Data Mining and Analytics II

Aik Choon Tan, Ph.D.
Associate Professor of Bioinformatics
Division of Medical Oncology
Department of Medicine
aikchoon.tan@ucdenver.edu
9/21/2018
http://tanlab.ucdenver.edu/labHomePage/teaching/BSBT6111/

Outline

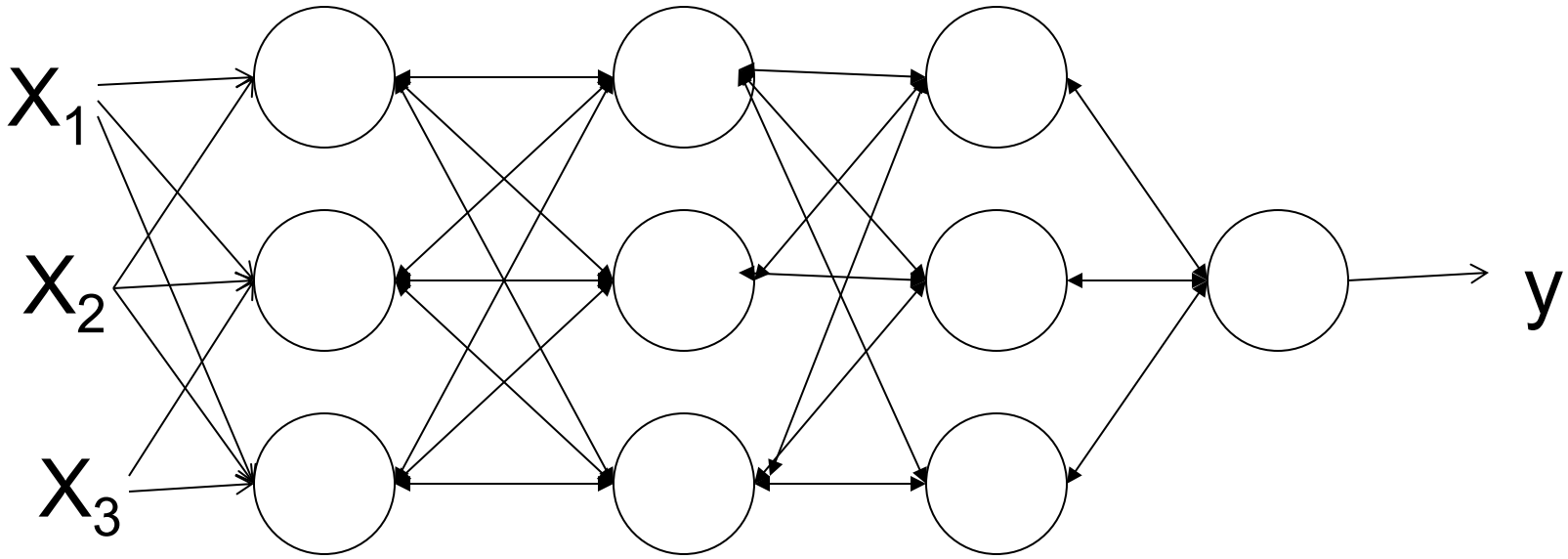
- Introduction
- (*Selected*) Machine Learning Approaches
  - Deep learning - AlphaGo
  - Naïve Bayes
  - Recommendation System
  - Ensemble Approach
  - Clustering
- Feature Selection
- Model evaluation

Deep Learning

Deep learning (also known as deep structured learning, hierarchical learning or deep machine learning) is a branch of machine learning based on a set of algorithms that attempt to model high-level abstractions in data by using a deep graph with multiple processing layers, composed of multiple linear and non-linear transformations.

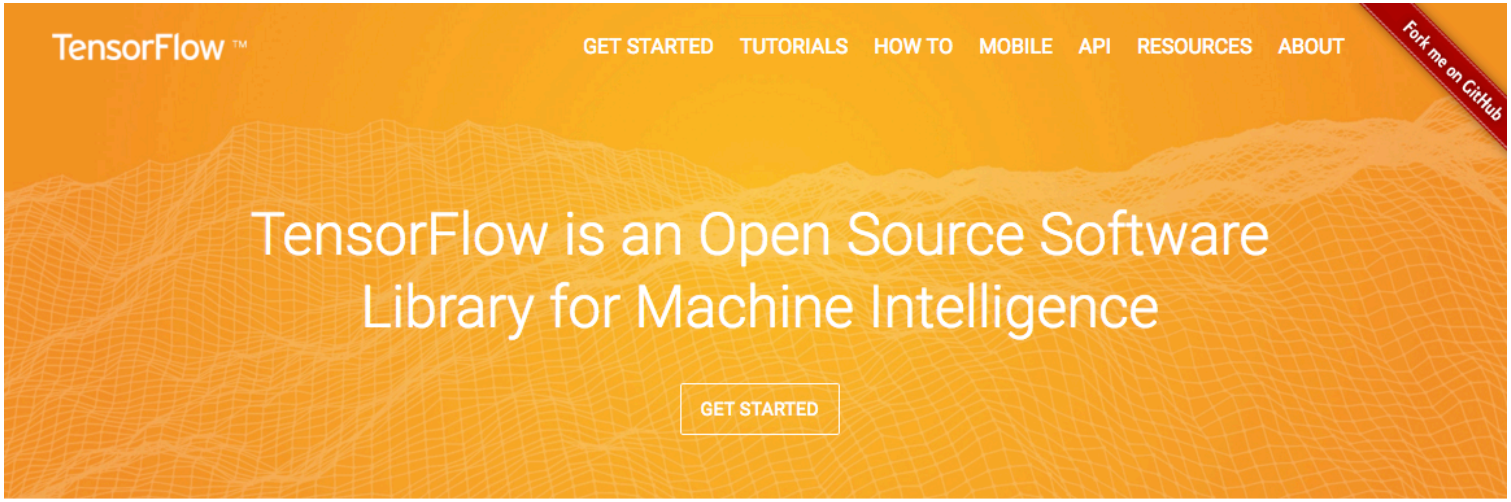
(From Wikipedia)

An illustration



Deep = more “nodes” and “hidden” layers

TensorFlow



https://www.tensorflow.org/

About TensorFlow

TensorFlow™ is an open source software library for numerical computation using data flow graphs. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them. The flexible architecture allows you to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API.
TensorFlow was originally developed by researchers and engineers working on the Google Brain Team within Google's Machine Intelligence research organization for the purposes of conducting machine learning and deep neural networks research, but the system is general enough to be applicable in a wide variety of other domains as well.

Example

http://playground.tensorflow.org/

https://www.youtube.com/watch?v=Iv0o9Lw3nz0

https://www.ted.com/talks/fei_fei_li_how_we_re_teaching_computers_to_understand_pictures?language=en#t-118437

AlphaGO-
A little bit more on Deep Learning

- AlphaGO – general purpose AI

ARTICLE

doi:10.1038/nature16961

Mastering the game of Go with deep neural networks and tree search

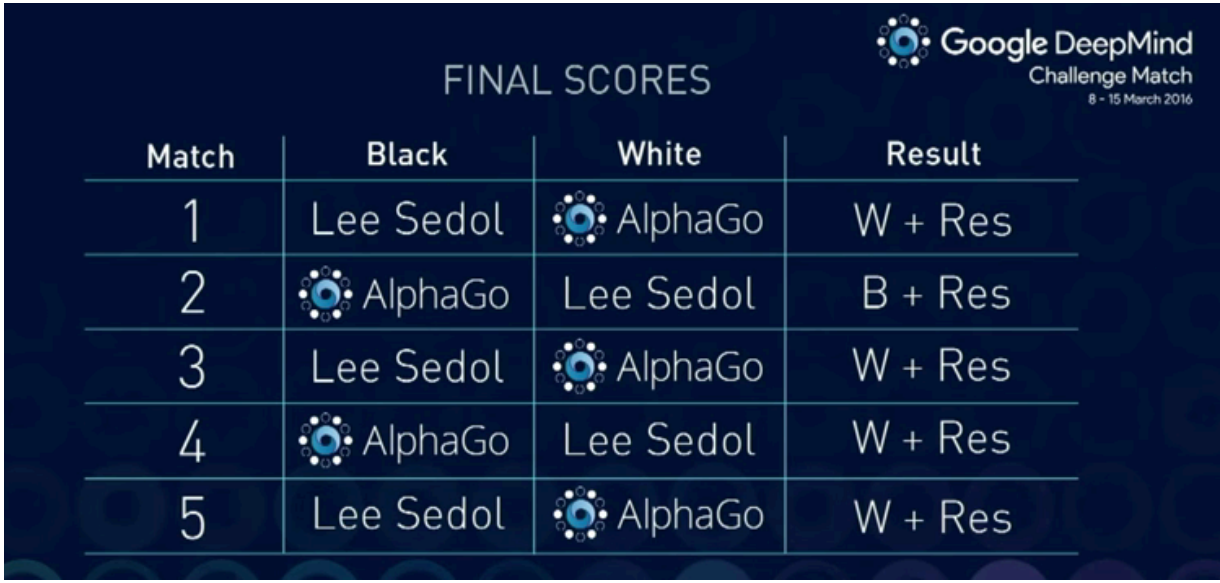
David Silver[1]*, Aja Huang[1]*, Chris J. Maddison[1], Arthur Guez[1], Laurent Sifre[1], George van den Driessche[1], Julian Schrittwieser[1], Ioannis Antonoglou[1], Veda Panneershelvam[1], Marc Lanctot[1], Sander Dieleman[1], Dominik Grewe[1], John Nham[2], Nal Kalchbrenner[1], Ilya Sutskever[2], Timothy Lillicrap[1], Madeleine Leach[1], Koray Kavukcuoglu[1], Thore Graepel[1] & Demis Hassabis[1]

The game of Go has long been viewed as the most challenging of classic games for artificial intelligence owing to its enormous search space and the difficulty of evaluating board positions and moves. Here we introduce a new approach to computer Go that uses 'value networks' to evaluate board positions and 'policy networks' to select moves. These deep neural networks are trained by a novel combination of supervised learning from human expert games, and reinforcement learning from games of self-play. Without any lookahead search, the neural networks play Go at the level of state-of-the-art Monte Carlo tree search programs that simulate thousands of random games of self-play. We also introduce a new search algorithm that combines Monte Carlo simulation with value and policy networks. Using this search algorithm, our program AlphaGo achieved a 99.8% winning rate against other Go programs, and defeated the human European Go champion by 5 games to 0. This is the first time that a computer program has defeated a human professional player in the full-sized game of Go, a feat previously thought to be at least a decade away.

https://www.youtube.com/watch?v=SUbqykXVx0A

AlphaGO-
A little bit more on Deep Learning

https://www.youtube.com/watch?v=TnUYcTuZJpM



Match	Black	White	Result
1	Lee Sedol	AlphaGo	W + Res
2	AlphaGo	Lee Sedol	B + Res
3	Lee Sedol	AlphaGo	W + Res
4	AlphaGo	Lee Sedol	W + Res
5	Lee Sedol	AlphaGo	W + Res

Deep Learning in Medicine

Research

JAMA | Original Investigation | INNOVATIONS IN HEALTH CARE DELIVERY

Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs

Varun Gulshan, PhD; Lily Peng, MD, PhD; Marc Coram, PhD; Martin C. Stumpe, PhD; Derek Wu, BS; Arunachalam Narayanaswamy, PhD; Subhashini Venugopalan, MS; Kasumi Widner, MS; Tom Madams, MEng; Jorge Cuadros, OD, PhD; Ramasamy Kim, OD, DNB; Rajiv Raman, MS, DNB; Philip C. Nelson, BS; Jessica L. Mega, MD, MPH; Dale R. Webster, PhD

IMPORTANCE Deep learning is a family of computational methods that allow an algorithm to program itself by learning from a large set of examples that demonstrate the desired behavior, removing the need to specify rules explicitly. Application of these methods to medical imaging requires further assessment and validation.

OBJECTIVE To apply deep learning to create an algorithm for automated detection of diabetic retinopathy and diabetic macular edema in retinal fundus photographs.

DESIGN AND SETTING A specific type of neural network optimized for image classification called a deep convolutional neural network was trained using a retrospective development data set of 128 175 retinal images, which were graded 3 to 7 times for diabetic retinopathy, diabetic macular edema, and image gradability by a panel of 54 US licensed ophthalmologists and ophthalmology senior residents between May and December 2015. The resultant algorithm was validated in January and February 2016 using 2 separate data sets, both graded by at least 7 US board-certified ophthalmologists with high intragrader consistency.

EXPOSURE Deep learning–trained algorithm.

MAIN OUTCOMES AND MEASURES The sensitivity and specificity of the algorithm for detecting referable diabetic retinopathy (RDR), defined as moderate and worse diabetic retinopathy, referable diabetic macular edema, or both, were generated based on the reference standard of the majority decision of the ophthalmologist panel. The algorithm was evaluated at 2 operating points selected from the development set, one selected for high specificity and another for high sensitivity.

RESULTS The EyePACS-1 data set consisted of 9963 images from 4997 patients (mean age, 54.4 years; 62.2% women; prevalence of RDR, 683/8878 fully gradable images [7.8%]); the Messidor-2 data set had 1748 images from 874 patients (mean age, 57.6 years; 42.6% women; prevalence of RDR, 254/1745 fully gradable images [14.6%]). For detecting RDR, the algorithm had an area under the receiver operating curve of 0.991 (95% CI, 0.988-0.993) for EyePACS-1 and 0.990 (95% CI, 0.986-0.995) for Messidor-2. Using the first operating cut point with high specificity, for EyePACS-1, the sensitivity was 90.3% (95% CI, 87.5%-92.7%) and the specificity was 98.1% (95% CI, 97.8%-98.5%). For Messidor-2, the sensitivity was 87.0% (95% CI, 81.1%-91.0%) and the specificity was 98.5% (95% CI, 97.7%-99.1%). Using a second operating point with high sensitivity in the development set, for EyePACS-1 the sensitivity was 97.5% and specificity was 93.4% and for Messidor-2 the sensitivity was 96.1% and specificity was 93.9%.

CONCLUSIONS AND RELEVANCE In this evaluation of retinal fundus photographs from adults with diabetes, an algorithm based on deep machine learning had high sensitivity and specificity for detecting referable diabetic retinopathy. Further research is necessary to determine the feasibility of applying this algorithm in the clinical setting and to determine whether use of the algorithm could lead to improved care and outcomes compared with current ophthalmologic assessment.

JAMA. 2016;316(22):2402-2410. doi:10.1001/jama.2016.17216
Published online November 29, 2016. Corrected on December 13, 2016.

Editorial pages 2366 and 2368
Supplemental content

Author Affiliations: Google Inc, Mountain View, California (Gulshan, Peng, Coram, Stumpe, Wu, Narayanaswamy, Venugopalan, Widner, Madams, Nelson, Webster); Department of Computer Science, University of Texas, Austin (Venugopalan); EyePACS LLC, San Jose, California (Cuadros); School of Optometry, Vision Science Graduate Group, University of California, Berkeley (Cuadros); Aravind Medical Research Foundation, Aravind Eye Care System, Madurai, India (Kim); Shri Bhagwan Mahavir Vitreoretinal Services, Sankara Nethralaya, Chennai, Tamil Nadu, India (Raman); Verily Life Sciences, Mountain View, California (Mega); Cardiovascular Division, Department of Medicine, Brigham and Women's Hospital and Harvard Medical School, Boston, Massachusetts (Mega).
Corresponding Author: Lily Peng, MD, PhD, Google Research, 1600 Amphitheatre Way, Mountain View, CA 94043 (lhpeng@google.com).

Table. Baseline Characteristics[a]

Characteristics	Development Data Set	EyePACS-1 Validation Data Set	Messidor-2 Validation Data Set
No. of images	128 175	9963	1748
No. of ophthalmologists	54	8	7
No. of grades per image	3-7	8	7
Grades per ophthalmologist, median (interquartile range)	2021 (304-8366)	8906 (8744-9360)	1745 (1742-1748)
Patient demographics			
No. of unique individuals	69 573[b]	4997	874
Age, mean (SD), y	55.1 (11.2)[c]	54.4 (11.3)	57.6 (15.9)
Female, No./total (%) among images for which sex was known	50 769/84 734 (59.9)[c]	5463/8784 (62.2)	743/1745 (42.6)
Image quality distribution			
Fully gradable, No./total (%) among images for which image quality was assessed	52 311/69 598 (75.1)[d]	8788/9946 (88.4)	1745/1748 (99.8)
Disease severity distribution classified by majority decision of ophthalmologists (reference standard)			
Total images for which both diabetic retinopathy and diabetic macular edema were assessed, No. (%)	118 419 (100)	8788 (100)	1745 (100)
No diabetic retinopathy	53 759 (45.4)	7252 (82.5)	1217 (69.7)
Mild diabetic retinopathy	30 637 (25.9)	842 (9.6)	264 (15.1)
Moderate diabetic retinopathy	24 366 (20.6)	545 (6.2)	211 (12.1)
Severe diabetic retinopathy	5298 (4.5)	54 (0.6)	28 (1.6)
Proliferative diabetic retinopathy	4359 (3.7)	95 (1.1)	25 (1.4)
Referable diabetic macular edema	18 224 (15.4)	272 (3.1)	125 (7.2)
Referable diabetic retinopathy[e]	33 246 (28.1)	683 (7.8)	254 (14.6)

[a] Summary of image characteristics and available demographic information in the development and clinical validation data sets (EyePACS-1 and Messidor-2). Abnormal images were oversampled for the development set for algorithm training. The clinical validation sets were not enriched for abnormal images.

[b] Unique patient codes (deidentified) were available for 89.3% of the development set (n = 114 398 images).

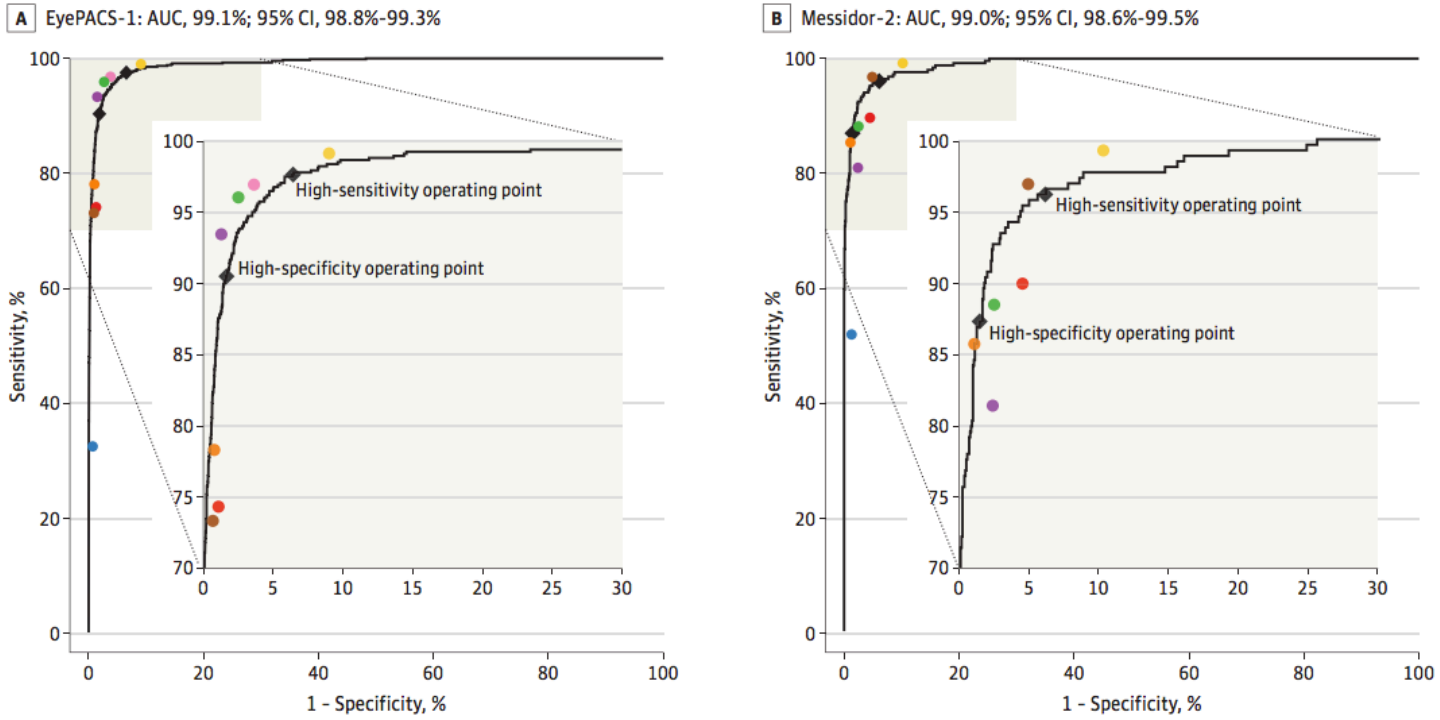
[c] Individual-level data including age and sex were available for 66.1% of the development set (n = 84 734 images).

[d] Image quality was assessed for a subset of the development set.

[e] Referable diabetic retinopathy, defined as the presence of moderate and worse diabetic retinopathy and/or referable diabetic macular edema according to the International Clinical Diabetic Retinopathy Scale,[14] was calculated for each ophthalmologist before combining them using a majority decision. The 5-point grades represent the grade that received the highest number of votes for diabetic retinopathy alone. Hence, the sum of moderate, severe, and proliferative diabetic retinopathy for the 5-point grade differs slightly from the count of referable diabetic retinopathy images.

Deep Learning in Medicine

Figure 2. Validation Set Performance for Referable Diabetic Retinopathy



Performance of the algorithm (black curve) and ophthalmologists (colored circles) for the presence of referable diabetic retinopathy (moderate or worse diabetic retinopathy or referable diabetic macular edema) on A, EyePACS-1 (8788 fully gradable images) and B, Messidor-2 (1745 fully gradable images). The black diamonds on the graph correspond to the sensitivity and specificity of the algorithm at the high-sensitivity and high-specificity operating points. In A, for the high-sensitivity operating point, specificity was 93.4% (95% CI, 92.8%-94.0%) and sensitivity was 97.5% (95% CI, 95.8%-98.7%); for the high-specificity operating point, specificity was 98.1% (95% CI, 97.8%-98.5%) and sensitivity was 90.3% (95% CI, 87.5%-92.7%). In B, for the high-sensitivity operating point, specificity was 93.9% (95% CI, 92.4%-95.3%) and sensitivity was 96.1% (95% CI, 92.4%-98.3%); for the high-specificity operating point, specificity was 98.5% (95% CI, 97.7%-99.1%) and sensitivity was 87.0% (95% CI, 81.1%-91.0%). There were 8 ophthalmologists who graded EyePACS-1 and 7 ophthalmologists who graded Messidor-2. AUC indicates area under the receiver operating characteristic curve.

Deep Learning in Medicine

Dermatologist-level classification of skin cancer

An artificial intelligence trained to classify images of skin lesions as benign lesions or malignant skin cancers achieves the accuracy of board-certified dermatologists.

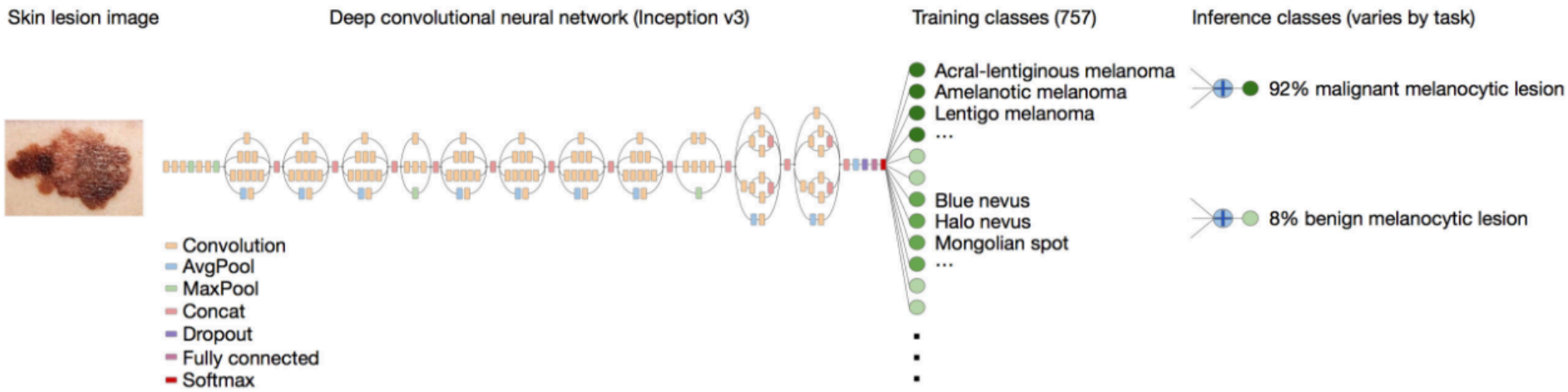
In this work, we pretrain a deep neural network at general object recognition, then fine-tune it on a dataset of ~130,000 skin lesion images comprised of over 2000 diseases.

FULL NATURE ARTICLE >

OPEN-ACCESS PDF >

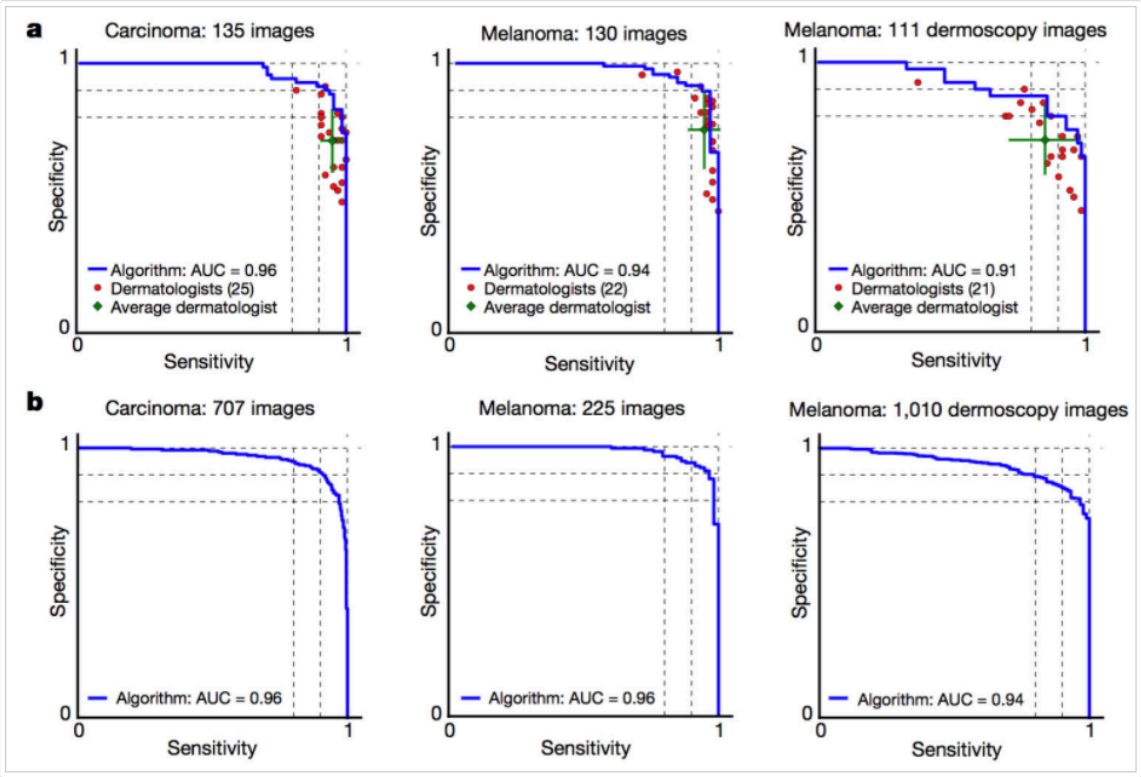
http://cs.stanford.edu/people/esteva/nature/

Deep Learning in Medicine



https://www.youtube.com/watch?time_continue=1&v=kCIvKNl0Wfc

Deep Learning in Medicine



Skin cancer classification performance of the CNN and dermatologists. a, The deep learning CNN outperforms the average of the dermatologists at skin cancer classification (keratinocyte carcinomas and melanomas) using photographic and dermoscopic images. For each test, previously unseen, biopsy-proven images of lesions are displayed, and dermatologists are asked if they would: biopsy/treat the lesion or reassure the patient. A dermatologist outputs a single prediction per image and is thus represented by a single red point. The green points are the average of the dermatologists for each task, with error bars denoting one standard deviation (calculated from n = 25, 22 and 21 tested dermatologists for carcinoma, melanoma and melanoma under dermoscopy, respectively). The CNN is represented by the blue curve, and the AUC is the CNN's measure of performance, with a maximum value of 1. The CNN achieves superior performance to a dermatologist if the sensitivity–specificity point of the dermatologist lies below the blue curve, which most do. b, The deep learning CNN exhibits reliable cancer classification when tested on a larger dataset. We tested the CNN on more images to demonstrate robust and reliable cancer classification. The CNN's curves are smoother owing to the larger test set.

Bayes Theorem

In machine learning we are interested to determine the best hypothesis *h(**x**)* from space *H*, based on the observed training data ***x***.

Best hypothesis = *most probable* hypothesis, given the data ***x*** with any initial knowledge about the prior probabilities of the various hypothesis in *H*.

Bayes theorem provides a way to calculate

(i) the probability of a hypothesis based on its prior probability Pr(*h(**x**)*)
(ii) the probabilities of the observing various data given the hypothesis Pr(***x***|h)
(iii) the probabilities of the observed data Pr(***x***)

We can calculate the posterior probability *h(**x**)* given the observed data ***x***, Pr(*h(**x**)*|***x***) using *Bayes theorem*.

$$\Pr(h(x) \mid x) = \frac{\Pr(x \mid h(x)) \Pr(h(x))}{\Pr(x)}$$

Training Data

Decision attributes (dependent)

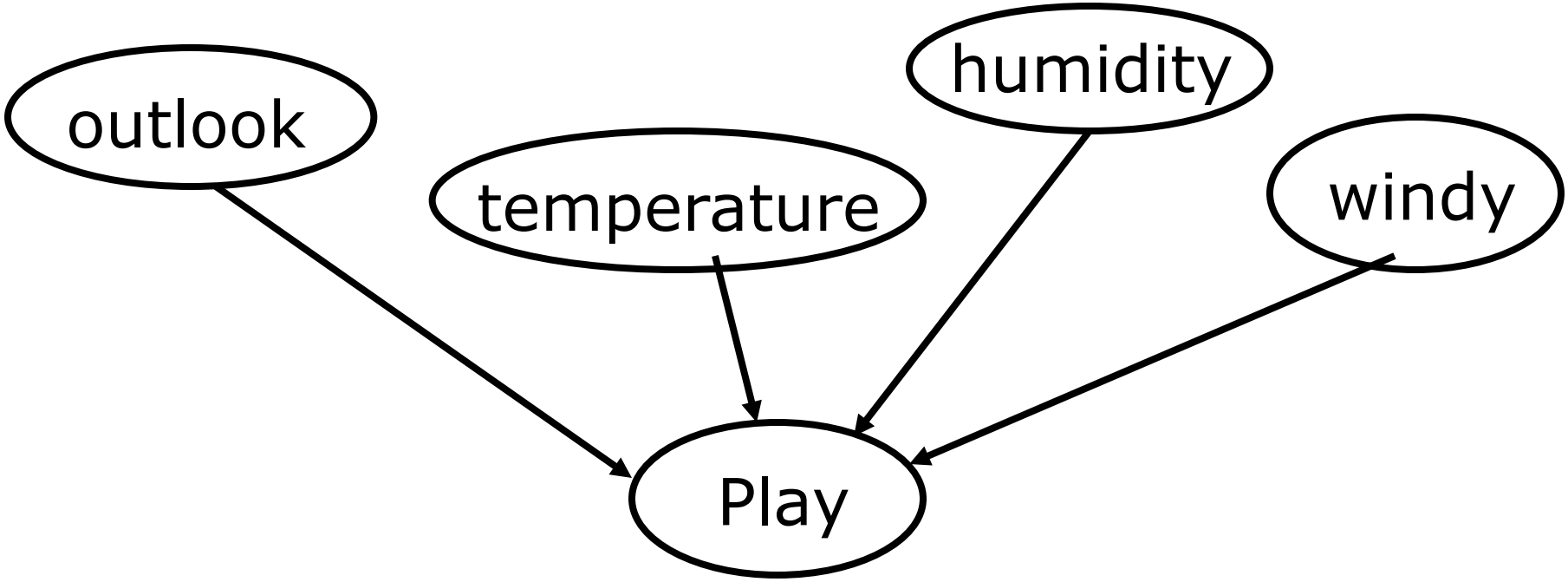
Independent condition attributes

Day	outlook	temperature	humidity	windy	play
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

Today	sunny	cool	high	TRUE	?

Naïve Bayes

(John & Langley, 1995)



To use all attributes and allow them to make contributions to the decision that are *equally important* and *independent* of one another, given the class.

Naïve Bayes Classifier

$$v_{NB} = \arg\max_{v_j \in V} \Pr(v_j) \prod_i \Pr(a_i \mid v_j)$$

Where v_{NB} denotes the target value output by the naïve Bayes classifer, $\Pr(v_j)$ is the probability of target value v_j occurs in the training data, $\Pr(a_i|v_j)$ is the conditionally independant probability of a_i given target value v_j.

Summary:

- The naïve Bayes learning method involves a learning step in which the various $\Pr(v_j)$ and $\Pr(a_i|v_j)$ terms are estimated, based on their frequencies over the training data.
- The set of these estimates corresponds to the learned hypothesis h(x).
- This hypothesis is then used to classify each new instance by applying the above rule.
- There is no explicit search through the space of possible hypothesis, instead the hypothesis is formed simply by counting the frequency of various data combinations within the training examples.

Naïve Bayes example

Today	sunny	cool	high	TRUE	?

Pr(Play = yes) = 9/14 = 0.64
Pr(Play = no) = 5/14 = 0.36

Pr(Outlook=sunny|Play = yes) = 2/9 = 0.22
Pr(Outlook=sunny|Play=no) = 3/5 = 0.60

Pr(Temperature = cool|Play = yes) =3/9 = 0.33
Pr(Temperature =cool|Play =no) =1/5 = 0.20

Pr(Humidity = high|Play = yes) = 3/9 =0.33
Pr(Humidity = high|Play = no) = 4/5 =0.80

Pr(Wind = TRUE|Play = yes) = 3/9 = 0.33
Pr(Wind = TRUE|Play = no) = 3/5 = 0.60

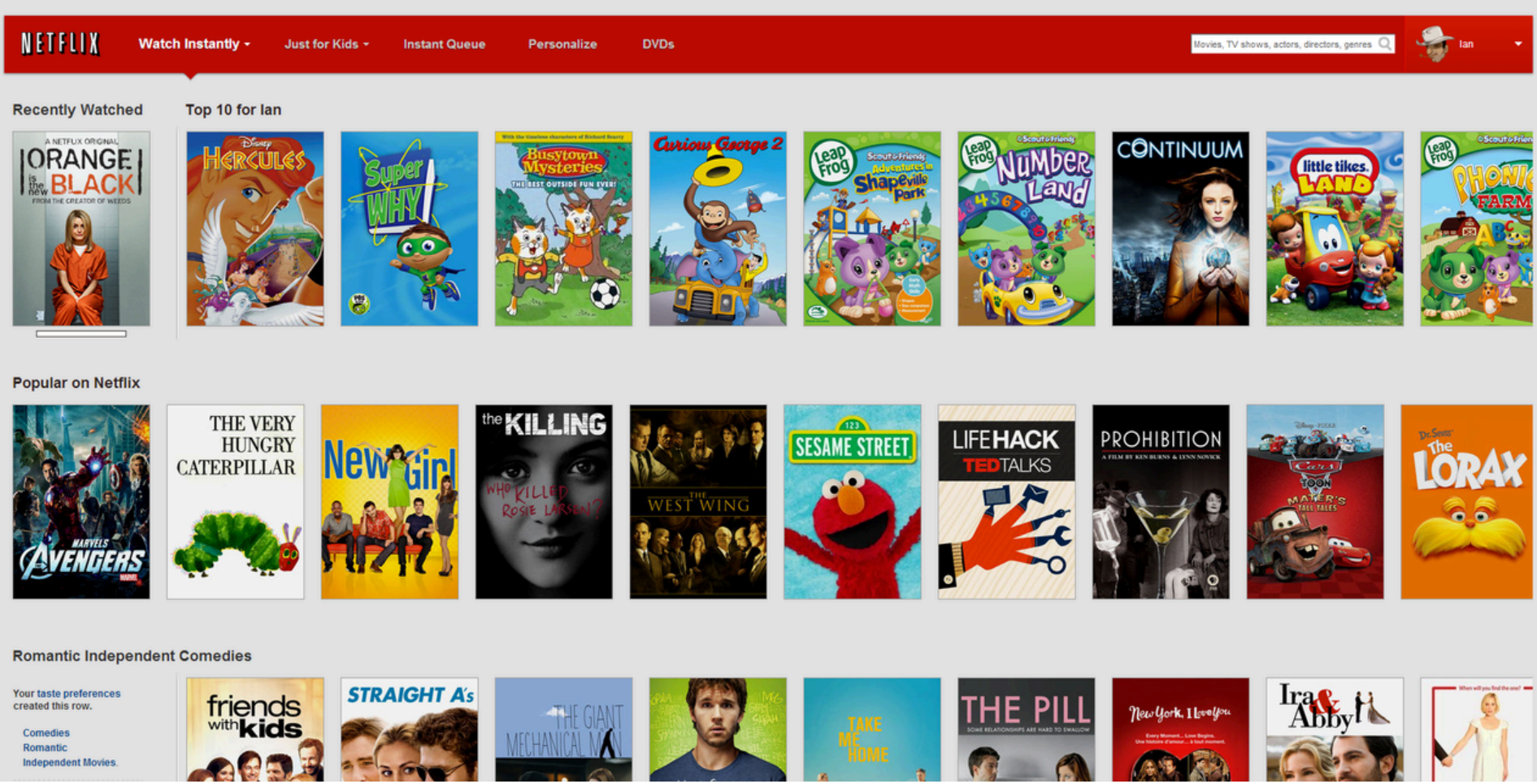
Pr(yes)Pr(sunny|yes)Pr(cool|yes) Pr(high|yes)Pr(TRUE|yes)= 0.64*0.22*0.33*0.33*0.33 = 0.0051

Pr(no)Pr(sunny|no)Pr(cool|no) Pr(high|no)Pr(TRUE|no)= 0.36*0.60*0.20*0.80*0.60 = 0.0207

Play = NO

Probability = 0.0207/(0.0207+0.0051) =0.80 (80%)

Example: Netflix Recommendation System https://www.youtube.com/watch?v=ImpV70uLxyw



Ensemble Approach

- “No Free Lunch Theory”
- Rationale – the combination of learning models increases the classification accuracy
- Idea – generate different learners (classifiers) from the training features that capture different “space”, combine these learners will provide a better classification
- Some approaches:
  - Boosting – combination of a set of “weak learners” to create a single “strong learner” – reducing bias and variance
    - AdaBoost
  - Bootstrap aggreating (Bagging) – to average noisy and unbiased models to create a model with low variance
  - Random Forest – large collection of decision trees generated with different features

Relative Expression Reversal Classifiers

BIOINFORMATICS ORIGINAL PAPER Vol. 21 no. 20 2005, pages 3896–3904
doi:10.1093/bioinformatics/bti631

Gene expression

Simple decision rules for classifying human cancers from gene expression profiles

Aik Choon Tan[1,*], Daniel Q. Naiman[1,2], Lei Xu[1], Raimond L. Winslow[1] and Donald Geman[1,2]

[1]Center for Cardiovascular Bioinformatics and Modeling, Whitaker Biomedical Engineering Institute, 3400 N. Charles Street, Baltimore, MD 21218, USA and [2]Department of Applied Mathematics and Statistics, Johns Hopkins University, 3400 N. Charles Street, Baltimore, MD 21218, USA

Received on May 9, 2005; revised on July 28, 2005; accepted on August 14, 2005
Advance Access publication August 16, 2005

- *Pairwise rank*-based *comparisons* (relative expression values *within each array*)
- Generates *accurate* and *simple* decision rules
  - ***TSP*** classifier: Top Scoring Pair
  - ***k-TSP*** classifier: *k*-disjoint Top Scoring Pairs
- *Data driven*, *parameter-free* learning algorithm
- *Performance comparable to or exceeds* that of other machine learning methods
- *Easy to interpret*, *facilitating follow-up study (small number of genes)*

k-TSP Algorithm

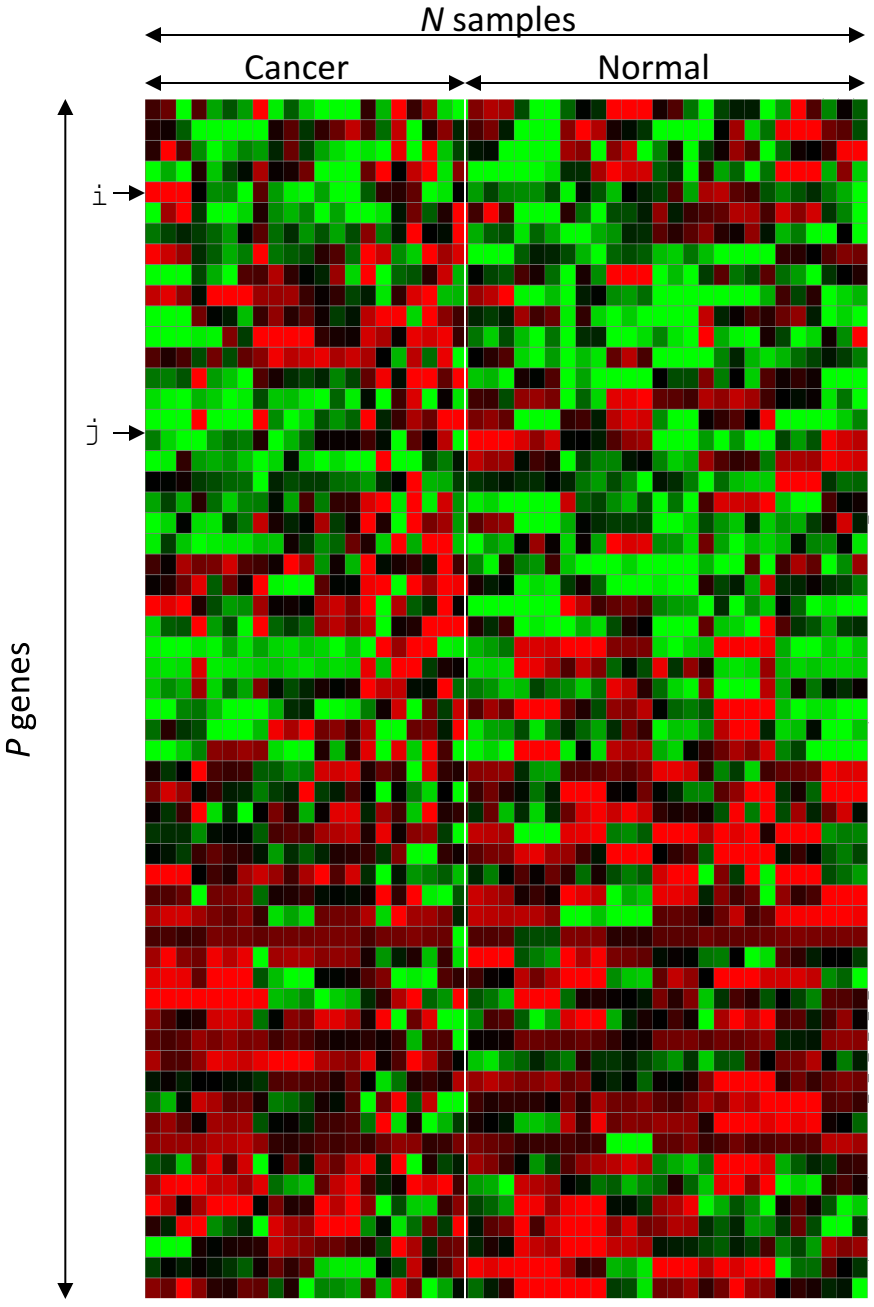
_k_-TSP Algorithm

Input: Training sample S of P genes and N arrays.

Output: k-TSP classifier $h_{k\text{-TSP}}$.

1. Set an upper bound ($k_{\max}$) on the number of top scoring pairs to be included in the final k-TSP classifier ($h_{k\text{-TSP}}$). ($k_{\max} = 10$ in this study.)
2. (Cross-validation) Repeat m times:
   a. Leave out n arrays from the training set S. ($n = 3$ and $m=N/3$ in this study.)
   b. Compute the score Δ_{ij} and the rank score Γ_{ij} on the current, reduced training set for every pair of genes (i, j), $1 \le i \ne j \le P$.
   c. Make an ordered list O of all of the gene pairs (i, j) from largest to smallest using the lexicographic ordering defined by setting $(i, j) \succ (i', j')$ whenever either $\Delta_{ij} > \Delta_{i'j'}$ or $\Delta_{ij} = \Delta_{i'j'}$ and $\Gamma_{ij} > \Gamma_{i'j'}$.
   d. Initialize Θ at the empty list and perform the following steps for k=1, 2,…, $k_{\max}$:
      i. Add the top pair (i, j) in the list O to Θ.
      ii. Remove every pair from O that involves either i or j.
      iii. If k is odd, compute the error rate for the classifier based on the k pairs in Θ.
3. Select the (odd) value of k whose average classification rate over the m loops in Step 2 is optimal and compute the classifier $h_{k\text{-TSP}}$ based on the top k scoring pairs as follows:
4. Make an ordered list O of gene pairs as in Steps 2b and 2c using the entire training set.
   a. Initialize Θ at the empty list.
   b. Repeat k times:
      i. Add the top pair (i, j) in O to Θ.
      ii. Remove every pair from O that involves either i or j.
5. Return $h_{k\text{-TSP}}$.

Basic concept of the Relative Expression Reversal algorithm



Goal: Find gene signature that can discriminate between cancer and normal samples

Pre-set number of k (k_max)

For each gene pair $(i,j) \subset P$, $(i \neq j)$:
Compute:
Prob (i>j | Cancer)
Prob (i>j | Normal)

Calculate a score:
$Score_{ij}$ =
|Prob(i>j|Cancer) - Prob(i>j|Normal)|

Note: $Score_{ij}$ = 1.0 is the highest score and it means in all the cancer cases,i>j; however, this gene expression pattern reversed in all the normal cases.

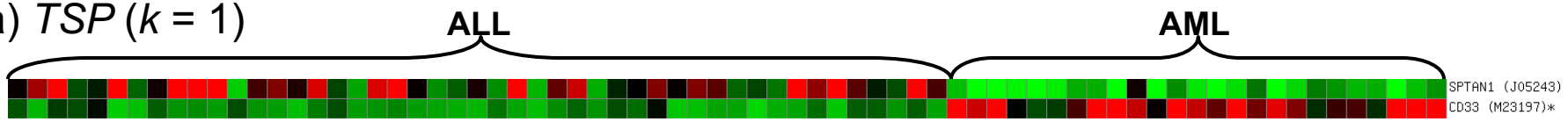
Sort the list of $Score_{ij}$ in descending order.

Pick the top pair as the top scoring pair (TSP). Second top pair as the second top scoring pair (k=2), etc.

Break ties with the maximum difference in signal intensity values between gene pair.

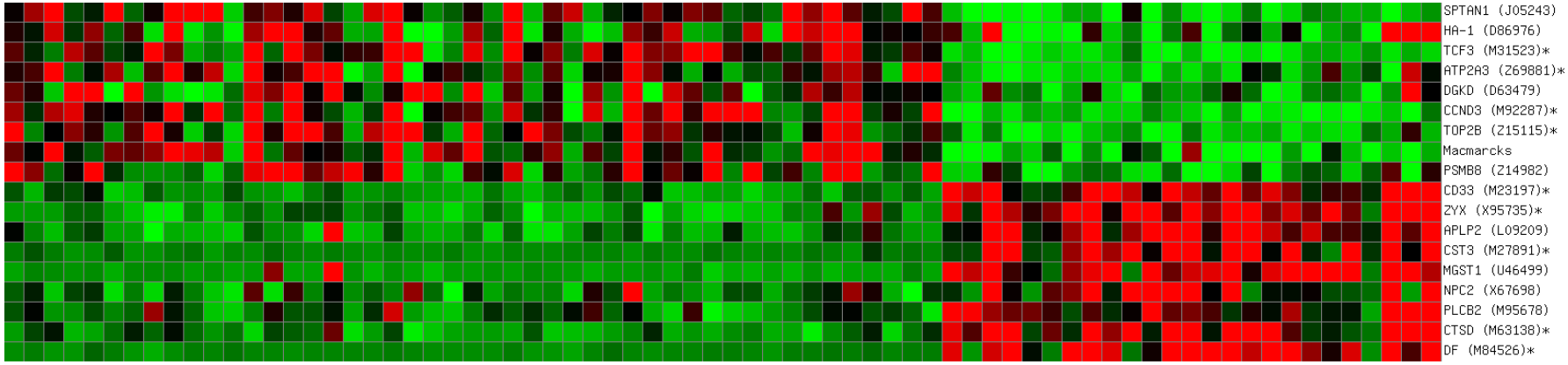
Repeat with LOOCV to find optimal k (k with lowest LOOCV error rate).

(a) *TSP* (*k* = 1)



IF SPTAN1 ≥ CD33* **THEN** ALL; **ELSE** AML Δ = 0.9787

(b) *k-TSP* (*k* = 9)



IF SPTAN1 ≥ CD33* **THEN** ALL; **ELSE** AML Δ = 0.9787
IF HA-1 ≥ ZYX* **THEN** ALL; **ELSE** AML Δ = 0.9787
IF TCF3* > APLP2 **THEN** ALL; **ELSE** AML Δ = 0.9574
IF ATP2A3* ≥ CST3* **THEN** ALL; **ELSE** AML Δ = 0.9387
IF DGKD > MGST1 **THEN** ALL; **ELSE** AML Δ = 0.9387
IF CCND3* ≥ NPC2 **THEN** ALL; **ELSE** AML Δ = 0.9387
IF TOP2B* > PLCB2 **THEN** ALL; **ELSE** AML Δ = 0.9387
IF Macmarcks ≥ CTSD* **THEN** ALL; **ELSE** AML Δ = 0.9362
IF PSMB8 ≥ DF* **THEN** ALL; **ELSE** AMLΔ = 0.9200

* Genes previously identified by Golub *et al* (1999)

(Tan *et al*., 2005, Bioinformatics, 21:3896-3904)

Results

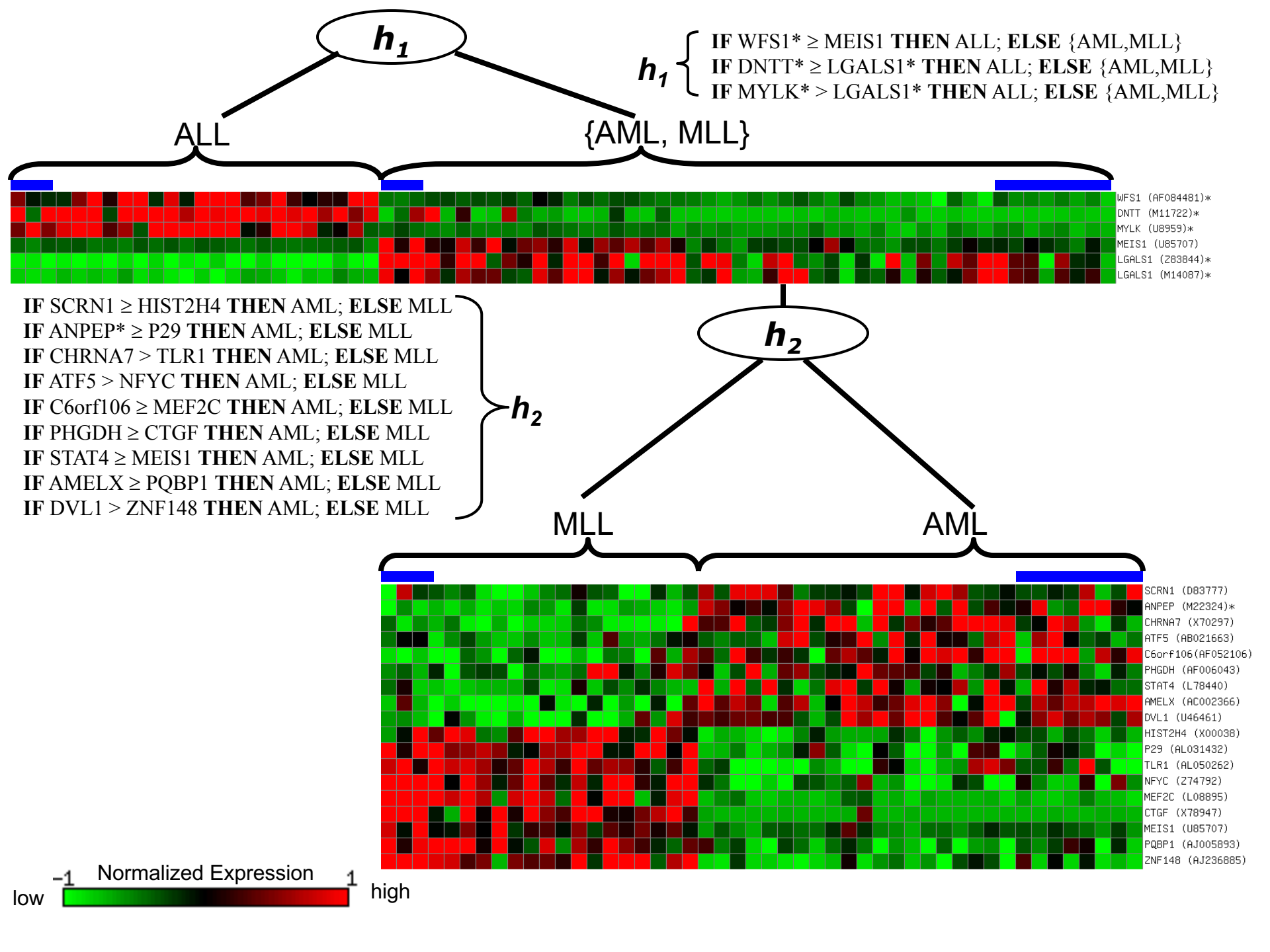
(LOOCV Binary Class Problems)

Method	Leukemia	CNS	DLBCL	Colon	Prostate1	Prostate2	Prostate3	Lung	GCM	Average
TSP	93.80	77.90	**98.10**	**91.10**	**95.10**	67.60	97.00	98.30	75.40	88.26
k-TSP	95.83	**97.10**	97.40	90.30	91.18	75.00	97.00	98.90	85.40	**92.01**
DT	73.61	67.65	80.52	80.65	87.25	64.77	84.85	96.13	77.86	79.25
NB	**100.00**	82.35	80.52	58.06	62.75	73.86	90.91	97.79	84.29	81.17
k-NN	84.72	76.47	84.42	74.19	76.47	69.32	87.88	98.34	82.86	81.63
SVM	98.61	82.35	97.40	82.26	91.18	76.14	**100.00**	**99.45**	**93.21**	91.18
PAM	97.22	82.35	85.71	85.48	91.18	**79.55**	**100.00**	**99.45**	79.29	88.91

Number of Informative Genes

Method	Leukemia	CNS	DLBCL	Colon	Prostate1	Prostate2	Prostate3	Lung	GCM
TSP	2	2	2	2	2	2	2	2	2
k-TSP	18	10	2	2	2	18	2	10	10
DT	2	2	3	3	4	4	1	3	14
PAM	2296	4	17	15	47	13	701	9	47

(Tan *et al.*, 2005, Bioinformatics, 21:3896-3904)


h1
ALL
{AML, MLL}
h1
IF WFS1* ≥ MEIS1 THEN ALL; ELSE {AML,MLL}
IF DNTT* ≥ LGALS1* THEN ALL; ELSE {AML,MLL}
IF MYLK* > LGALS1* THEN ALL; ELSE {AML,MLL}
WFS1 (AF084481)*
DNTT (M11722)*
MYLK (U8959)*
MEIS1 (U85707)
LGALS1 (Z83844)*
LGALS1 (M14087)*
IF SCRN1 ≥ HIST2H4 THEN AML; ELSE MLL
IF ANPEP* ≥ P29 THEN AML; ELSE MLL
IF CHRNA7 > TLR1 THEN AML; ELSE MLL
IF ATF5 > NFYC THEN AML; ELSE MLL
IF C6orf106 ≥ MEF2C THEN AML; ELSE MLL
IF PHGDH ≥ CTGF THEN AML; ELSE MLL
IF STAT4 ≥ MEIS1 THEN AML; ELSE MLL
IF AMELX ≥ PQBP1 THEN AML; ELSE MLL
IF DVL1 > ZNF148 THEN AML; ELSE MLL
h2
h2
MLL
AML
SCRN1 (D83777)
ANPEP (M22324)*
CHRNA7 (X70297)
ATF5 (AB021663)
C6orf106(AF052106)
PHGDH (AF006043)
STAT4 (L78440)
AMELX (AC002366)
DVL1 (U46461)
HIST2H4 (X00038)
P29 (AL031432)
TLR1 (AL050262)
NFYC (Z74792)
MEF2C (L08895)
CTGF (X78947)
MEIS1 (U85707)
PQBP1 (AJ005893)
ZNF148 (AJ236885)
low
−1
Normalized Expression
1
high

Results

(Test Accuracy for Multi-Class Problems)

Method	Leuk1	Lung1	Leuk2	SRBCT	Breast	Lung2	DLBCL	Leuk3	Cancers	GCM	Average
HC-*TSP*	**97.06**	71.88	80.00	95.00	66.67	83.58	83.33	77.68	74.32	52.17	78.17
HC-*k-TSP*	**97.06**	78.13	**100**	**100**	66.67	94.03	83.33	82.14	82.43	**67.39**	85.12
DT	85.29	78.13	80.00	75.00	73.33	88.06	86.67	75.89	68.92	52.17	76.35
NB	85.29	81.25	**100**	60.00	66.67	88.06	86.67	32.14	79.73	52.17	73.20
k-NN	67.65	75.00	86.67	30.00	63.33	88.06	93.33	75.89	64.86	34.78	67.96
1-vs-1-*SVM*	79.41	**87.50**	**100**	**100**	83.33	97.01	**100**	84.82	83.78	65.22	**88.11**
PAM	**97.06**	78.13	93.33	95.00	**93.33**	**100**	90.00	**93.75**	**87.84**	56.52	**88.50**

Number of Informative Genes

Method	Leuk1	Lung1	Leuk2	SRBCT	Breast	Lung2	DLBCL	Leuk3	Cancers	GCM
HC-*TSP*	4	4	4	6	8	8	10	12	20	26
HC-*k-TSP*	36	20	24	30	24	28	46	64	128	134
DT	2	4	2	3	4	5	5	16	10	18
PAM	44	13	62	285	4822	614	3949	3338	2008	1253

(Tan *et al.*, 2005, Bioinformatics, 21:3896-3904)

Feedback from *k-TSP* user

From	Nathan Price <nprice@systemsbiology.org>
Sent	Friday, March 10, 2006 1:29 pm
To	AIK CHOON TAN <actan@jhu.edu>
Subject	ktsp

Aik Choon,

Hi again. :-)

As you know, we are big fans of your KTSP method here at ISB. I have used it now in two collaborations of mine, one with MD Anderson and one with the Hutch. In one study, KTSP outperformed SVMs etc. significantly, and in the other it outperformed SVMs etc. dramatically. For both data sets, the LOOCV is very small. So, I am to the point where this is by far my favorite approach to classification. I think the concept of relevant expression reversals is a brilliantly simple idea for getting around so many of the vagaries associated with data normalization and standardization across populations.

……

All the best,
Nathan

Nathan D. Price, Ph.D.
American Cancer Society Postdoctoral Fellow
Hood Lab
Institute for Systems Biology
1441 N. 34th Street
Seattle, WA 98103
Tel: (206) 732-1452
Fax: (206) 732-1299
http://personal.systemsbiology.net/nprice

Highly accurate two-gene classifier for differentiating gastrointestinal stromal tumors and leiomyosarcomas

Nathan D. Price*, Jonathan Trent†, Adel K. El-Naggar‡, David Cogdell‡, Ellen Taylor‡, Kelly K. Hunt§, Raphael E. Pollock§, Leroy Hood*¶, Ilya Shmulevich*, and Wei Zhang‡||

*Institute for Systems Biology, Seattle, WA 98103; and Departments of †Sarcoma Medical Oncology, ‡Pathology, and §Surgical Oncology, University of Texas M. D. Anderson Cancer Center, Houston, TX 77030

PNAS (2007) 104: 3414-3419.

Contributed by Leroy Hood, December 28, 2006 (sent for review November 29, 2006)

IF *OBSCN* > *C9ORF65 (PRUNE2)* THEN GIST, ELSE LMS

Gastrointestinal stromal tumor (GIST) has emerged as a clinically distinct type of sarcoma with frequent overexpression and mutation of the *c-Kit* oncogene and a favorable response to imatinib mesylate [also known as STI571 (Gleevec)] therapy. However, a significant diagnostic challenge remains in the differentiation of GIST from leiomyosarcomas (LMSs). To improve on the diagnostic evaluation and to complement the immunohistochemical evaluation of these tumors, we performed a whole-genome gene expression study on 68 well characterized tumor samples. Using bioinformatic approaches, we devised a two-gene relative expression classifier that distinguishes between GIST and LMS with an accuracy of 99.3% on the microarray samples and an estimated accuracy of 97.8% on future cases. We validated this classifier by using RT-PCR on 20 samples in the microarray study and on an additional 19 independent samples, with 100% accuracy. Thus, our two-gene relative expression classifier is a highly accurate diagnostic method to distinguish between GIST and LMS and has the potential to be rapidly implemented in a clinical setting. The success of this classifier is likely due to two general traits, namely that the classifier is independent of data normalization and that it uses as simple an approach as possible to achieve this independence to avoid overfitting. We expect that the use of simple marker pairs that exhibit these traits will be of significant clinical use in a variety of contexts.

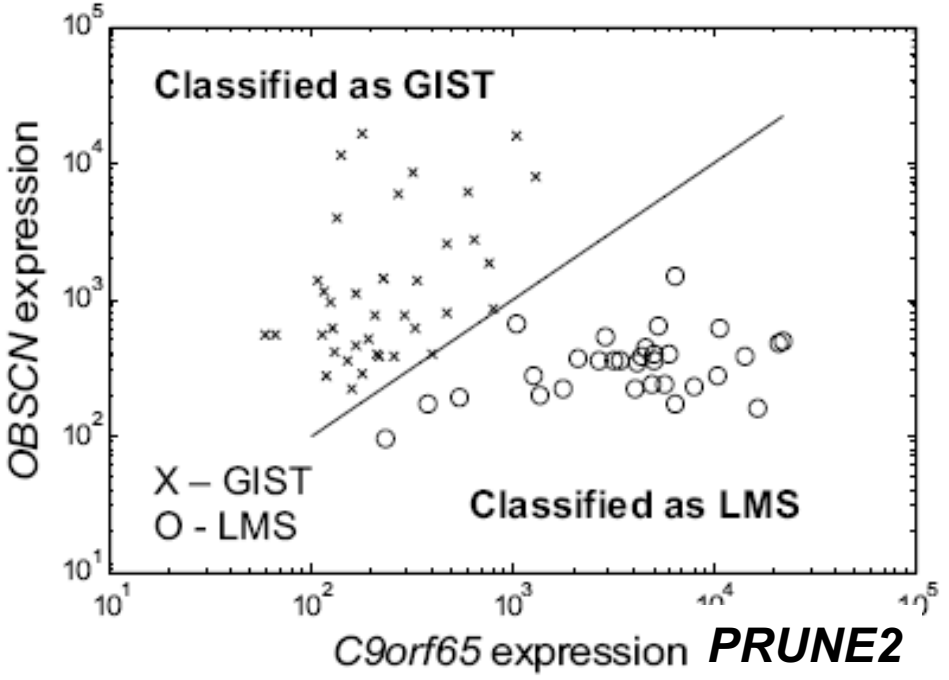
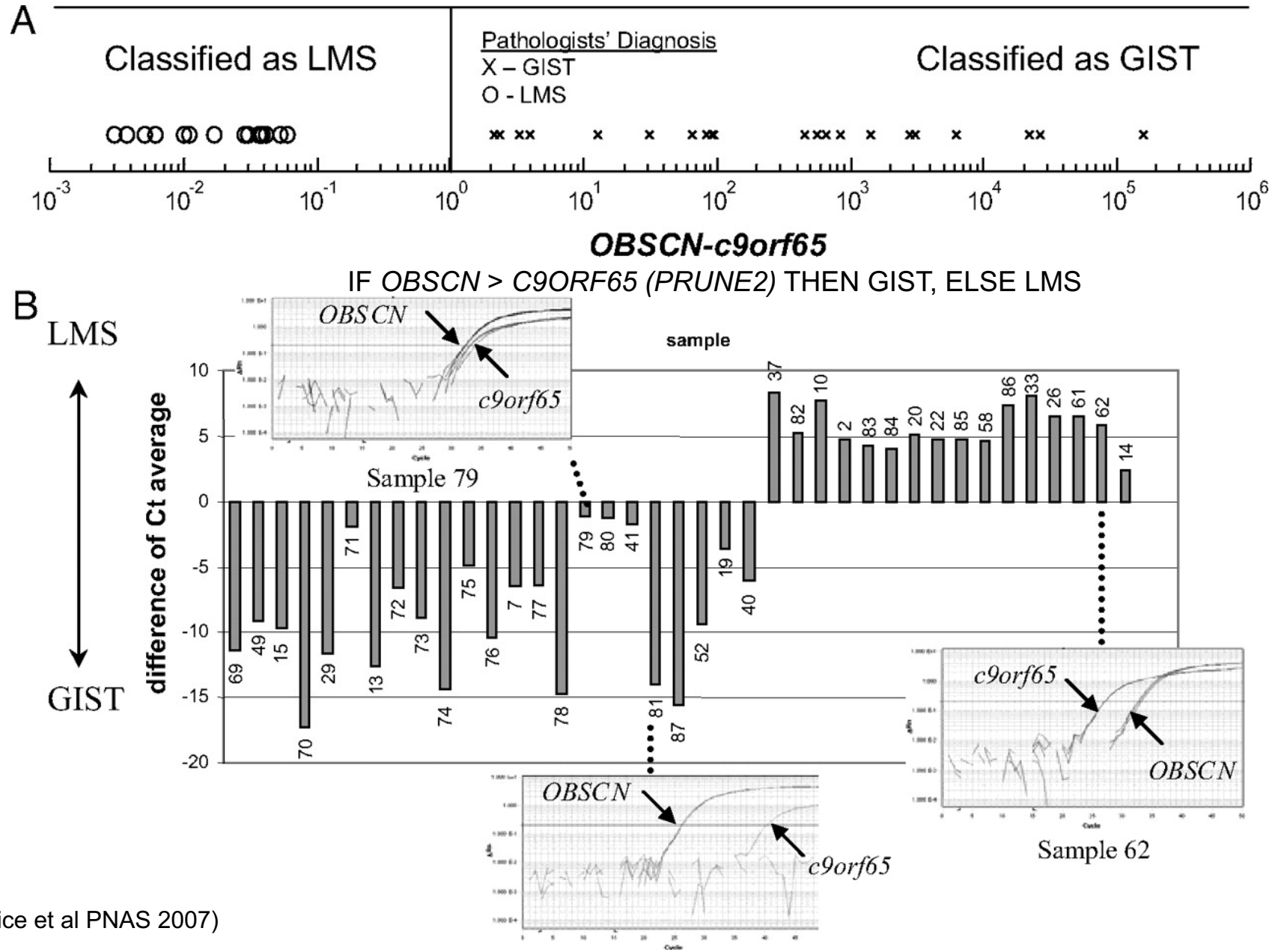


Fig. 2. Expression values of the two genes involved in the TSP classifier on the Agilent microarrays after quantile normalization. (Note: The classification is independent of normalization, because the decision is based only on which gene is higher, but the magnitude of the expression shown does vary somewhat with normalization technique.) The separating line (slope = 1) represents the cutoff for which gene is more highly expressed. It is not a fit to the data.

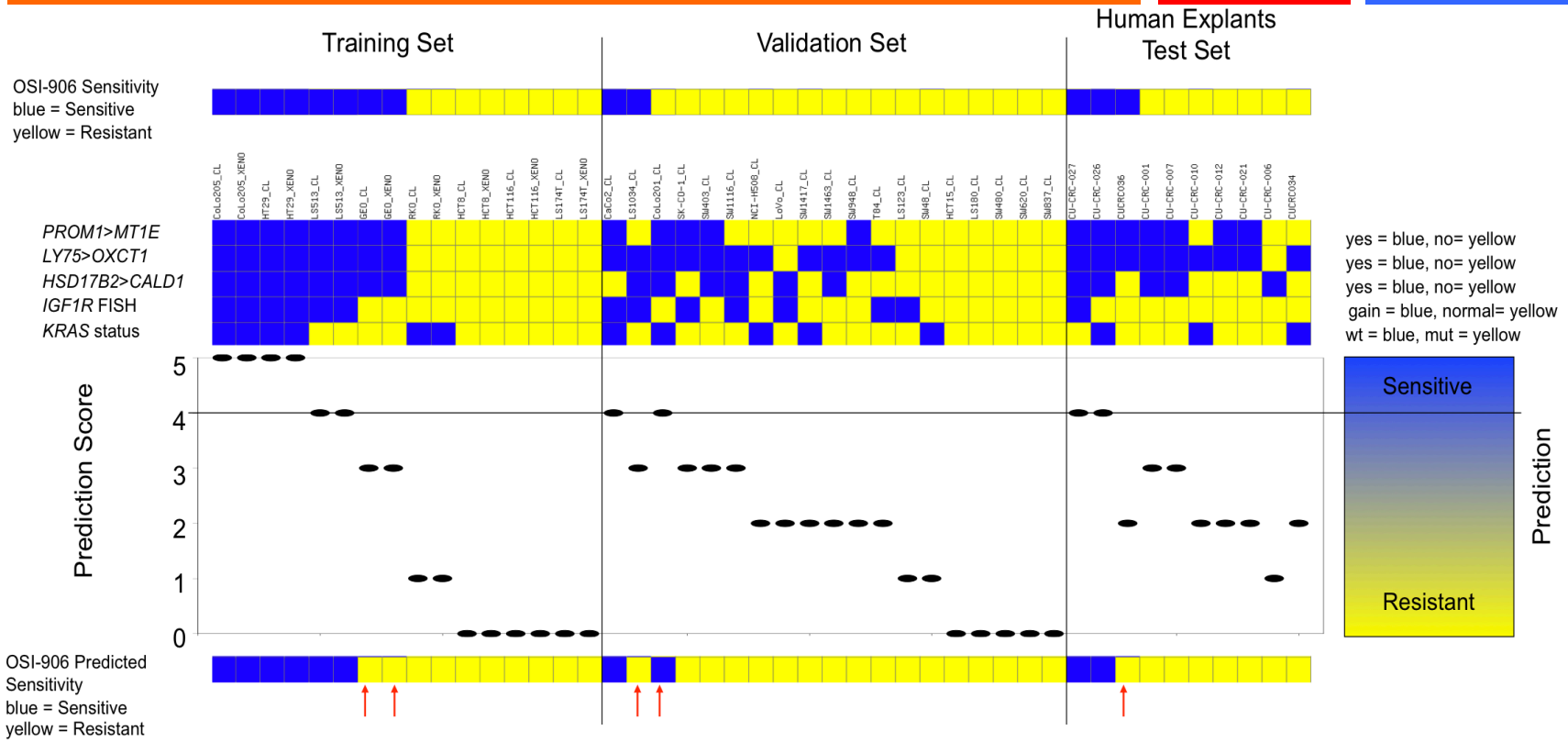
We thank Dr. Aik Choon Tan for helpful assistance with code for the k-TSP algorithm. This work was supported by National Institutes of Health (NIH)/National Cancer Institute (NCI) Grant R01 CA098570-01 (to W.Z.), a grant from the Commonwealth Foundation for Cancer Research (to W.Z. and J.T.), and NIH/National Institute of General Medical Sciences Grant P50 GM076547 (to L.H.). N.D.P.

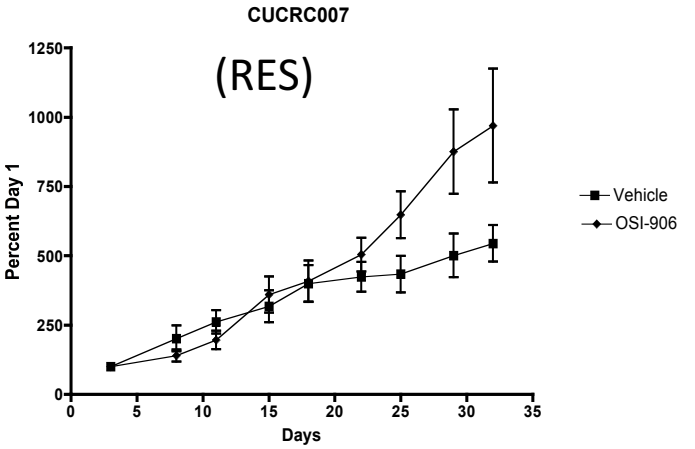
RT-PCR Validation

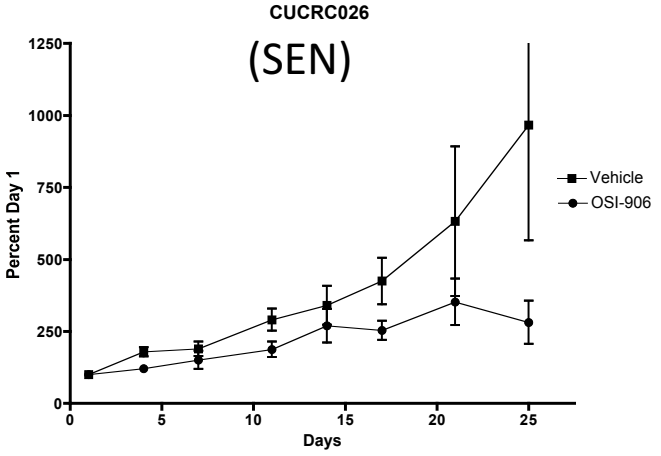


(Price et al PNAS 2007)

Integrative Genomic Classifier for IGF1R/IR TKI (OSI-906) in CRC







Translational Bioinformatics: Clinical Trials

OSI-906 and Irinotecan in Patients With Advanced Cancer

This study is currently recruiting participants.

Verified December 2012 by University of Colorado, Denver

Sponsor:
University of Colorado, Denver

Information provided by (Responsible Party):
University of Colorado, Denver

ClinicalTrials.gov Identifier:
NCT01016860

First received: November 18, 2009
Last updated: December 28, 2012
Last verified: December 2012
History of Changes

Full Text View | Tabular View | No Study Results Posted | Disclaimer | How to Read a Study Record

Purpose

This study plans to learn more about an investigational drug called OSI-906. OSI-906 is being looked at to see if it could be a treatment for advanced cancer. "The FDA is the U.S. government agency that reviews the results of research of drugs and decides if it can be sold in the U.S. OSI-906 has been given to over 185 people with cancer.

Condition	Intervention	Phase
Colorectal Cancer	Drug: Treatment with OSI-906 and/or irinotecan	Phase 1

Study Type: Interventional
Study Design: Endpoint Classification: Safety/Efficacy Study
Intervention Model: Single Group Assignment
Masking: Open Label
Primary Purpose: Treatment

Official Title: A Phase I/IB Study of OSI-906 and Irinotecan in Patients With Advanced Cancer With Expanded Cohorts of Patients With Colorectal Cancer Stratified by the OSI-906 Integrated Classifier

PI: Stephen Leong, M.D.
University of Colorado

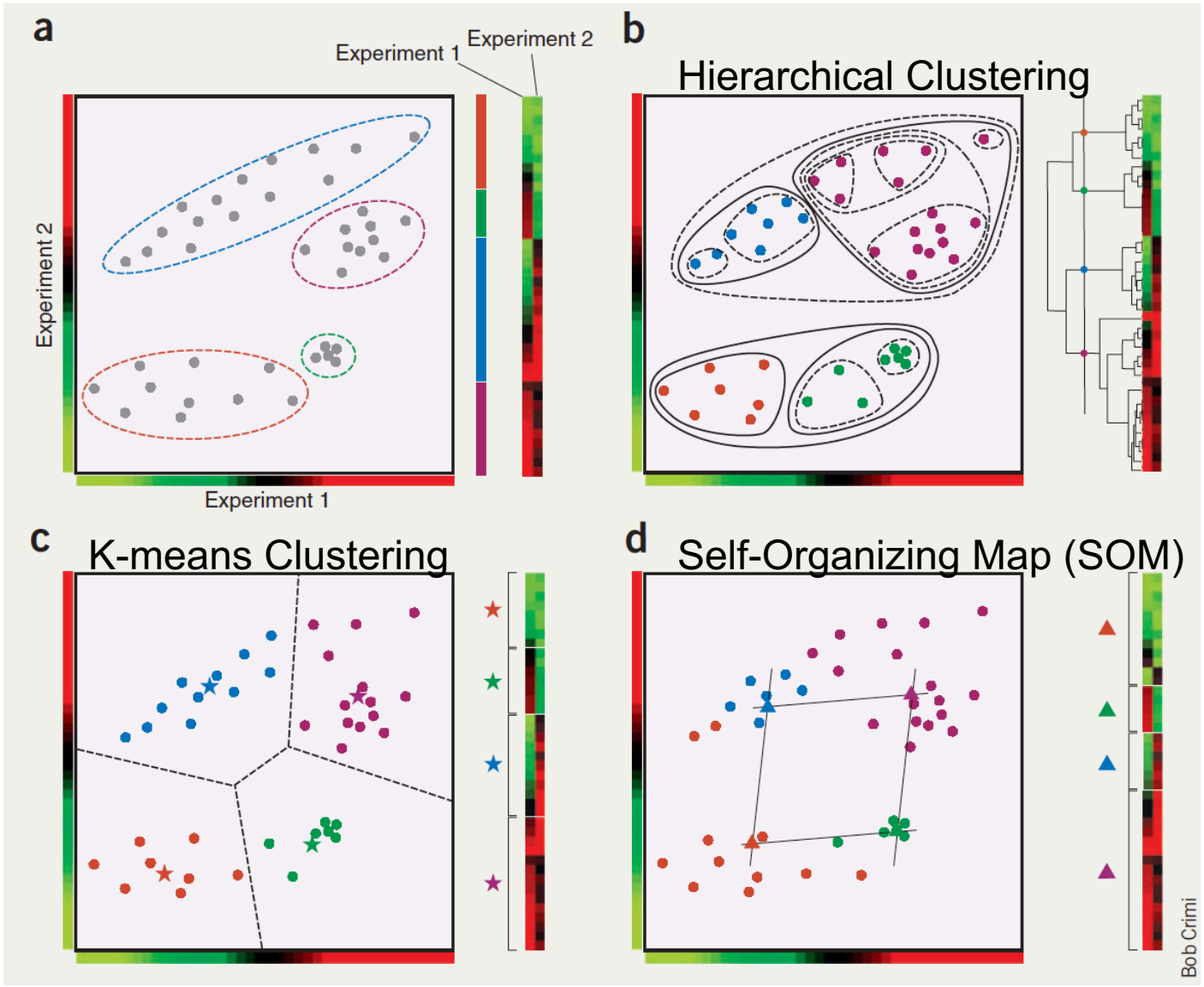
Clustering

- A method of grouping together data / samples that are ***similar*** in some way – based on certain criteria
- ***Unsupervised learning*** – no prior knowledge about the grouping
- Arranging objects into groups according to certain ***properties*** (e.g. expressions, mutations etc)
- Group members share certain properties in common and it is hoped that the resultant classification will provide some insight
- Useful for ***data exploration***
- Could be used to assign new samples into "clusters" –similarities of the new sample to one of the clusters.

Underlying Concepts

- Clustering depends on
  - **Similarity** determines how closely the objects resemble each other. Dissimilarity is the inverse of this, and this is related to the concept of **distance** .
  - **Distance measure** (e.g. Euclidian, correlation, etc)
  - Definition of **distance between clusters** (e.g. single linkage, average linkage etc)
  - **Number of clusters** (user-defined or computationally determined)

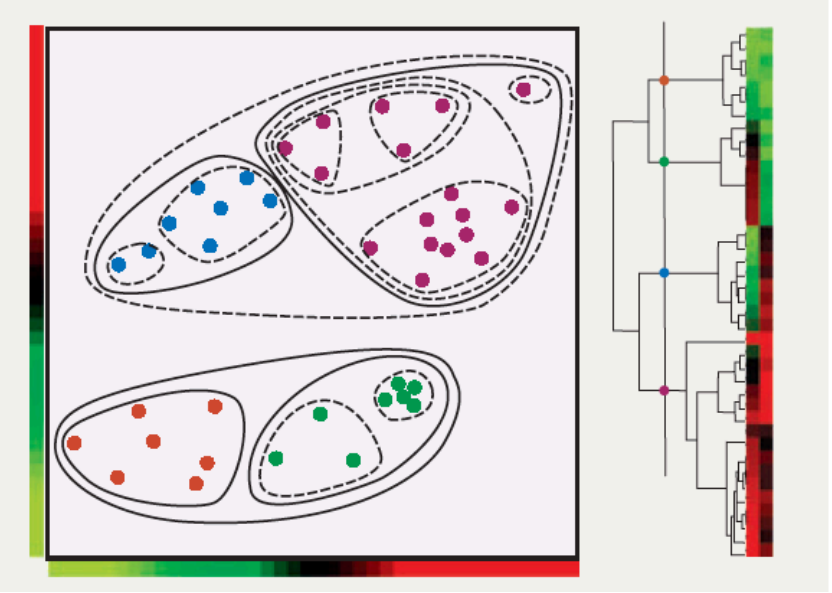
Common Clustering Methods



(Adapted from D'haeseleer 2005)

Hierarchical Clustering

- Step 1: Start every data point in a separate cluster.
- Step 2: Find pairs of data that are similar, merge into one cluster
- Step 3: Repeat Step 2 until one big cluster left



- Hierarchical clustering is a bottom-up or agglomerative method.
- Hierarchical clustering produces a binary tree or dendrogram.
- The final cluster is the root and each data point is a leaf.
- The height of the bars (braches) indicate how close (distance) between clusters

Similarity Measures

Table 1 Gene expression similarity measures

Manhattan distance (city-block distance, L1 norm)	$d_{fg} = \sum_c \lvert e_{fc} - e_{gc} \rvert$
Euclidean distance (L2 norm)	$d_{fg} = \sqrt{\sum_c (e_{fc} - e_{gc})^2}$
Mahalanobis distance	$d_{fg} = (e_f - e_g)' \Sigma^{-1} (e_f - e_g)$, where Σ is the (full or within-cluster) covariance matrix of the data
Pearson correlation (centered correlation)	$d_{fg} = 1 - r_{fg}$, with $r_{fg} = \frac{\sum_c (e_{fc} - \bar{e}_f)(e_{gc} - \bar{e}_g)}{\sqrt{\sum_c (e_{fc} - \bar{e}_f)^2 \sum_c (e_{gc} - \bar{e}_g)^2}}$
Uncentered correlation (angular separation, cosine angle)	$d_{fg} = 1 - r_{fg}$, with $r_{fg} = \frac{\sum_c e_{fc} e_{gc}}{\sqrt{\sum_c e_{fc}^2 \sum_c e_{gc}^2}}$
Spellman rank correlation	As Pearson correlation, but replace e_{gc} with the rank of e_{gc} within the expression values of gene g across all conditions $c = 1 \ldots C$
Absolute or squared correlation	$d_{fg} = 1 - \lvert r_{fg} \rvert$ or $d_{fg} = 1 - r_{fg}^2$

d_{fg}, distance between expression patterns for genes f and g. e_{gc}, expression level of gene g under condition c.

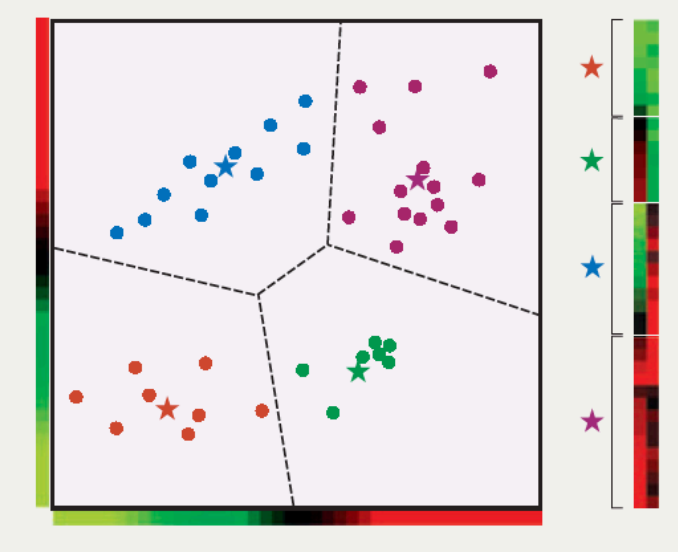
(Adapted from D'haeseleer 2005)

Linkage Methods

Method	Description
Single Linkage	• Minimum of all pairwise distances between points in the two clusters. • Tends to produce long, “loose” clusters.
Complete Linkage	• Maximum of all pairwise distances between points in the two clusters. • Tends to produce very tight clusters.
Average Linkage	• Average of all pairwise distances between point in the two clusters
Centroid Linkage	• Each cluster is associated with a mean vector which is the mean of all the data points in the cluster. • Distances between two mean vectors.

K-means Clustering

- An iterative method that creates K clusters.



Step 1: define number of clusters k

Step 2: initialize cluster centers

  - Pick k data points and set cluster centers to these points
  - Or randomly assign points to clusters and take means of clusters

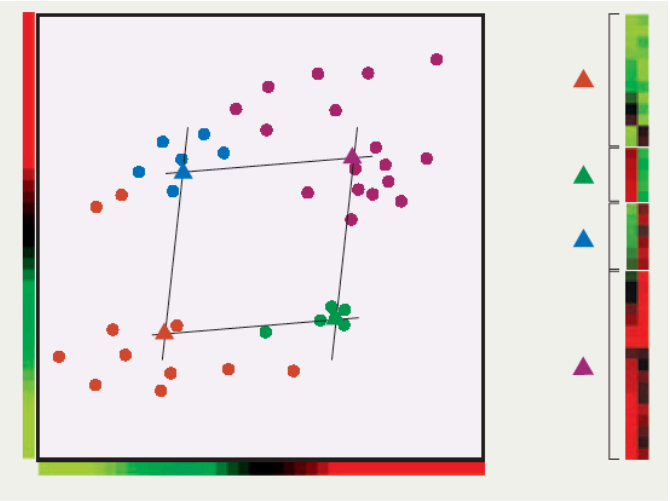
Step 3: For each data point, compute the cluster center closest to it and assign the data point to this cluster

Step 4: Re-compute cluster centers

Stop when there are no new re-assignments.

Self-Organizing Maps

- It requires pre-define number of clusters centroids and pre-specify a topology – a 2D grid that gives the geometric relationships between the clusters.
- For each data point, SOM algorithm moves the cluster centroids to its closest data point, but maintaining the topology specified by the 2D grid.
- At the end of the process, nearby data points tend to map to nearby cluster centroids.



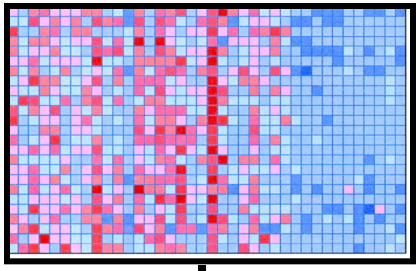
Comparisons of the Clustering Methods

Hierarchical Clustering	K-means Clustering	Self-Organizing Map (SOM)
• Easy to implement • Provide intuitive results (dendrogram) • Hard to decide the stopping criteria	• Easy to implement • Need to pre-specify number of k clusters • Unstable – due to random assignment in different runs	• Complicated and lots of parameters for "tweaking" • Defining the topology in high-dimensional is not obvious • Need to pre-specify number of k clusters

Classifying Microarray Gene Expression data

- Different from other problems because the characteristics of microarray data:
- *Large p small n problem*
  - Number of candidate features (p) greatly exceeds the number of samples (n) (p>>n)
  - Typical data: p > 10,000, n < 100
  - Hypothesis space H is very large (any combination of p has high possibility to be a good classifier)
  - Easily overfit the training examples (n)

Gene (Feature Subset) Selection



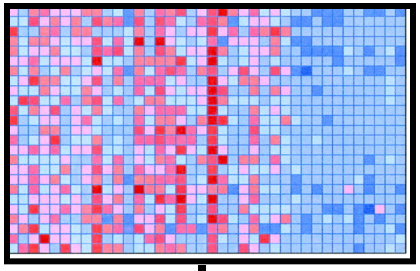
All Genes
(Features)

Feature
Subset
Selection

(Search
Algorithms)

Optimal genes

Learning
Algorithm



Feature
Subset
Selection

Feature Subset
Generation
(Search Algorithms)

Selected
genes

Evaluation

Learning
Algorithm

Optimal
genes

Optimal genes

Learning Algorithm

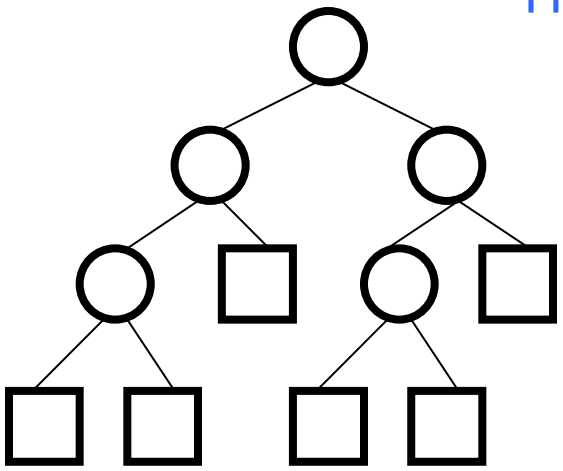
(a) Filter approach

(b) Wrapper approach

Overfitting

Overfitting : A classifier that performs good on the training examples but poor on unseen instances.

Low Training-set error: % errors on training data
High Generalisation error: % errors on unseen data



DT1

Train and test on same data → good classifier with massive overfitting

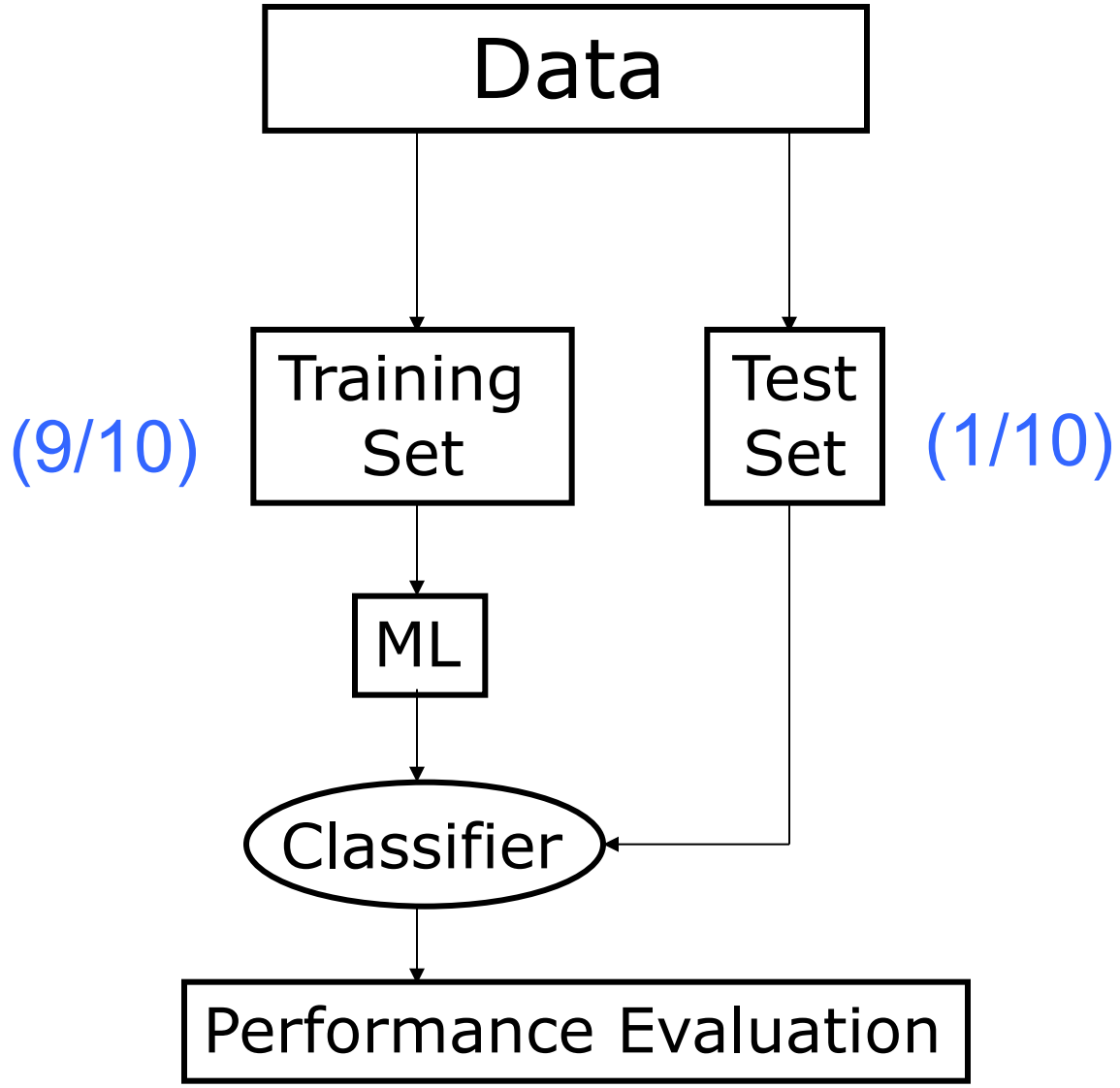
To avoid overfitting:

- Pruning the model
- Cross-validation (Computational expensive)
- Simpler model (Occam's razor)

Comparison between classifiers

- Size (Complex? Simple?)
- Sensitivity, specificity?
- Coverage?
- Compression?
- Receiver Operating Characteristic (ROC) Curve

10-Fold Cross-validation



Confusion matrix / Contingency Table

		Predicted		
		Positive	**Negative**	
Actual	**Positive**	**TP**	**FN**	**Positive Examples**
	Negative	**FP**	**TN**	**Negative Examples**

True Positives(TP): $x \in X+$ and $h(x)$ = TRUE
True Negatives(TN): $x \in X-$ and $h(x)$ = FALSE
False Positives(FP): $x \in X-$ and $h(x)$ = TRUE
False Negatives(FN): $x \in X+$ and $h(x)$ = FALSE

Performance measurements

Accuracy

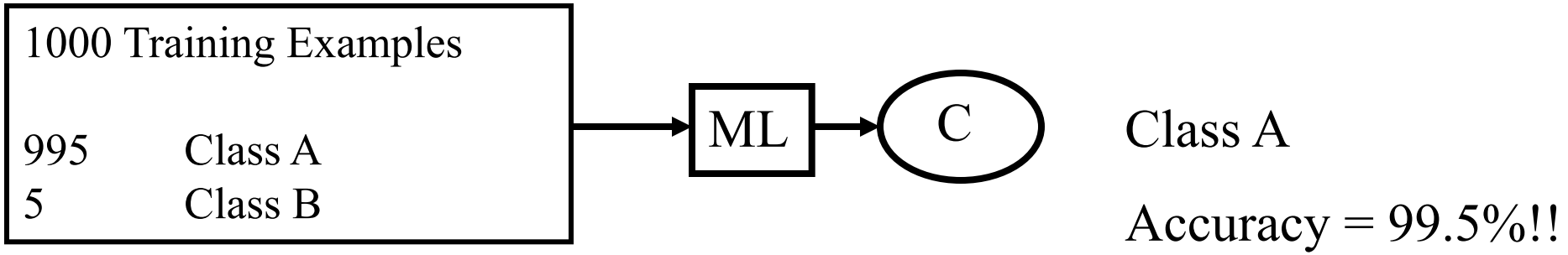
$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

$0 \leq \text{Accuracy} \leq 1$

Accuracy Error, ε = 1 - Accuracy

NOT the good measurement for evaluating classifier's performance!!

IF the classes are unequally represented in the training examples



Prediction Reliability

Reliability of Positive Prediction (Positive Predicted Value / Precision)

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

$0 \leq \text{PPV} \leq 1$

Reliability of Negative Prediction (Negative Predicted Value)

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

$0 \leq \text{NPV} \leq 1$

More measurements …

TP-rate (Sensitivity / Recall)

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

$0 \leq \text{Sn} \leq 1$

TN-rate (Specificity)

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

$0 \leq \text{Sp} \leq 1$

FP-rate

$$FP - rate = \frac{FP}{FP + TN}$$

$0 \leq \text{FP-rate} \leq 1$

FN-rate

$$FN - rate = \frac{FN}{TP + FN}$$

$0 \leq \text{FN-rate} \leq 1$

Other Statistical Measurements

F – measure (van Rijsbergen)

$$F-measure = \frac{2 \times recall \times precision}{recall + precision} = \frac{2TP}{2TP + FP + FN}.$$

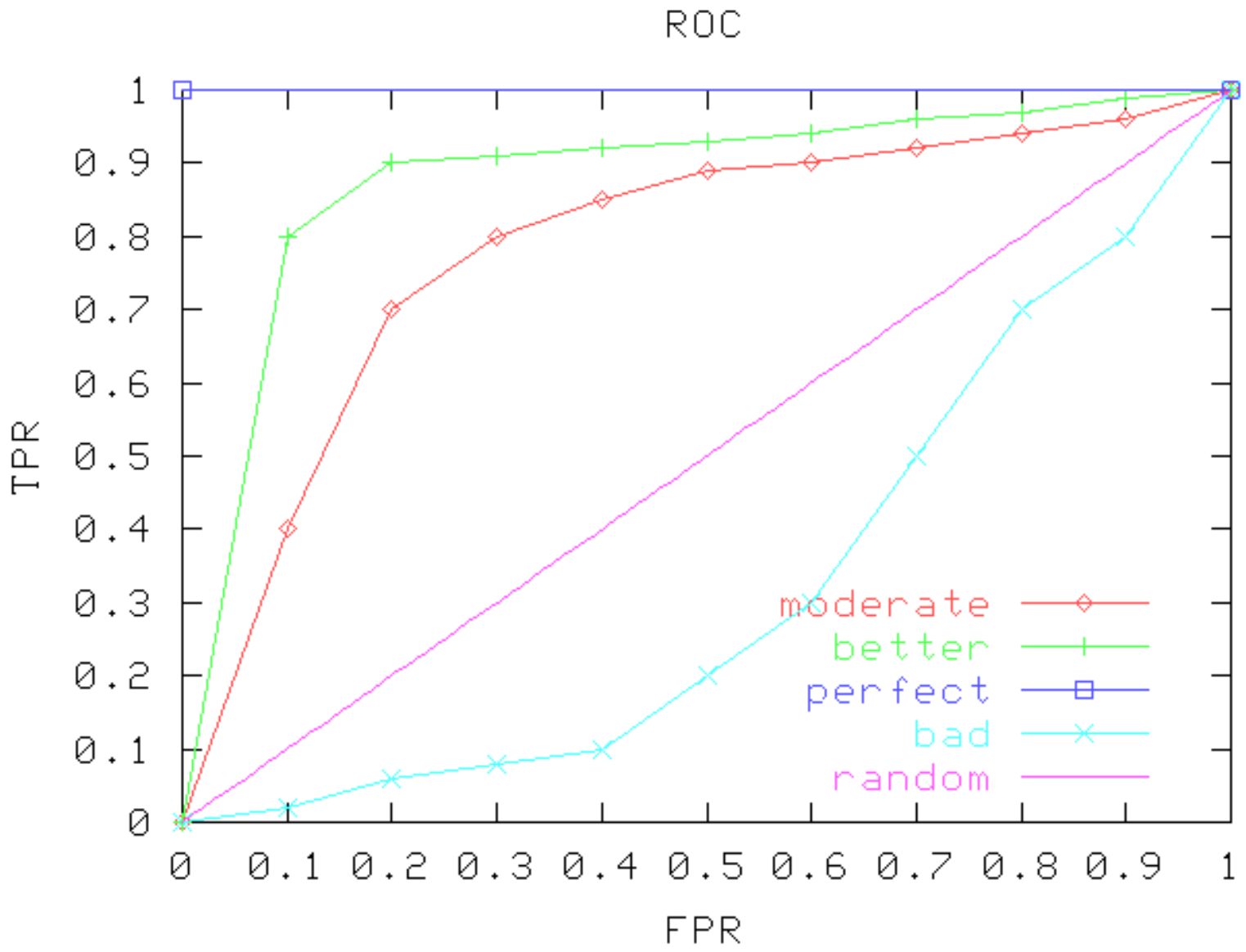
Coefficient Correlation

$$cc = \frac{(TP * TN - FP * FN)}{\sqrt{(TP + FP) * (FP + TN) * (TN + FN) * (FN + TP)}}$$

$-1 \leq cc \leq 1$

cc
- 1.0 no FP or FN
- 0.0 when *f* is random with respect to S+ and S-
- -1.0 only FP and FN

Receiver Operating Curve (ROC)



Area Under Curve (AUC)

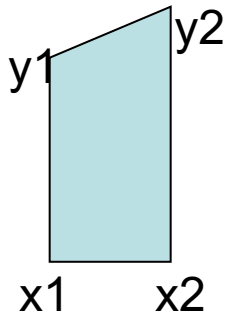
Which classifier performs better?

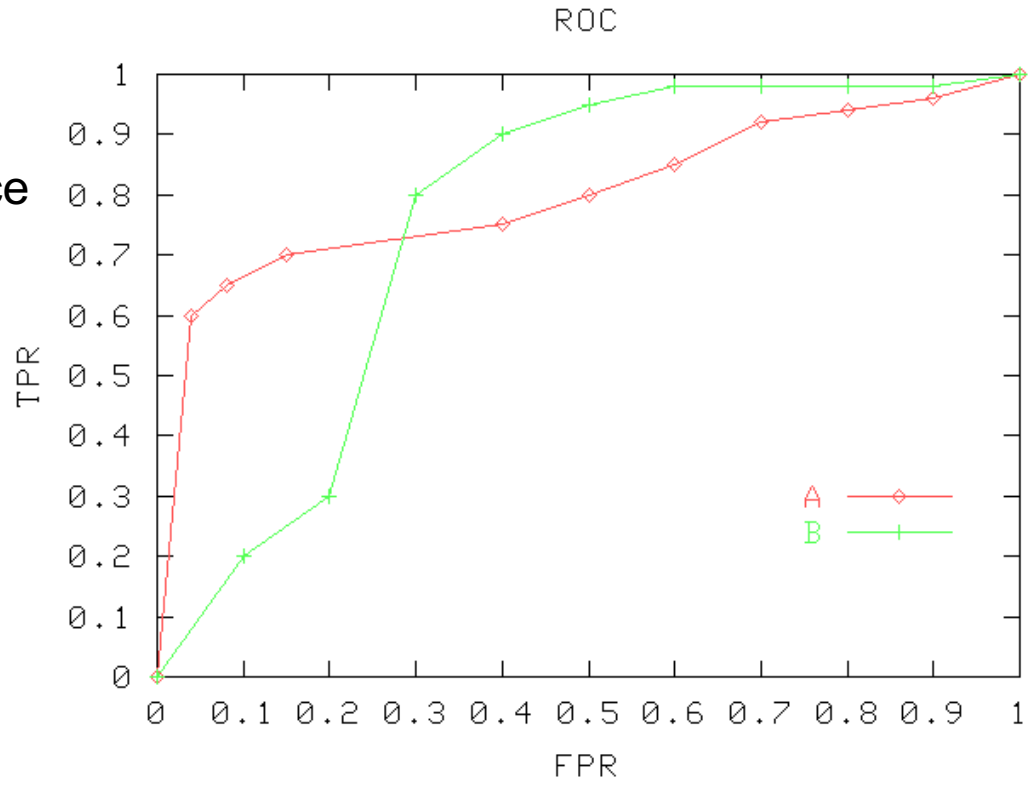
Area Under Curve (AUC) as a Measure of a classifier's performance

Area of trapezoid

The area of a trapezoid is simply the average height times the width of the base.

1. **function** *trap_area*(x1;x2; y1; y2)
2. Base = |x1-x2|
3. $Height_{avg}$ = (y1+y2)/2
4. **return** Base*$Height_{avg}$
5. **end function**





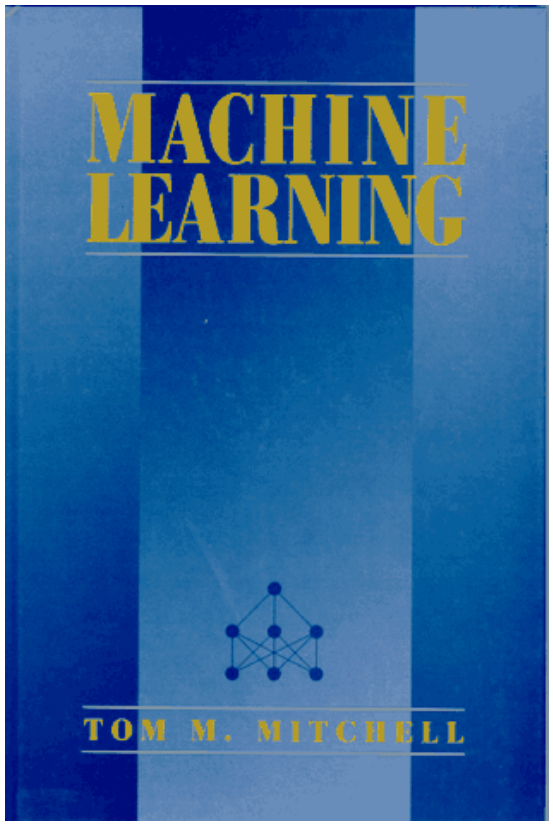
A, AUC = 0.8

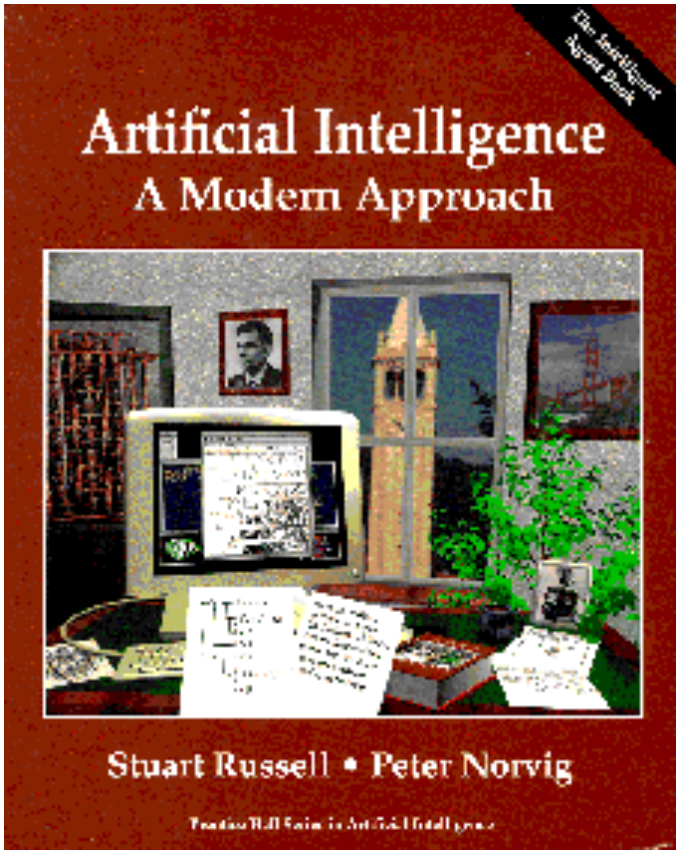
B, AUC = 0.757

Take home message

- Machine learning has been widely applied in bioinformatics, especially in the classification and clustering of high-dimensional data
- Need to understand the “problem” (task) and choose the appropriate machine learning technique
- Do compare with different methods
- The ultimate goal is to interpret the data

References


MACHINE
LEARNING
TOM M. MITCHELL


Artificial Intelligence
A Modern Approach
Stuart Russell • Peter Norvig