{i,f}/r_f$, where $r_f = \max_i |x_{i,f}|$ $\Rightarrow$ range of $x'_{i,f}$'s in [-1,+1]
- Same transformation applied to both training set and test data
- Aside: example normalization: $x'_i = \frac{x_i}{\|x_i\|}$ sometimes also can be applied

Feature transformations

- Combining features into a “meta-feature”, e.g. $x_{\text{no}} \cdot x_{\text{excellent}}$
  - Useful for e.g. Perceptron learners
- In general, $\binom{d}{k}$ mega-features if allowed to combine k features
- Computationally cheaper alternative:
  - train a decision tree, use the meta-feature induced by leaves
- Logarithmic feature transformation
  - $x'_f \leftarrow \log_2(x_f)$ (“excellent” word count: 1->2 vs. 10->11
  - $x'_f \leftarrow \log_2(x_f + 1)$





Classification metrics beyond error rate

Confusion matrix

- E.g. activity recognition

- $P(\hat{y} = \text{skip} \mid y = \text{jump}) = 11\%$

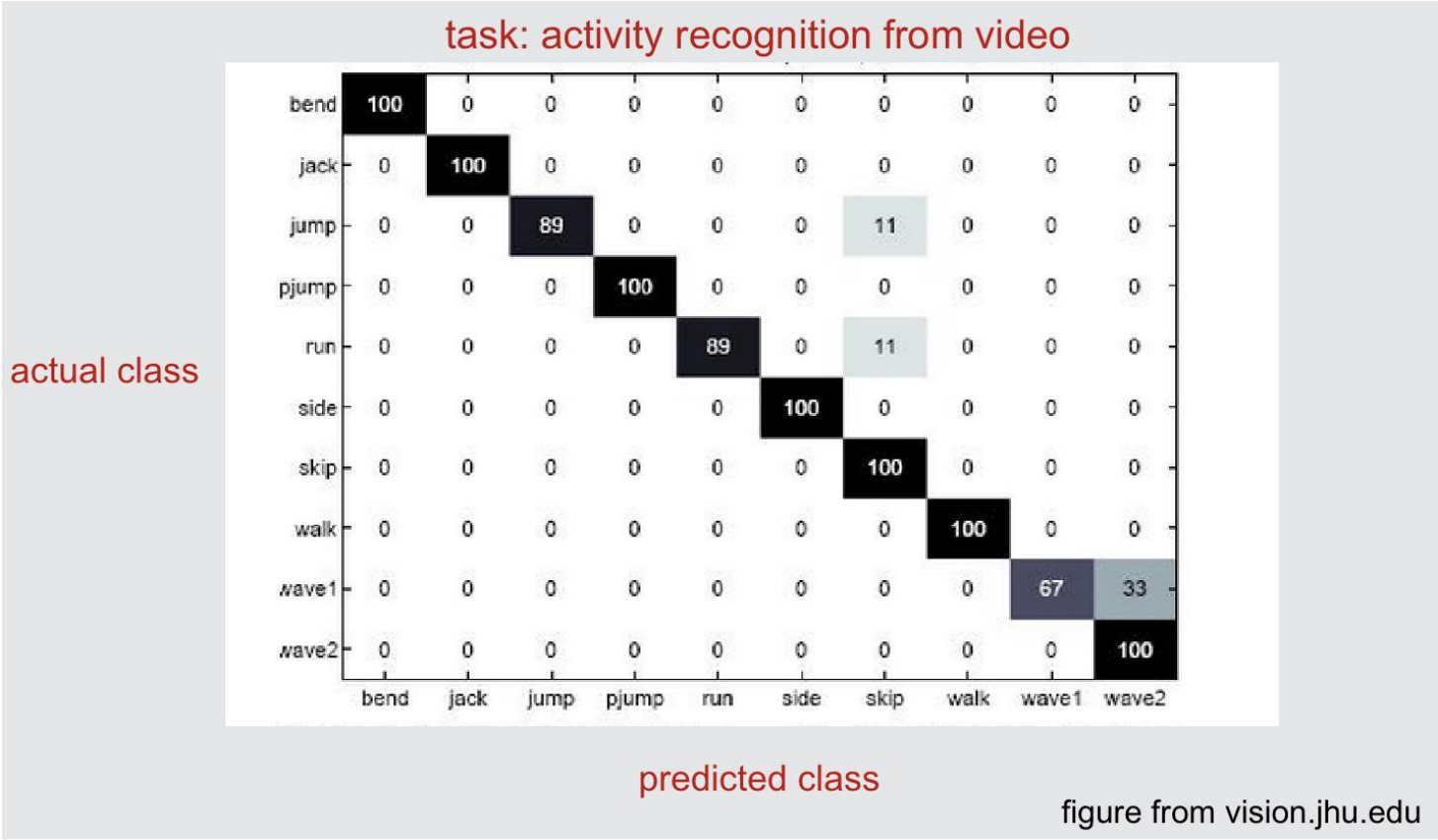


figure from vision.jhu.edu

Class imbalance problem

- E.g., 5% pos, 95% negative.
- Baseline: always predict majority class
- Implicit assumption:
  misclassifying positive example is more costly than misclassifying negative examples

- Standard ML algorithms aims at finding h that minimizes unweighted training error

$$\sum_{i=1}^{n} I(h(x_i) \neq y_i)$$

- 2 alternatives:
  - Duplicate the minority class to make the positive and negative class balanced
    repeat every positive example w times, where $w = P(y = -1)/P(y = +1)$
  - Importance weighted classification: minimize $\sum_{i=1}^{n} w_i \, I(h(x_i) \neq y_i)$,
    where $w_i = 1$ when $y_i = -1$, $w_i = w$ when $y_i = +1$

New measures of classification performance

- True positive rate (TPR)

  $= \frac{\mathrm{TP}}{\mathrm{P}} = \frac{P(\hat{y}=+1, y=+1)}{P(y=+1)}$

  (aka recall, sensitivity)
- True negative rate (TNR) = $\frac{\mathrm{TN}}{\mathrm{N}}$

  (specificity)
- False positive rate (FPR) = $\frac{\mathrm{FP}}{\mathrm{N}}$
- False negative rate (FNR) = $\frac{\mathrm{FN}}{\mathrm{P}}$
- Precision = $\frac{\mathrm{TP}}{\mathrm{P}-called} = \frac{P(\hat{y}=+1, y=+1)}{P(\hat{y}=+1)}$, $\mathrm{P} - called = \mathrm{TP} + \mathrm{FP}$

		actual class: positive	actual class: negative
predicted class	positive	true positives (TP)	false positives (FP) Type I error
predicted class	negative	false negatives (FN) Type II error	true negatives (TN)

P = TP + FN N = FP + TN

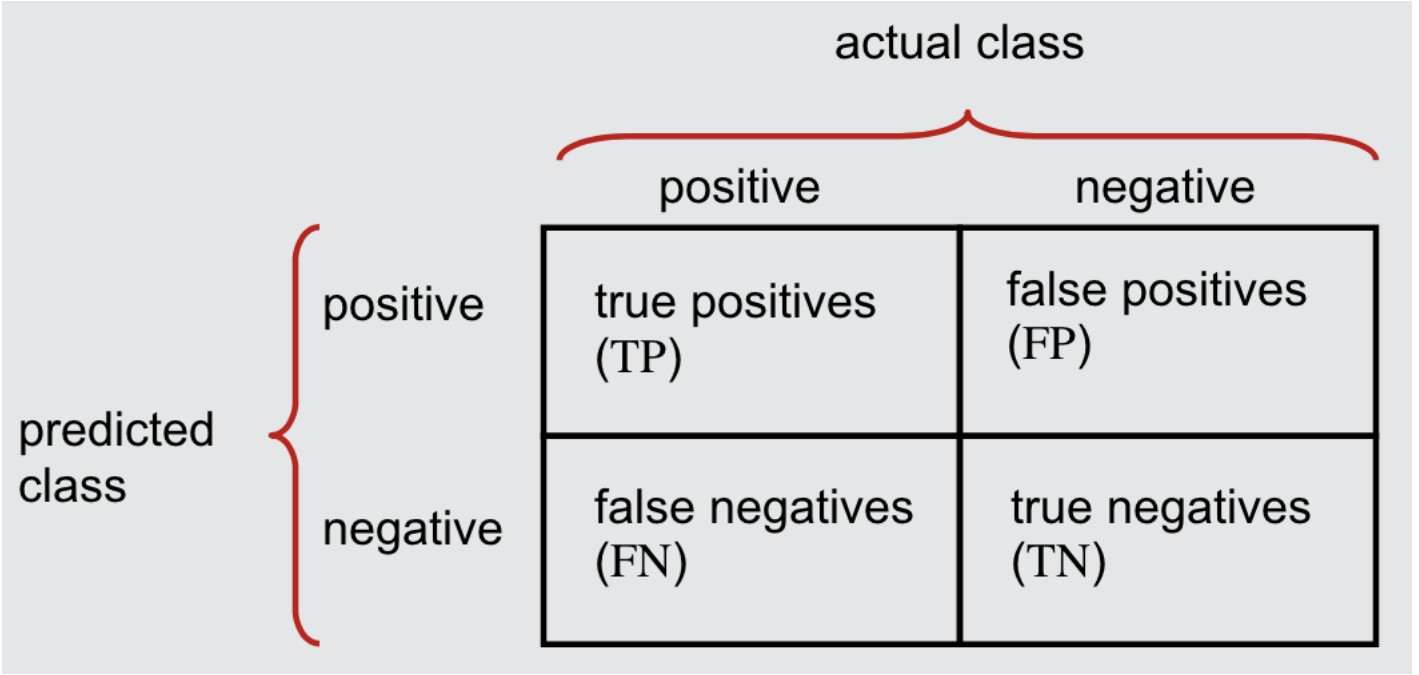
Applications:

- Search engine: precision & recall
- Cancer classification: FNR vs. FPR

Adjusting TP, FP, TN, FN via thresholding

- Decision values (classification scores)

$c(x_i)$	y_i
.99	+
.98	+
.72	-
.51	-
.24	+

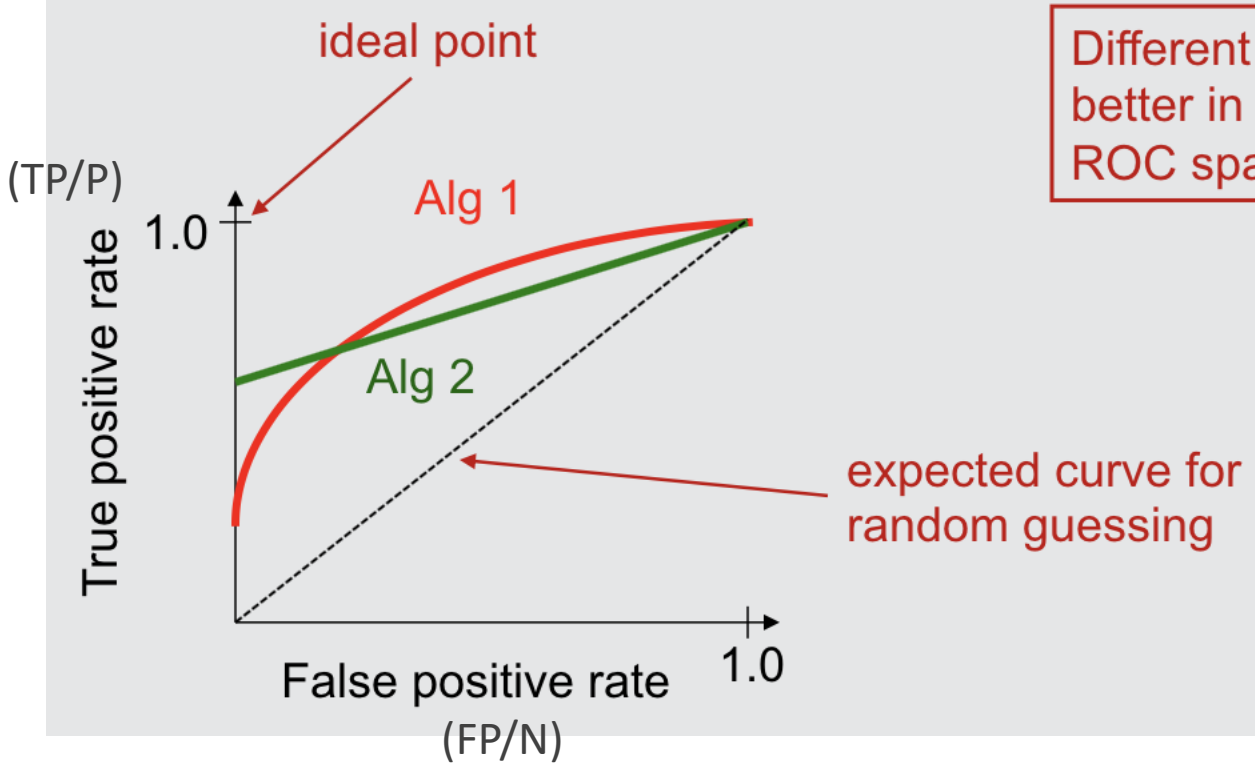


P = TP + FN N = FP + FN

- $h_t(x) = I(c(x) \geq t)$
- Choice of threshold t:
  - $t = \infty$: $h_t \equiv -1 \Rightarrow \mathrm{TPR} = 0,\ \mathrm{FPR} = 0$
  - $t = 0$: $h_t \equiv +1 \Rightarrow \mathrm{TPR} = 1,\ \mathrm{FPR} = 1$

ROC curve

A *Receiver Operating Characteristic* (*ROC*) curve plots the TP-rate vs. the FP-rate as a threshold on the confidence of an instance being positive is varied



Different methods can work better in different parts of ROC space.

	$c(x_i)$	y_i
	.99	+
	.98	+
	.72	+
	.51	-
	.24	-

ROC curve

- Conceptually, consider every possible threshold, put a dot for each, and connect them.

- Actually, just need to care about when the ‘correct class’ changes
  - results in staircase shape, but diagonal line can still happen.

- A popular alternative: just plot when going from + to -. (what’s shown here)

instance	confidence positive		correct class
Ex 9	.99		+
Ex 7	.98	TPR= 2/5, FPR= 0/5	+
Ex 1	.72		-
Ex 2	.70		+
Ex 6	.65	TPR= 4/5, FPR= 1/5	+
Ex 10	.51		-
Ex 3	.39		-
Ex 5	.24	TPR= 5/5, FPR= 3/5	+
Ex 4	.11		-
Ex 8	.01	TPR= 5/5, FPR= 5/5	-

Calculating ROC curve

let $\left(\left(y^{(1)}, c^{(1)}\right)\cdots\left(y^{(m)}, c^{(m)}\right)\right)$ be the test-set instances sorted according to predicted confidence $c^{(i)}$ that each instance is positive

let num_neg, num_pos be the number of negative/positive instances in the test set

$TP = 0,\ FP = 0$

$last\_TP = 0$

for $i = 1$ to m

// find thresholds where there is a pos instance on high side, neg instance on low side

if $(i > 1)$ and ($c^{(i)} \neq c^{(i-1)}$) and ($y^{(i)} ==$ neg) and ($TP > last\_TP$)

$FPR = FP / num\_neg,\quad TPR = TP / num\_pos$

output (FPR, TPR) coordinate

$last\_TP = TP$

if $y^{(i)} ==$ pos

$++TP$

else

$++FP$

$FPR = FP / num\_neg,\ TPR = TP / num\_pos$

output (FPR, TPR) coordinate

instance	confidence positive		correct class
Ex 9	.99		+
Ex 7	.98	TPR= 2/5, FPR= 0/5	+
Ex 1	.72		-
Ex 2	.70		+
Ex 6	.65	TPR= 4/5, FPR= 1/5	+
Ex 10	.51		-
Ex 3	.39		-

ROC curve examples

task: recognizing genomic units called operons

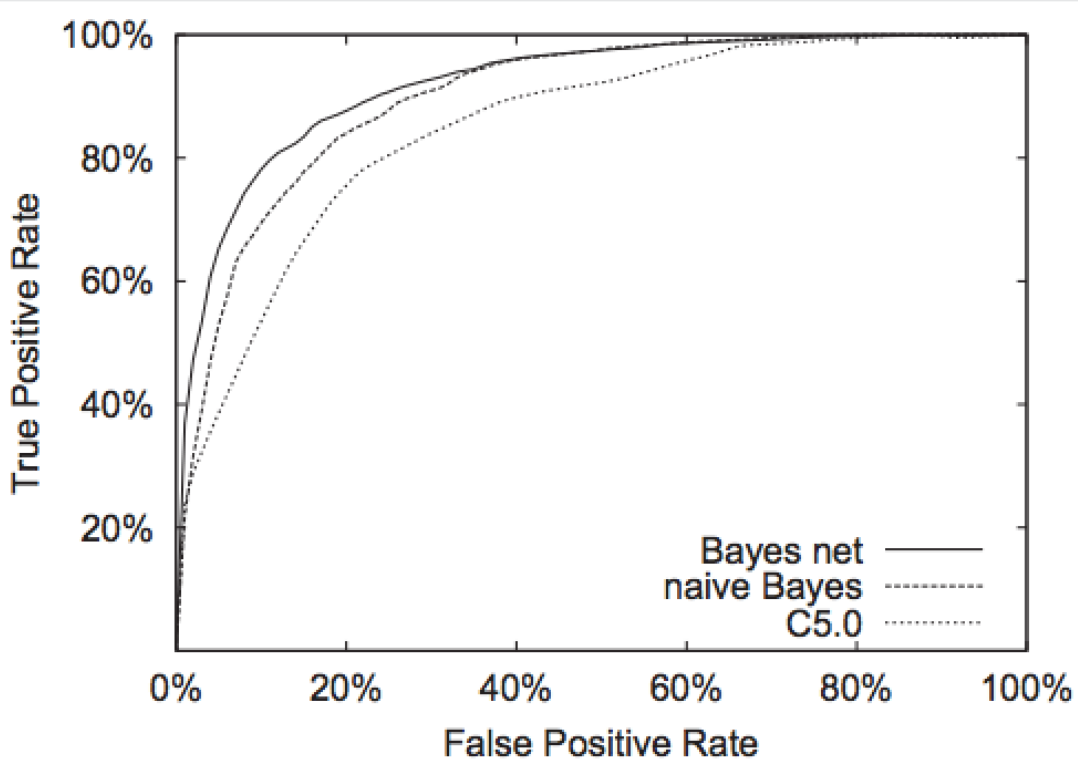
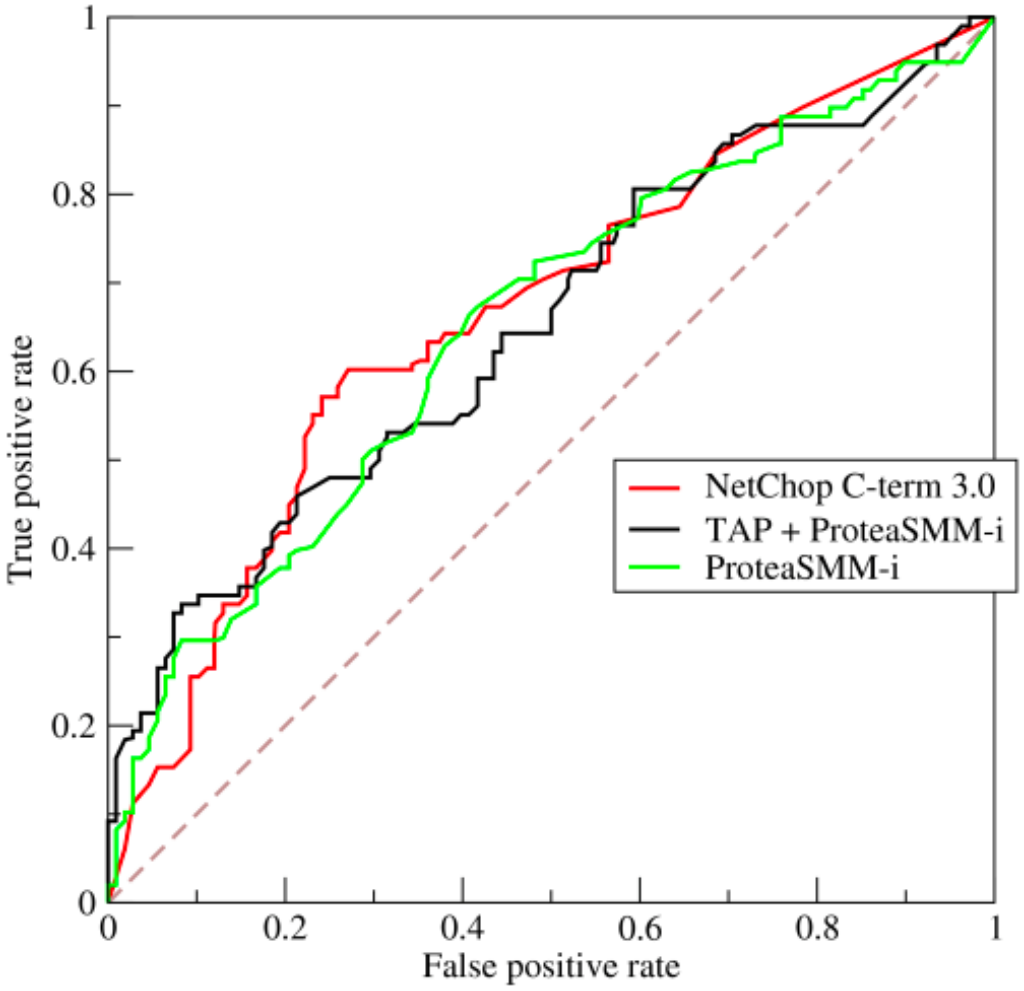


figure from Bockhorst et al., *Bioinformatics* 2003



from Wikipedia

Area under ROC curve

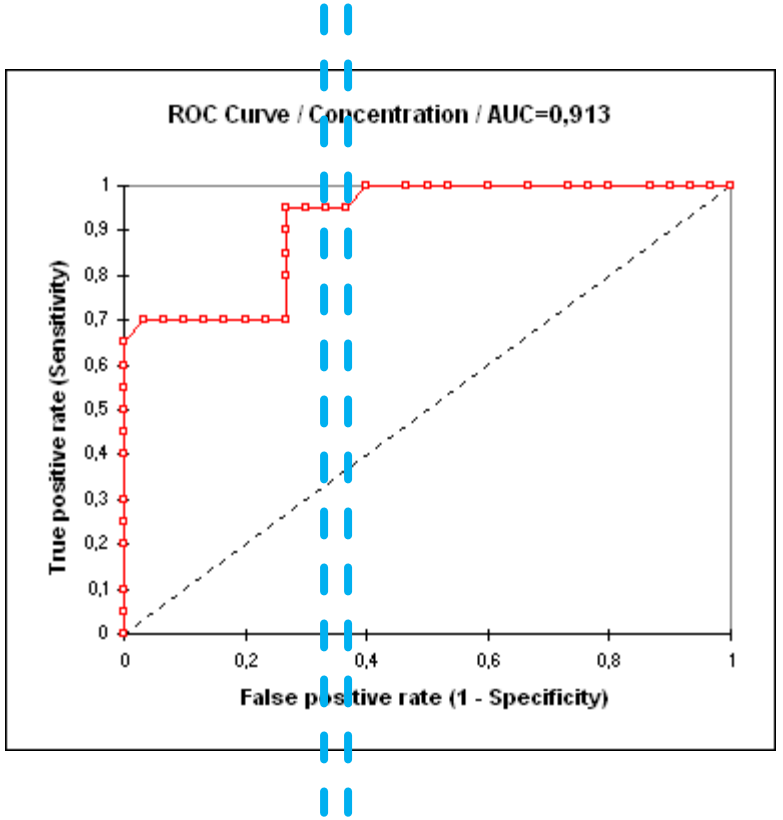
- The boss says “could you just give me one number?”
- **AUC**: Area Under the ROC curve:

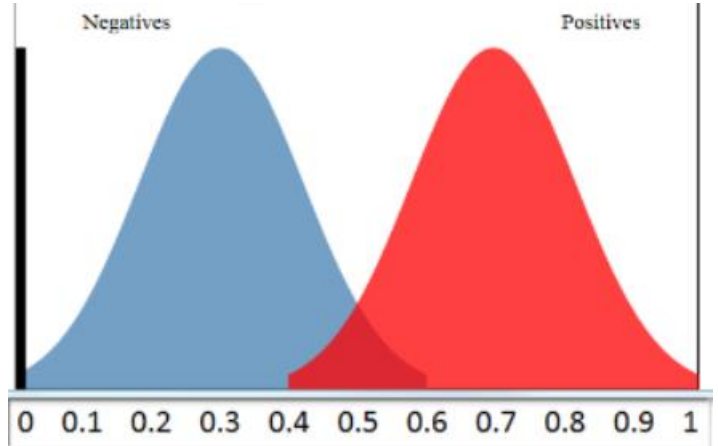
$$\mathrm{AUC}(c) := \frac{\sum_{(x_-,-1)\in S_-} \sum_{(x_+,+1)\in S_+} I(c(x_+)>c(x_-))}{N_- \cdot N_+}$$

- c(x): decision value of x
- S_-: negative examples, S_+: positive examples
- Idea: the slice corresponds to x_- has area

$$\frac{1}{N_-} \cdot \frac{\sum_{(x_+,+1)\in S_+} I(c(x_+)>c(x_-))}{N_+}$$

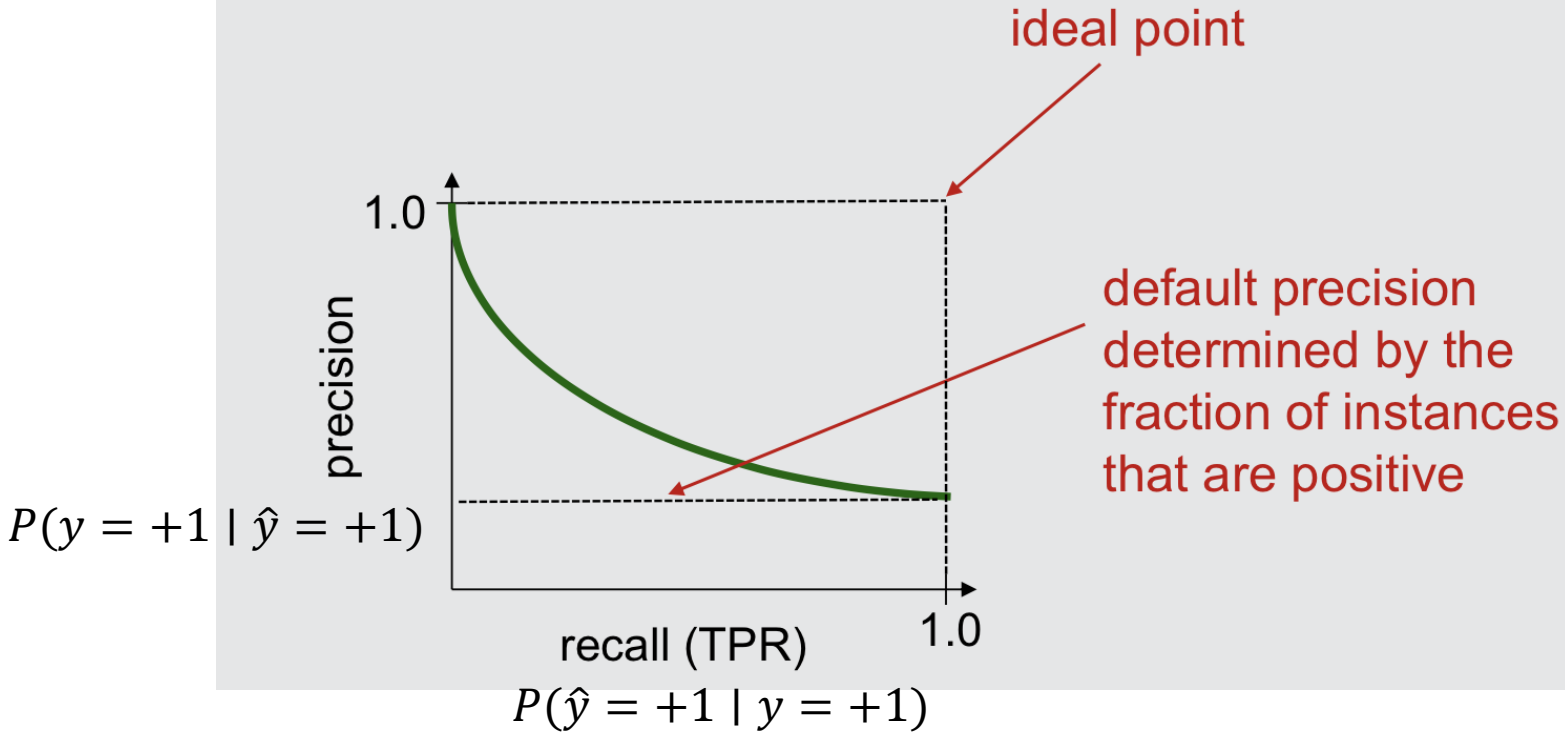
- Interpretation: “how well does c distinguish between + and -?”





Precision-Recall (PR) curve

A *precision/recall curve* plots the precision vs. recall (TP-rate) as a threshold on the confidence of an instance being positive is varied



$c(x_i)$	y_i
.99	+
.98	+
.72	+
.51	-
.24	-

- This is usually a trade-off curve: $t \downarrow \Rightarrow$ recall $\uparrow$, precision $\downarrow$

PR-curve example

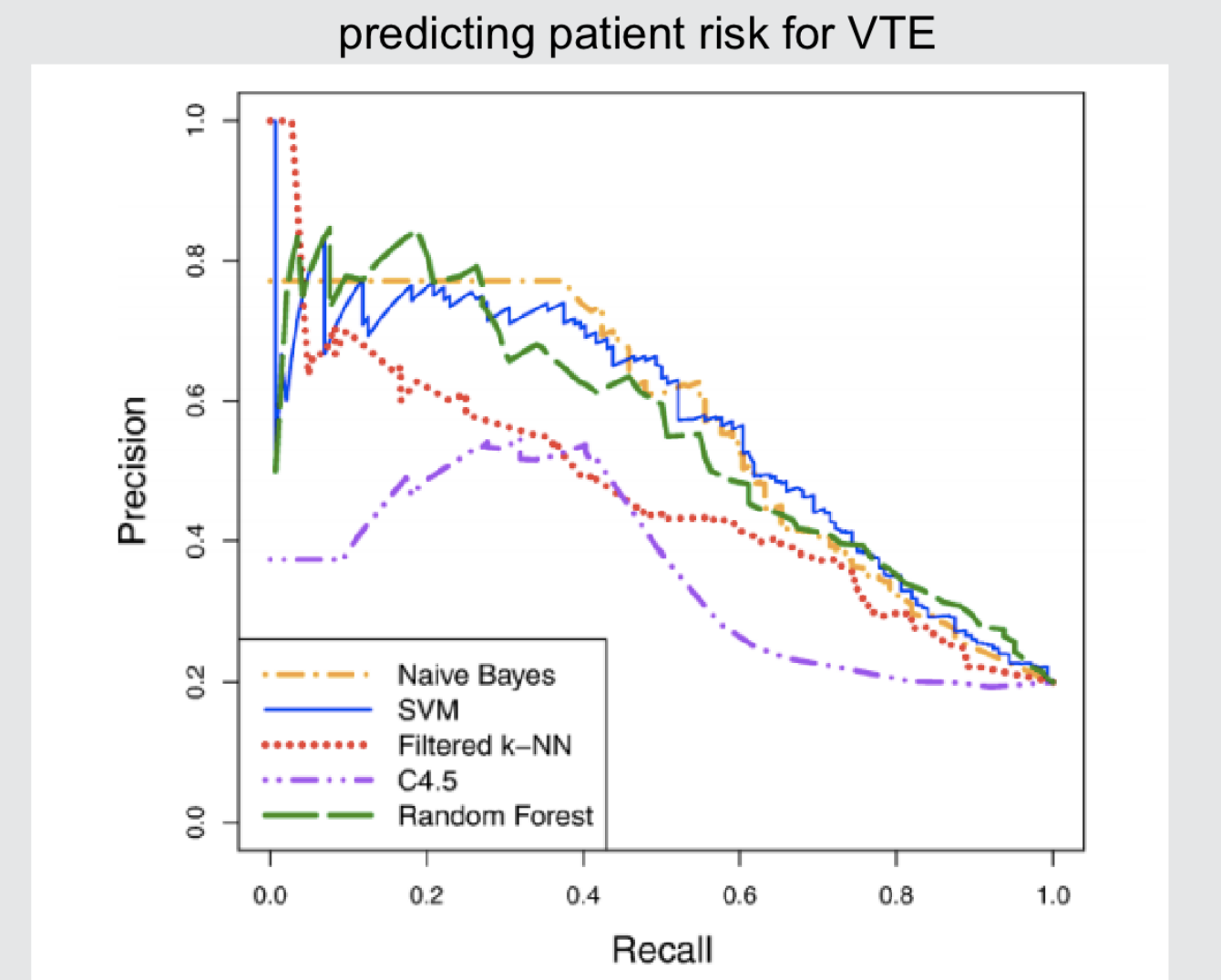


figure from Kawaler et al., *Proc. of AMIA Annual Symosium,* 2012

Summary of precision-recall

- Reporting one number
- Take the harmonic mean: **F1 score**

- Recall: minimum of the two <= harmonic mean <= geometric mean <= arithmetic mean

$$F_1 = \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}} :$$

- Emphasizes the smaller measure
  - E.g. $\text{recall} = 0.1, \text{precision} = 0.9 \Rightarrow F_1 = 0.18$

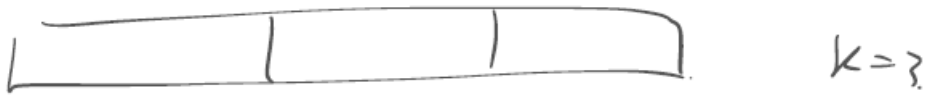
- Area under PR-curve is also a popular metric

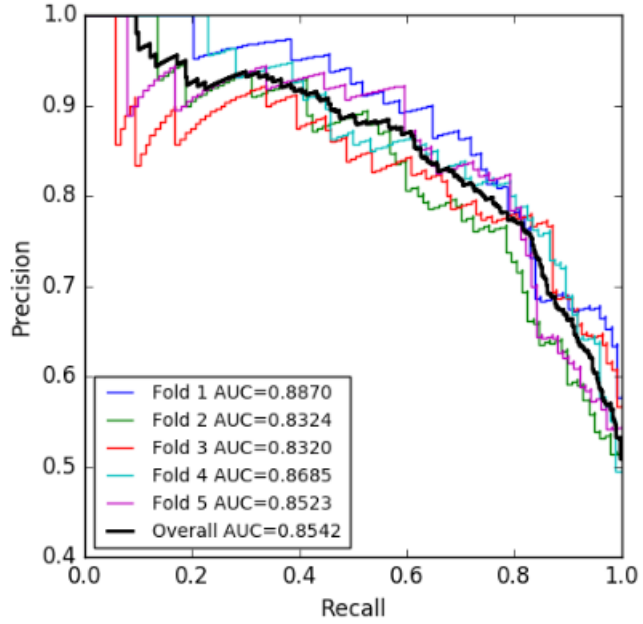
	0.0	0.2	0.4	0.6	0.8	1.0
0.0	0.00	0.00	0.00	0.00	0.00	0.00
0.2	0.00	0.20	0.26	0.30	0.32	0.33
0.4	0.00	0.26	0.40	0.48	0.53	0.57
0.6	0.00	0.30	0.48	0.60	0.68	0.74
0.8	0.00	0.32	0.53	0.68	0.80	0.88
1.0	0.00	0.33	0.57	0.74	0.88	1.00

Table 5.2: Table of f-measures when varying precision and recall values.

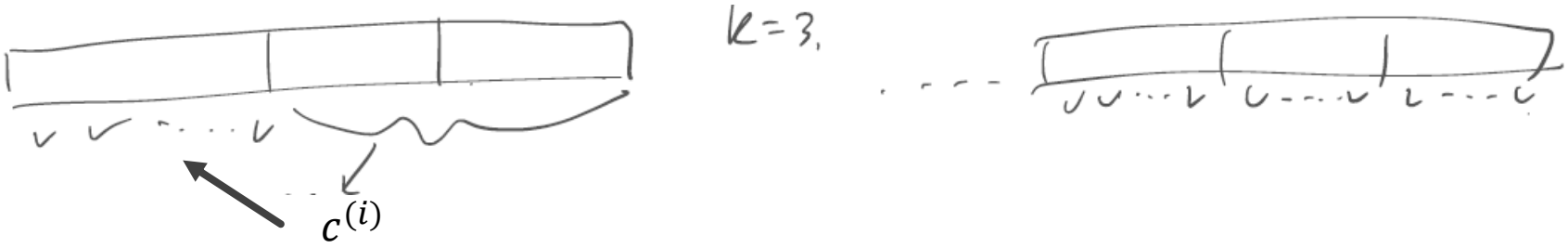
How to plot ROC/PR curve when training set is small?

- k-fold CV:
  - Obtain k curves and plot them all





- **Pooled prediction** from k-fold CV.



Next lecture (9/14)

- General model performance evaluation & comparison: hypothesis testing, bootstrapped confidence intervals

- Linear models revisited

- Assigned reading: CIML Section 5.7, Sections 7.1-7.3

Hypothesis testing and Confidence Interval

Motivation: evaluating & comparing ML models

- Setting 1:
  - Your ML model f has test set error = 6.9%
  - How confident are we to conclude that f has generalization error < 7%?

- Setting 2:
  - Your ML model f has test set error = 6.9%
  - Gabe's ML model g has test set error = 6.8%
  - How confident are we to conclude that g has smaller generalization error than that of f?

- Intuition: test set size matters
- These questions can be answered rigorously using hypothesis testing and confidence interval
- Disclaimer: we only focus on the key ideas (standard stats courses spend >= 5 lectures on this)

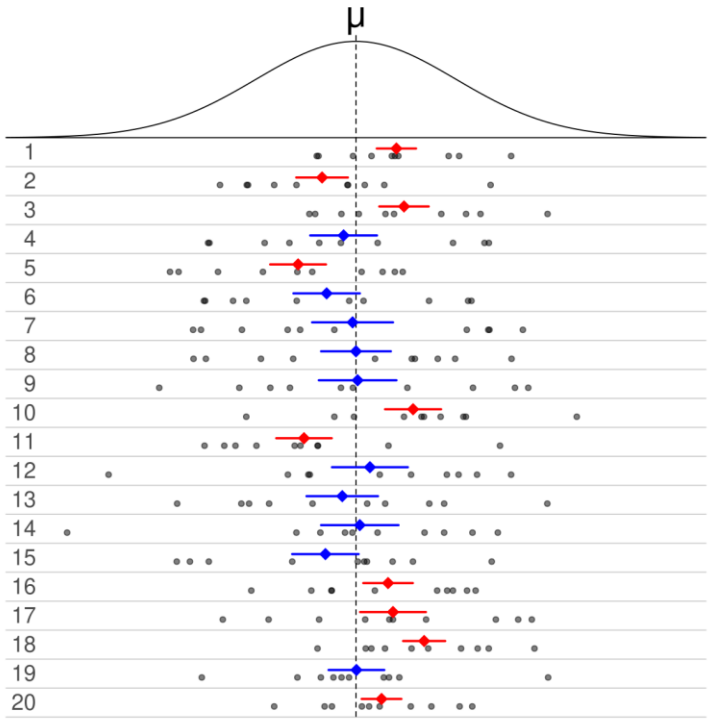
Confidence interval (CI): definition

- Given distribution family D_θ: $\theta \in \Theta$
- Sample $S = (X_1, \ldots, X_n)$ drawn iid from distribution D_θ
- A mapping I is said to be a $(1-\alpha)$-confidence interval construction for θ, if

$$P_{S \sim D_\theta^n}\big(\theta \in I(S)\big) \geq 1 - \alpha$$

- A 50% confidence interval construction

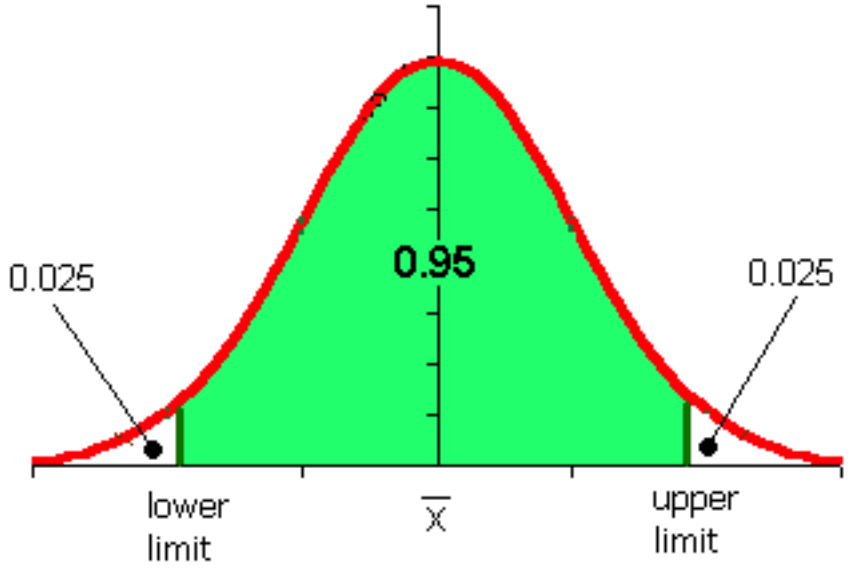
  for the mean parameter μ in $D_\mu = N(\mu, 1)$



CI construction

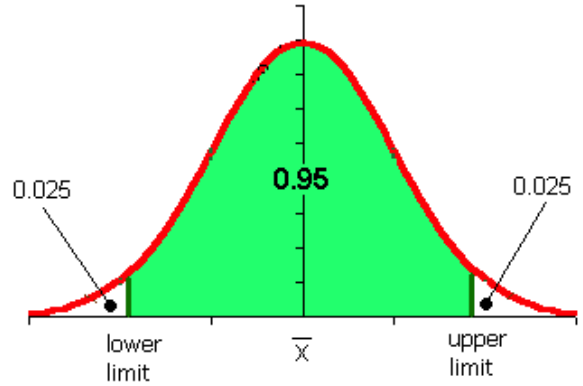
- A standard recipe:
  - Construct an estimator for θ based on S -- call it $\hat{\theta}_S$
  - Let $I(S) \coloneqq [\hat{\theta}_S - w, \hat{\theta}_S + w]$, where w is chosen such that for all θ,
$$P_{S \sim D_\theta^n}\left(\theta \in [\hat{\theta}_S - w, \hat{\theta}_S + w]\right) \geq 1 - \alpha$$
  - Sometimes choose $I(S) \coloneqq [\hat{\theta}_S - w_L, \hat{\theta}_S + w_R]$ with different w_L, w_R's

- Important example: confidence interval for normal mean
  - $D_\mu = N(\mu, 1), S = (X_1, \ldots, X_n) \sim D_\mu^n$
  - Define $\hat{\mu}_S = \frac{1}{n}\sum_{i=1}^n X_i$
  - $\hat{\mu}_S - \mu \sim N\left(0, \frac{1}{n}\right)$
  - How to choose w such that $P(|\hat{\mu}_S - \mu| \leq w) \geq 1 - \alpha$?

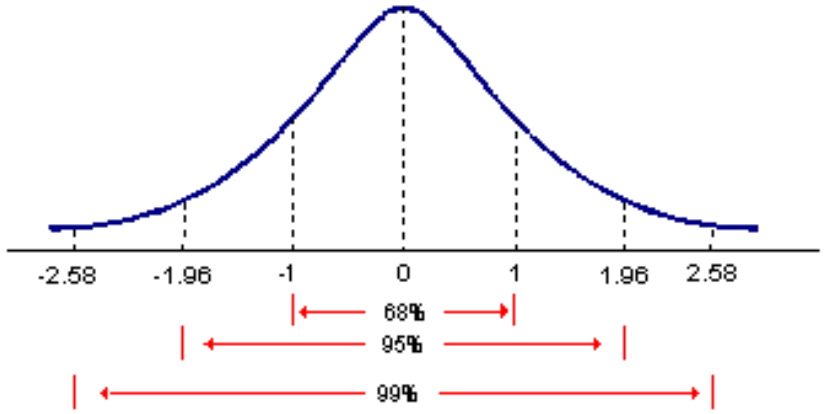


CI for normal mean (cont'd)

- $\hat{\mu}_S - \mu \sim N\left(0, \frac{1}{n}\right)$
- How to choose w such that $P(|\hat{\mu}_S - \mu| \leq w) \geq 1 - \alpha$?



- Note: $Z = \sqrt{n}\,(\hat{\mu}_S - \mu) \sim N(0,1)$
- Suffices to find z_α such that $P(|Z| \leq z_\alpha) \geq 1 - \alpha$, and let $w = \frac{z_\alpha}{\sqrt{n}}$



- Final $(1-\alpha)$-confidence interval construction for μ: $I(S) = \left[\hat{\mu}_S - \frac{z_\alpha}{\sqrt{n}}, \hat{\mu}_S + \frac{z_\alpha}{\sqrt{n}}\right]$
- E.g. 95%-confidence interval for μ: $I(S) = \left[\hat{\mu}_S - \frac{1.96}{\sqrt{n}}, \hat{\mu}_S + \frac{1.96}{\sqrt{n}}\right]$

CI for means of binary random variables (r.v.'s)

- Important example: estimating generalization error using error on test set S

With approximately C% probability, the true error lies in the interval

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

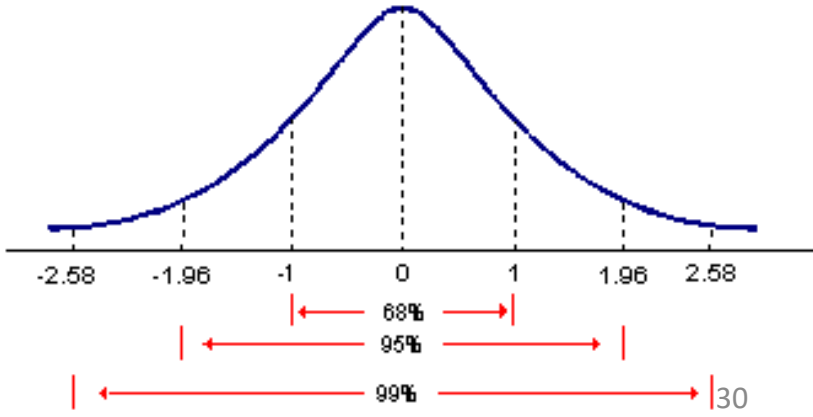
variance

where z_C is a constant that depends on C (e.g. for 95% confidence, z_C =1.96)

- Python code for computing z_C

```
import scipy.stats as st
alpha = 0.05
st.norm.ppf(1-alpha/2)
=> 1.959963984540054
```

// alpha = 1-C
// ppf: inverse of the Gaussian CDF



CI for means of binary r.v.'s (cont'd)

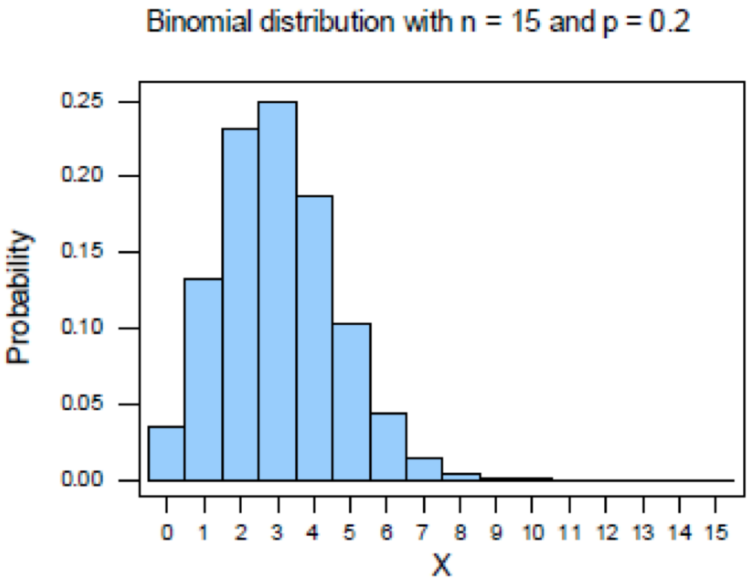
- $D_\mu = \mathrm{Ber}(\mu), \quad S = (X_1, \ldots, X_n) \sim D_\mu^n$

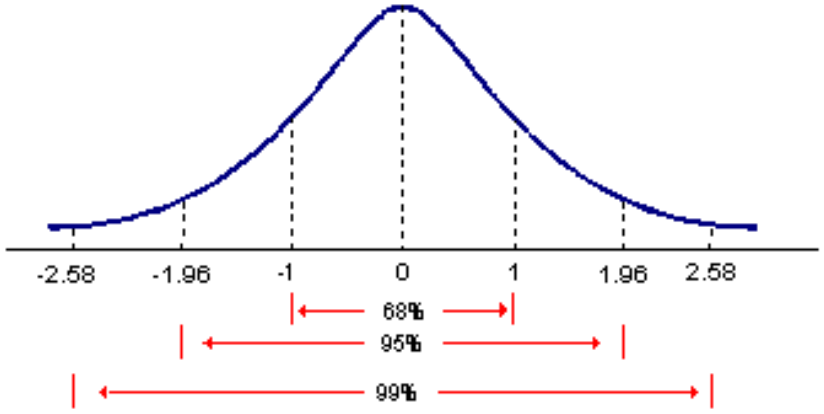
- Define $\hat{\mu}_S = \frac{1}{n}\sum_{i=1}^{n} X_i$
- Central limit theorem:

$$Z = \frac{\sqrt{n}(\hat{\mu}_S - \mu)}{\sqrt{\mu(1-\mu)}} \Rightarrow N(0,1)$$

- Can define $I(S) = \left[\hat{\mu}_S \pm \frac{\sqrt{\mu(1-\mu)}z_\alpha}{\sqrt{n}}\right]$
- Issue: μ is unknown
- Implementable alternative: $I(S) = \left[\hat{\mu}_S \pm \frac{\sqrt{\hat{\mu}_S(1-\hat{\mu}_S)}z_\alpha}{\sqrt{n}}\right]$

- Note: exact confidence interval possible, and the computational complexity is not that high. (log(1/precision_level) evaluations of Binomial cdf)



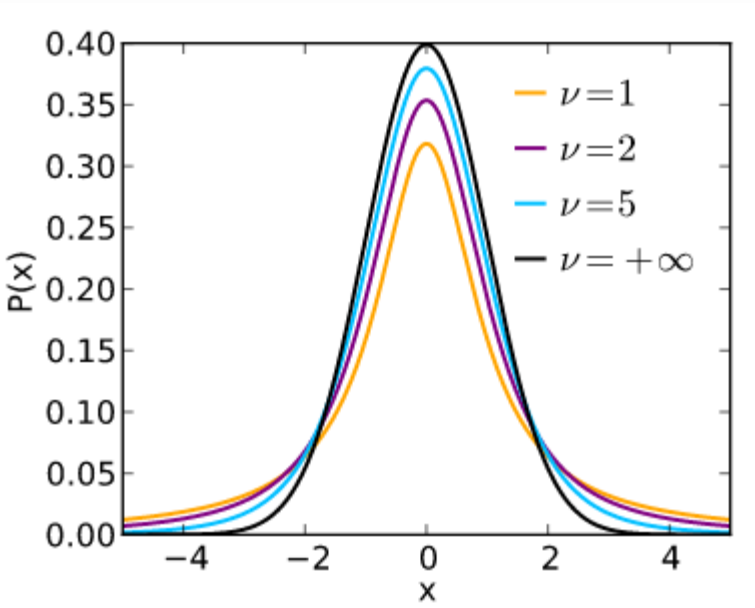


CI for means of general distributions, *unknown* variance

- Given D_θ with mean parameter θ with *unknown* variance
- $\hat{\sigma}_n^2 \coloneqq \frac{\sum_{i=1}^n (X_i - \hat{\mu}_n)^2}{n-1} \Longrightarrow$ unbiased estimator of $\mathrm{var}(D_\theta)$
- (Thm) Let $X_1, \ldots, X_n \sim N(\mu, \sigma^2)$, and $\hat{\mu}_n \coloneqq \frac{1}{n}\sum_{i=1}^n X_i$

$$\sqrt{n}\frac{\hat{\mu}_n - \mu}{\hat{\sigma}_n} \sim \text{student-t (mean 0, scale 1, degrees of freedom} = n-1)$$

- CI: $\left[\hat{\mu}_n \pm \frac{\hat{\sigma}_n \cdot t_\alpha}{\sqrt{n}}\right]$




```
import scipy.stats as st
alpha = 0.05
st.t.ppf(1-alpha/2,df=2)
=> 4.302652729911275

st.t.ppf(1-alpha/2,df=5)
=> 2.5705818366147395

st.t.ppf(1-alpha/2,df=10)
=> 2.2281388519649385

st.t.ppf(1-alpha/2,df=30)
=> 2.0422724563012373

st.t.ppf(1-alpha/2,df=100)
=> 1.9839715184496334
```

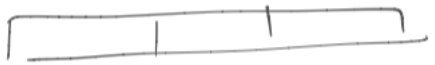
CI for complex evaluation functions

- So far, each trial was one test point, and the score of interest takes explicit “average”
  - each test point is i.i.d., so it was “easy” to compute CI’s

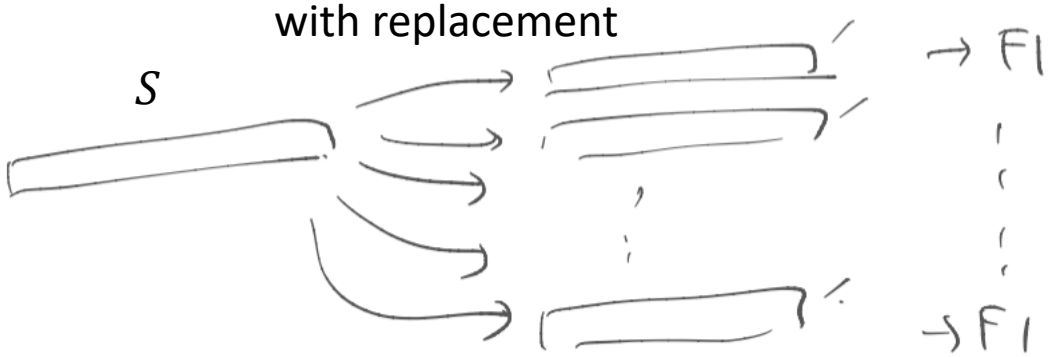
- What about other evaluation functions, e.g. F1 score?

$$F_1 = \frac{2}{\mathrm{recall}^{-1} + \mathrm{precision}^{-1}}$$

- (1) k-fold split:
  - $S_f, f = 1, \ldots, k$ are iid



- (2) bootstrap



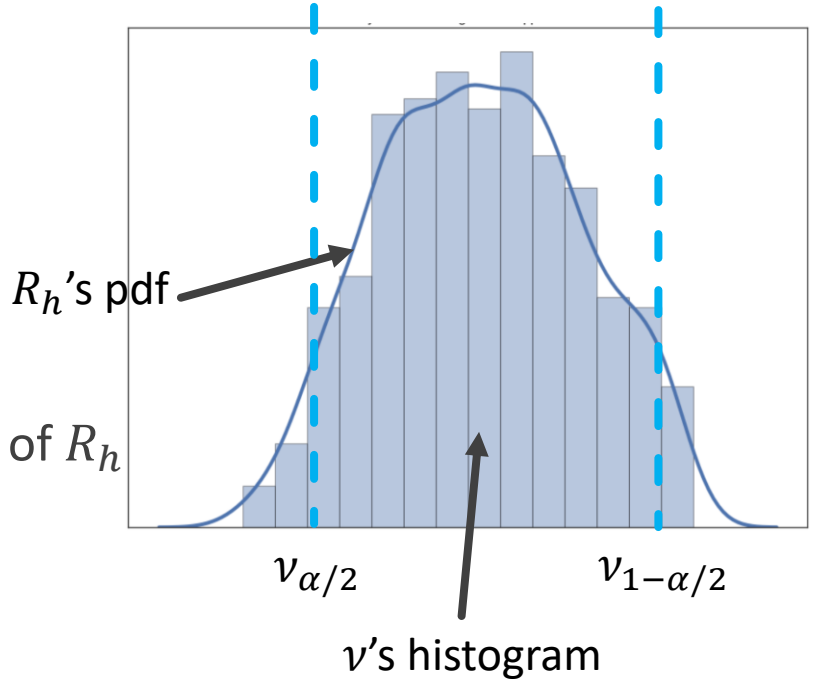
Bootstrapping CI

empirical distribution of $\{X_1, \dots, X_n\}$:
$\frac{1}{n}\sum_{i=1}^{n} \delta_{X_i}$ where δ_X is a dirac delta function

- Goal: estimate property h of D (:=$h(D)$) using confidence intervals, using sample S (e.g. h=F1 of model f)

- Idea: estimate the distribution of $h(S) - h(D)$, denoted by R_h

  by *bootstrapping* (resampling)
  - perform n times of "sampling with replacement" from S
  - repeat B times (e.g., B $\approx 10^4$) to obtain $S_1, \dots, S_B$
  - take $\nu :=$ **empirical distribution** of $\{h(S_b) - h(S_0)\}_{b=1}^{B}$, as the **'shape'** of R_h



- Assumption: $h(S) - h(D) \sim R_h \approx \text{emp\_distribution}\left[\{h(S_b) - h(S_0)\}_{b=1}^{B}\right]$

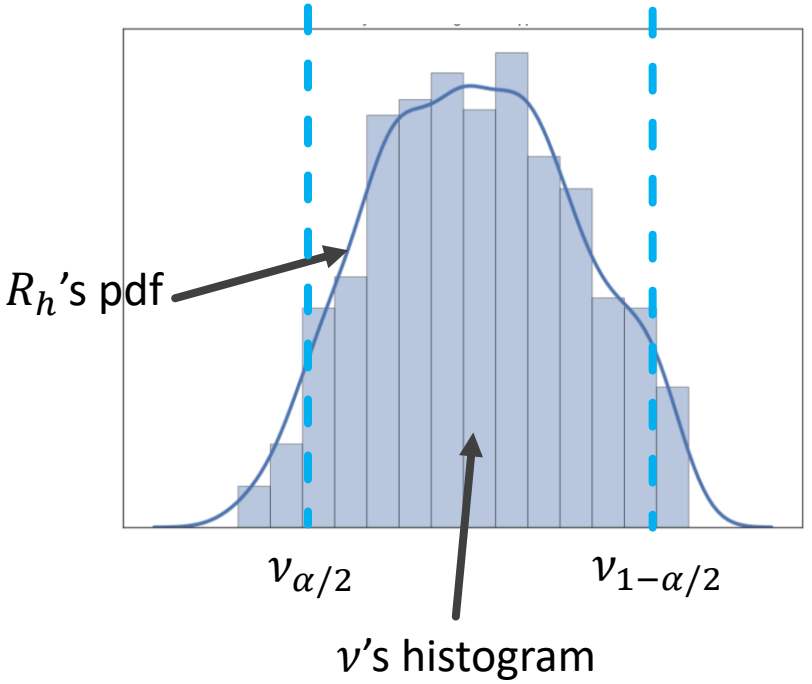
  Quantile interval: sort values and take top/bottom-quantiles (see next slide)

- With prob. $\approx 1 - \alpha$, $h(S) - h(D) \in [\nu_{\alpha/2}, \nu_{1-\alpha/2}] \Longrightarrow I(S) = [h(S) + \nu_{\alpha/2}, h(S) + \nu_{1-\alpha/2}]$

Bootstrapping CI: Implementation

- From bootstrapping, obtain $\{h(S_b) - h(S)\}_{b=1}^{B}$

- How to calculate its empirical distribution's quantiles?
  - Sort them in increasing order; say v[0..(B-1)]
  - $\nu_{1-\alpha/2}$ ≔ the top 0.025 (i.e., v[int(0.975*B)])
  - $\nu_{\alpha/2}$ ≔ the bottom 0.025 (i.e., v[int(0.025*B)])



Hypothesis testing: motivation

- How to claim your new system A is better than existing one B
- Ex1: each test data point => take prediction from A & B => record correct/not

- Ex2: each evaluator => a random keyword is picked, and then both systems pick top 10 relevant documents and rank them => the evaluator provides rating (1-5) for both lists.

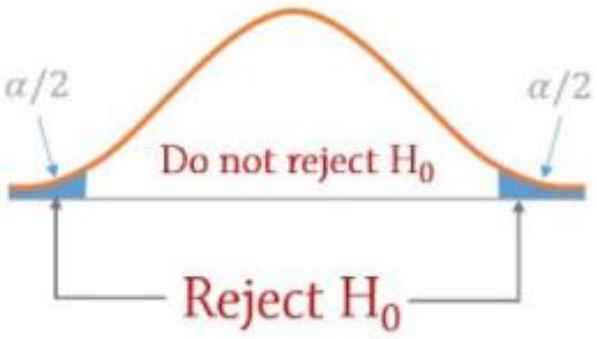
Evaluator	1	2	3	4	5	6	…
A	5	2	2	5	4	2	…
B	4	1	1	4	3	1	…

One-sample hypothesis testing: definition

- Given D_θ with parameter θ
- Sample $S = (X_1, \dots, X_n)$ drawn iid from distribution D_θ
- Equality test version:
  - Null hypothesis $H_0: \theta = \theta_0$
  - Alternative hypothesis $H_1: \theta \neq \theta_0$

- E.g. $D_\mu = \mathrm{Ber}(\mu)$, $H_0: \mu$=7%;
  $D_\mu = N(\mu, 1)$, $H_0: \mu$=23;

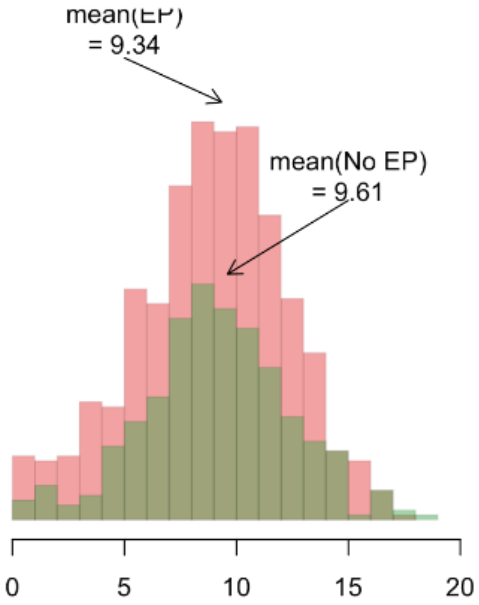
$$H_0: \mu = 23$$
$$H_1: \mu \neq 23$$



- Hypothesis test T: maps S to $\{0,1\}$
  - $T(S) = 0/1$: accept / reject the null hypothesis H_0
  - Goal: minimize type-II error $P_{H_1}(T(S) = 0)$
    s.t. type-I error $P_{H_0}(T(S) = 1) \leq \alpha$:= significance level

Two-sample hypothesis testing: definition

- Given D_θ with parameter θ
- Samples $S_X = (X_1, \dots, X_n)$ and $S_Y = (Y_1, \dots, Y_n)$ drawn iid from distribution D_{θ_X} and D_{θ_Y}, respectively

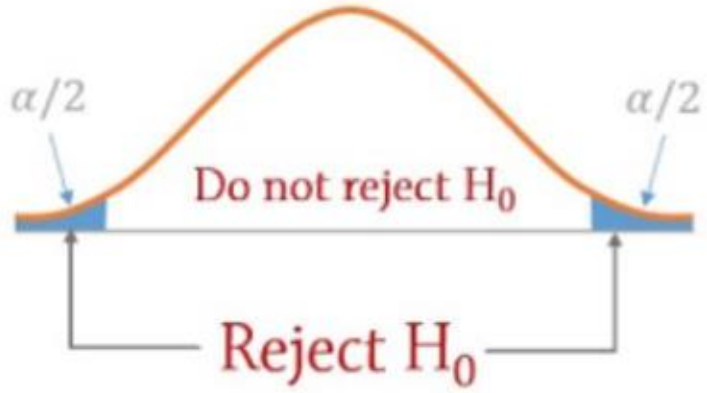


- Equality test version:
  - Null hypothesis H_0: $\theta_X = \theta_Y$
  - Alternative hypothesis H_1:$\theta_X \neq \theta_Y$
- E.g. $D_\mu = \mathrm{Ber}(\mu)$, H_0: $\mu_X = \mu_Y$
- Similarly, design hypothesis tester T such that the two types of errors are controlled

Paired t-test

- $S_X = (X_1, \dots, X_n)$ and $S_X = (Y_1, \dots, Y_n)$ drawn iid from distribution $D_{\theta_X} = N(\mu_X, \sigma_X^2)$ and $D_{\theta_Y} = N(\mu_Y, \sigma_Y^2)$, respectively
  - $H_0: \mu_X = \mu_Y$
  - $H_1: \mu_X \neq \mu_Y$

- Let $\delta_i := X_i - Y_i$, for all $i = 1, \dots, n$
- Let $\bar{\delta}_n := \frac{1}{n}\sum_{i=1}^{n} \delta_i$



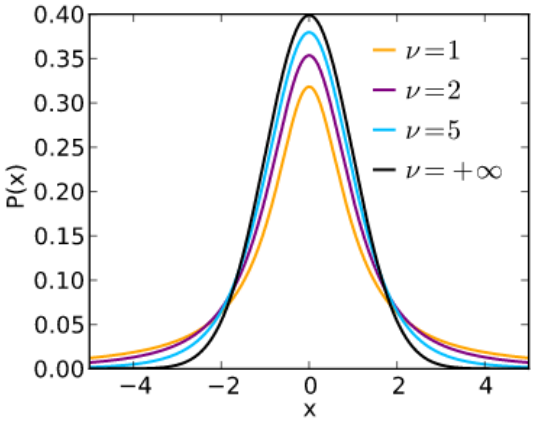
- Design hypothesis test T so that $P_{H_0}(T(S) = 0) \geq 1 - \alpha$
- Intuition: reasonable to reject if $|\bar{\delta}_n|$ is large

Paired t-test

- Under H_0, $\delta_i \sim N(0, \sigma^2), i = 1, \dots, n$, where $\sigma^2 = \sigma_X^2 + \sigma_Y^2$

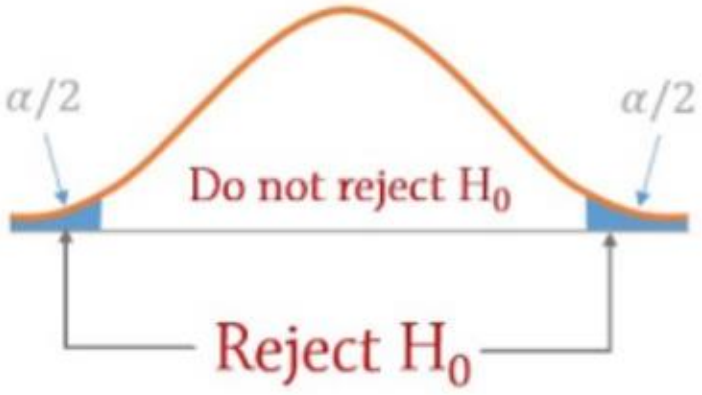
- Recall Thm: Let $\delta_1, \dots, \delta_n \sim N(0, \sigma^2)$, and $\bar{\delta}_n := \frac{1}{n}\sum_{i=1}^{n} \delta_i$, $\hat{\sigma}_n^2 := \frac{\sum_{i=1}^{n}(\delta_i - \bar{\delta}_n)^2}{n-1}$

$$Z = \sqrt{n}\frac{\bar{\delta}_n}{\hat{\sigma}_n} \sim \text{student-t (mean 0, scale 1, degrees of freedom = } n-1)$$



- Let's ask "under H_0, what is a plausible range of values of Z with failure rate $\alpha = 0.05$?"
  - Find the 0.025, 0.975-quantiles of Z => $t_{0.025}, t_{0.975}$
  - Hypothesis tester

$$T(S) = I(Z \notin [t_{0.025}, t_{0.975}]) = I\left(\sqrt{n}\frac{\bar{\delta}_n}{\hat{\sigma}_n} \notin [t_{0.025}, t_{0.975}]\right)$$



Testing for non-paired data: Permutation test

- A/B testing: Say you are a Netflix ML engineer. Spare 5% of the traffic, and randomly split it into two (2.5% each) and test recommendation system X vs Y. Each score is 0/1.
- You have now tested A **991** times (thus 991 scores) and B **1011** times (1011 scores).
- You hardly get “paired” data, when you deploy your system!

- H_0: the two system’s scores we saw came from the same distribution (=: null distribution)
- H_1: the two system’s scores we saw came from different distributions

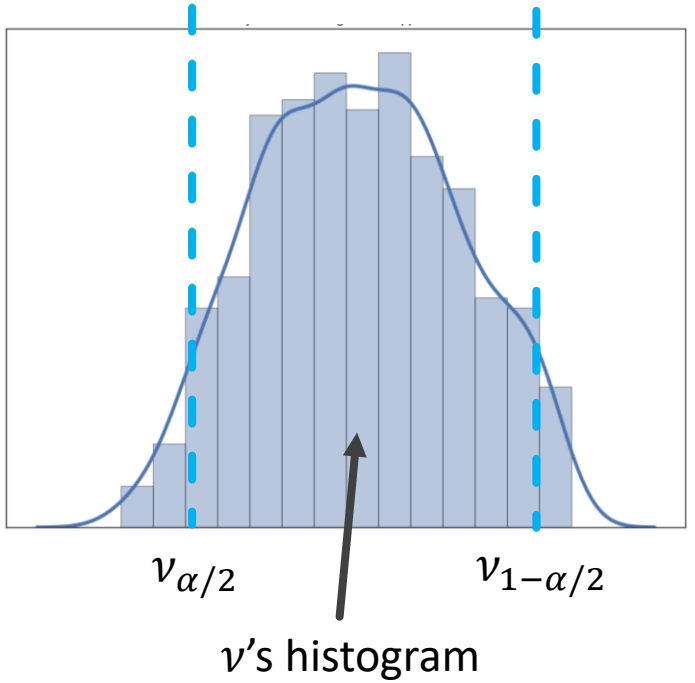
- Intuition: calculate f(X)-f(Y) with some ``discriminator’’ f, reject if its value is ``atypical’’
- How to define the set of ``typical’’ values?

(bootstrap version also available, but permutation test is simpler)

Permutation test

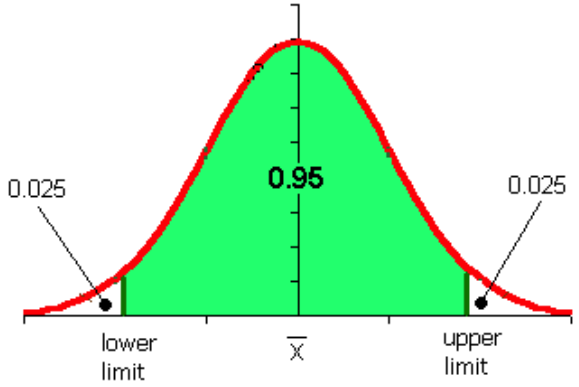
[python code]

- Given: set of scores X (size N1), set of scores Y (size N2)
  evaluation function f(), significance level α (default 0.05)
- 1. Estimate the null distribution:
  For $i = 1, \ldots, B$ (e.g., $B = 10^4$)
  - Concatenate X and Y; call it Z (size N1+N2)
  - Shuffle Z (e.g., Z = Z[numpy.random.permutation(N1+N2)])
  - Split Z into X' (size N1) and Y' (size N2) (e.g., Xp = Z[:N1]; Yp = C[N1:])
  - Let δ_i = f(X') – f(Y') (e.g., delta[i-1] = f(X') - f(Y'))
- 2. Compute the quantiles
  - Sort $\{\delta_i\}$ in decreasing order.
  - U := top $\frac{\alpha}{2}$ quantile (e.g., delta[int((N1+N2)*alpha/2)])
  - L := bottom $\frac{\alpha}{2}$ quantile (e.g., delta[int((N1+N2)*(1-alpha/2))])
- 3. Does [L,U] contain f(X) – f(Y) ? Yes => passed the test; No => failed the test
- Key idea: under H_0, f(X) – f(Y) should have the same distribution as the δ_i's



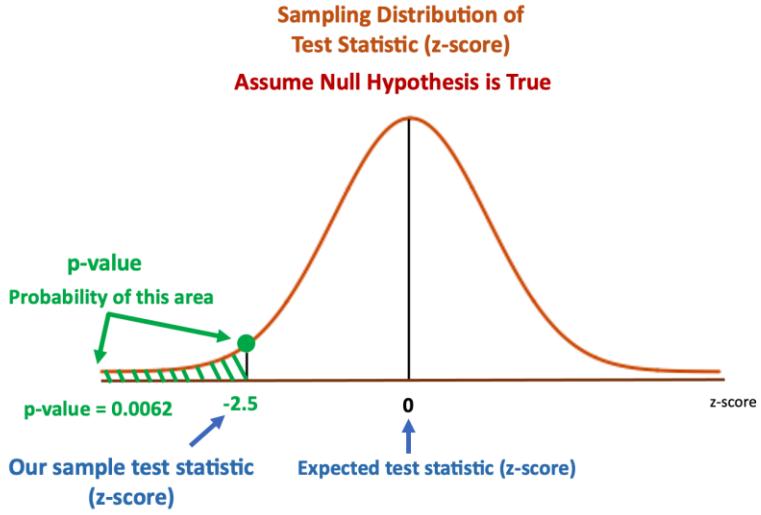
Hypothesis testing: additional remarks

- Confidence intervals can be used for hypothesis testing
  - $S = (X_1, \ldots, X_n)$ drawn iid from distribution D_μ
  - $H_0: \mu = 0$
  - $H_1: \mu \neq 0$
  - I is a $(1-\alpha)$-CI construction for μ => hypothesis test $T(S) = I(0 \notin I(S))$ has significance α



- p-value: given dataset S, and a family of hypothesis tests T_α's with different significance α's

  $p =$ the smallest α with which you can still reject H_0



Other materials

- Bootstrap test: https://ocw.mit.edu/courses/mathematics/18-05-introduction-to-probability-and-statistics-spring-2014/readings/MIT18_05S14_Reading24.pdf

- Permutation test: https://www.jwilber.me/permutationtest/

- STAT 566 lecture slides (at UA): https://www.math.arizona.edu/~jwatkins/stat566s20s.html

Next lecture (9/19)

- Linear models revisited: classification, regression, loss minimization formulations

- Assigned reading: CIML Sections 7.4-7.6