Applications of Statistics to Genomic Clinical Group Classification

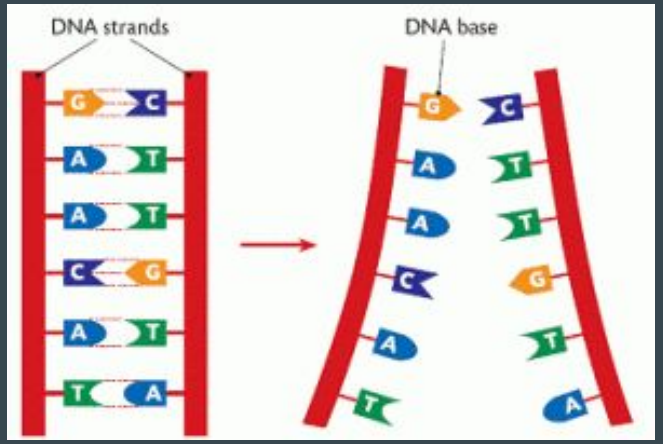
by Jake Sauter

Project Goal

- The goal of this project was to train a machine learning classifier to accurately predict the true class of a genetic sample, given a training set to extrapolate from
  - These genetic samples were from patients with two different types of cancers that must be treated very differently, though are difficult to differentiate from more classical methods
  - A total of 72 samples were available from the data set, all pre-normalized with randomized training and testing sets
    - 47 ALL Cases and 25 AML Cases

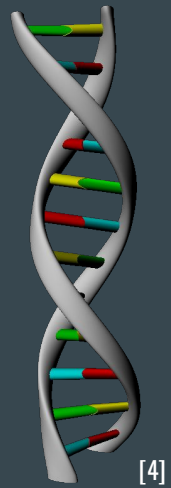
Biological Background

- A key idea that we will need moving forward is that of **complementary DNA**, which is when two single strand DNA (ssDNA) strands have sequences of compatible bases that allow them to **hybridize** (pair) and form a complete DNA strand



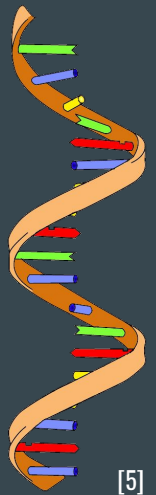
[2]

Central Dogma of Molecular Biology



[4]

Transcription



[5]

Translation

[6]

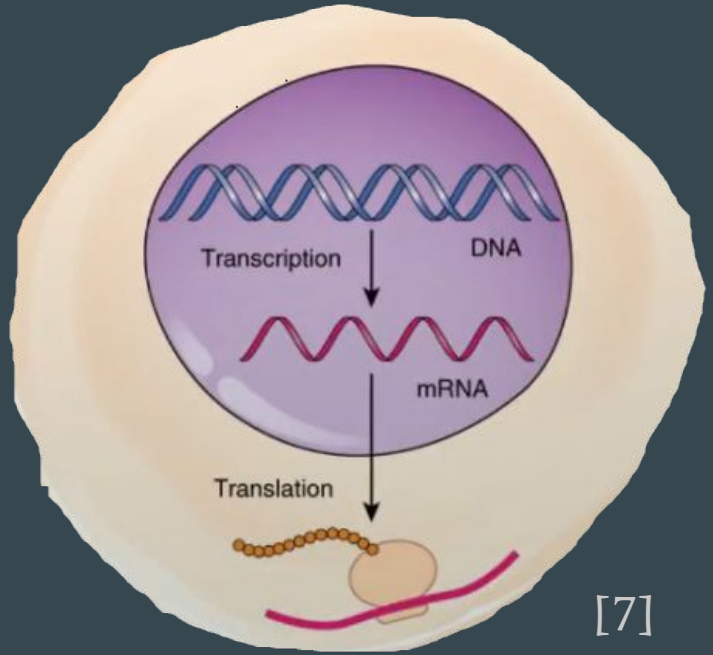
DNA

RNA

Protein

Biological Background

- Thus the **genome** is the entire set of genes in an organism, and gives rise to the name of the study of **genomics**
- However DNA, which is contained in the **nucleus** of a cell, cannot leave this nucleus due to a non-DNA-permeable membrane
- Instead, **mRNA** (messenger RNA) copies desired DNA in the nucleus and permeates through the membrane



[7]

Biological Background

- In a last step before this sequence can be analyzed, mRNA is **reverse transcribed** into ssDNA
- Now this ssDNA can be used in conjunction with **Microarrays** to measure **gene expression level** (the amount of impact a particular gene has on the final protein structure)
- Several factors can lead to different gene expression levels
    - transcription
    - RNA splicing
    - translation
    - post-translational modification of a protein

Biological Background

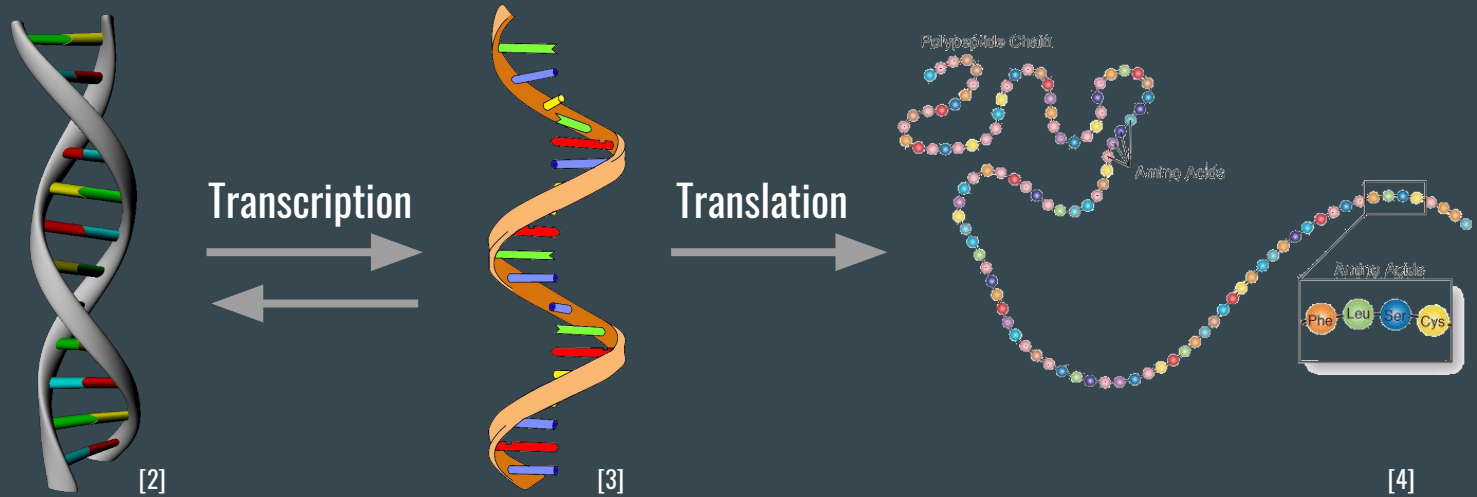
- The total gene expression level from three of the four of these expression level regulators can measured from Microarrays, with the exception of post-translational modification of a protein, as this happens in protein synthesis
- We are interested in these genetic expression levels as they are measurable characteristic of different biological outcomes
    - If the gene expression levels for a gene that regulate cellular replication is modified in a particular way, that person has cancer!

Microarrays

By Jake Sauter

- Background
- Introduction
- Fabrication
  - Deposition Techniques
  - In Situ Synthesis
- Applications
- Challenges
  - Noise
  - Variability
  - . . .

Central Dogma of Molecular Biology

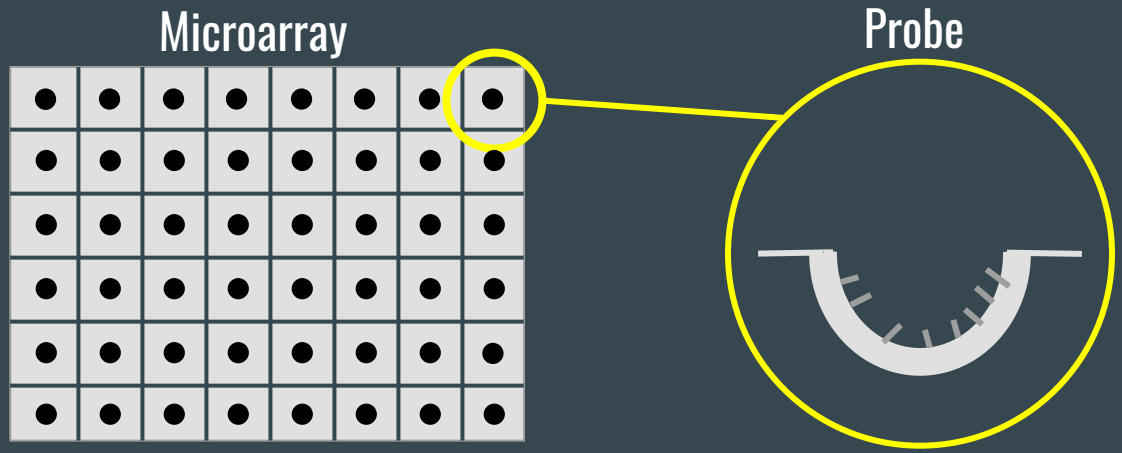


[2] [3] [4]

DNA RNA Protein

Microarray Concept

- DNA Microarray - Usually a substrate (nylon membrane, glass or plastic) on which one deposits single stranded DNAs (ssDNAs) with various sequences.
- Different configurations of deposits (in the probe) can be implemented depending on the purpose of the study



Measuring Gene Expression with Microarrays

- Probes contain DNA material complementary to the the target DNA
- Target DNA material is fluorescently labelled and will hybridize (pair to) the complementary DNA (cDNA) in the wells
- The level of hybridization (fluorescence level) of each probe can easily be measured in a scanner and can indicate the level of expression of a gene corresponding to the cDNA in the well.
- Expression levels from DNA samples of different tissues can be compared using multiple microarrays or on the same microarray (competitive hybridization)

Fabrication Techniques

- Two fabrication types are widely used
  - Deposition
  - In Situ Synthesis
- Each has its pros/cons that we will discuss later
- Deposition tends to result in longer chains of DNA
- In Situ synthesis allows for man made oligonucleotides (strands of nucleotides) but tend to be shorter (~20-50 bp) to prevent errors

Deposition Techniques

- DNA is prepared away from the chip
- Robots dip thin pins into the solution containing DNA material and then touch the pins onto the surface of the array
- Spotted arrays use small sequences to whole genes and even PCR products (clones)
- Gene expression in most eukaryotes (nucleus containing cells) is studied by utilizing complementary DNA (cDNA) clones, which allow for amplification of sufficient of sufficient quantities of DNA for deposition
    - Introns are also removed in this process as prokaryotic DNA does not contain introns

Deposition Techniques - Cloning

- Mature RNA (mRNA) is reverse transcribed into short cDNAs and introduced into bacterial hosts, which are grown isolated, then selected out if they carry foreign DNA
- The bacteria are prokaryotic and do not contain introns in their DNA
  - introns are intragenic material that does not make it through the transcription process to mRNA
- Two methods of cloning are used
  - ESTs - (expressed sequence tags) are cheap single pass sequences of entire clone libraries and result in partial sequences of clone inserts that are long enough to uniquely identify gene fragments
  - PCR - (polymerase chain reactions) are used to amplify clones containing desired fragments, after which they are purified and result in better clones than ESTs

In Situ Synthesis

- In Situ is latin for "on site" or "locally"
- Short synthesized oligonucleotides are attached to the solid support of the microarray
- Probes can be designed to detect multiple variant regions of a transcript (splice variants) and can be short enough to detect specific exons (intragenic material than is transcription to mRNA)
- This measurement of the abundance of splice variants is not possible with spotted DNA arrays due to probes of variable length which more than one different splice variant may hybridize to

In Situ Synthesis - Strengths

- Probe selection is based on sequence information alone. This means that every probe synthesized on the array is know in contrast to cDNA arrays, which deal with ESTs, and in many cases the function of the sequence to a spot is unknown

- This technology can distinguish and quantitatively monitor closely related genes for the simple fact that it can avoid identical sequences among gene family members due to the precisely synthesized oligonucleotides

In Situ Synthesis - Manufacturing Techniques

- Photolithographic (Affymetrix)
  - Uses photolithographic masks for each base. The mask allows for the correct base to be deposited in all locations that require it
  - Allows for high density arrays but the length of DNA sequence constructed is limited due to a non-zero probability of error at each step
  - **The microarrays used in the observed study were produced from Affymetrix and were high-density oligonucleotide microarrays**
- Ink Jet
  - Employs the technology of ink-jet color printers
  - Four cartridges contain the standard nucleotides (A, C, G, T) and the print head moves across the substrate, depositing nucleotides as needed
- Electrosynthesis
  - Electrodes in each substrate are turned on when the nucleotide solution is currently on the chip is needed to bind to that specific well

In Situ Synthesis - Novel Assay Technique

- A company Illumina makes a DASL - c**D**NA mediated **A**nnealing, **S**election, **E**xtension and Ligation assay that is revolutionary
  - Due to the processes involved in the assay, short (~50 bp) sequences can be used, allowing for the use of degraded samples, but this is not novel for in situ synthesis
  - There is estimated to be ~400 million FFPE (usually degraded) samples archived in North America for cancer alone, with associated clinical outcomes.
  - the sensitivity of the platform is enhanced by PCR amplification using common primers and by having 30 replicate beads per probe

Comparison of Techniques

Deposition	In Situ Synthesis
Long sequences	Short sequences due to limitation of synthesis technology
Spot unknown sequences	Spot known sequences
More variability in the system	More reliable data
Easier to analyze with appropriate experimental design	More difficult to analyze

* Though the field is dynamic and shifts can occur rapidly

Applications

- Sequencing
- SNP detection
- Genotyping
- Disease Association
- Genetic Linkage
- Genomic Loss and Amplification
- Detection of chromosomal rearrangement

Challenges

- Noise
  - Introduced at each step in the complex process
- Normalization
  - Not always performed in the same manner
- Experimental Design
  - Not always thoughtfully designed
- Large number of genes
  - Sometimes finding the one influential gene in excess of 7000 is equivalent to finding a needle in a haystack
- Significance
  - Classical techniques (e.g. χ squared test) cannot be applied because the number of variables is much greater than the number of experiments

Challenges

- Biological factors
  - The expression level is the amount of protein produced, not the amount of mRNA (what is detected by the microarray)
  - Other tools can simply not be replaced by microarrays
  - Gene regulation to biological impact is a complicated non-linear mapping
- Array quality assessment
  - Sources of variability can include mRNA preparation, transcription and labelling processes

References

[1] Book

[2] Vong, Andrew. “3D DNA Strand Rendering in MAYA.” av_designs, 27 Mar. 2015, www.andrewvong.com/3d-dna-strand-rendering-in-maya/.

[3]“Ribonucleic Acid (RNA).” RNA Extraction, Quinnipiac University, qu.edu.iq/bt/wp-content/uploads/2015/12/RNA-extraction.pdf.

[4] charity-stanton. “What Organic Molecule Is DNA?” SlideServe, 12 Nov. 2014, www.slideserve.com/charity-stanton/what-organic-molecule-is-dna.

Reliability and Reproducibility in Microarray measurements

By Jake Sauter

- Introduction
- Sensitivity
- Accuracy
- Reproducibility
- Cross - Platform Consistency
- MAQC Project

Microarray Results

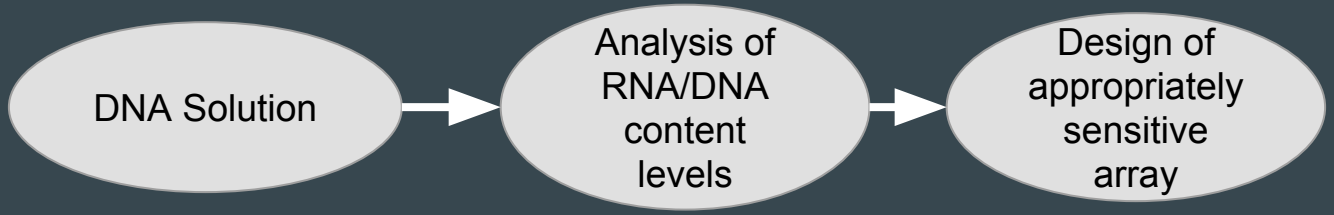
- Results are still marred by several technical issues that are often neglected
- Each manufacturing technique has its own weaknesses
- Different arrays have different results concerning accuracy, sensitivity, specificity and robustness
    - Claims have been made of reduced reproducibility of results from one platform to another
- These issues became much more important when microarrays were proposed as a diagnostic tool in molecular disease classification
    - Regulatory agencies such as the FDA require solid, empirically supported data about the accuracy, sensitivity, specificity, reproducibility and reliability of microarrays for clinical use

Fabrication Issues

- Nucleotide probes synthesized on the chip are not 100% accurate due to base-skipping
- Deposition based techniques can lead to incorrect cDNA intermixed on the probe
- Analysis is based on the assumption that most microarray probes produce specific signals under a single, rather lenient hybridization process
    - Probably not true as testified by widespread cross-hybridization of transcripts on microarrays

Microarray Sensitivity

- Determining the sensitivity of microarray measurements is essential in order to define the concentration range in which accurate measurements can be made



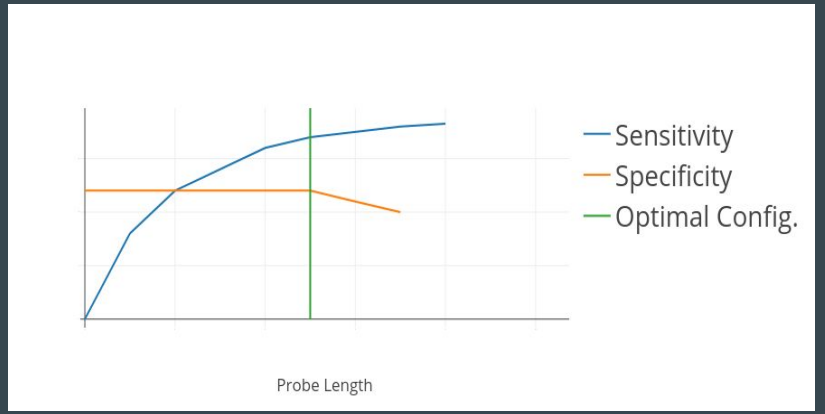
- Results from using microarrays with suggested solution concentrations were verified for many manufacturers, however any lower than the recommended concentrations caused non-meaningful measurements

Sensitivity Issues

- 40-50% of present transcripts from RNA samples are estimated to be below sensitivity threshold levels
- Failure to detect a highly relevant gene EGFR (important for cancer diagnosis) encourages reflection of using microarrays in the diagnosis pipeline

Probe Length and Detection Limits

- Probe length can provide trade-offs in sensitivity and specificity
- Sensitivity increases with probe length, however after ~30-mer probes specificity starts to decline for less of an increase in sensitivity
- Detection limit of current microarray technology appears to be between 1 and 10 copies of mRNA per cell.
    - Might still be insufficient for detection of relevant changes in low abundance genes (e.g. transcription factors that control cell replication)



Microarray Accuracy

- Microarrays can measure
  - Absolute transcript concentration
  - Relative transcript concentration (expression ratios, comparing two samples)
- Expression ratios can be measured with a high accuracy and are usually favored
- Probes in a given Affymetrix probe set (probes to detect the same gene transcript) while producing significantly different intensities may still produce consistent ratio values across the very same probe set when two RNA samples are compared

Validation of Accuracy

- Assessing the accuracy of microarray measurements requires that the true concentrations or concentration ratios are known for a large amount of transcripts
- True concentrations can be obtained by
  - Spike-in or dilution experiments (small amounts of genes per test)
  - Independent means such as quantitative RT-PCR or Northern Blots (costly process so done for a small number of genes)
- At most 42 genes can be assessed per spike-in study, much less than the 10-30k possibly informative genes
- Usually only 3-10 genes are usually verified using independent quantitative techniques per study

Key Points for Accuracy of Microarrays

1. In their appropriate dynamic range, microarray measurements accurately reflect the existence and direction of expression changes in approximately 70-90% of genes.
2. Microarrays measure expression ratios more accurately than absolute expression levels
    a. High correlation in expression ratio measurements has been seen with more robust experimental methods
3. This relatively good correlation is not perfect due to compressed (underestimated) microarray expression ratios

Reproducibility

- A platform can have excellent reproducibility without producing any accurate or cross-platform consistent measurements
- "Good" reproducibility requires that a given probe bind the same number of labelled transcripts in repeated measurements of the same sample
- Badly designed probes that perhaps cross-hybridize with a number of other transcripts besides the target transcript can easily provide highly reproducible yet useless data

Cross-Platform Consistency

- If cross-platform consistency and reproducibility were high, one could use appropriately normalized data regardless of the platform it was collected on
  - This would also reduce the need to replicate experiments and allow researcher to build a universal gene expression database
- Cross-platform consistency has become a top required characteristic of most platforms for reliability
  - Though this is not a sound technique as it is required but not sufficient to validate microarray technology
- Incorrect probe matching can contribute to low cross-platform consistency
  - Usually during analysis there are only 1-2 thousand highly cross-platform consistent genes
  - Up to 50% of contradictory cDNA to oligo arrays can be explained by incorrect clones on cDNA probe

Causes of Inaccuracy and Inconsistencies in Microarrays

- The transcript hybridization signal is composed of 3 signals that can be difficult to isolate
  - Specific signal from target
  - cross-hybridization
  - background (present when absence of sequence similarity)
- The relationships between probe sequences, target concentration and probe intensity is poorly understood
  - Some probes require more energy to bind then others, meaning less target sequences would bind than other target in the solution even if there is the same quantity
- Splice variants are problematic
  - More than 50% of human genes are alternatively spliced
  - Probes must be designed to bind to all variants of a desired target

Microarray Quality Control Project (MAQC)

- FDA spawned project initiated September 2006 with the goals of
  - Provide quality control, tools to the microarray community to avoid procedural failures
  - Develop guidelines for microarray data analysis by providing the public with larger reference data sets along with readily accessible reference RNA samples
  - Establish QC metrics and thresholds for objectively assessing the performance achievable by various microarray platforms
  - Evaluate the advantages and disadvantages of various data analysis methods
- Goals of this project update periodically once the direction wished has been sufficiently explored

References

[1] Drăghici Sorin. Statistics and Data Analysis for Microarrays: Using R and Bioconductor. Chapman and Hall, 2012.

Multiple Comparisons

By Jake Sauter

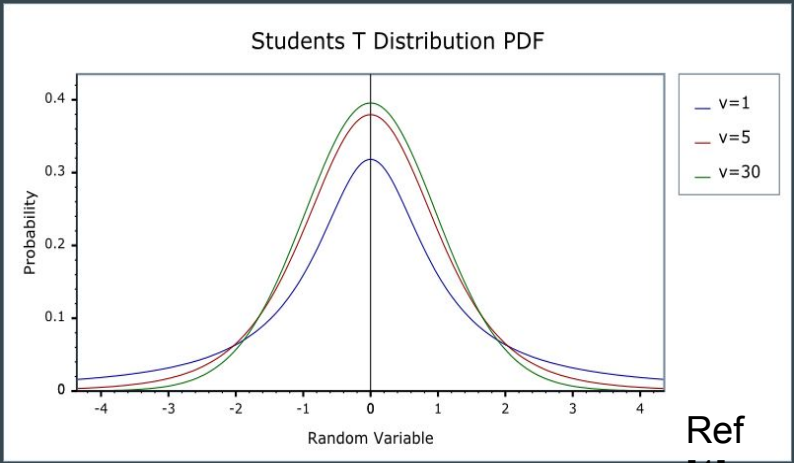
- Hypothesis Testing
- T Test
  - Application to Microarray Data
- Multiple Comparisons
  - Šidák correction
  - Bonferroni correction
  - Step-Wise correction
  - FDR correction
  - Permutation correction
  - SAM

Hypothesis Testing

- Hypothesis Test - A statistical test in which a null hypothesis (H_0) is assessed through the use of a test statistic, comparing this test statistic to a critical value
  - In most cases if the test statistic is larger than the critical value, then H_0 is rejected.
- P Value - The probability of rejecting a null hypothesis H_0 when it is true due to random chance / sampling error
- Significance Level - A significance level α is the acceptable p value such that H_0 is rejected in a statistically significant manner
- Critical Value - A quantity derived from the statistical distribution of the test statistic and the accepted p value.
- Test Statistic - A quantity derived from a sample, compared to a critical value to test for the truth value of the null hypothesis

T Test

- Student's T distribution - A statistical distribution that can be used in comparing means of normally distributed data sets in situations with unknown standard deviation σ and a small sample size. This distribution comes from dividing a Normal Distribution with a Chi-Square Distribution



Ref [1]

T Test

- The T test is used to test if there is a difference in the mean of two groups
- The test statistic for the T test, denoted T here, follows the Student's T distribution and is defined to be the following

$$T = \frac{\text{signal}}{\text{noise}} = \frac{\text{diff between group means}}{\text{variability of groups}} = \frac{|\bar{x_1} - \bar{x_2}|}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

- If the test statistic falls above the critical value on the distribution, it indicates that the means of the two groups are statistically significantly different

Applying the T-test to Microarray Data

- The T-test can be useful in microarray analysis to analyze if gene expression levels are different for a gene from a group of control individuals and group of patients
- This test is not ideal, as applying the t-test with α=.05 significance level to a list of 10,000 genes will produce approximately 500 genes that appear to be differentially expressed even if they are random
- Let's look at some simple probabilities

$$P(\text{correct}) = (1-p) \quad \text{(Eq. 1)}$$

$$P(\text{globally correct}) = (1-p) * (1-p) * \ldots * (1-p) = (1-p)^R \text{for R genes} \quad \text{(Eq. 2)}$$

$$P(\text{wrong somewhere}) = 1 - P(\text{globally correct}) = 1 - (1-p)^R \quad \text{(Eq. 3)}$$

Multiple Comparisons

- Family-Wise Error Rate (FWER) - Probability of having a Type 1 error (false positive) in any comparison in the data set
- We would like to control this overall rate of false positives, framing Eq. 2 in terms of significance levels, we arrive at

$$\alpha_e = 1 - (1 - \alpha_c) \quad \text{(Eq. 4)}$$

  where α_e is the fp rate at experiment level and α_c is the fp rate at single gene level

- Solving eq 4 for α_c results in the Šidák correction for multiple comparisons

$$\alpha_c = 1 - \sqrt[R]{1 - \alpha_e} \quad \text{(Eq. 5)}$$

Corrections for Multiple Comparisons

- Bonferroni notes that for small *p*, Eq. 4 can be approximated by taking only the first two terms of the binomial expansion of $(1-p)^R$, resulting in the Bonferroni correction for multiple comparisons

$$\alpha_c = \frac{\alpha_e}{R} \quad \text{(Eq. 6)}$$

- Unfortunately, both mentioned corrections require a very small significance level for any reasonable R value.
- These corrections are sufficient but not necessary for declaring a gene differentiable between control and patient groups

Step-Wise Correction

- The Holm's step-wise group of methods allow less conservative adjustments of the *p*-values, ordering genes in increasing order of their *p*-value and making successive smaller adjustments
- Procedure:
  - Choose the experiment-level significance α_e
  - Order the genes in the increasing order of individual *p*-values
  - Compare the *p*-values of each gene with a threshold that depend on the position of the gene in the list of ordered values. The thresholds are as follows:

    $\frac{\alpha_e}{R}$ for the first gene, $\frac{\alpha_e}{R-1}$ for the second gene, and so on
  - Let *k* be the largest *i* for which $p_i < \frac{\alpha_e}{R-i+1}$. Reject the null hypothesis for *i* = 1, 2, ... , k

False Discovery Rate (FDR)

- The FDR correction procedure allows for some dependencies between variables, while the previous methods act on the assumption of independence
- FDR correction procedure:
  - Choose the experiment-level significance α_e
  - Order the genes in the increasing order of individual *p*-values
  - Compare the *p*-values of each gene with a threshold that depends on the position of the gene in the list of ordered values. The thresholds are as follows:

    $\frac{1}{R}\alpha_e$ for the first gene, $\frac{2}{R}\alpha_e$ for the second gene, and so on
  - Let *k* be the largest *i* for which $p_i < \frac{i}{R}\alpha_e$. Reject the null hypothesis for *i* = 1, 2, ... , k

Permutation Correction

- The Westfall and Young (W-Y) step-down correction is a more general method that adjusts the *p*-value whole taking into consideration the possible correlations
- This method permutes the classes individuals thousands of times, running a T-test after every permutation
- The *p*-value for a gene *i* will the the proportion of times the value of *t* calculated for the real labels is less than or equal to the value of *t* calculated for a random permutation

$$p\text{-value for gene } i : \frac{\text{number of permutations for which } t_j^{(b)} \geq t_i}{\text{total number of permuatations}}$$

where $t_j^{(b)}$ are the calculated t-values from gene j and permutation b

Significance Analysis of Microarrays (SAM)

- Tusher et al. have reported that permutation testing was still too stringent for their microarray data. In response, they formulated their own method
- SAM assigns a score to each gene taking into consideration the relative change of each gene expression level with respect to the standard deviation of repeated measurements
- SAM uses a test statistic similar to *T*, in that it expresses the difference between means in terms of standard deviations

$$d_i = \frac{\bar{x_i}_1 - \bar{x_i}_2}{s_i + s_0}$$

SAM contd.

- SAM calculates a gene-by-gene variance which will allow for the selection of the appropriate genes independently of their expression levels
- SAM uses the same permutation idea to estimate the percentage of genes identified just by chance
- FDR in SAM:
  - Fix a threshold for differentially expressed genes
  - Count how many genes are reported as differentially expressed in each permutation
  - Calculate the median number of false positives across all permutations
  - Calculate the FDR as the number of false positives divided by the number of genes in the original data

References

[1] Maddock, John, et al. “Students t Distribution.” Boost C++ Libraries, Boost Software, 2008, www.boost.org/doc/libs/1_36_0/libs/math/doc/sf_and_dist/html/math_toolkit/dist/dist_ref/dists/students_t_dist.html.

[2] Drăghici Sorin. Statistics and Data Analysis for Microarrays: Using R and Bioconductor. Chapman and Hall, 2012.

Principal Component Analysis

Jake Sauter

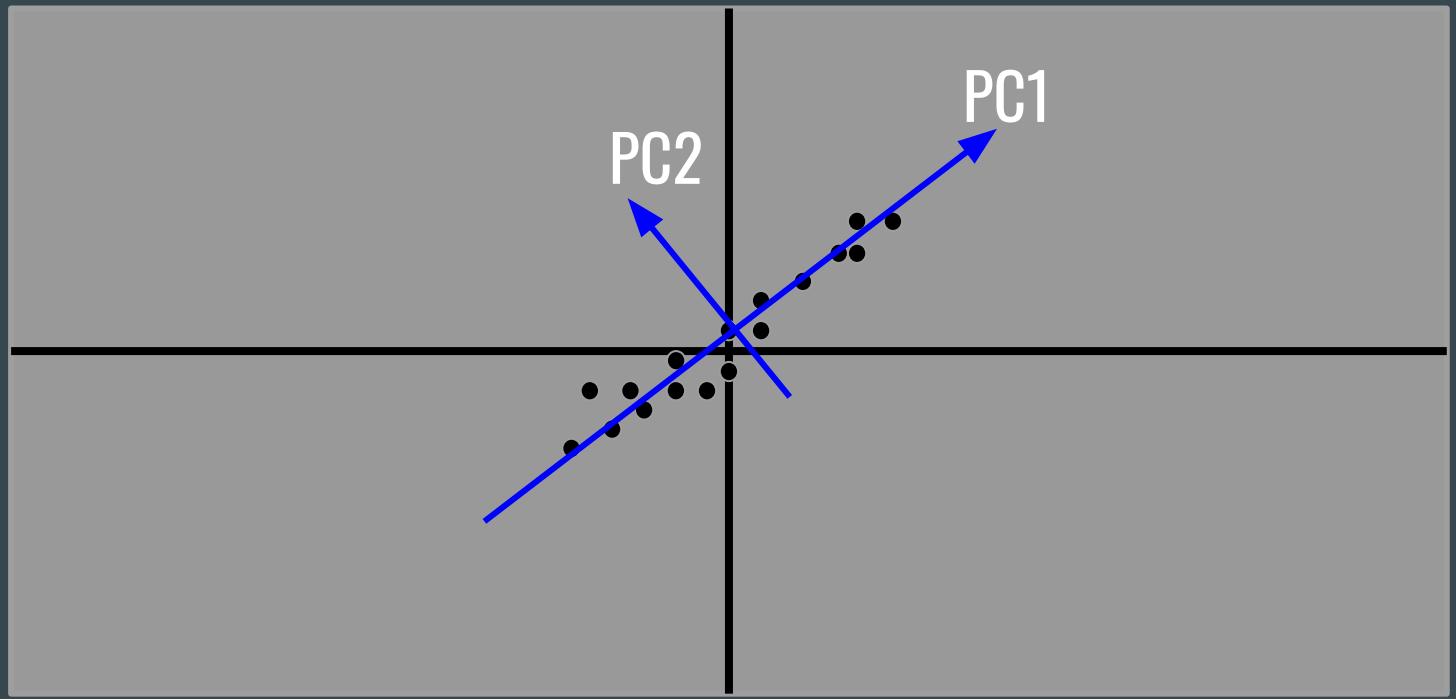
Motivation

- In some data sets, especially in DNA microarrays, there can be thousands of features per sample
- It is difficult to classify samples and isolate which of these features is most important in such a high dimensional feature space
- A form of dimensionality reduction would be very helpful, in which the dimensions of the data set are reduced, while the majority of variance and the relation of samples in the space is preserved
- Principal Component Analysis (PCA) is a form of dimensionality reduction

Principal Components

- The result of PCA is a list of principal components, or the most varying directions of the data set, such that all principal components are orthogonal
- A principal component can be realized as a weight of values for each feature
- Principle components are named in order of significance, being how much variability of the data they make up (PC1 , PC2 , ... , PCN)
- PCs can aid in data visualization, by plotting the data points on the new axes of the principal components

Visualization



Background - Covariance Matrix

- A covariance matrix of a data set is a matrix that expresses how each feature varies with every other feature
    - As such it captures the shape of the data set

$$\begin{bmatrix} cov(x_1, x_1) & cov(x_1, x_2) & \dots & cov(x_1, x_n) \\ cov(x_1, x_2) & cov(x_2, x_2) & \dots & cov(x_2, x_n) \\ . & & & . \\ . & . & & . \\ . & & . & . \\ cov(x_1, x_n) & \dots & & cov(x_n, x_n) \end{bmatrix}$$

PCA - How it Works

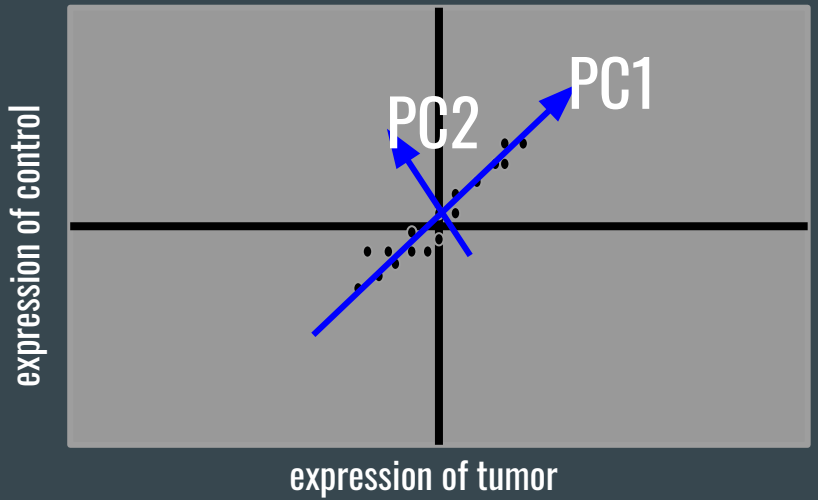
- PCA is a procedure that uses an orthogonal transformation to reduce dimensions, and this is done by finding the eigenvectors of the covariance matrix of the data
- If A is the covariance matrix of the data, then the eigenvector z_1 and eigenvalue λ_1 would form the transformation

$$Az_1 = \lambda_1 z_1$$

- The eigenvalue with the largest absolute value will indicate that the data have the largest variance along its vector

Caution

- In Microarray analysis and other fields, variance along one axis may be expected
- In DNA Microarrays, this expected variance comes along in the form of expression level, it is known that different genes will express at different levels, but we are interested in the **ratio** of the expression levels in comparative analysis



PCA in R

- Two functions are available for PCA in R, being **prcomp()** and **princomp()**
  - Difference in how PCA is calculated
- **Princomp()** follows the method that we have described
- **Prcomp()** uses a process called singular value decomposition (svp), with very little difference in output, and is a preferred computational method

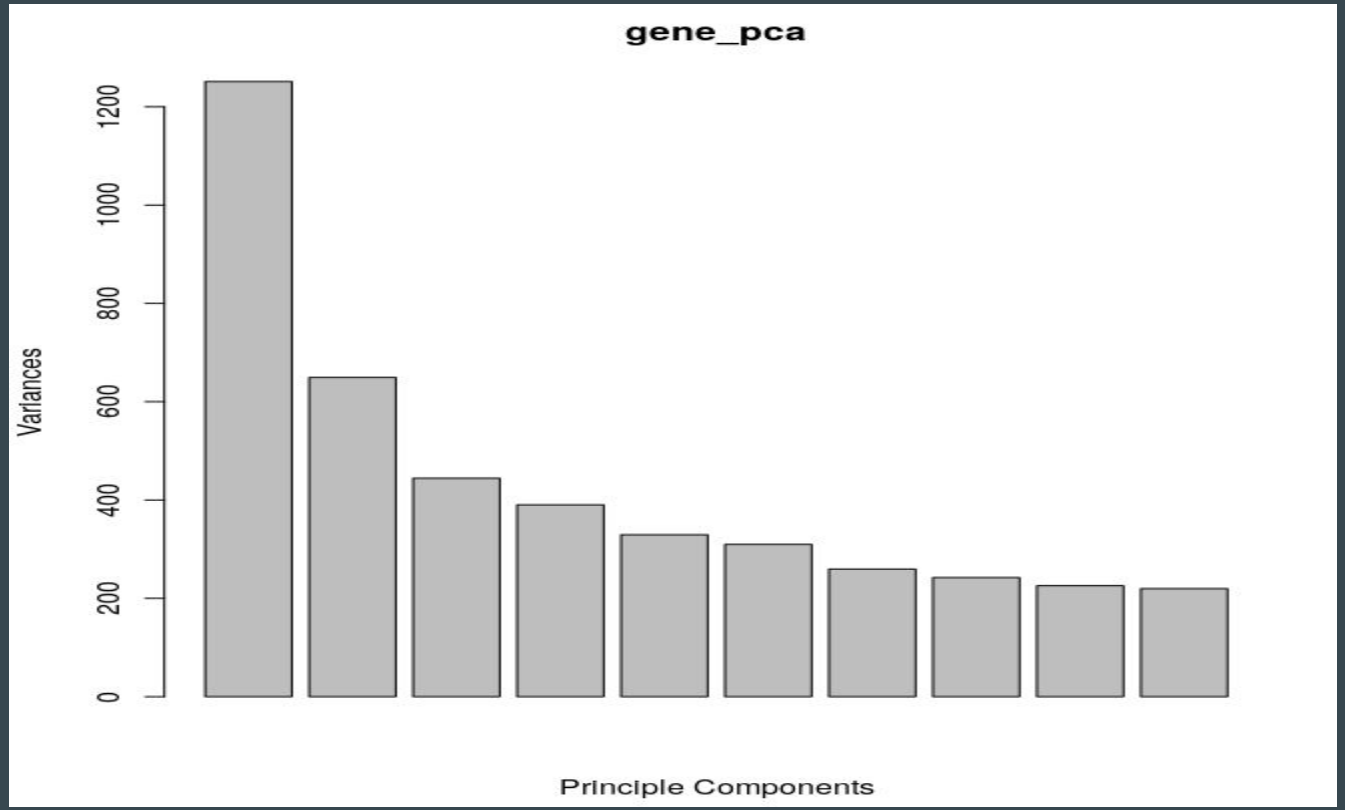
PCA in R

- **Prcomp()** options:
  - center: 0-center the data
  - scale: scale the data to have unit variance
- Usage:

```
gene_pca = prcomp(cleaned_data, scale=T, center=T)
```

where cleaned_data is a matrix of samples x features

PCA Results


gene_pca
Variances
0
200
400
600
800
1000
1200
Principle Components

PCA Results

```
> summary(gene_pca)
Importance of components:
                           PC1     PC2      PC3      PC4      PC5      PC6
Standard deviation     24.4378 21.4323 16.12834 13.71839 13.45205 12.06632
Proportion of Variance  0.1525  0.1173  0.06643  0.04806  0.04621  0.03718
Cumulative Proportion   0.1525  0.2698  0.33623  0.38429  0.43050  0.46768
                           PC7      PC8      PC9     PC10     PC11     PC12
Standard deviation     11.64198 11.48964 10.86800 10.19896 9.82568 9.6743
Proportion of Variance  0.03461  0.03371  0.03016  0.02656 0.02465 0.0239
Cumulative Proportion   0.50229  0.53600  0.56616  0.59272 0.61738 0.6413
                          PC13    PC14    PC15    PC16    PC17    PC18    PC19
Standard deviation     9.36009 8.90415 8.86479 8.64271 8.45269 8.33057 8.16112
Proportion of Variance 0.02237 0.02025 0.02007 0.01907 0.01825 0.01772 0.01701
Cumulative Proportion  0.66365 0.68390 0.70396 0.72304 0.74128 0.75900 0.77601
                          PC20    PC21    PC22    PC23    PC24    PC25    PC26
Standard deviation     7.91017 7.72997 7.70600 7.56953 7.49546 7.47426 7.31963
Proportion of Variance 0.01598 0.01526 0.01516 0.01463 0.01435 0.01427 0.01368
Cumulative Proportion  0.79199 0.80725 0.82241 0.83705 0.85139 0.86566 0.87934
                          PC27    PC28    PC29    PC30   PC31    PC32    PC33
Standard deviation     7.22077 7.12840 7.06897 7.03548 6.6804 6.61997 6.37643
Proportion of Variance 0.01331 0.01298 0.01276 0.01264 0.0114 0.01119 0.01038
Cumulative Proportion  0.89265 0.90563 0.91839 0.93103 0.9424 0.95362 0.96400
                          PC34    PC35    PC36   PC37      PC38
Standard deviation     6.24080 5.96485 5.86086 5.6656 1.085e-14
Proportion of Variance 0.00995 0.00909 0.00877 0.0082 0.000e+00
Cumulative Proportion  0.97395 0.98303 0.99180 1.0000 1.000e+00
```

Plotly Library

- **Plotly** is a free online tool for plotting
- They have developed an R library to allow easy interfacing
- Setup is quick and easy! All one has to do is make an account and set username and password environmental variables in R
- Then a plot can be created through the **plot_ly()** command and posted to an account through the **api_create()** command

```
p <- plot_ly(df, x = ~PC2, y = ~PC3, z = ~PC4, color = ~group, colors = c('#BF382A', '#0C4B8E')) %>% add_markers()

api_create(p, type="scatter3d", filename="scatter3d-my_8")
```

Plotly Output

- The plotting of training data on PC1, PC2 and PC3
- The plotting of training data on PC2, PC3, and PC4
- The plotting of testing data on PC1, PC2 and PC3 generated from training data
- The plotting of testing data on PC2, PC3 and PC4 generated from training data

References

[1] Drăghici Sorin. Statistics and Data Analysis for Microarrays: Using R and Bioconductor. Chapman and Hall, 2012.

Cluster Analysis

Jake Sauter

Background

- Cluster analysis is the most frequently used multivariate technique for analyzing gene sequence expression data
- Clustering is appropriate where there is no a priori knowledge about the data (unsupervised technique)
- In this situation, the only possible approach is to study the similarity between different samples or experiments
- Clustering has become so popular in this field that most authors presenting results obtained with microarrays feel the need to include some type of clustering diagram in their papers

Background

- Clustering is the process of grouping together similar entities

  input: n-dimensional vector

  argument: measure of similarity / distance / metric

  output: many different types, but mostly groups of similar inputs

- The input space of the problem is a n-dimensional space, where n can be the number of samples or the number of feautures

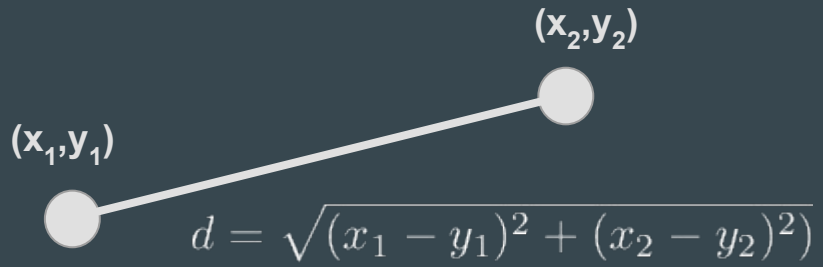
Distances

- A distance metric *d* is a function that takes as arguments two points x and y in an n-dimensional space R^n and has the following properties:
  - Symmetry: The distance should be symmetric such that d(x,y) = d(y,x)
  - Positivity: The distance between any two points should be a real number greater than or equal to 0
  - Triangle Inequality: The distance between two points x and y should be shorter than or equal to the sum of the distances from x to a third point z and y to z (the distance should be the shortest path between two points).

Distances

- There are many different valid distance measures, we will go over a few of them here
- Euclidean Distance: What is thought of normally as "distance", a very intuitive distance metric

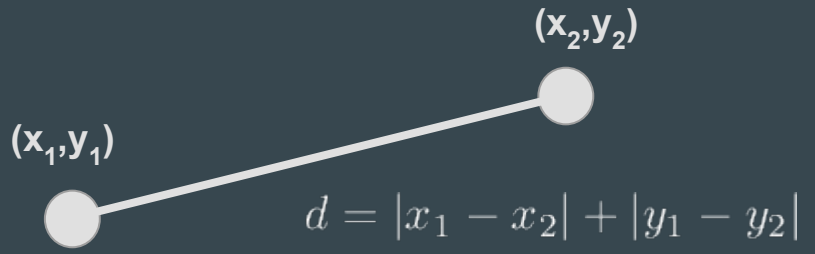
$$d_E(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2}$$
$$= \sum_{i=1}^{n} (x_i - y_i)^2$$



Distances

- Manhattan Distance: Can easily be thought of as how many city blocks must be walked to get from point a to point b on a blocked city design. Only movements along axis directions are allowed
  - This distance measure slightly emphasized outliers as a change of one unit in one coordinate direction leads to a 14% change with respect to Euclidean distance

$$d_m(x,y) = |x_1 - y_1| + |x_2 - y_2| + \cdots + |x_n - y_n|$$
$$= \sum_{i=1}^{n} |x_i - y_i|$$



Distances

- Pearson Correlation distance: Will be proportional to the covariance of two coordinates.
  - This is effective when the points are experiments and dimensions are genes, allowing experiments with very highly correlated genes to be close together
  - This can be used for testing the reliability of equipment or experimental conditions

$$d_R(x, y) = 1 - r_{xy} \quad \text{where}$$

$$r_{xy} = \frac{s_{xy}}{\sqrt{s_x}\sqrt{x_y}} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}}$$

Distances

- The Pearson distance can be a very bad measure if a gene is incorrectly measured!
  - The jackknife correlation aims to solve this issue with leaving out one dimension each iteration
  - The selected value is then the minimum correlation value

$$d_j(x,y) = min\{d_R^1(x,y), d_R^2(x,y) \cdots d_R^n(x,y)\}$$

where d^k_R is d_R with the k-th element removed

- However, this measure is only robust to 1 outlier, so it is still not a great measure

Clustering

- The results of clustering algorithms differs, though it is usually a form of a set of clusters
- Clustering is not necessarily deterministic, the same clustering algorithm applied to the same data may produce different results
    - Some clustering algorithms start with a random choice of clusters
- Membership of a pattern to a cluster should be taken with a grain of salt and further analysed
- The fact that two patterns belong to the same cluster does not necessarily mean that are close to one another

Warnings

- ANYTHING can be clustered
- Given enough patterns, they will always cluster
- There is no scientific value in that there are genes that behave in a similar way, given the amount of genes in the genome and common sample sizes
    - The Scientific value should come from what can be said about the genes that fall in the same cluster and what can be done with said genes
- In most cases, clustering is highly dependent on the distance metric used
    - Changing the distance metric may dramatically affect the number and membership of the clusters, as well as the relationship between them

K-Means Clustering

- K-Means is one of the simplest, fastest and most widely used clustering algorithms
- K-Means takes the number of desired clusters (k) as an input argument
- K Means clustering algorithm :
  - Randomly assign k points as the centers of the clusters
  - Calculate the distance from every point to every cluster center
  - Assign each point to a cluster
  - Reassign cluster centers as the mean of each cluster
  - Recalculate the centroid of each cluster
  - Repeat this process until no pattern moves from one cluster to another

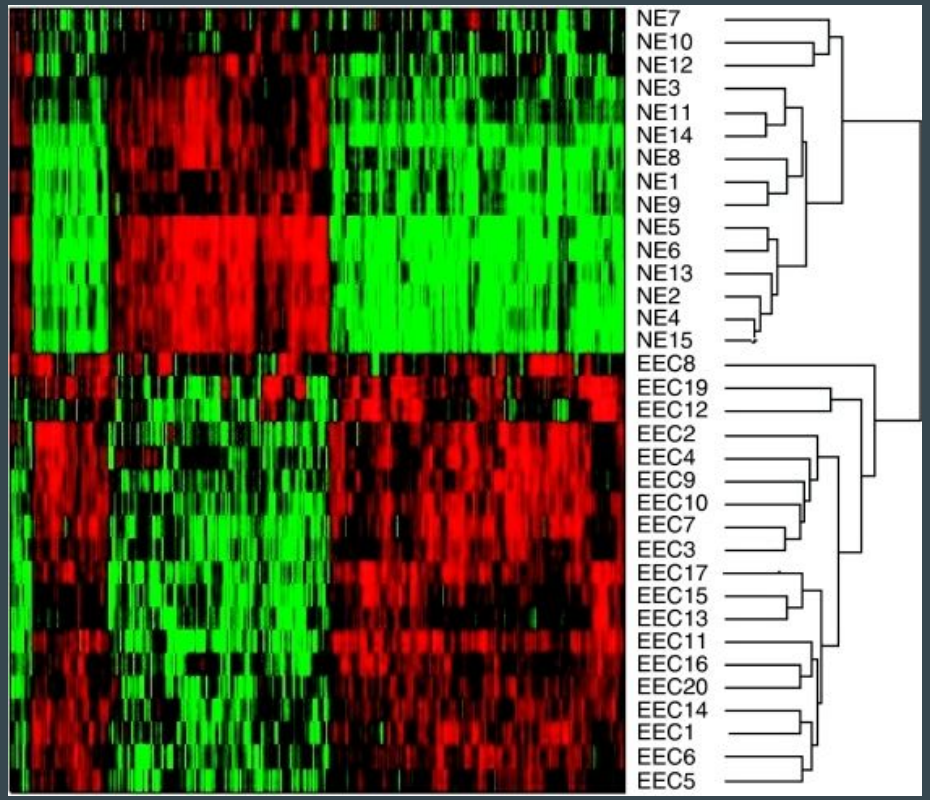
K-Means Clustering

- With K-Means clustering, care should be taken in centroid initialization so that a cluster is not initialized far away from all points, leaving an empty cluster
- A common practice is to initialize centroids as k points chosen randomly from the existing patterns
- This ensures that
  - The starting cluster centers are in an area populated by data
  - Each cluster will have at least one pattern

Hierarchical Clustering

- Hierarchical clustering has been used since the very beginning of the microarray field
- This method aims at the more ambitious task of providing the definitive clustering that characterized a set of patterns the context of a given distance metric
- The result of hierarchical clustering is a complete tree with individual patterns as leaves and the root as the convergent point of all branches, called a dendrogram
- This dendrogram represents a hierarchy of categories based on the degrees of similarity

Hierarchical Clustering



[1]

Hierarchical Clustering

- This method is deterministic and can be applied in a bottom-up (agglomerative) or top-down (divisive) method
- Bottom Up Hierarchical Clustering :
  - Assign n clusters, each containing one pattern
  - Compute the distance from each cluster to every other cluster
  - Merge the two most similar clusters
  - Repeat distance calculation and merging until only one cluster remains

Hierarchical Clustering

- Top Down Hierarchical Clustering
  - Consider the whole set of patterns to be clustered, and use any of a large number of non-hierarchical clustering algorithm to divide the set into two clusters
    - K-Means with k=2 is a possible choice
  - Recursively repeat this process on each of the smaller clusters as they are obtained
  - Terminate when all small clusters contain a single pattern

Partitioning Around Medoids (PAM)

- PAM Clustering starts with computing a dissimilarity matrix from the original data structure with a distance measure of choice
- After this dissimilarity matrix is computed, the resulting distance matrix is mapped into a specified number of clusters
- This algorithm is almost the same as K-Means, with a difference being **cluster centers must be a present data point (medoid)**
- The medoids are representations of the cluster centers, which are robust with respect to outliers (in the same way median is robust to outliers)
    - This is particularly important in the common situation in which many elements do not have a clear-cut membership to any specific cluster

Biclustering

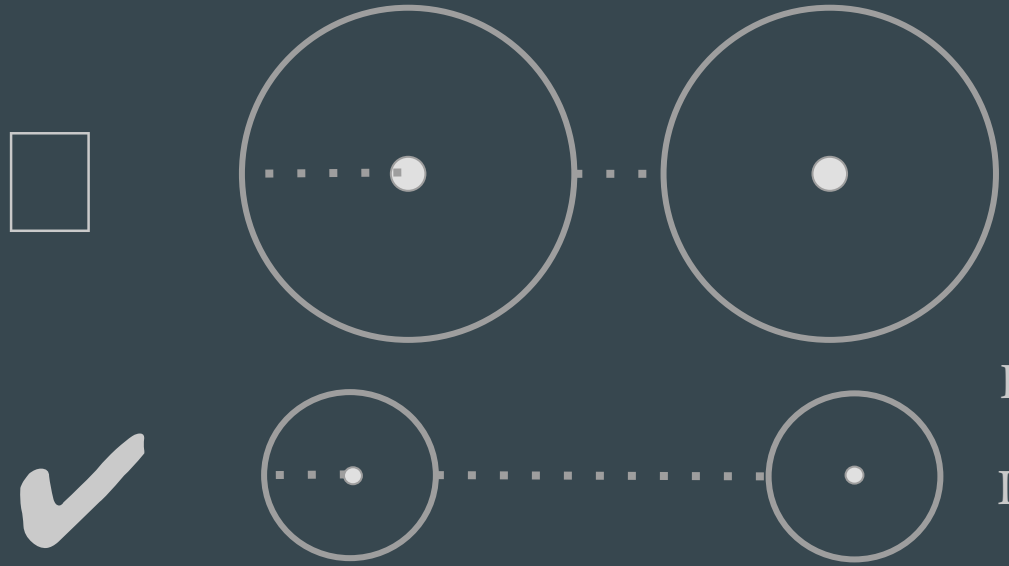
- One can observe that in microarray data, the activities of genes are not independent of each other.
  - It is important to study groups of genes and not single genes
- K-Means and Hierarchical Clustering assume that related geneses should have similar expression profiles across all samples
  - Though this assumption does not hold in all experiments
- Biclustering was proposed to overcome these limitations

Biclustering

- A Bicluster can be defined as a subset of genes that are correlated under a subset of samples
- Biclustering refers to simultaneously clustering both rows and columns of a given matrix of patterns
    - This helps in discovering local patterns that cannot be identified by the standard one-way clustering algorithms
- Biclustering has been used in several applications such as clustering microarray data, protein interactions, collaborative filtering and text mining.

Assessing "Goodness" of Clusters

- One way to assess the goodness of fit of a given clustering is to compare the size of the clusters vs. the distance to the nearest cluster



Assessing "Goodness" of Clusters

- Another possible quality indicator is the average of the distances between the members of a cluster and the center, very similar to before but slightly more robust
    - This is normally done by summing the square of the distances from every point to the center, called Total Sum of Squares
    - The total sum of squares can be taken between clusters and within clusters, and the proportion of this value can be assessed
- The diameter of the smallest sphere including all members of a given cluster may also be used as a quality assessment,
    - Though this is a sensitive measure

Confidence in Cluster Assignment

- How confident can we be that the pattern falls in a given cluster?
    - We can follow a gene through several clusterings to ensure it belongs with its group
- This can also be addressed using a bootstrapping approach, where a goodness of fit measure is based on many repeats of the same experiment on slightly different data sets all constructed on from the available data
- Essentially clustering many times and the confidence of a pattern belonging to a cluster is inversely proportional to the amount of times is moves to a different cluster

Results

Training data cluster, all features

	ALL	AML
1	0.1481481	0.7272727
2	0.8518519	0.2727273

```
> train_cluster$withinss / train_cluster$betweenss
[1] 4.390243 3.480078
```

Testing data cluster, all features

	ALL	AML
1	0.65	0.6428571
2	0.35	0.3571429

```
test_cluster$withinss / test_cluster$betweenss
] 0.9207035 1.1718051
```

Training data cluster, PCs

	ALL	AML
1	0.7037037	0.09090909
2	0.2962963	0.90909091

```
> train_cluster$withinss / train_cluster$betweenss
[1] 3.480078 4.390243
```

Testing data cluster, PCs

	ALL	AML
1	0.2	0.5714286
2	0.8	0.4285714

```
> test_cluster$withinss / test_cluster$betweenss
[1] 1.1718051 0.9207035
```

References

[1] Bignotti, E et al. “Trefoil Factor 3: A Novel Serum Marker Identified by Gene Expression Profiling in High-Grade Endometrial Carcinomas.” British Journal of Cancer 99.5 (2008): 768–773. PMC. Web. 4 Oct. 2018.

[2] Drăghici Sorin. Statistics and Data Analysis for Microarrays: Using R and Bioconductor. Chapman and Hall, 2012.

Clustering Continued

By Jake Sauter

Kohonen Self Organizing Feature Maps (SOFM)

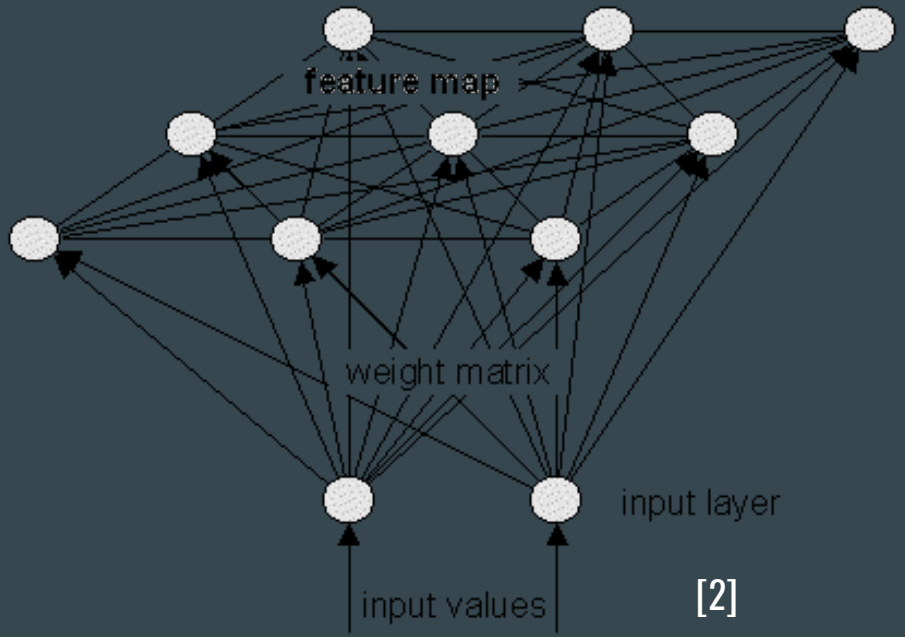
- SOFM is a type of clustering
- Novelty in that it the relationship between the clustered patterns actually contains information about the relationships and reciprocal positions of the patterns in the input space
- SOFM is designed to plot similar patterns next to one another, creating a **feature map**
- A feature map has the property that the distances and relationships measured on the feature map are proportional ot distances and relationships between patterns according to the similarity metric chosen

SOFMs

- The SOFM is actually a **neural network** technique
- Neural networks are graphs connecting simple processing nodes (**neurons**), with each connection possessing a specific connection strength (**weight)**
- Common SOFM architectures implement one dimensional and two dimensional networks

SOFMs

- The **input layer** of the network represents the dimensionality of the input space, and the **output layer** of the network represents the dimensionality of the desired output space



[2]

SOFM Training

- In order to form the output space, the SOFM must be **trained** with respect to a **training rule**
- The **training rule** in combination with a **learning rate** will inform us on how to update the weights of the network as training occurs

SOFM Training

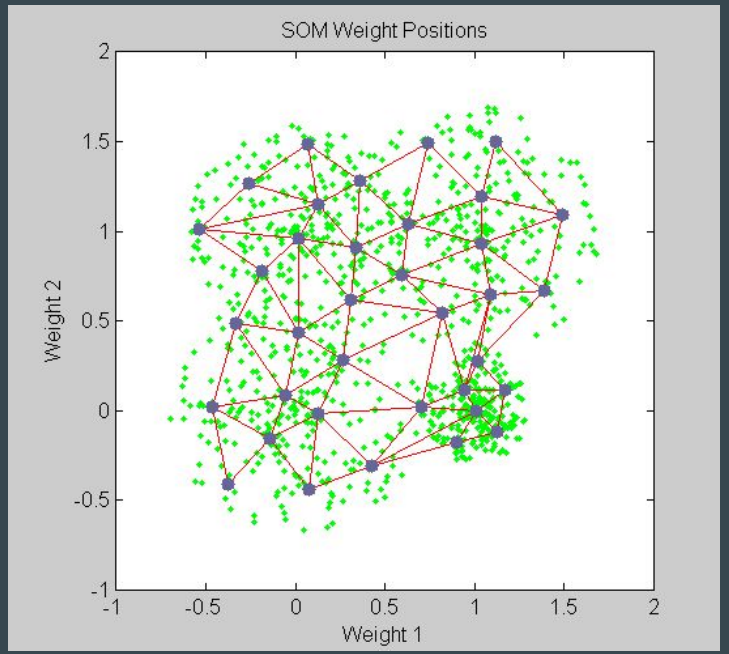
- The process of training occurs the following way:
  - An input is presented to the input layer of the neural network
  - Each neuron calculates the distance from the input to its weight vector
  - The neuron that possesses the smallest distance will be updated such that the weights are made more similar to the input profile
    - Depending on the training rule, weight of surrounding neurons may be updated as well
  - Each input is presented to the network in this way
  - After each input is presented, the learning rate and/or neighborhood distance is decreased and this process is repeated

SOFM Output

- The SOFM provides three benefits
  - Each neuron of the SOFM will represent the set of common features extracted from the input patterns by the neuron
  - The SOFM yields a set of clusters, with all inputs activating the same unit being clustered together
  - The relationship between the neurons activated by specific genes will be closely related to the relationship between the genes

SOFM Use

- SOFMs can be used as a clustering or visualization tool
- One could simply plot the clusters generated from the SOFM
- The prototypes (weights of the units) could be plotted in the input space, with the topology of the network showing the links between the units



[3]

K Means Results

- The K-Means clustering algorithm was implemented from scratch to allow for different distance metrics to be used
- A function to produce meaningful output for supervised learning sets, with any amount of clusters, was crafted
    - This technique could be applied to determine how well classification can even be implemented

K Means Results

Custom Implementation

Library Function

Euc.

```
 between_ss / total_ss =  62.23644 %

Cluster Results:
     ALL         AML
[1,]   1 0.09090909
[2,]   0 0.90909091
```

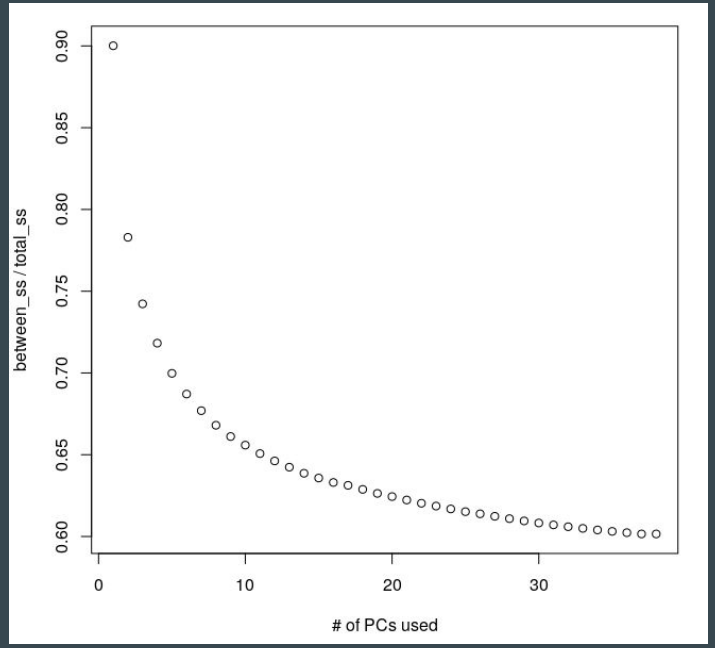
```
between_ss / total_ss =  11.30524 %

Cluster Results:
           ALL       AML
[1,] 0.8518519 0.2727273
[2,] 0.1481481 0.7272727
```

Man.

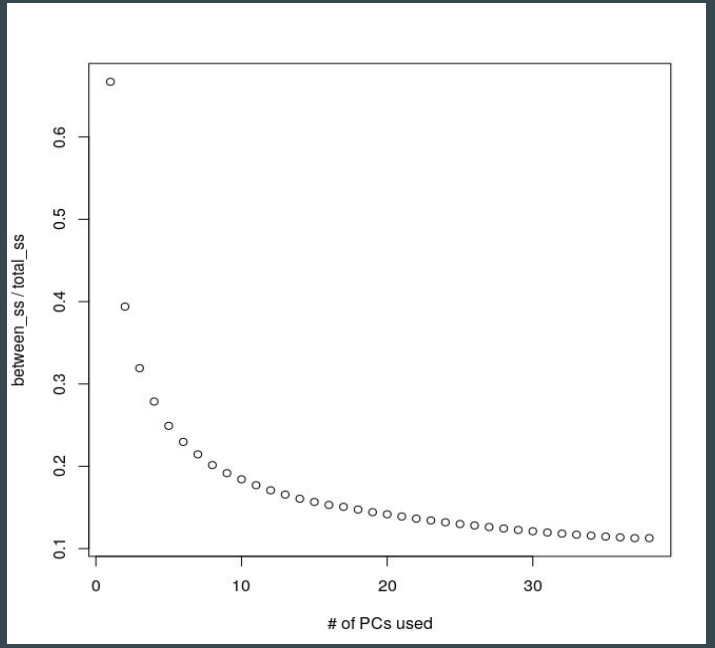
```
 between_ss / total_ss =  60.78344 %

Cluster Results:
           ALL AML
[1,] 0.4074074   1
[2,] 0.5925926   0
```

K Means of Principle Components (Euc)

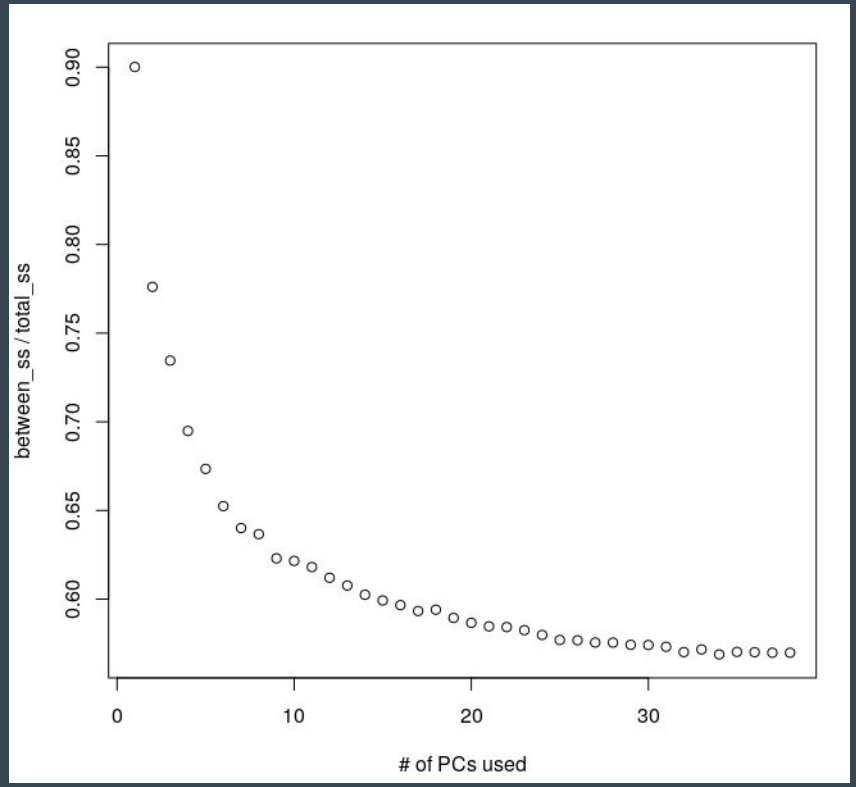
Custom Implementation



Library Function



K Means of Principle Components (Man)



K Medoids Results

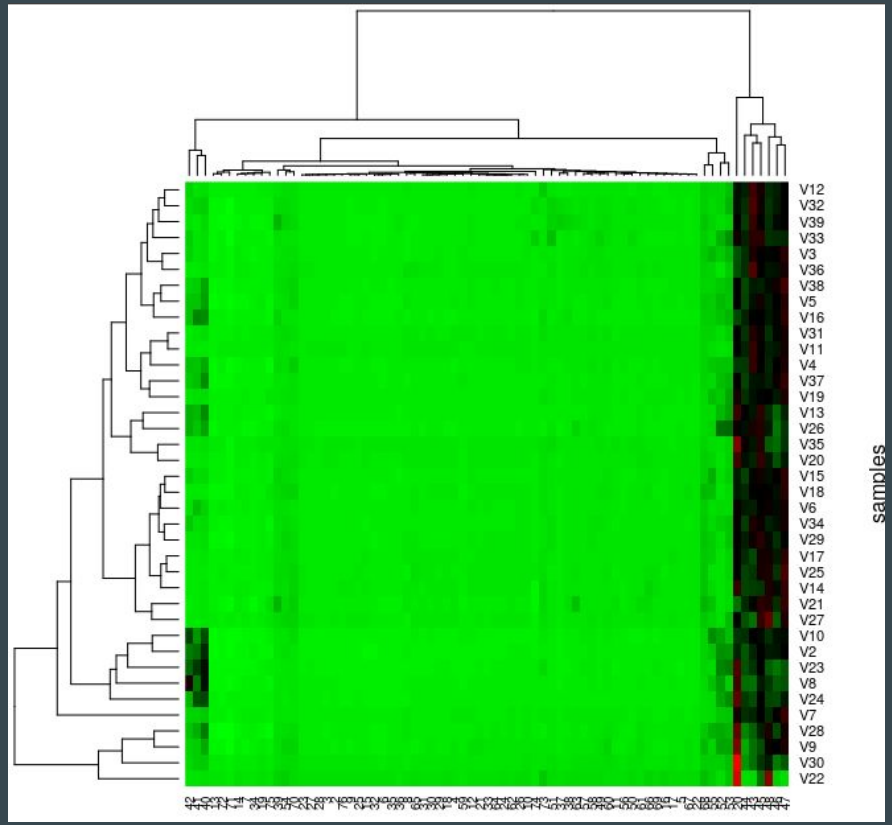
Euclidean

```
Cluster Results:
            ALL       AML
[1,] 0.8888889 0.1818182
[2,] 0.1111111 0.8181818
```

Manhattan

```
Cluster Results:
            ALL       AML
[1,] 0.92592593 0.1818182
[2,] 0.07407407 0.8181818
```

Hierarchical Clustering



Rerenerences

[1] Draǧhici Sorin. Statistics and Data Analysis for Microarrays: Using R and Bioconductor. Chapman and Hall, 2012.

[2] “Kohonen Feature Map.” Kohonen Feature Map - Neural Networks with Java, www.nnwj.de/kohonen-feature-map.html.

[3] “Cluster with Self-Organizing Map Neural Network.” Cluster with Self-Organizing Map Neural Network - MATLAB & Simulink, www.mathworks.com/help/deeplearning/ug/cluster-with-self-organizing-map-neural-network.html;jsessionid=38782e43df7fa2b45736dd152319.

Selecting Differentially Expressed Genes

by Jake Sauter

Motivation

- In all comparative studies (healthy vs disease, treated vs untreated, drug A vs drug B) a very important problem is to determine the genes that are **differentially expressed** (DE) in the two samples being compared
- This task is simple in principle, though becomes more complex in reality due to numerous sources of fluctuation and noise
    - For spotted cDNA arrays, non-negligible ~0.05 probability that hybridization of any spot will not reflect the presence of mRNA
    - The probability that a single spot will provide a signal even if mRNA is not present is ~0.10

Combative Production Methods

- Affymetrix technology tries to respond to the challenge of poor reliability by using a set of multiple probes to represent a gene
    - Each gene is represented by two probes, one being a perfect match (PM) and the other having a mismatch (MM) in the middle of the sequence
    - The average difference in PM and MM is taken as representative of the for the expression level of the gene
- Illumina microarrays have a large number of beads carrying the same DNA sequence, constituting technical replications (allow the estimation of variance)

Criteria

In order to assess the performance of a gene selection method, quantifiable criteria must be devised to analyze the outcome of the selection process

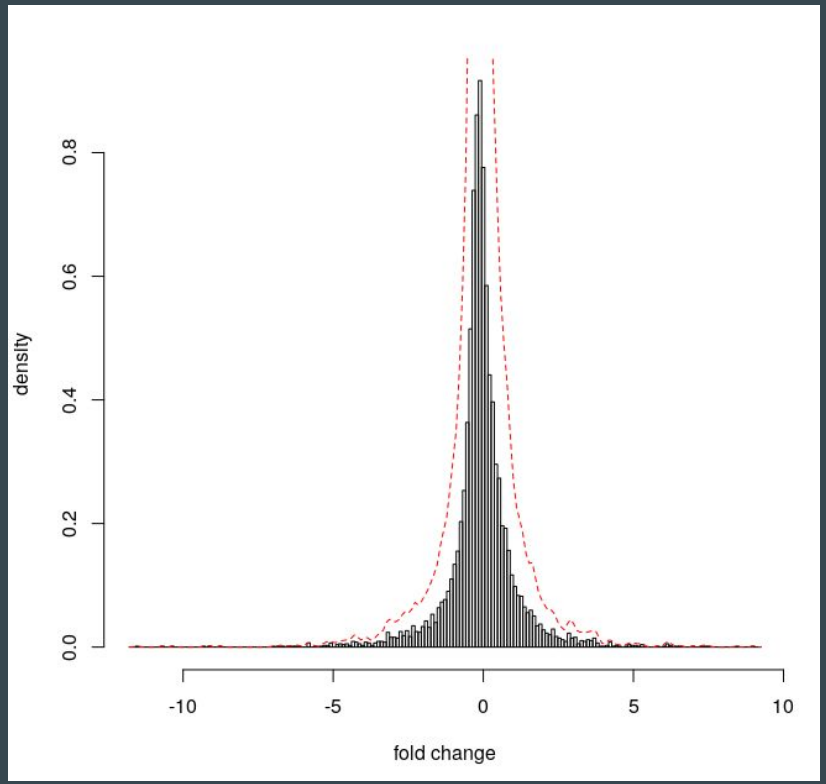
	True		
Reported	changed	unchanged	
changed	TP	FP	Positive predicted value $\frac{TP}{TP+FP}$
unchanged	FN	TN	Negative predicted value $\frac{TN}{TN+FN}$
	Sensitivity $\frac{TP}{TP+FN}$	Specificity $\frac{TN}{TN+FP}$	Accuracy $\frac{TP+TN}{TP+TN+FP+FN}$

Fold Change

- **Fold Change** is the simplest and most intuitive approach to finding genes that are DE
- The log of this ratio is normally taken to provide better distribution characteristics
- Genes are selected as DE if they have a fold change greater than an arbitrary selected threshold

Fold Change

- The log of these fold change ratios can be plotted as a histogram, with the x-axis in fold change units. Selecting DE genes relates to selecting ratios on the tails of the distribution

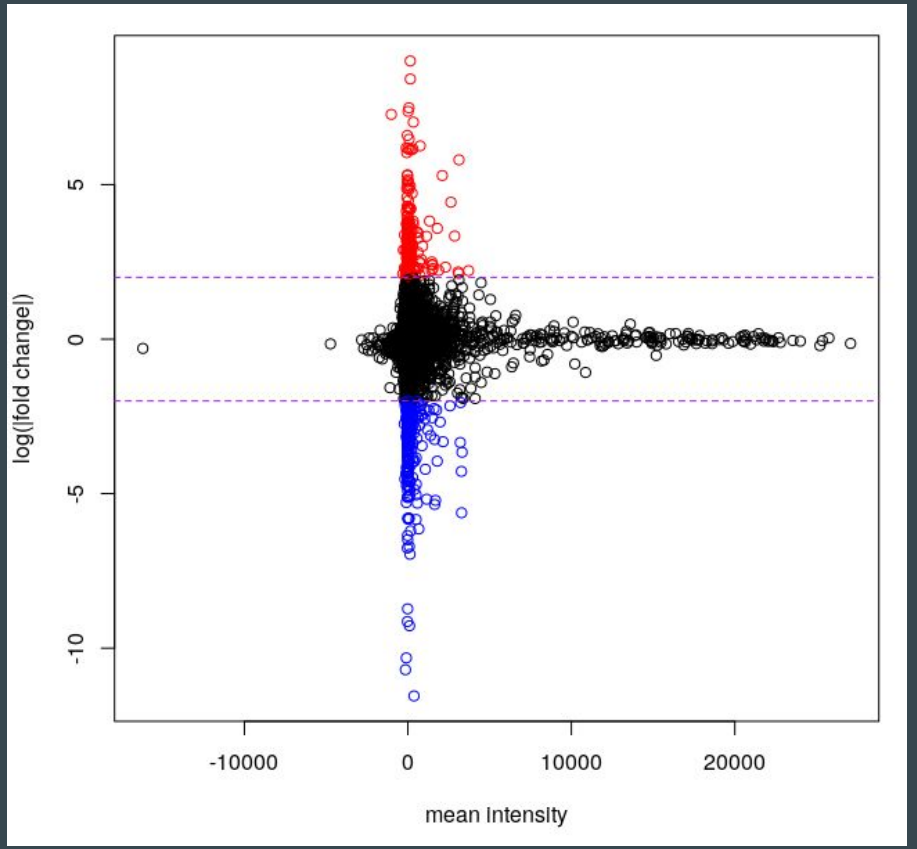


Fold Change

- Fold change is often used because it is simple and intuitive, however it has important disadvantages
    - The fold threshold is chosen arbitrarily and may often be inappropriate (no genes or non-DE genes can be selected)
- Microarrays tend to have low signal to noise ratios for genes with low expression levels (low intensity values have higher variance, high intensity values have lower variance)
    - Constant fold change threshold for all genes will introduce false positives at low expression levels (decreasing specificity) and miss true positives at high expression levels (decreasing sensitivity)

Fold Change Applied

- The log of the absolute values of the fold change ratios had to be taken due to the normalization techniques applied to the data
- 547 genes were selected as DE with a fold change threshold of 2



Unusual Ratio

- This method is superior to the fold-change method while still being simple and intuitive
- The cut-off threshold is automatically adjusted
    - Thresholds on how different the experiment/control ratio of a gene is with respect to the mean of all ratios is used instead of thresholds on the values of the ratios themselves
- No matter how many genes are up/down regulated, and no matter by how much they are regulated by, this method will always pick the genes that are affected the most

Unusual Ratio

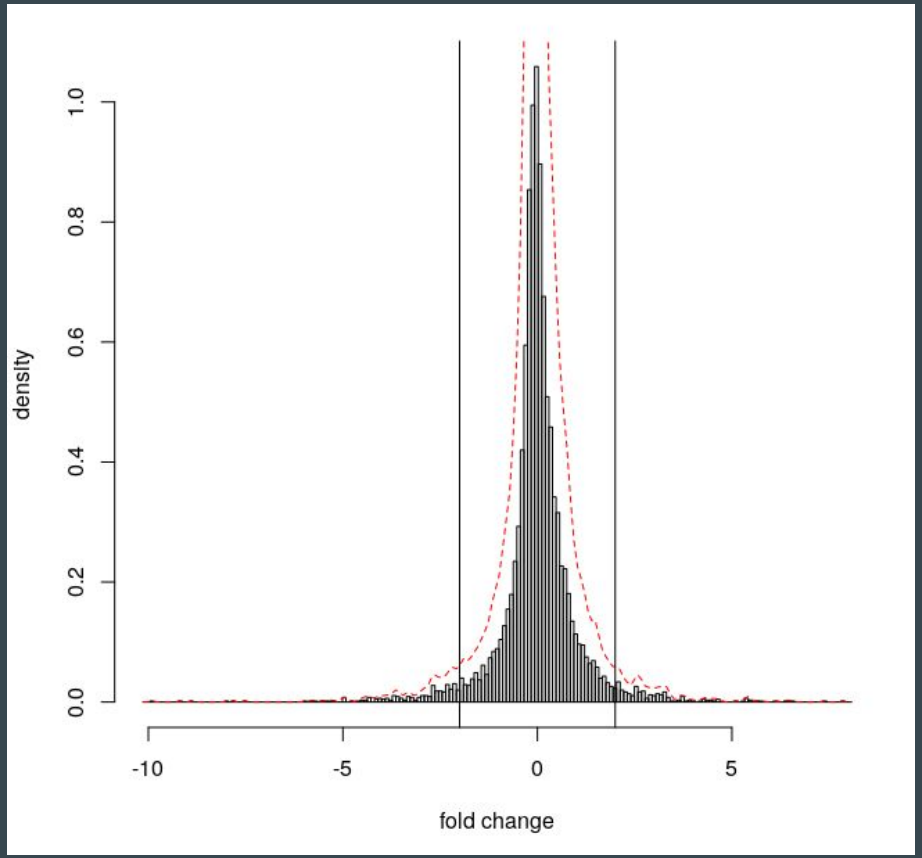
- Also a widely used tactic for selecting DE genes
- For this method, a z-transform is applied to the log ratio values
- Selects genes that are a certain distance from the mean experiment/control ratio
    - Typically this distance is taken to be $\pm 2\sigma$

Unusual Ratio

- This method still has important intrinsic drawbacks
  - Will report 5% of the genes as DE even if no DE genes are present
  - Will report 5% of the genes as DE even if many more genes are in fact DE
  - Continues to use cutoff boundaries unrelated to the high variance of low expression levels and low variance of high expression levels
- A variation of this method selects those genes for which the absolute difference in the average expression intensities is much larger than the estimated standard error computed for each gene using array replicates (if present)

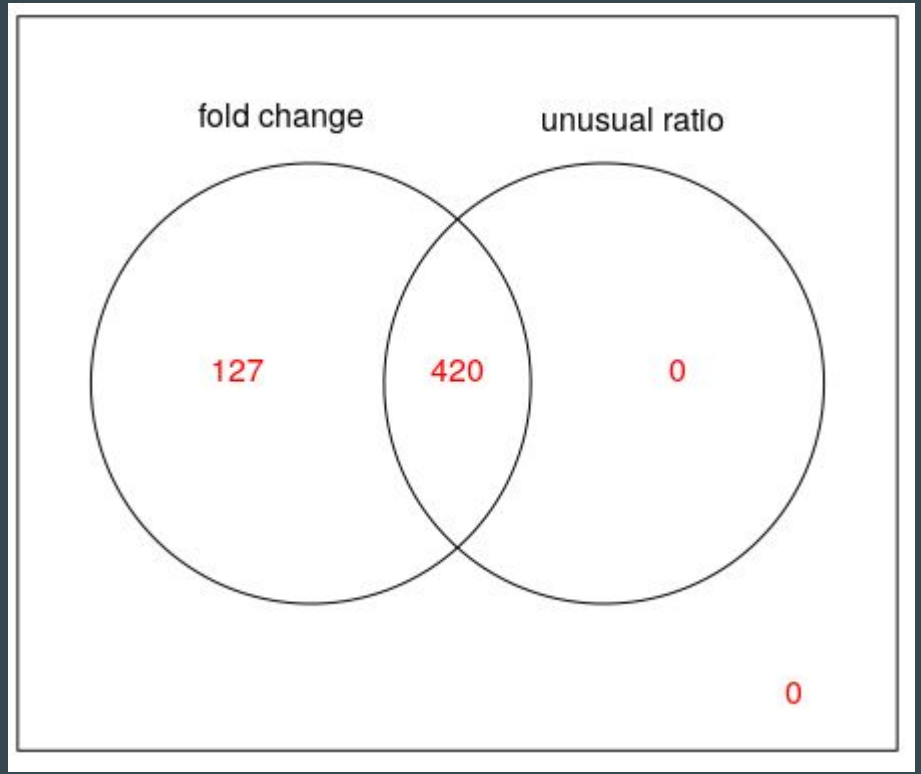
Unusual Ratio Applied

- The log of the absolute values of the ratios were z-scored before this method was performed
- 420 genes were selected as DE, being 2σ from the mean ratio



Fold Change and Unusual Ratio

- All genes selected by the unusual ratio method were also selected in the fold change method
- With a fold change threshold of 2 (really 4 as we have taken the base 2 log), we see that 127 more genes are selected than are in the unusual ratio method with 2σ threshold



Hypothesis Testing

- Another possible approach to selecting DE genes is to use a univariate statistical test (e.g. t-test)
- Significance level corrections for multiple comparisons must be made when doing simultaneous hypothesis tests
    - The Bonferroni and Šidák corrections for multiple comparisons have been discussed previously
    - The Holm stop-down group of methods, false discovery rate (FDR) and significance analysis of microarrays (SAM) are all suitable methods for multiple comparison corrections in the context of microarray data

Hypothesis Testing

- The drawback of hypothesis testing for finding DEs is that they tend to be conservative
  - There may just be insufficient data to reject the null hypothesis
  - Though the genes that are selected as DE are very likely to be so
- The classical hypothesis testing approach assumes that the genes are independent, which is clearly untrue in the analysis of genetic data sets
  - Combining classical hypothesis testing approaches with a re-sampling or bootstrapping approach (step-down methods or SAM) tends to make the test less conservative and take the dependencies into account

Hypothesis Testing - Gene Filtering

- As discussed, the problem with hypothesis testing is that when performing multiple tests, out significance level must change with the number of tests
    - The best way to avoid the issues that arise with multiple comparisons is to perform only as many tests are necessary
- To prevent too many tests from being performed, it is best to first filter out genes that that are array pike controls and probes that are either expressed at a very low level or exhibit little variability between samples
    - To avoid any bias being introduces, label/group information cannot be used in this filtering step

Hypothesis Testing - Gene Filtering

- Common filters ensure that
  - There are at least a few samples in the group that have significant expression values
  - The ratio of maximum / minimum intensity is at least 1.5 (the gene is variable)
- When the Golub (1999) data was filtered only for max / min intensity difference of 1.5, only 2012 of the original 7129 genes remained

Hypothesis Testing - ANOVA

- ANOVA can be used to build and explicit model of all sources of variance that affect the measurements and use the data to estimate the variance of each individual variable in the model
    - ANOVA requires a complex experimental setup so is not normally performed
- To see the complexity, the Kerr and Churchill model is as follows

$$log(y_{ijkg}) = \mu + A_i + D_j + G_g + (AD)_{ij} + (VG)_{kg} + (DG)_{jg} + \epsilon_{ijkg}$$

where the noise of all experimental interactions are modelled to produce the log ratio for gene *g* of variety *j* measure on array *i* using dye *j*

Hypothesis Testing - Noise Sampling

- A variation of ANOVA can be used to identify DE genes using spot replicates on single chips
  - Noise is estimated and confidence levels for gene regulation can be calculated
- This method modifies the Kerr-Churchill model as follows

$$logR(gs) = \mu + G(g) + \epsilon(g, s)$$

logR(gs) - log ratio for gene *g* on spot *s*

μ - average log ratio over the whole array

G(g) - a term for the differential regulation of gene *g*

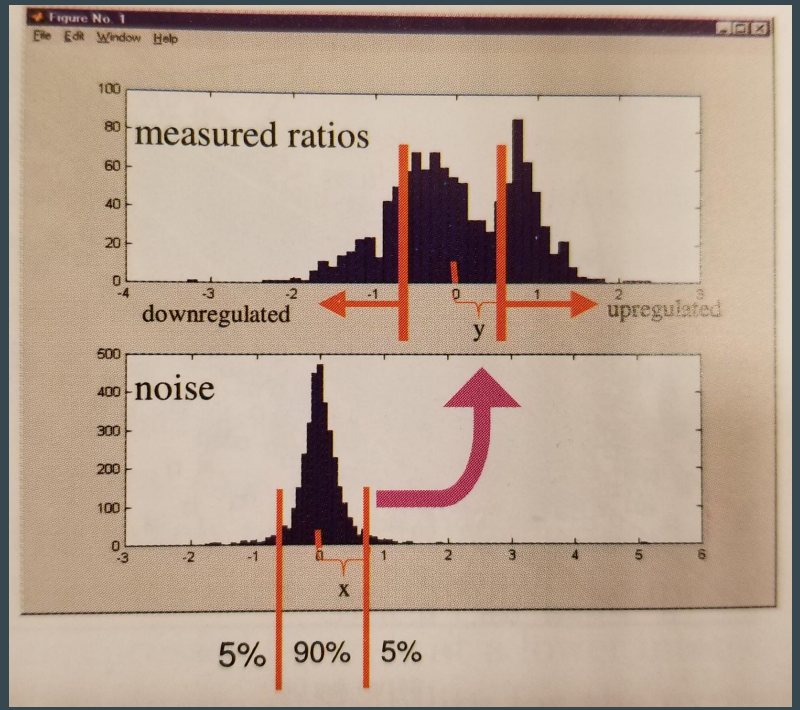
ϵ(g,s) - zero-mean noise terms

Hypothesis Testing - Noise Sampling

- In this model, estimates of the terms can be formulated to provide an estimate for the noise of each gene at each spot
  - The noise samples from each spot can be collected to yield an empirical noise distribution
  - A given confidence level can be associated with a deviation from the mean of this distribution
  - This distance on the noise distribution then can be corresponded to a distance on the measured distribution by bootstrapping
  - Dependency between intensity and variance can be taken into account by constructing several models covering the entire intensity range

Hypothesis Testing - Noise Sampling

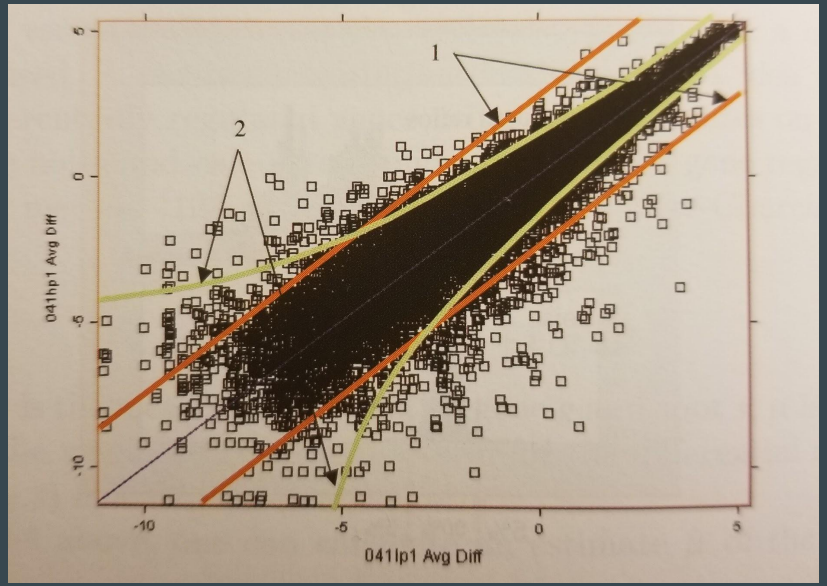
Visualization of correlation of noise distribution to gene expression ratio distribution



[1]

Hypothesis Testing - Noise Sampling

- The noise sampling method is advantageous as it possesses nonlinear selection boundaries that adapt automatically both to various amounts of regulation and different amount of noise for a given confidence level



(1) - Fold Change

(2) - Noise Sampling

[1]

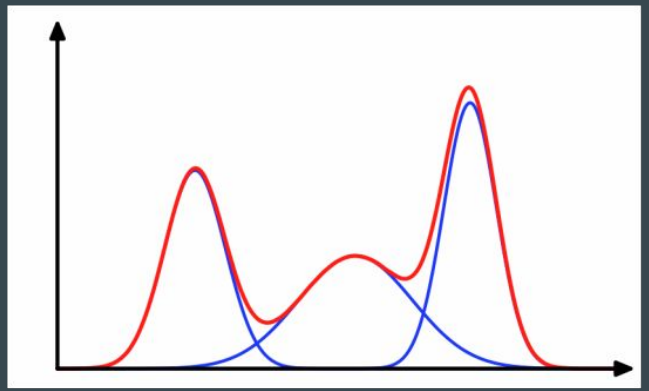
Hypothesis Testing - SAM

- SAM is software designed by Stanford specifically for applications in microarray analysis
- Uses a "relative difference" statistic d_i for gene i
  - Very similar to a t statistic with equal variance, except that the "gene-specific scatter" (std. dev. of the difference) s_i in the denominator is offset by a "fudge factor" s_0
  - This s_0 is an "exchangeability" factor chosen to minimize the coefficient of variation of d_i
  - This small positive exchangeability factor stabilizes d_i for genes with low expression levels
- A permutation test is used to assess significance of d_i, as well as estimate the FDR

Model-based MLE

- If p_1 , p_2 and p_3 are all a priori probabilities of a gene being expressed as up, down or not regulated; f_{up} f_{down} and $f_{unchanged}$ are all pdf's of observing value y ; then f_j (y) is a mixture model of three distributions for the probability of the observed value of the gene occuring

$$f_j(y) = p_1 \cdot f_{up_j}(y) + p_2 \cdot f_{down_j}(y) + p_3 \cdot f_{unchanged_j}(y)$$



[2]

Model Based MLE

- Then in using Bayes Theorem, we can calculate the probability of the gene in question being DE (The even E_g)

$$Pr\{E_g \mid Y_{gj} = y\} = \frac{p_1 \cdot f_{E_j}(y)}{f_j(y)}$$

Model Based MLE

- This method assumes that the model is a mixed normal pdf, so only mean and variance must be estimated for all distributions
  - These parameters can be estimated numerically using a maximum likelihood approach, which searches various combinations of parameters until the obtained equation fits the data as best as possible
- This approach has been used to provide several important facts about studies of this type
  - Any spot can provide erroneous results, thought the probability of three or more spots being erroneous is negligible
  - The intensities do seem to be normally distributed
  - The probability of a false negative is as high as 5% for any single replicate
  - The probability of a false positive is as high as 10% for any single replicate

Model Based MLE

- This approach is very general and powerful
  - The MLEs become unbiased minimum variance estimators as the sample size increases
- However this approach has many disadvantages
  - Requires solving complex nonlinear equations (computationally intensive)
  - The experimental method for isolating the distributions is complicated, and involves three a priori probabilities
  - Results become quickly unreliable as sample size decreases

References

[1] Drăghici Sorin. Statistics and Data Analysis for Microarrays: Using R and Bioconductor. Chapman and Hall, 2012.

[2] Kagan, Michael. Machine Learning: Lecture 1, CERN, 9 July 2018, indico.cern.ch/event/726959/attachments/1683504/2705968/Kagan_Lecture1.pdf.

Noise Sampling Estimates

$$logR(gs) = \mu + G(g) + \epsilon(g, s)$$

$$\hat{\mu} = \sum_{g} log(R(g, s))$$

$$\hat{G(g)} = \frac{1}{m} \sum_{g} log(R(g, s)) - \hat{\mu}$$

$$\hat{\epsilon(g, s)} = log(R(g, s)) - \hat{\mu} - \hat{G(g)}$$

Selecting DE Genes Continued

...

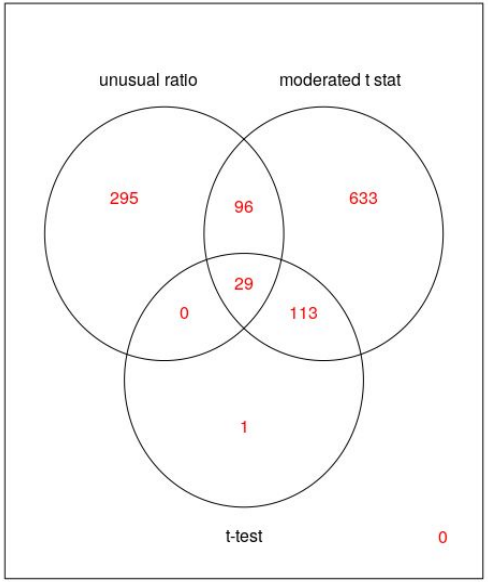
by Jake Sauter

Covered This Week

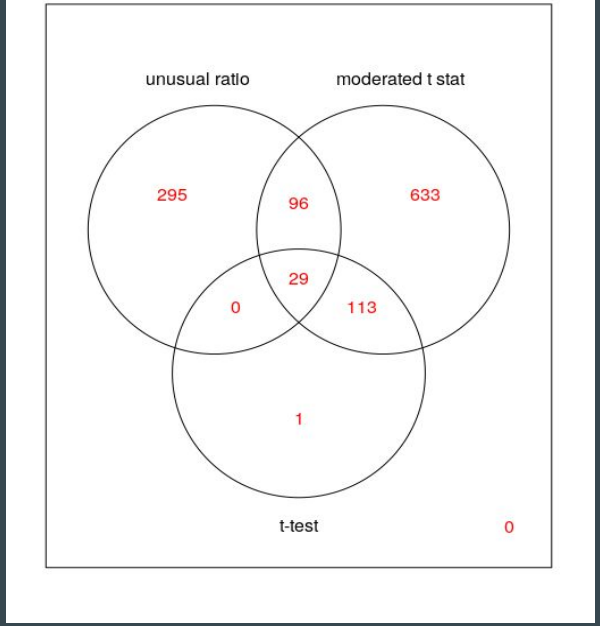
- Implemented Holms and FDR Family Wise Error Rate corrections for t-test
- Use of SAM in R
- Implemented permutation testing for t-test

T-test Results

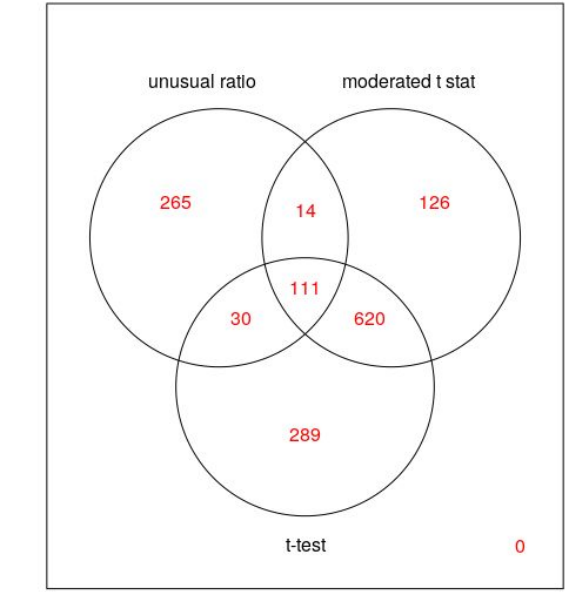
Bonferroni



Holms



FDR

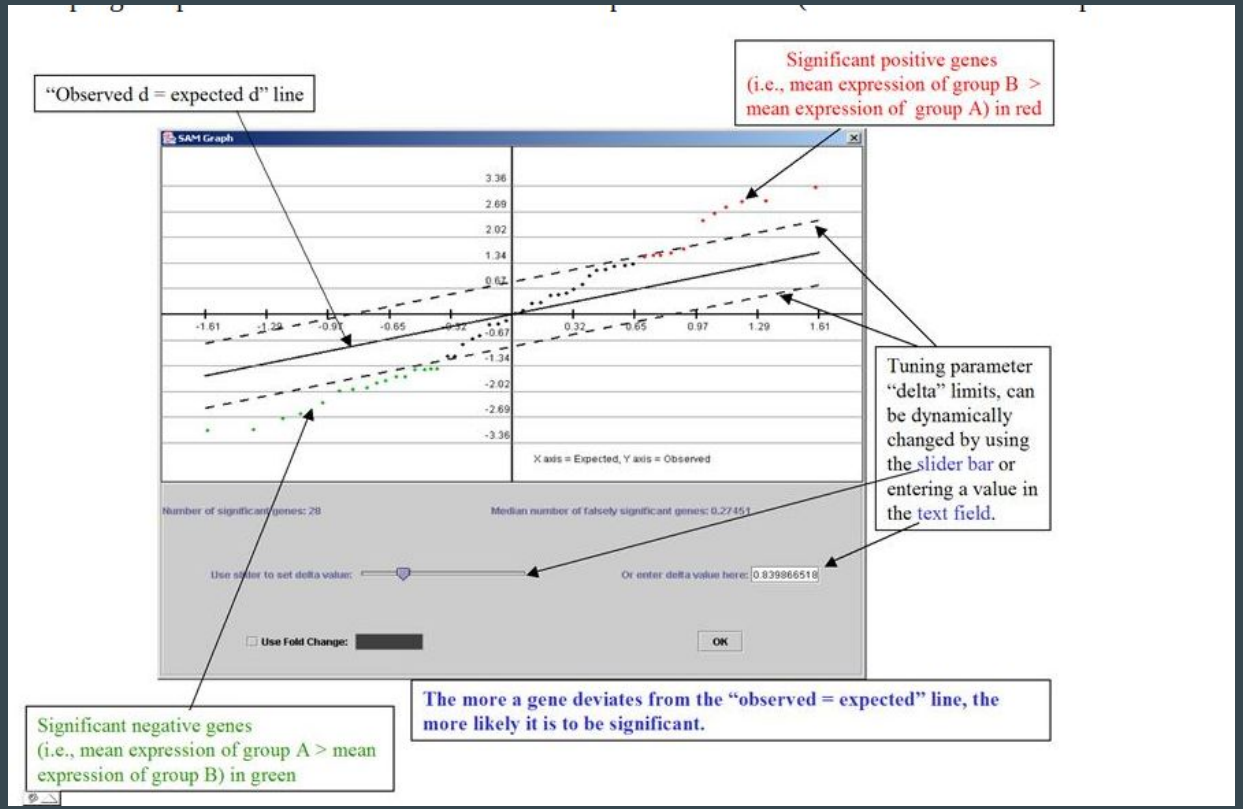


SAM Results

- The way to control for FDR in SAM is through the **delta** parameter
- SAM results allow for a table to be printed that shows how many genes are selected, and various FDR statistics associated with a particular delta value

```
> delta.table[, c(1,4,5,6)]
         delta # called  median FDR 90th perc FDR
 [1,] 0.000000000     2127 0.504348388   0.527150150
 [2,] 0.001772127     2126 0.503690508   0.527142359
 [3,] 0.007088508     2118 0.502512485   0.526771709
 [4,] 0.015949144     2110 0.499779422   0.525728268
 [5,] 0.028354033     2091 0.496259888   0.524004677
 [6,] 0.044303177     2061 0.491611968   0.522425104
 [7,] 0.063796575     2048 0.484511353   0.520192628
 [8,] 0.086834227     2024 0.478973931   0.516179702
 [9,] 0.113416133     1994 0.472546391   0.512766006
[10,] 0.143542293     1972 0.463618762   0.508257404
[11,] 0.177212707     1945 0.450486464   0.501000211
[12,] 0.214427376     1861 0.428748663   0.492703063
[13,] 0.255186298     1774 0.401349687   0.478891857
[14,] 0.299489475     1693 0.375429673   0.466830259
[15,] 0.347336906     1595 0.345829794   0.455152233
[16,] 0.398728591     1485 0.305725077   0.435594161
[17,] 0.453664530     1372 0.269479568   0.417217779
[18,] 0.512144724     1248 0.231122297   0.399203319
[19,] 0.574169171     1179 0.199223692   0.370823543
[20,] 0.639737873     1058 0.158026708   0.349200482
[21,] 0.708850829     1004 0.129700857   0.313664871
[22,] 0.781508039      937 0.106189736   0.281083651
[23,] 0.857709503      850 0.082516695   0.248509582
[24,] 0.937455221      780 0.060993632   0.206541863
[25,] 1.020745193      719 0.044994468   0.177104274
[26,] 1.107579420      656 0.035225416   0.151013431
[27,] 1.197957901      586 0.023659938   0.120712077
[28,] 1.291880635      545 0.016461086   0.086794817
[29,] 1.389347624      488 0.010027524   0.064844658
[30,] 1.490358867      438 0.004965431   0.047668134
[31,] 1.594914365      385 0.004236738   0.026973896
```

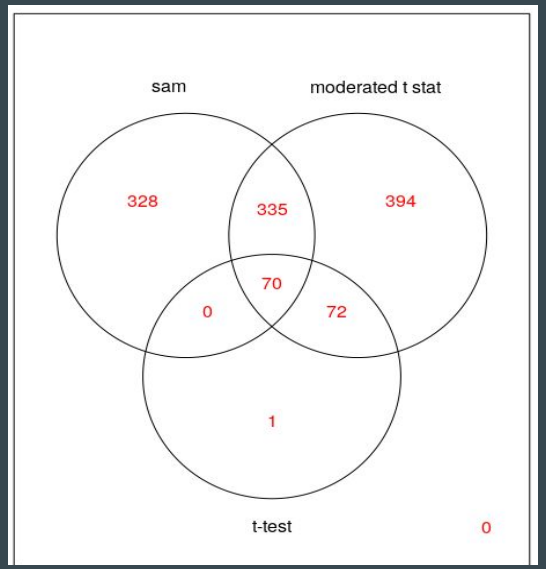
SAM Results



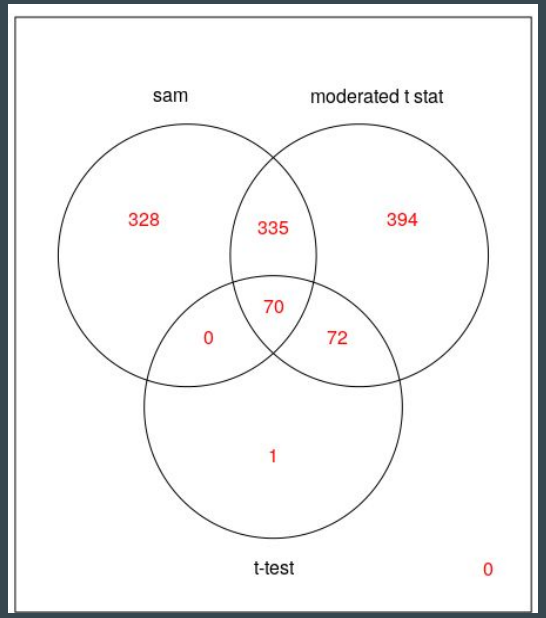
[1]

SAM Results

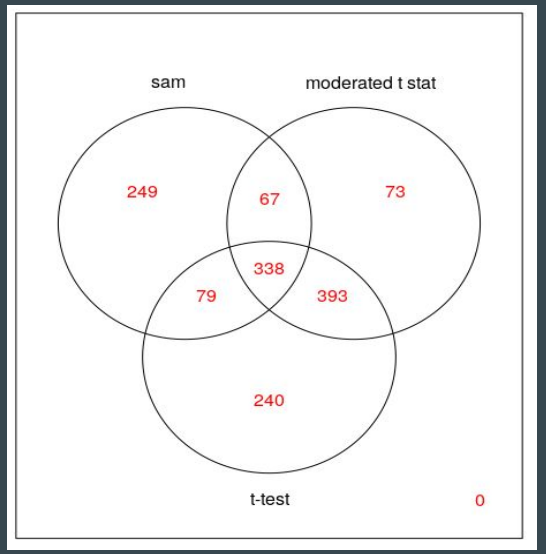
Bonferroni



Holms



FDR

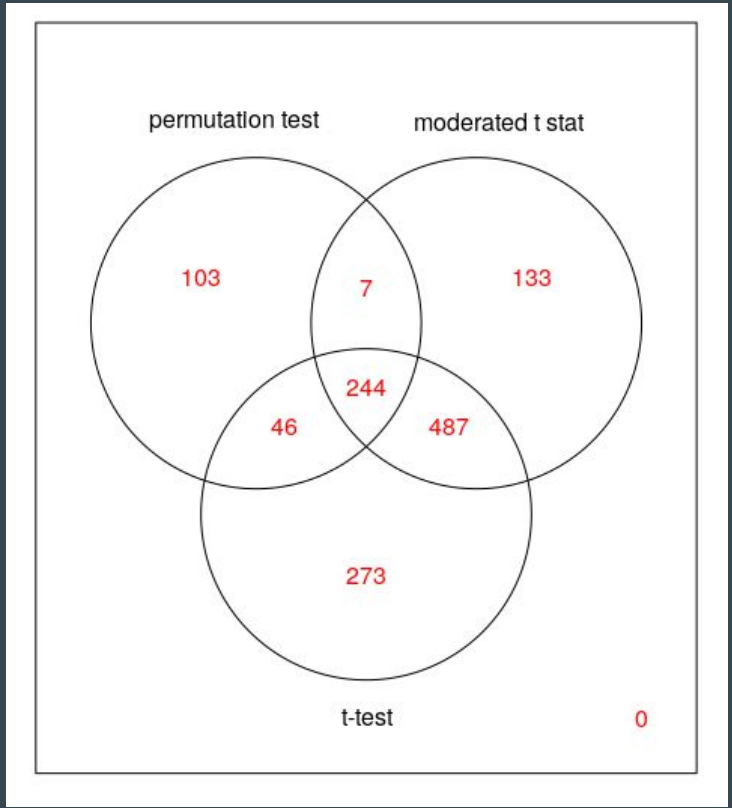


Permutation Testing

- For permutation testing, our goal is to determine the distribution of our test statistic by computing the test statistic for all or some amount of permutations of the data
    - With the distribution of our test-statistic, we can produce a p-value with our actual test statistic, counting how many statistics in our distribution are greater than or equal to our actual test statistic
- If there are too many possible permutations to account for, we can choose permutations randomly (Monte Carlo Sampling) [2]
    - The more random permutations we perform, the closer our analysis become to an exact test

Permutation Testing

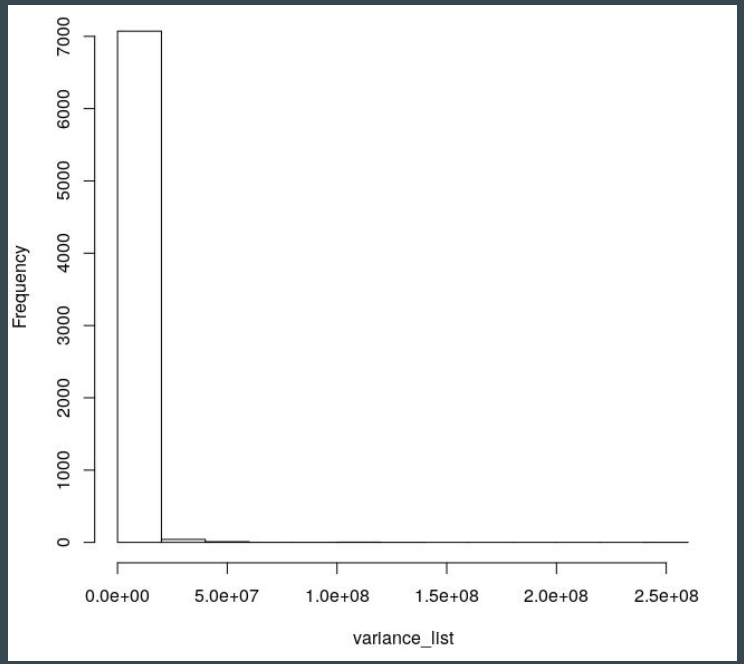
- 10,000 tests were performed for each gene
- Only 1 value was below the .05 threshold
- The 10 lowest p-values were 0.0324, 0.0694, 0.0741, 0.0775, 0.1013, 0.1033, 0.1103, 0.1186, 0.1205, 0.1284



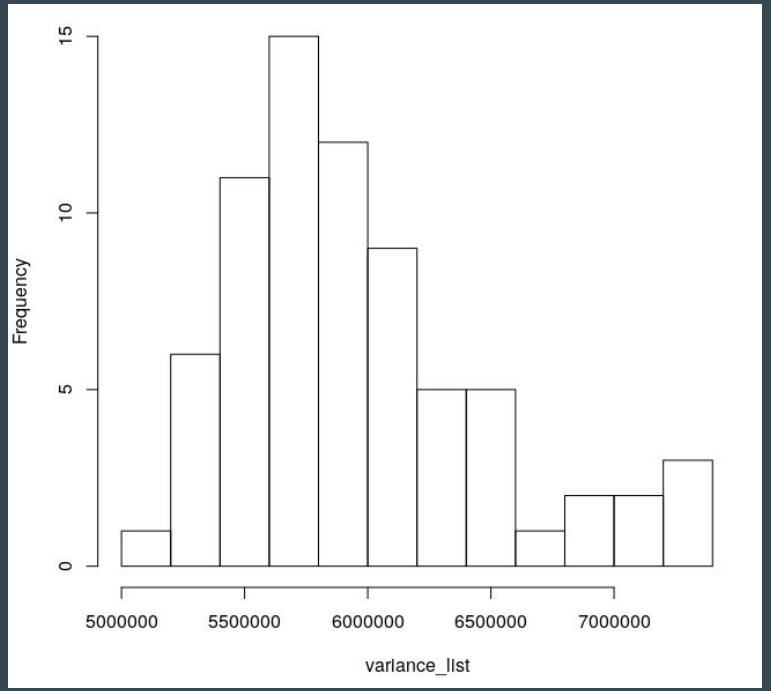
Selecting the 400 lowest p-values of permutation testing

Variance Observations

Genes

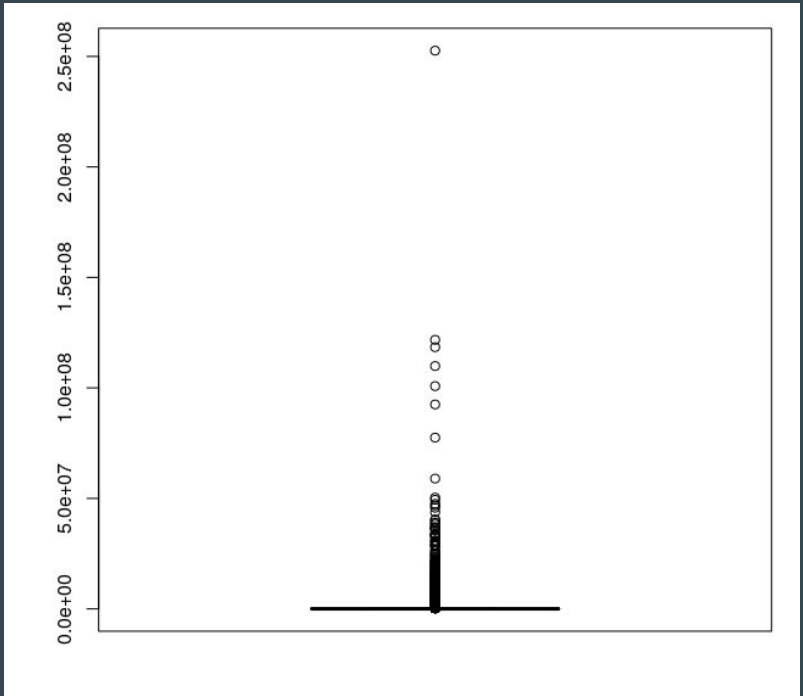


People

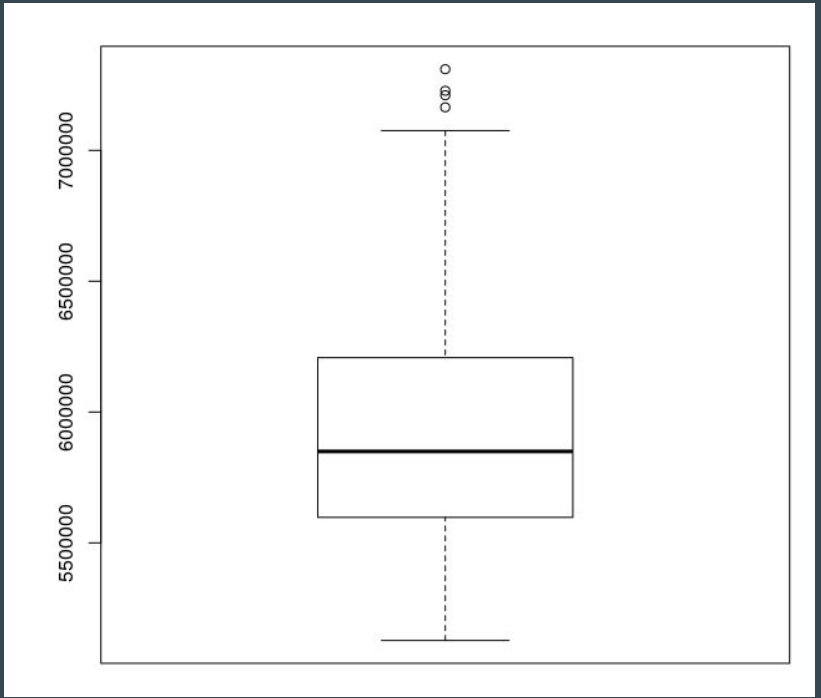


Variance Observations

Genes



People



References

[1] Cui, Yan. “Data File: Stanford_Large.Txt.” Microarray Data Analysis II, compbio.uthsc.edu/microarray/lecture2.htm.

[2] Jianqiang, MA. “Permutation Test & Monte Carlo Sampling.” Permutation Test & Monte Carlo Sampling, 18 Mar. 2009, www.let.rug.nl/nerbonne/teach/rema-stats-meth-seminar/presentations/Permutation-Monte-Carlo-Jianqiang-2009.pdf.

[3] Drăghici Sorin. Statistics and Data Analysis for Microarrays: Using R and Bioconductor. Chapman and Hall, 2012.

Selecting DE Genes: Moderated *t*-Statistic

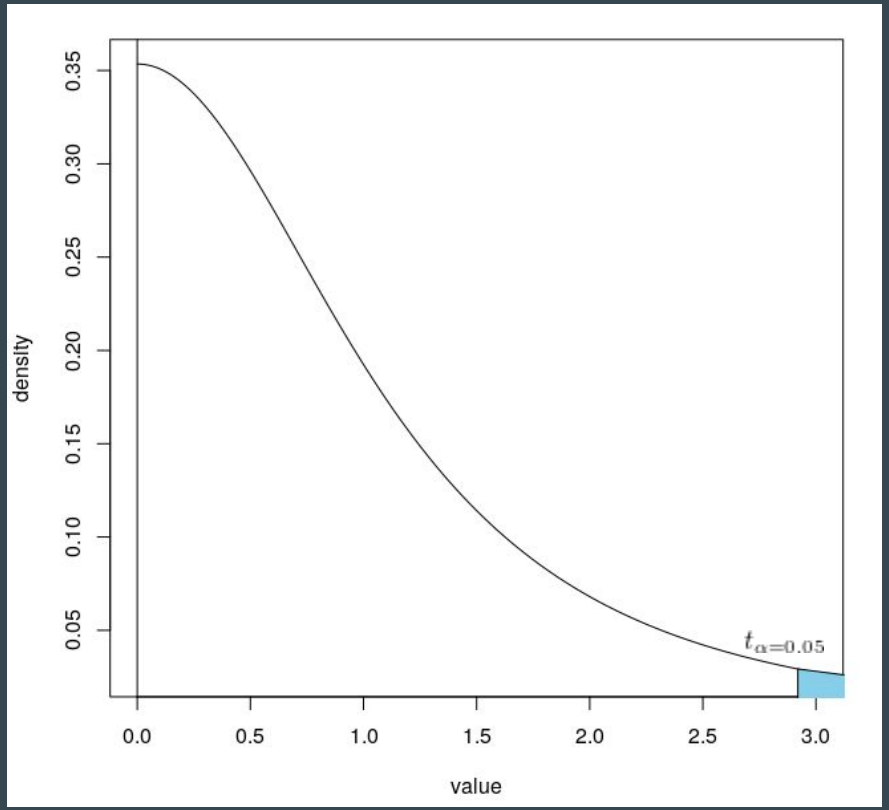
by Jake Sauter

Background: *t*-Statistic

- We have previously discussed the common hypothesis test known as the *t*-Test
- The statistic for this test is

$$t = \frac{\hat{B}_{gj}}{\hat{\sigma}/\sqrt{n}}$$

- g $j = 1$
- g $j = 2$
- $\cdots$
- g $j = n$



Background: Empirical Bayes Methods

- Empirical Bayes Methods are **procedures for statistical inference** in which the prior distribution is estimated from the data
- This family of methods is an approach to setting hyperparameters of known distribution to best fit the data

Background: General Linear Model

- For the General Linear Model (GLM) we assume that the observations Y_i can be modelled by a constant followed by linear scaling factors of various variables, plus an error rate for the observed sample

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_p X_{ip} + \epsilon_i$$

Background: Hierarchical Models

- A Hierarchical Linear Model or a Multilevel Model is statistical model of parameters that vary at more than on level
  - Generally, individual analysis and group analysis parameters share a relationship

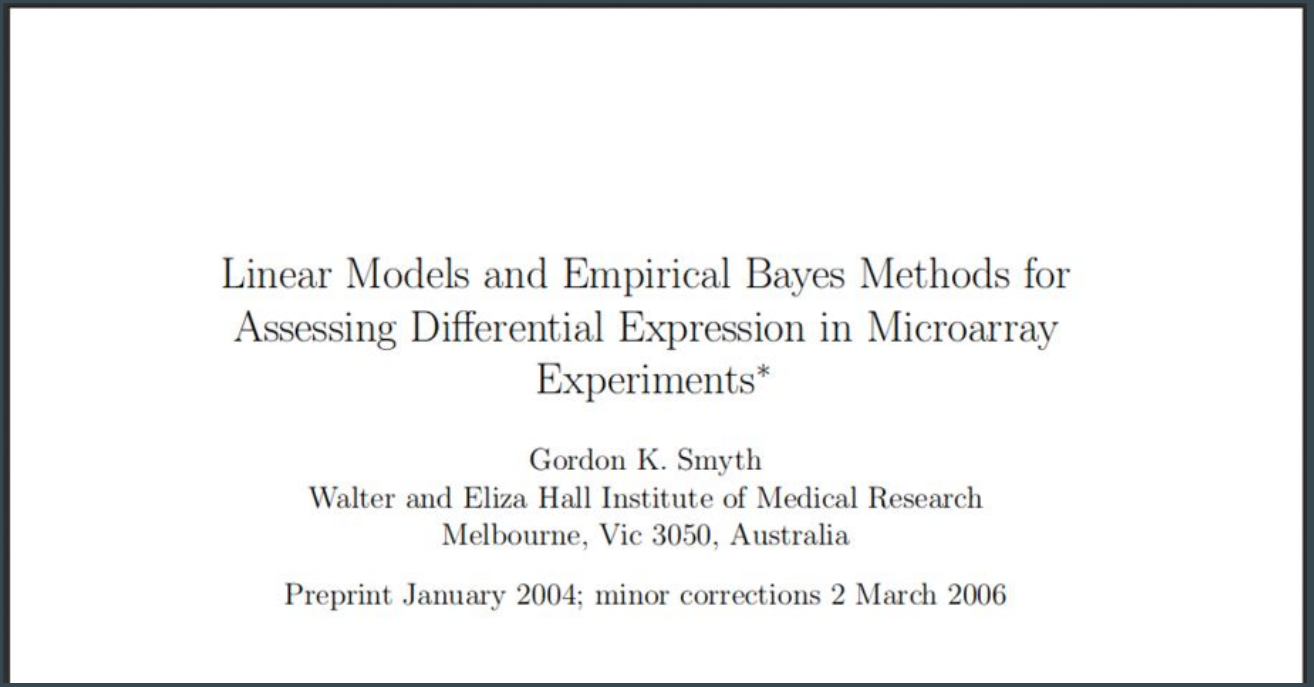
$$\textbf{[Level 1]} \quad Y_{ij} = \beta_{0j} + \beta X_{ij} + r_{ij}$$

$$\downarrow$$

$$\textbf{[Level 2]} \quad \beta_{0j} = \gamma_{00} + \gamma_{01} W_j + v_{0j}$$

[2]

Origin of Moderated *t* Statistic



Linear Models and Empirical Bayes Methods for Assessing Differential Expression in Microarray Experiments*

Gordon K. Smyth
Walter and Eliza Hall Institute of Medical Research
Melbourne, Vic 3050, Australia

Preprint January 2004; minor corrections 2 March 2006

[1]

Linear Model Setup

- Assume that we have n microarrays with an expression vector :

$$y_g^T = (y_{g1}, \cdots, y_{g2})$$

being a vector of log-ratios or log intensities

- The general linear model is assumed as :

$$E(y_g) = X\alpha_g \text{ with } var(y_g) = W_g\sigma_g^2$$

where X is a design matrix, α_g is a coefficient vector, and W_g is a known weight matrix

Linear Model Setup

- Arbitrary contrasts of biological interest β_g can be extracted from the coefficient vector α_g :

$$\beta_g = C^T \alpha_g$$

- This is done with the contrast matrix C

$$C = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

Linear Model Estimation

- With this linear model setup **for each gene**, the model is fit and generates estimators $\hat{\alpha_g}$ of α_g, s^2 of σ^2, and $var(\hat{\alpha_g})$
- The estimators of contrast $\hat{\beta_g}$ and its variance estimators $var(\hat{\beta_g})$ can be derived from the model above using :

$$\beta_g = C^T \alpha_g$$

- Two assumptions about the underlying distributions is made here :
  - The contrast estimators $\hat{\beta_g}$ are normally distributed
  - The residual variances s_g^2 follow a scaled chi-square distribution

Linear Modelling

- At this point, an ordinary t statistic can be derived for the contract of interest β_{gj} being the j-th contrast for the g-th gene through the contrast estimators $\hat{\beta}_g$ and its variance estimators
- The null hypothesis $H_0 : B_{gj} = 0$ can be tested
- This process of modelling is a gene wise model fitting ignoring the parallel structure of the dependent gene expression
    - A hierarchical Bayes model can now be set up to take advantage of such information in the assessment for DE genes

Linear Modelling

- Under the assumptions that we have made about the data, the ordinary t-statistic can be calculated as :

$$t = \frac{B_{gj}}{s_g / \sqrt{v_{gj}}}$$

Hierarchical Bayes Model

- Given the large number of gene-wise linear model fits needed in a microarray experiment, it would be advantageous to make use of the parallel structure of the data
- To make use of this parallel structure, the same model is fitted to each gene
    - The key is to describe how the unknown coefficients Bgj and unknown variances σ^2 vary across genes
- In order to describe how these coefficients and variancances vary across genes, prior distributions for these sets of parameters are assumed

Hierarchical Bayes Model

- The prior information assumes that σ_g^2 is equivalent to a prior estimator s_0^2 with d_0 degrees of freedom:

$$\frac{1}{\sigma_g^2} \sim \frac{1}{d_0 s_0^2} \chi^2_{d_0}$$

- This describes how the variances are expected to vary across genes

Hierarchical Bayes Model

- For any given *j* , we assume that B_{gj} is non zero with known probability

$$P(B_{gj} \neq 0) = p_j$$

- Where p_j is just the expected proportion of differentially expressed genes
- For those which are non zero, prior information on the coefficient is assumed equivalent to a prior observation equal to zero with unscaled variance v_{0j}

$$\beta_{gj} \mid \sigma_g^2, \beta_{gj} \neq 0 \sim N(0, v_{0j}\sigma_g^2)$$

- This describes the expected distribution of the contrast for genes which are differentially expressed

Hierarchical Bayes Model

- Under the previously described hierarchical model, the posterior mean of σ_g^{-2} given s_g^2 is

$$\tilde{s}_g^2 = \frac{d_0 s_0^2 - d_g s_g^2}{d_0 + d_g}$$

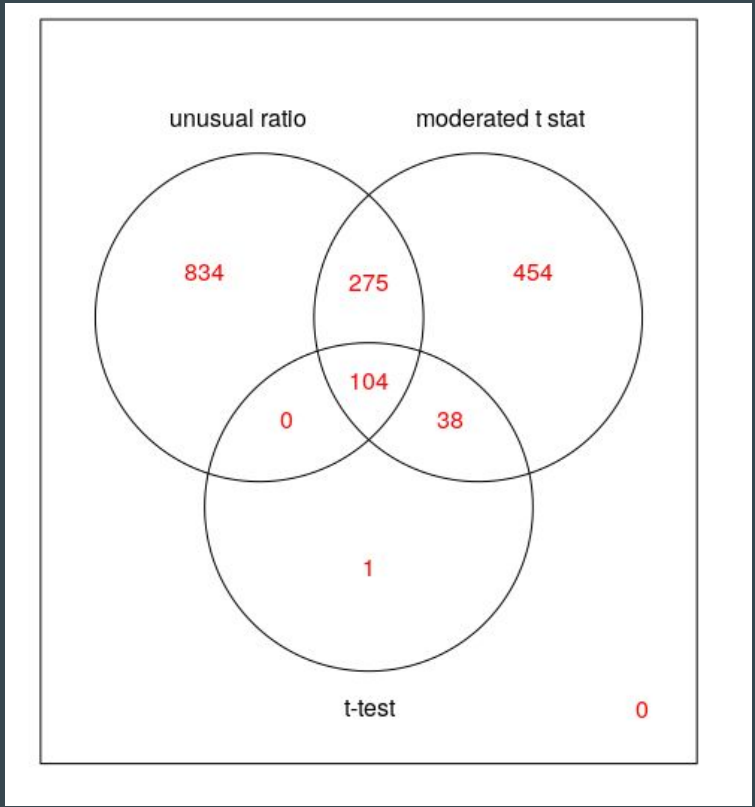
- This newly derived variance derived from our empirical bayes model can now be substituted into our t-statistic to form a more robust model

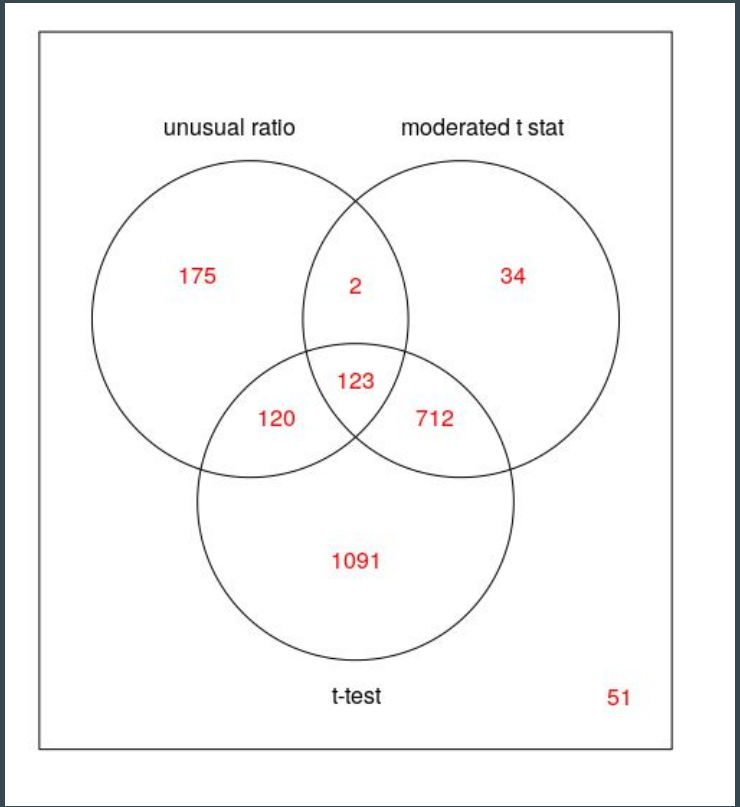
Hierarchical Bayes Model

- This model can be generated through specifying prior distributions for the unknown parameters B_{gj} and σ_g^2 in the previously described linear model
- The meta-parameters introduced in the prior distributions can be estimated from the data through an empirical Bayesian process
- The posterior residual standard deviation $\hat{s_g^2}$ can be derived from the above models, and the moderated t statistic can be defined as :

$$\tilde{t} = \frac{\hat{B_{gj}}}{\tilde{s_g}/\sqrt{v_{gj}}}$$

Results


unusual ratio
moderated t stat
834
275
454
104
0
38
1
t-test
0


unusual ratio
moderated t stat
175
2
34
123
120
712
1091
t-test
51

Discussion

- We have seen that the moderated *t* statistic approach produces more robust results than simpler non-hypothesis testing methods and hypothesis driven methods alike
- The moderated t-statistic approach is much more transparent, and better performing than SAM as it is a parametric approach with more power
- The moderated t-statistic draws power from estimating the global mean for the standard deviation and contrast parameters

References

[1] G. Smyth et al. Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, 3(1):1027, 2004.

[2] “Introduction to Multi-Level Modeling.” YouTube, Duke, 6 Feb. 2017, www.youtube.com/watch?v=m4fx_mzlBQI.

[3] Drăghici Sorin. Statistics and Data Analysis for Microarrays: Using R and Bioconductor. Chapman and Hall, 2012.

Selecting DE Genes: Moderated *t*-Statistic

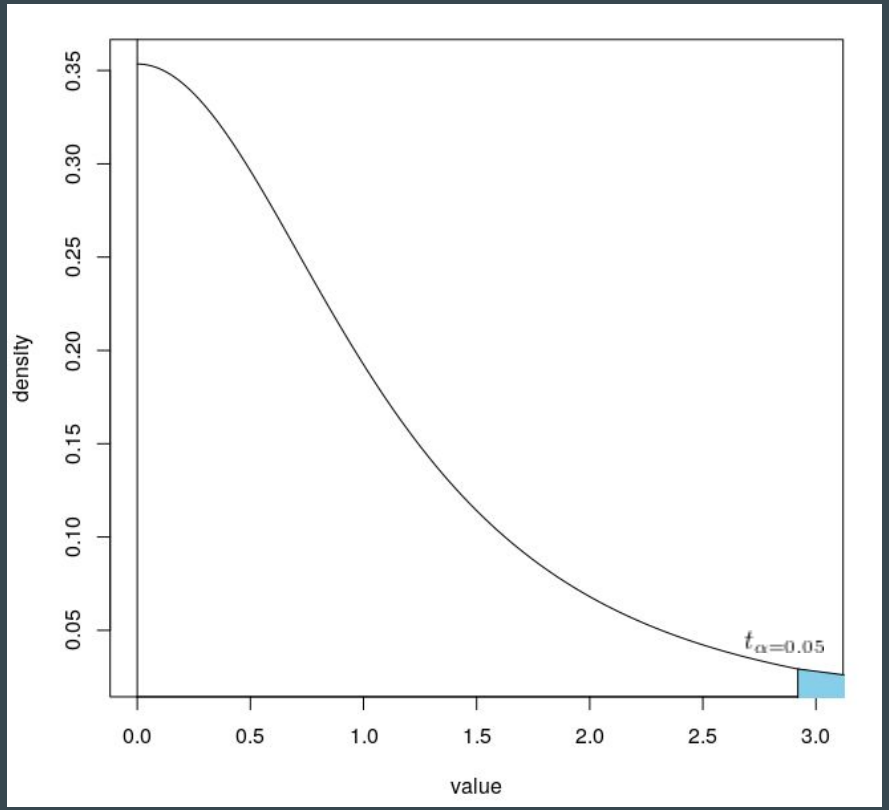
by Jake Sauter

Background: *t*-Statistic

- We have previously discussed the common hypothesis test known as the *t*-Test
- The statistic for this test is

$$t = \frac{\hat{B}_{gj}}{\hat{\sigma}/\sqrt{n}}$$

- g $j = 1$
- g $j = 2$
- $\cdots$
- g $j = n$



Background: Empirical Bayes Methods

- Empirical Bayes Methods are **procedures for statistical inference** in which the prior distribution is estimated from the data
- This family of methods is an approach to setting hyperparameters of known distribution to best fit the data

Background: General Linear Model

- For the General Linear Model (GLM) we assume that the observations Y_i can be modelled by a constant followed by linear scaling factors of various variables, plus an error rate for the observed sample

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_p X_{ip} + \epsilon_i$$

Background: Hierarchical Models

- A Hierarchical Linear Model or a Multilevel Model is statistical model of parameters that vary at more than on level
  - Generally, individual analysis and group analysis parameters share a relationship

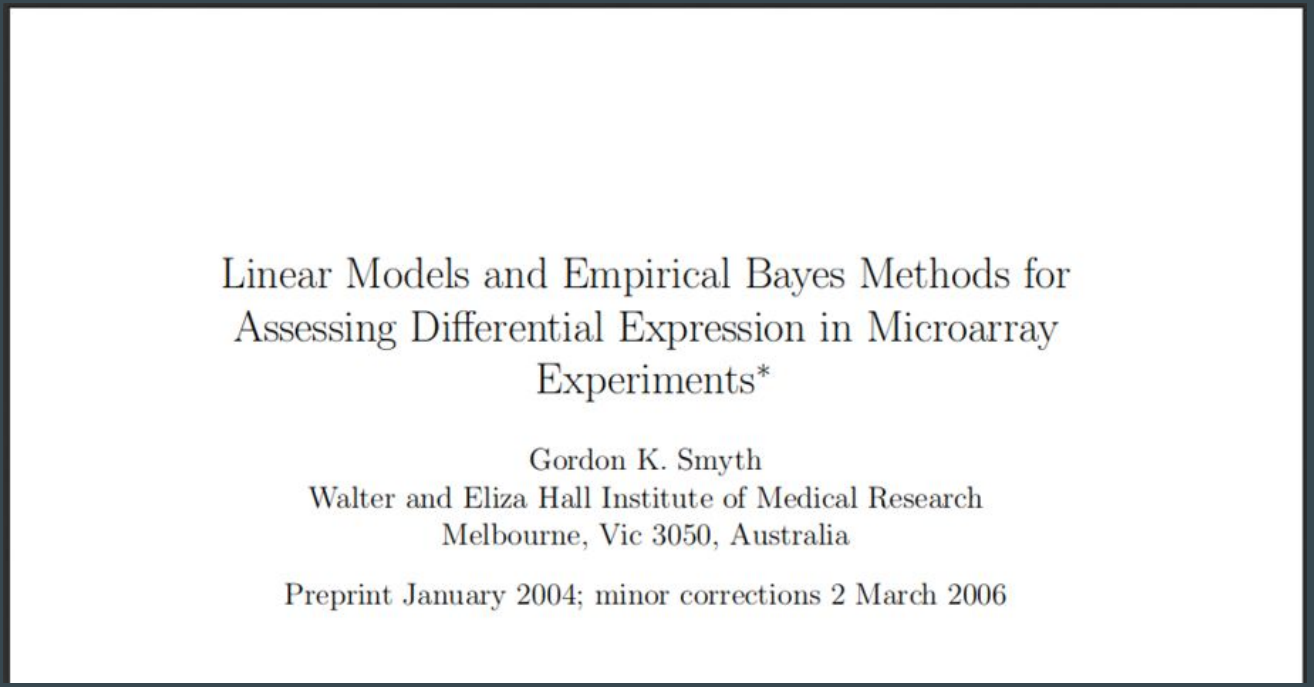
$$\textbf{[Level 1]} \quad Y_{ij} = \beta_{0j} + \beta X_{ij} + r_{ij}$$

$$\downarrow$$

$$\textbf{[Level 2]} \quad \beta_{0j} = \gamma_{00} + \gamma_{01} W_j + v_{0j}$$

[2]

Origin of Moderated *t* Statistic



Linear Models and Empirical Bayes Methods for Assessing Differential Expression in Microarray Experiments*

Gordon K. Smyth
Walter and Eliza Hall Institute of Medical Research
Melbourne, Vic 3050, Australia

Preprint January 2004; minor corrections 2 March 2006

[1]

Linear Model Setup

- Assume that we have n microarrays with an expression vector :

$$y_g^T = (y_{g1}, \cdots, y_{g2})$$

being a vector of log-ratios or log intensities

- The general linear model is assumed as :

$$E(y_g) = X\alpha_g \text{ with } var(y_g) = W_g\sigma_g^2$$

where X is a design matrix, α_g is a coefficient vector, and W_g is a known weight matrix

Linear Model Setup

- Arbitrary contrasts of biological interest β_g can be extracted from the coefficient vector α_g :

$$\beta_g = C^T \alpha_g$$

- This is done with the contrast matrix C

$$C = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

Linear Model Estimation

- With this linear model setup **for each gene**, the model is fit and generates estimators $\hat{\alpha_g}$ of α_g, s^2 of σ^2, and $var(\hat{\alpha_g})$
- The estimators of contrast $\hat{\beta_g}$ and its variance estimators $var(\hat{\beta_g})$ can be derived from the model above using :

$$\beta_g = C^T \alpha_g$$

- Two assumptions about the underlying distributions is made here :
  - The contrast estimators $\hat{\beta_g}$ are normally distributed
  - The residual variances s_g^2 follow a scaled chi-square distribution

Linear Modelling

- At this point, an ordinary t statistic can be derived for the contract of interest β_{gj} being the j-th contrast for the g-th gene through the contrast estimators $\hat{\beta}_g$ and its variance estimators
- The null hypothesis $H_0 : B_{gj} = 0$ can be tested
- This process of modelling is a gene wise model fitting ignoring the parallel structure of the dependent gene expression
    - A hierarchical Bayes model can now be set up to take advantage of such information in the assessment for DE genes

Linear Modelling

- Under the assumptions that we have made about the data, the ordinary t-statistic can be calculated as :

$$t = \frac{B_{gj}}{s_g / \sqrt{v_{gj}}}$$

Hierarchical Bayes Model

- Given the large number of gene-wise linear model fits needed in a microarray experiment, it would be advantageous to make use of the parallel structure of the data
- To make use of this parallel structure, the same model is fitted to each gene
    - The key is to describe how the unknown coefficients Bgj and unknown variances σ^2 vary across genes
- In order to describe how these coefficients and variancances vary across genes, prior distributions for these sets of parameters are assumed

Hierarchical Bayes Model

- The prior information assumes that σ_g^2 is equivalent to a prior estimator s_0^2 with d_0 degrees of freedom:

$$\frac{1}{\sigma_g^2} \sim \frac{1}{d_0 s_0^2} \chi^2_{d_0}$$

- This describes how the variances are expected to vary across genes

Hierarchical Bayes Model

- For any given *j* , we assume that B_{gj} is non zero with known probability

$$P(B_{gj} \neq 0) = p_j$$

- Where p_j is just the expected proportion of differentially expressed genes
- For those which are non zero, prior information on the coefficient is assumed equivalent to a prior observation equal to zero with unscaled variance v_{0j}

$$\beta_{gj} \mid \sigma_g^2, \beta_{gj} \neq 0 \sim N(0, v_{0j}\sigma_g^2)$$

- This describes the expected distribution of the contrast for genes which are differentially expressed

Hierarchical Bayes Model

- Under the previously described hierarchical model, the posterior mean of σ_g^{-2} given s_g^2 is

$$\tilde{s}_g^2 = \frac{d_0 s_0^2 - d_g s_g^2}{d_0 + d_g}$$

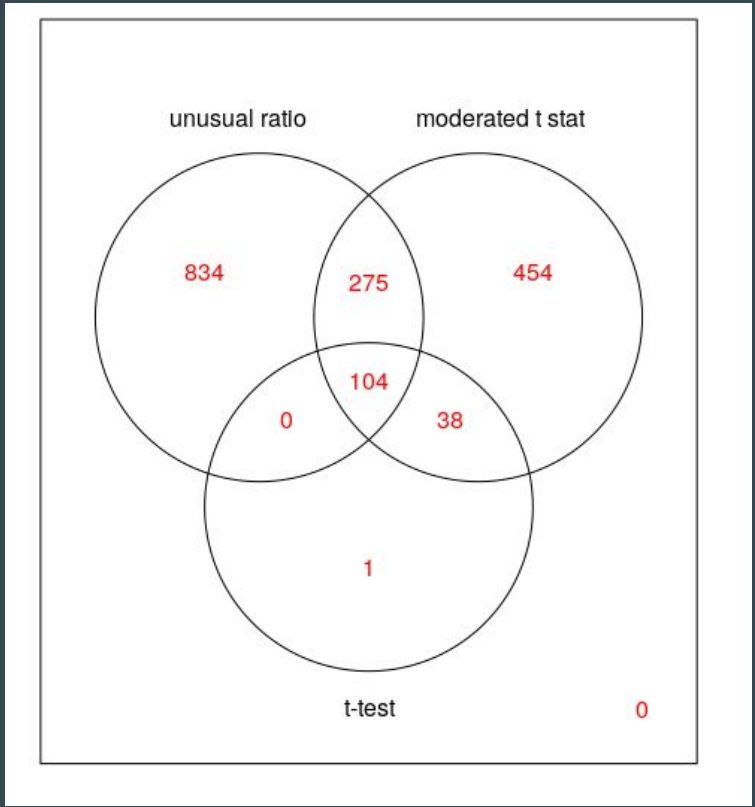
- This newly derived variance derived from our empirical bayes model can now be substituted into our t-statistic to form a more robust model

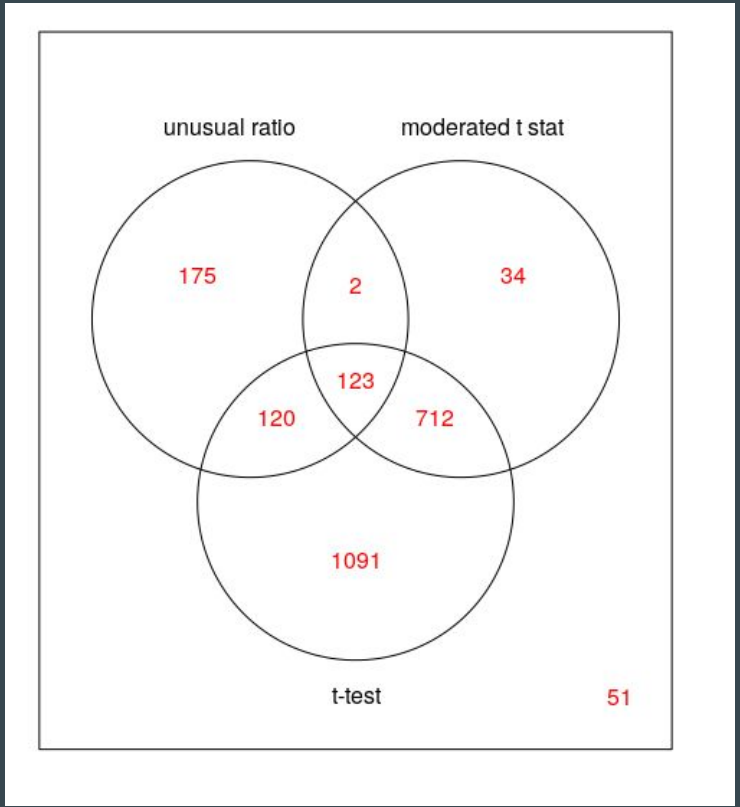
Hierarchical Bayes Model

- This model can be generated through specifying prior distributions for the unknown parameters B_{gj} and σ_g^2 in the previously described linear model
- The meta-parameters introduced in the prior distributions can be estimated from the data through an empirical Bayesian process
- The posterior residual standard deviation $\hat{s_g^2}$ can be derived from the above models, and the moderated t statistic can be defined as :

$$\tilde{t} = \frac{\hat{B_{gj}}}{\tilde{s_g}/\sqrt{v_{gj}}}$$

Results


unusual ratio
moderated t stat
834
275
454
104
0
38
1
t-test
0


unusual ratio
moderated t stat
175
2
34
123
120
712
1091
t-test
51

Discussion

- We have seen that the moderated *t* statistic approach produces more robust results than simpler non-hypothesis testing methods and hypothesis driven methods alike
- The moderated t-statistic approach is much more transparent, and better performing than SAM as it is a parametric approach with more power
- The moderated t-statistic draws power from estimating the global mean for the standard deviation and contrast parameters

References

[1] G. Smyth et al. Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, 3(1):1027, 2004.

[2] “Introduction to Multi-Level Modeling.” YouTube, Duke, 6 Feb. 2017, www.youtube.com/watch?v=m4fx_mzlBQI.

[3] Drăghici Sorin. Statistics and Data Analysis for Microarrays: Using R and Bioconductor. Chapman and Hall, 2012.

Machine Learning

by Jake Sauter

Introduction

- Machine learning refers to a set of topics dealing with the creation and evaluation of algorithms that facilitate pattern recognition, classification, and prediction, based on models derived from existing data [4]
- During supervised learning, objects are classified using a set of features. The classes are known in advance and the goal is to build a model that can extrapolate from the labelled examples to successfully classify new samples or predict the next continuous value
- During unsupervised learning, unlabelled data are examined with the goal of discovering similarities between objects

Applications

- A perceptron was employed to define criteria for start sites in *E. coli.* during the early work on the analysis of translation initiation
- Gene expression data has successfully been used to classify patients in different clinical groups and identify new disease groups
- Genetic code allowed for prediction of the protein secondary structure
- Continuous variable prediction with machine learning algorithms have been used to estimate bias in cDNA microarray data

Definitions

- If only a few labelled samples are available in a data set, then semi-supervised learning can be employed in which class labels of known samples are imposed on the most similar samples to them in feature space
- Machine learning problems can be divided into three different categories
  - Class prediction
  - Class comparison
  - Class discovery

Class Prediction

- In a class prediction problem, classes are defined in advance and labelled data is available for each class. Each data point is usually a vector of values for a number of features

$$example_{ci} =< val(feature_1), val(feature_2), \cdots, val(feature_n) >$$

- The goal of a class prediction problem is to build a classifier which will be able to assign a previously unseen input vector and correctly assign it to the class it belongs to

Class Comparison

- In a class comparison problem, the classes are still predefined such as in a class prediction problem, however now the goal is to find **all** the features that distinguish the classes
- Class comparison problems serve the role of classical discriminant analysis

Class Discovery

- In a class discovery problem, the classes are not known in advance
  - Input vectors are unlabelled
- The goal of the class discovery problem is to identify the input vectors that share certain features, essentially clustering them

Relatedness of Problem Type

- Class prediction problems require a feature selection step, in which useful discerning features of the input vectors are selected to be used in the creation of the classifier
    - This is very similar to the class comparison task however is to only to identify the features needed to successfully discern the classes
    - Ex. The only feature needed to discern Kubota tractors from John Deere tractors is the color, while if one was interested in comparing the two types of tractors before a purchase many more factors would be taken into consideration [4]

Relatedness of Problem Type

- The class discovery task is very different from both previous problem types, as the classes are not known in advance, and in most cases different classes or clusters can be obtained from the same data by applying different methods
- It is important that the type of problem is identified in advance of exploring solutions, as solutions to the different problem types invoke very different machinery

Supervised Learning

- Assume that we wish to classify a collection of $i = 1, \ldots, n$ objects into K predefined classes
    - For example distinguishing different types of tumors based on gene expression values
- A classifier $C(x)$ may be viewed as K discriminant functions $g_c(x)$ such that the object with feature vector x will be assigned to class c for which $g_c(x)$ is maximized over class labels c in $\{1, \ldots, K\}$
- The feature space X is thus partitioned by the classifier $C(x)$ into K disjoint subsets

Supervised Learning

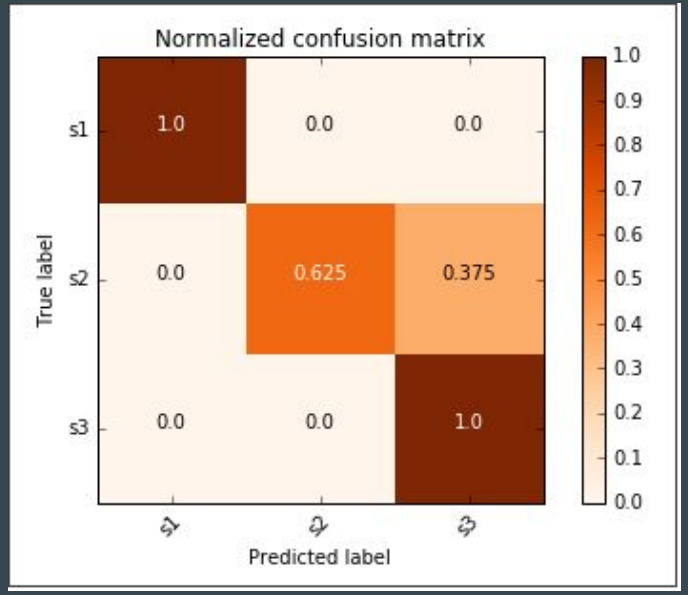
- Two main approaches to the identification of the discriminant functions $g_c(x)$ are
  - Assume knowledge of the underlying class conditional probability density functions and assign $g_c(x) = f(p(x \mid y = c))$ where f is a monotonically increasing function such as ln
  - Class boundaries can also be directly estimated without explicitly calculating the probability density functions
- Intuitively the probability based classifier will classify object *x* to its most probable class, though in practice $p(x \mid y = c)$ is unknown and must be estimated from the training set
  - Both parametric and nonparametric methods for density estimations can be used for this purpose

Supervised Learning

- Concerning probability density based classifiers
  - Parametric approach: **linear and quadratic discriminants**
  - Nonparametric approach: **k-nearest neighbors**
- Concerning classifiers that directly estimate class boundaries
  - **Decision trees, support vector machine, neural networks**

Error Estimation and Validation

- In order to assess how well our classifier is doing, we must first discuss some metrics and how they can be applied
- We have previously discussed a **confusion matrix** for the two class case, though we will show what this would look like for the three class case



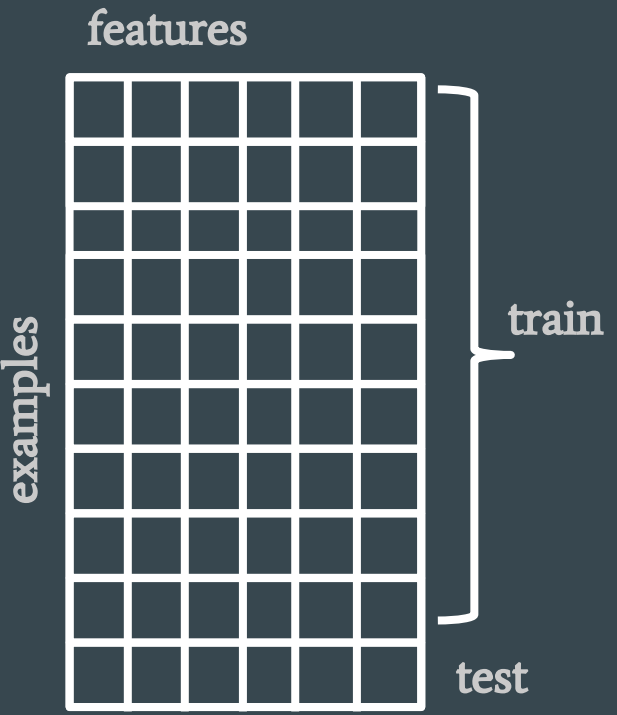
[1]

Error Estimation and Validation

- The goal behind developing a classifier is for use in predicting the true class of new samples, so determining the classification accuracy of the model using the same samples that were trained us will not be informative
- A better method of evaluating the effectiveness of ***C(x)*** would be **hold out** a portion of the given labelled data, and to use this to compose a **testing / validation set**
- This is a very effective testing procedure, though to achieve the best possible classifier we must use all of the provided data

Error Estimation and Validation

- Since we wish to use all possible data for building the model, but also need to evaluate how well the model will do on new data, we can implement **n-fold cross validation**
    - In this method, the data set is divided into **n-folds** with training and testing occurring n times
    - On each run, ***C(x)*** will be trained with n-1 folds and tested on the held out data
    - During each of the n runs, performance measures are calculated and at the end these performance measures are averaged to estimate how well the model will generalize when train with all n folds



Error Estimation and Validation

- In some situations the data may even be too scarce for n-fold cross validation and **leave-one-out (LOO) cross validation** must be implement
  - This is nothing but n fold cross validation with n being equal to the number of data points
  - The error obtained from LOO cross validation has a low bias but may have a large variance

Error Estimation and Validation

- For some problems, the number of samples may be very different between the K classes and the data is said to be **unbalanced**
- In these situations it is best to make sure that the training set is balanced as many classifiers will favor the richer data set
    - If the ratio of examples in each category is different between the training set and the testing set, the training and testing sets are said to be **stratified**

Feature Selection

- A common mistake particularly to this field is to try to build a model with too many features and not enough examples. In general there should be many more examples than features
    - About 10x more examples than features is a generally a desired ratio
- Thus in microarray experiments with thousands of features and tens of examples, **feature selection** is necessary
    - Wrapper methods: training on different sets of features and determining the best set
    - Filter Methods: Test for mutual information using correlation coefficient or statistical significance tests

Quadratic and Linear Discriminants

- **Quadratic Discriminants** is a standard classification approach applicable to continuous features and assumes that for each class c, x follows a multivariate normal distribution with mean μ_c and variance σ_c
- Using the multivariate normal probability density function and replacing the true class mean and covariance matrices with sample derived estimates (m_c and s_c^2 respectively), the discriminant function can be computed as

$$g_c(x) = -(x - m_c)\hat{\sigma_c}^{-1}(x - m_c)^T - log(|\hat{\sigma}_c|)$$

$$m_c = \frac{1}{n_c}\sum_{i=1}^{n_c} x_i \qquad \hat{\sigma}_c = \frac{1}{n_c}(x_i - m_c)(x_i - m_c)^T$$

Quadratic and Linear Discriminants

- The discriminant functions $g_c(x)$ are monotonically related to the densities $p(x|y = c)$, yielding higher values for larger densities
- The values of the discriminant functions will differ from one class to another only on the basis of the estimation of the class mean and covariance matrix
- After the formation of the discriminant functions, a new object z will be classified to the class for which the discriminant function is largest
- This classification approach produces nonlinear (quadratic) class boundaries, giving the name **quadratic discriminant rule** or **Gaussian classifier**

Quadratic and Linear Discriminants

- An alternative to the quadratic classifier is to assume that the class covariance matrices σ_c for $c = 1, \ldots, K$ are all the same
- In this case, a single pooled covariance matrix is used and is especially useful when the number of samples in each class is too low to produce a reliable estimate
- The resulting classifier uses hyperplanes as class boundaries, endowing the name **normal-based linear discriminant**

Quadratic and Linear Discriminants

- To further cope with situations where the number of features is comparable to the number of samples, a further simplification of the covariance matrix can be used by setting all off-diagonal elements to be 0
    - This method neglects co-variation between features and was found to outperform other types of classifiers on a variety of microarray analyses

K-Nearest Neighbor Classifier

- The KNN classifier can be seen as a nonparametric method of density estimation and makes no assumptions about the underlying distribution of the data besides the continuity of features
- There is no training involved in the KNN classifier at all. When a new object z must be classified, the distance between the new object and all other objects in the training set are calculated
- The samples are then ordered closest to furthest from the new object and the k closest samples are retained

K-Nearest Neighbor Classifier

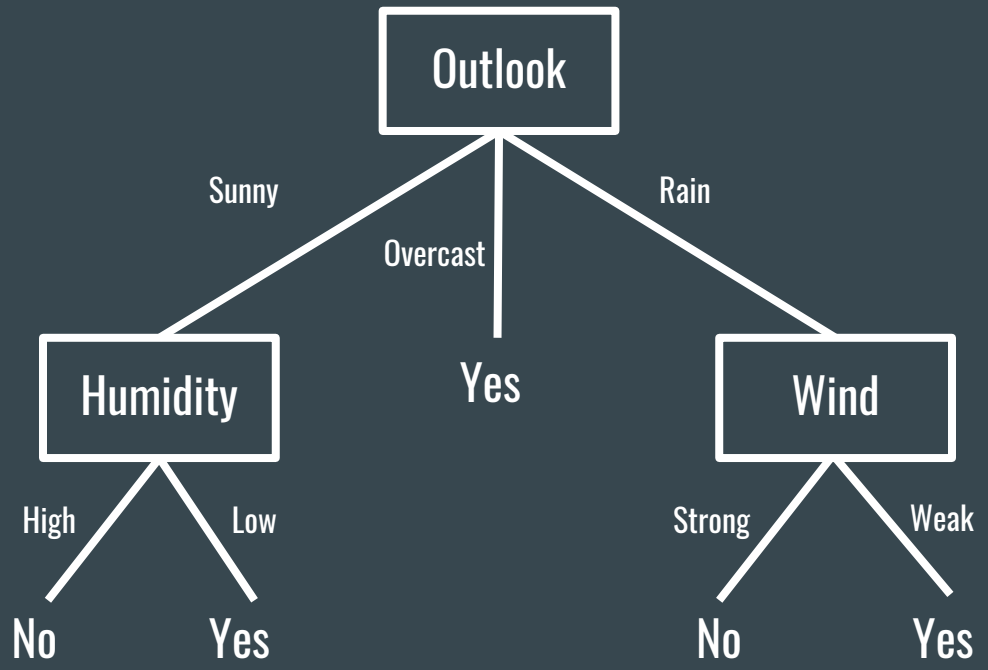
- The number of objects in each distinct class in the k remaining objects is then calculated (n_c), and the class that is most represented is chosen to be the class of the new object
- The KNN discriminant function can be written as

$$g_c(x) = n_c$$

- Note for the KNN classifier all of the computation is shifted to the classification phase, and none is needed for the training phase
    - This is very computationally and memory inefficient as the entire training set must be maintained and a large amount of computations must be performed for every classification

Decision Trees

- A **Decision Tree** is constructed by an iterative selection of individual features that are most salient at each node
- At each level, the input space $\boldsymbol{X}$ is repeatedly split into descendant subsets starting with $\boldsymbol{X}$ itself



Decision Trees

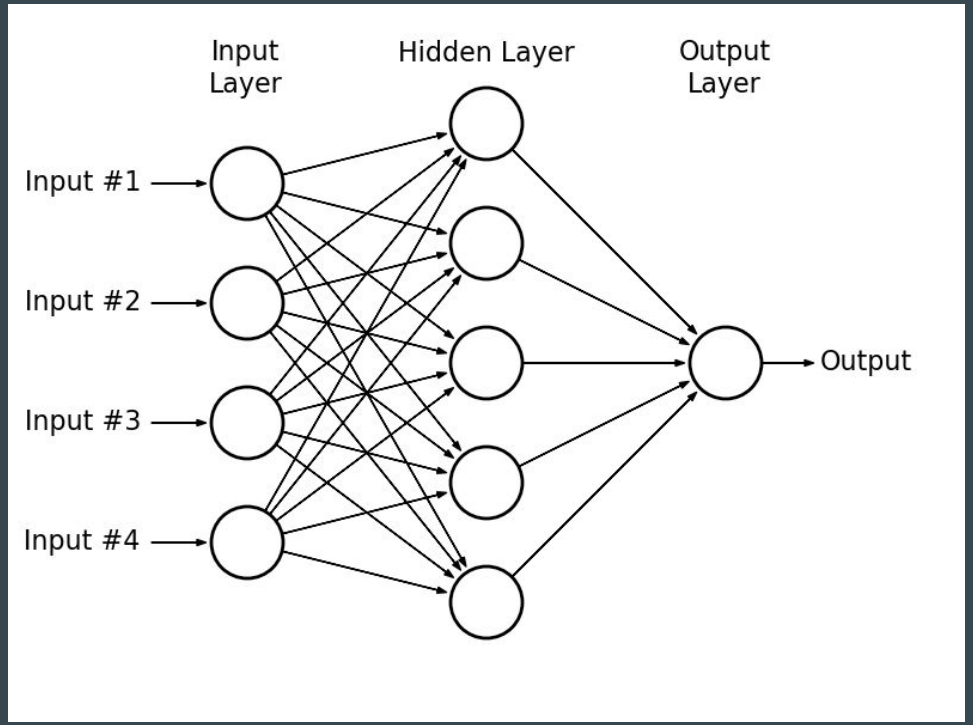
- Normally decision trees are constructed top-down, beginning at the root node and successively partitioning the feature space. This partitioning involves 3 main steps
  - Selecting a splitting rule for each internal node. This involves determining the feature together with the threshold value to split at
  - Determining which nodes are terminal nodes (this means that for each node we must determine to continue to split or to stop and assign a class label)
  - Assign a class label to terminal nodes by minimizing the estimated error rate

Decision Trees

- The most common implementation of decision trees is the binary implementation
  - A single feature is used for each node, resulting in decision boundaries that are parallel to features axes
  - Even though this implementation is suboptimal, they are extremely easy to interpret the set of rules leading to the chosen class label
- The creation of decision trees can be very computationally expensive, as finding the threshold value that creates the best split requires testing many different thresholds for a single feature
  - When performing this for all features in a high dimensional space, the number of computations is very large

Neural Networks

- The most common architecture used in classification problems is a three layered structure of **nodes** in which the signals are propagated from the **input layer** to the **output layer** via the **hidden layer**



[2]

Neural Networks

- The **hidden layer** is called "hidden" because it has no connections outside of the model
- Each hidden unit weights differently all output of the input layer, adds a bias term, and transforms the result using a nonlinear function
    - A common nonlinear function used is the logistic sigmoid function

$$\sigma(z) = \frac{1}{1 + exp(z)}$$

Neural Networks

- Similarly to the hidden layer, the **output layer** processes the output of the hidden layer
- A simple architecture uses one output unit for each class. The discriminant function implemented by the *k*-th output unit of such a network can be written as

$$g_k(x) = \sigma[\sum_j \alpha_{j,k}\sigma(\sum_i x_i w_{i,j} + b_j^h) + b_k^0]$$

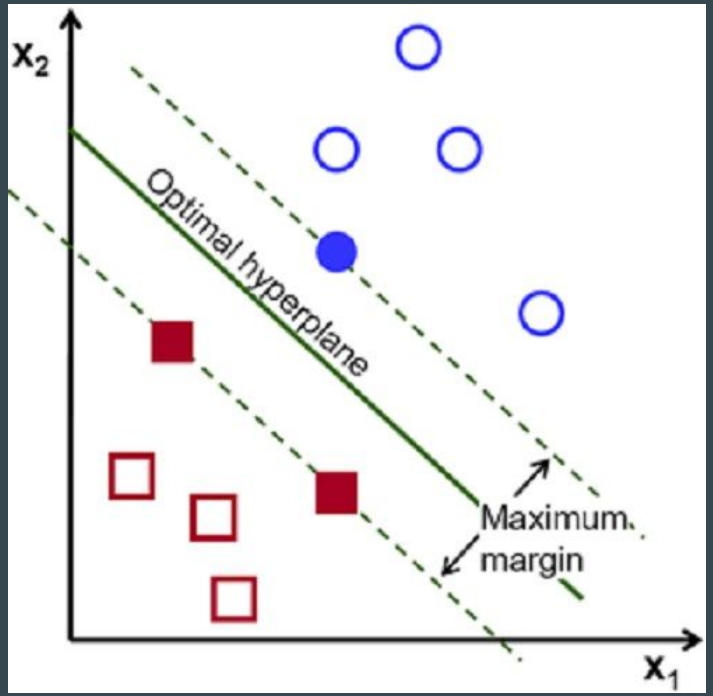
where $\alpha_{j,k}$ is the weight from the j-th hidden unti to the kth output node, b_j^k is the bias term of the h-th hidden unit, b_k^0 is the bias term of the k-th output unit

Neural Networks

- The error of the neural network on the training set can be computed as the sum of the error over all samples
- When a sample belongs to class k, it is desired that the output unit fires 1, while all other units fire 0
- Training is usually performed via **back-propagation** in which the weights are adjusted by the error vector of the network times the partial derivative of the weight with respect to the value of the output nodes

Support Vector Machines

- A classification problem is said to be **linearly separable** if it can be separated by a line (2d) plane (3d) or hyperplane(4d+)
- If a classification boundaries can be made this way, a natural question may be if all of the ways of forming the classification boundary are equally good, or if there is an optimal class boundary
- **Support Vector Machines** find the decision boundary that achieves the maximum margin between classes



[3]

Support Vector Machines

- In the case of an m-class classification problem, the procedure of finding the optimal decision boundary between two classes is repeated several times
- Using the labelled points, the SVM finds a function $f : R^n \rightarrow R^n$ such that for a given input x, $f(x) \geq 0$ if x belongs to the class denotes by 1, otherwise $f(x) < 0$
- The equation $f(x) = 0$ defines a hyperplane that is used for classification of unknown samples
- When the input consists of linearly separable classes, it is easy to find such a hyperplane

$$f(x) = \langle w \cdot x \rangle + b = \sum_{k=1}^{n} w_k x_k + b$$

Support Vector Machines

$$f(x) = \langle w \cdot x \rangle + b = \sum_{k=1}^{n} w_k x_k + b$$

where ***w*** is the normal vector of the hyperplane defined by ***f(x)=0*** and ***b*** is the offset from the origin. If ***b=0*** , the hyperplane passes through the origin

- Finding such a function is more complex in the case of non linearly separable data

  and will be covered in future research

Application

- I have begun to apply these concepts to the data from Golub (1999)
- For preprocessing and feature selection :
  - Use SAM to select around the top 500 differentially expressed genes
  - Perform PCA on these top DE genes
  - The top principal components can be used as features

Application: KNN

	k = 3	k = 5	k = 10
all genes	knn_output ALL AML ALL 20 0 AML 4 10	knn_output ALL AML ALL 19 1 AML 5 9	knn_output ALL AML ALL 20 0 AML 8 6
de genes	knn_output ALL AML ALL 19 1 AML 1 13	knn_output ALL AML ALL 19 1 AML 2 12	knn_output ALL AML ALL 20 0 AML 5 9
PCs	knn_output ALL AML ALL 20 0 AML 0 14	knn_output ALL AML ALL 20 0 AML 1 13	knn_output ALL AML ALL 20 0 AML 3 11
top 5 PCs	knn_output ALL AML ALL 19 1 AML 2 12	knn_output ALL AML ALL 19 1 AML 1 13	knn_output ALL AML ALL 19 1 AML 2 12

References

[1] unutbu. “How Can I Make My Confusion Matrix Plot Only 1 Decimal, in Python?” Stack Overflow, 2016, stackoverflow.com/questions/40264763/how-can-i-make-my-confusion-matrix-plot-only-1-decimal-in-python.

[2] Minnaar, Alex. Deep Learning Basics: Neural Networks, Backpropagation and Stochastic Gradient Descent, Machine Learning at University College London, 2015, alexminnaar.com/implementing-the-distbelief-deep-neural-network-training-framework-with-akka.html.

[3] Eliot, Lance. “Support Vector Machines (SVM) for AI Self-Driving Cars.” AI Trends, 19 Jan. 2018, aitrends.com/ai-insider/support-vector-machines-svm-ai-self-driving-cars/.

[4] Drăghici Sorin. Statistics and Data Analysis for Microarrays: Using R and Bioconductor. Chapman and Hall, 2012.

Machine Learning Continued

...

by Jake Sauter

Logistic Regression

- Logistic Regression is one of the most popular and widely used classification algorithms
- It can be applied in situations in which the desired prediction is a discrete class (spam / not spam ; malignant / benign)
- It is called logistic regression as it makes use of the **logistic function** also known as the **sigmoid function**

Logistic Regression

- Logistic Regression must be used in these situations where we would like discrete class output, as if we used the linear regression prediction model our class bounds can be exceeded
- The hypothesis function $h_\theta(x)$, where θ is the vector of parameters fit to the model of the training data is usually (in linear regression) described as

$$h_\theta(x) = \theta^T x$$

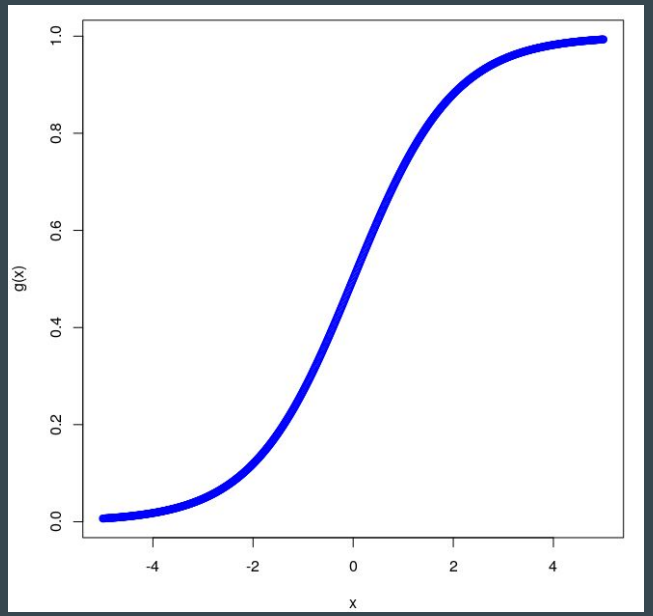
Logistic Regression

- A slight difference is implemented in logistic regression, letting the hypothesis function be

$$h_\theta(x) = g(\theta^T x)$$

where ***g(x)*** is the sigmoid function defined as

$$g(x) = \frac{1}{1 + e^{-z}}$$



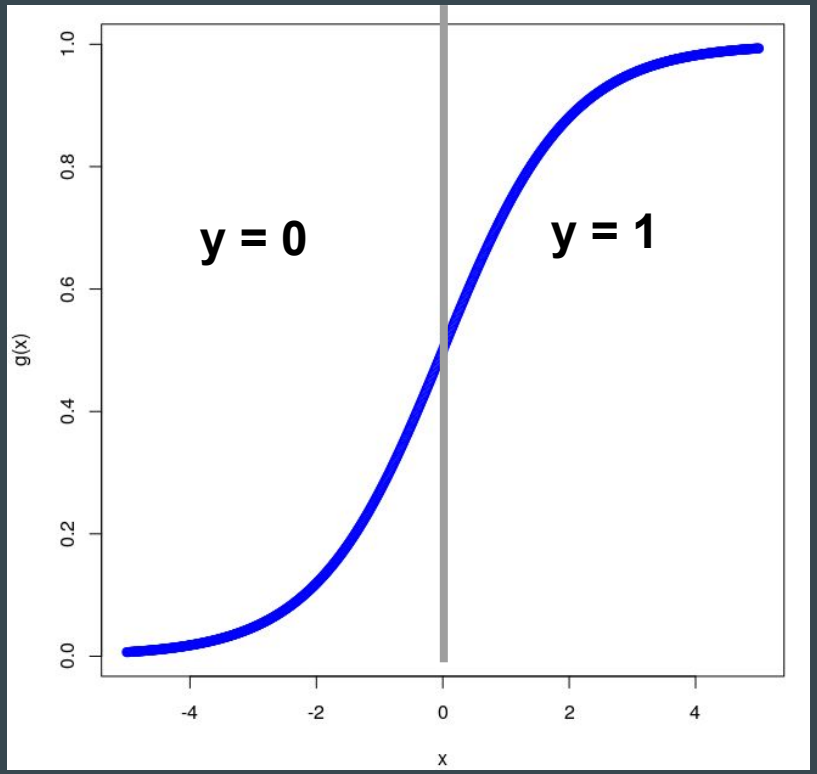
Logistic Regression

- The interpretation of the output of $h_\theta(x)$ is the probability of the sample with feature vector x belonging to the positive class $(y = 1)$

$$h_\theta(x) = p(y = 1 \mid x; \theta)$$

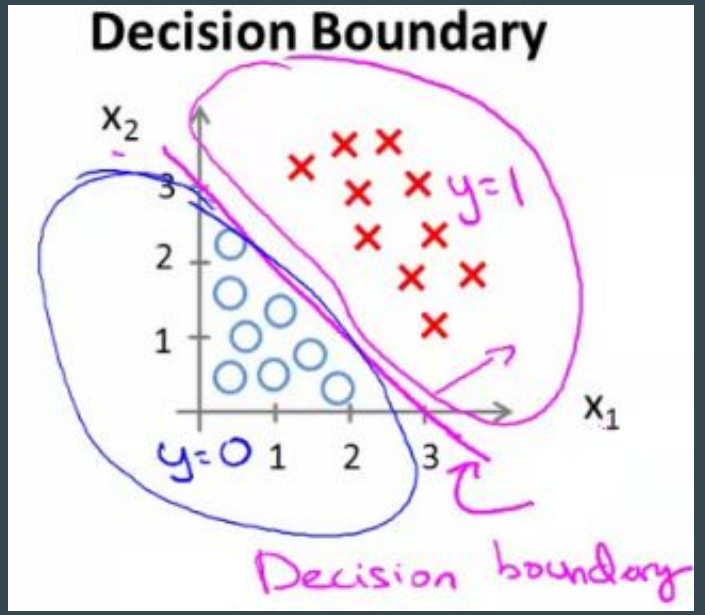
- The **decision boundary** of the hypothesis can be intuitively derived
    - We need to decide when we will predict $y = 1$ and $y = 0$, an intuitive decision could be assigning y to 1 if $h_\theta(x) > 0.5$ and y to 0 if $h_\theta(x) \leq 0.5$
    - Since $g(x) = 0.5$ at $x = 0$, these conditions can be written as $y = 1$ when $\theta^T(x) \geq 0$ and $y = 0$ when $\theta^T(x) \leq 0$

Logistic Regression Visualized



Logistic Regression: Linear Decision Boundaries

- Simple implementations of logistic regression implement a **linear decision boundary**, which in terms of our parameter vector and hypothesis simply means that the parameter coefficients for each given feature are linear

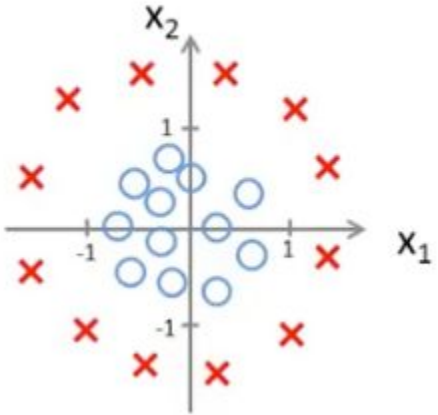


[2]

Logistic Regression: Non-Linear Decision Boundary

- Since in logistic regression we do not limit the form of this feature vector, we may also introduce non-linear terms to form a **non-linear decision boundary**

Non-linear decision boundaries



$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2)$$

[2]

Logistic Regression : Fitting Parameters

- So far logistic regression has seemed to be intuitive, though we have not seen how to actually calculate the parameter vector θ to make the decisions that we have prompted
- Briefly, a **cost function** is a function that provided a **training set** for fitting model parameters, will provide us with how well the model's predictions match the true class of each training example
- We must begin with the **cost function** of linear regression
    - In simple linear regression, we defined the cost function to be

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{2} (h_\theta(x^{(i)}) - y^{(i)})^2$$

Logistic Regression: Cost Function

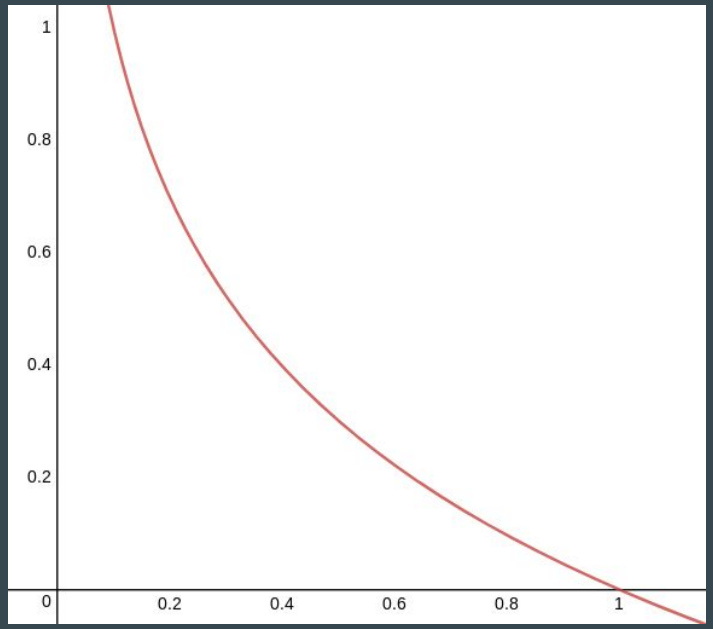
- Though this is not ideal for us as this is a non-convex cost function with respect to our new hypothesis function, meaning that **gradient descent** (our method of finding parameters θ) will not work
- To adapt this cost function into a convex function for logistic regression, we can begin by splitting the cost function into two cases

$$Cost(h_\theta(x), y) = \begin{cases} -log(h_\theta(x)); y = 1 \\ -log(1 - h_\theta(x)); y = 0 \end{cases}$$

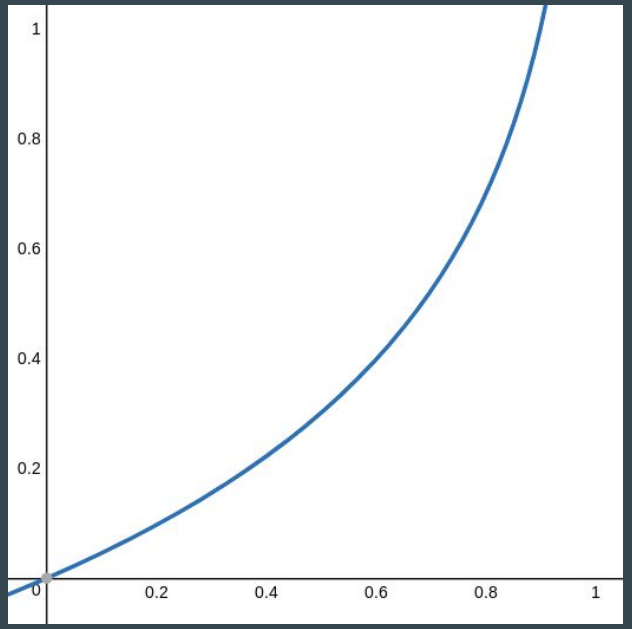
Logistic Regression

- Graphically, these two functions look like

If y=1



If y=0



Logistic Regression: Cost Function

- These two functions can be written in a single form

$$Cost(h_\theta(x), y) = -ylog(h_\theta(x)) - (1 - y)log(1 - h_\theta(x))$$

  as ***y*** strictly takes on the values 0 and 1 in the two possible cases of class membership

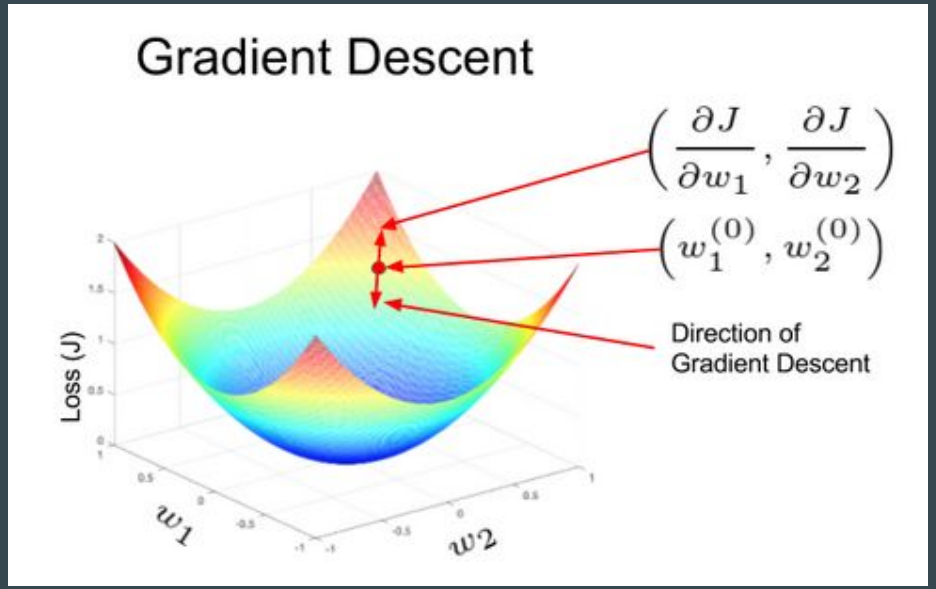
- This cost function can be derived from Maximum Likelihood Estimation, though also intuitively has many desirable properties for a cost function

Logistic Regression: Cost Function

- Properties of the cost function:
  - Convex for the logistic hypothesis function
  - Tends towards infinity as predicted class gets closer to incorrect class
  - Is 0 when predicted class is the correct class

Logistic Regression: Finding θ

- How can we use the cost function that we have obtained to generate the optimal parameters for prediction?
    - Gradient Descent
- **Gradient Descent** is a repetitive process that can be used to find the minimum value of a function, and and which point in the native space this minimum occurs



[1]

Logistic Regression: Gradient Descent

- We can apply gradient descent to find the parameter vector θ in the following way

$$\text{Repeat}\{$$
$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$
$$\}$$

where α is the **learning rate**, which is simply how far in the gradient direction we will step at each iteration. If α is too large, we may overstep the minimum and if α is too small, convergence may take a very long time

Gradient Descent

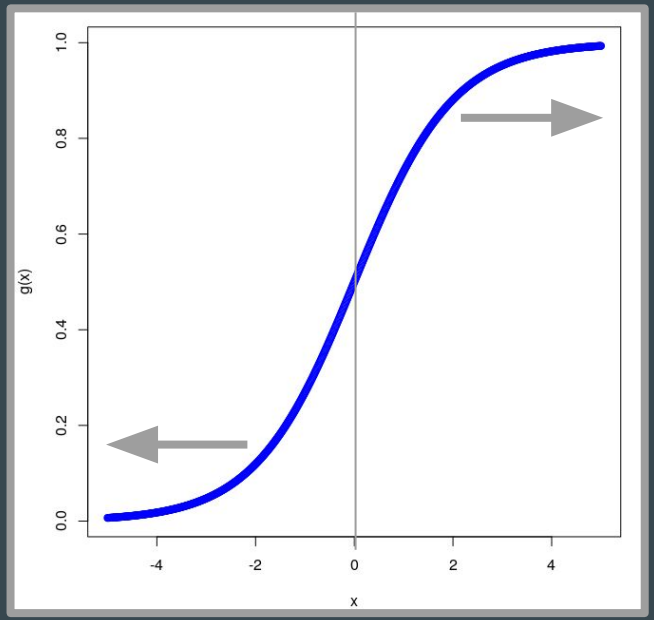
- One may note that this update rule is identical to that of linear regression, with the only difference being the hypothesis function
- Another important note is that advanced optimization techniques exist that can automatically select the learning rate such as
  - Conjugate Gradient
  - BFGS
  - L-BFGS
- One final note is that logistic regression can be used to predict multiclass problems by developing a model for each class and using them as discriminant functions

Support Vector Machines

- Support Vector Machines (or SVMs) can sometimes provide a cleaner and more powerful way of learning complex non-linear functions compared to logistic regression and neural networks
- In reviewing the SVM, we can actually see it as a modification of logistic regression, in which we can estimate the cost function in a simpler way and obtain a larger classification margin

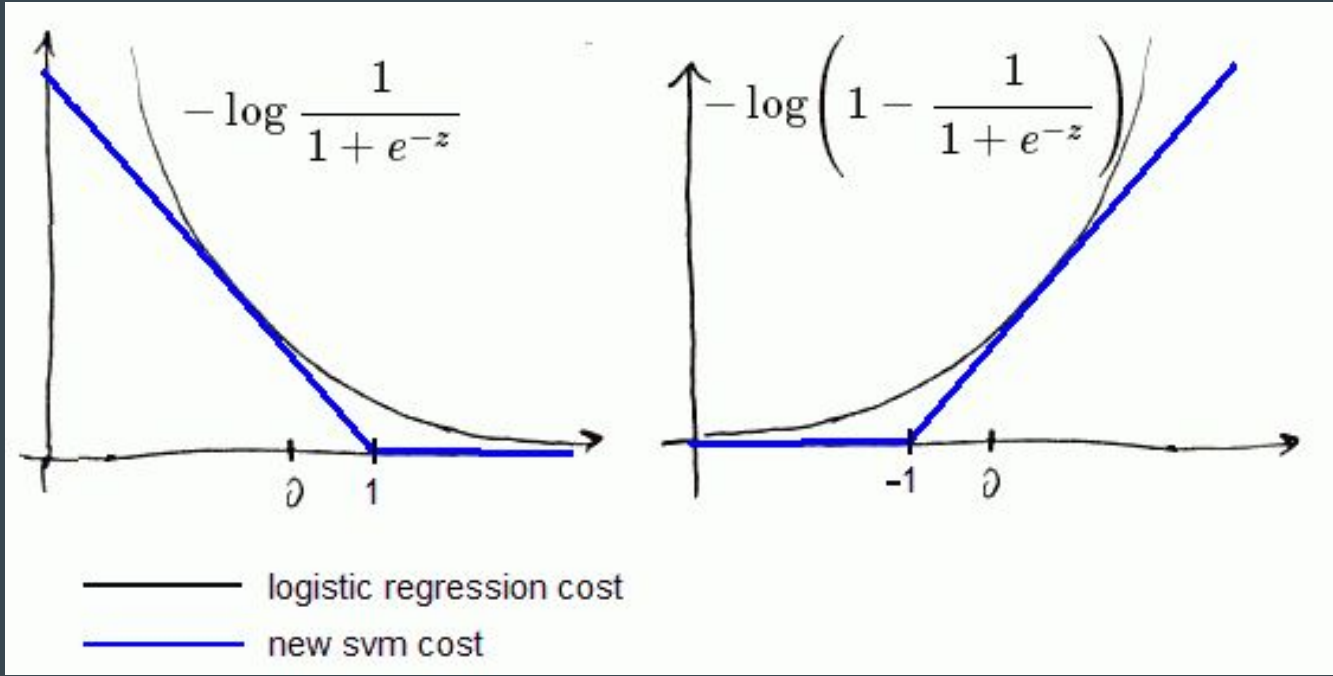
Support Vector Machines

- If we view logistic regression in the light of constructing a large margin classifier, if y=1, we want $h_\theta \simeq 0$, which implies that $\theta^T x >> 0$



$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

Support Vector Machine: Cost Function Estimation



[2]

Support Vector Machine: Cost Functions

- These cost functions add a margin to classification, as the cost is not 0 just when the example is classified correctly, but when it is classified correctly by a large margin (a margin of 1)
- This characteristic is what makes SVMs a **large margin classifier**

Aside: Regularization

- **Regularization** refers to the process of scaling features to avoid overfitting, which can be facilitated by adding a regularization term to the cost function
- This will provide an incentive for lower feature scalings as the magnitude of features will directly influence the cost function
- Often the added regularization term is of the form $\lambda \sum_{i=1}^{m} \theta_j^2$ where λ is the regularization parameter

SVM: Cost function

Logistic Regression cost function:

$$J(\theta) = \frac{1}{m}[\sum_{i=1}^{m} y^{(i)}(-log h_\theta(x^{(i)})) + (1 - y^{(i)})(-log(1 - h_\theta(x^{(i)})))] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

$$J(\theta) = \frac{1}{m}[\sum_{i=1}^{m} y^{(i)} cost_1(\theta^T x^{(i)}) + (1 - y^{(i)})\text{cost}_0(\theta^T x^{(i)})] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

SVM Cost Function

- Usual modifications to this cost function include
  - Reparameterizing the regularization coefficient from A + **λ** B to CA + B
  - dropping all **1/m** terms for simplification

$$J(\theta) = C[\sum_{i=1}^{m} y^{(i)} cost_1(\theta^T x^{(i)}) + (1 - y^{(i)}) \text{cost}_0(\theta^T x^{(i)})] + \frac{1}{2} \sum_{j=1}^{n} \theta_j^2$$

Support Vector Machines: Output Interpretation

- SVMs do not output probabilities, but output predictions

$$h_\theta(x) = \begin{cases} 1 \text{ if } \theta^T x \geq 0 \\ 0 \text{ otherwise} \end{cases}$$

Support Vector Machine: Mathematical Intuition

- So how does this optimization objective lead to our classifier having a large margin?
- First we must modify the cost function

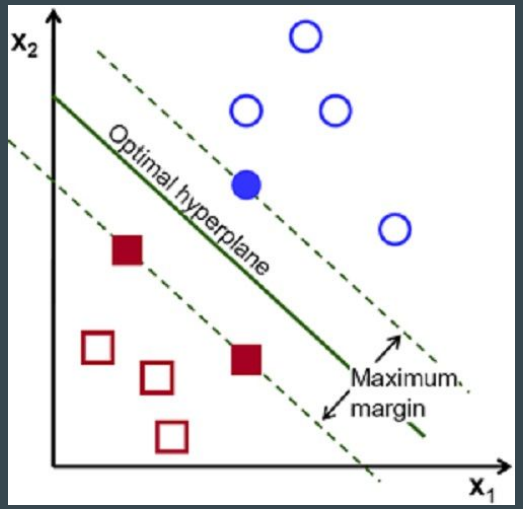
$$J(\theta) = \frac{1}{m}[\sum_{i=1}^{m} y^{(i)} cost_1(\theta^T x^{(i)}) + (1 - y^{(i)}) \text{cost}_0(\theta^T x^{(i)})] + \frac{\lambda}{2m} \sum_{j=1}^{n} \theta_j^2$$

$$\Downarrow$$

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{n} \theta_j^2 \quad s.t. \quad \begin{array}{l} \theta^T x^{(i)} \geq 1 \text{ if } y^{(i)} = 1 \\ \theta^T x^{(i)} \leq -1 \text{ if } y^{(i)} = 0 \end{array}$$

Support Vector Machine: Mathematical Intuition

- Now to obtain some geometric intuition of how this optimization objective translates to the standard image in explanation of SVMs, we will perform a simple conversion



[2]

$$J(\theta) = \frac{1}{2}\sum_{j=1}^{n}\theta_j^2 = \frac{1}{2}(\theta_1^2 + \theta_2^2 + \cdots \theta_n^2) = \frac{1}{2}(\sqrt{\theta_1^2 + \theta_2^2 + \cdots \theta_n^2} = \frac{1}{2}||\theta||^2$$

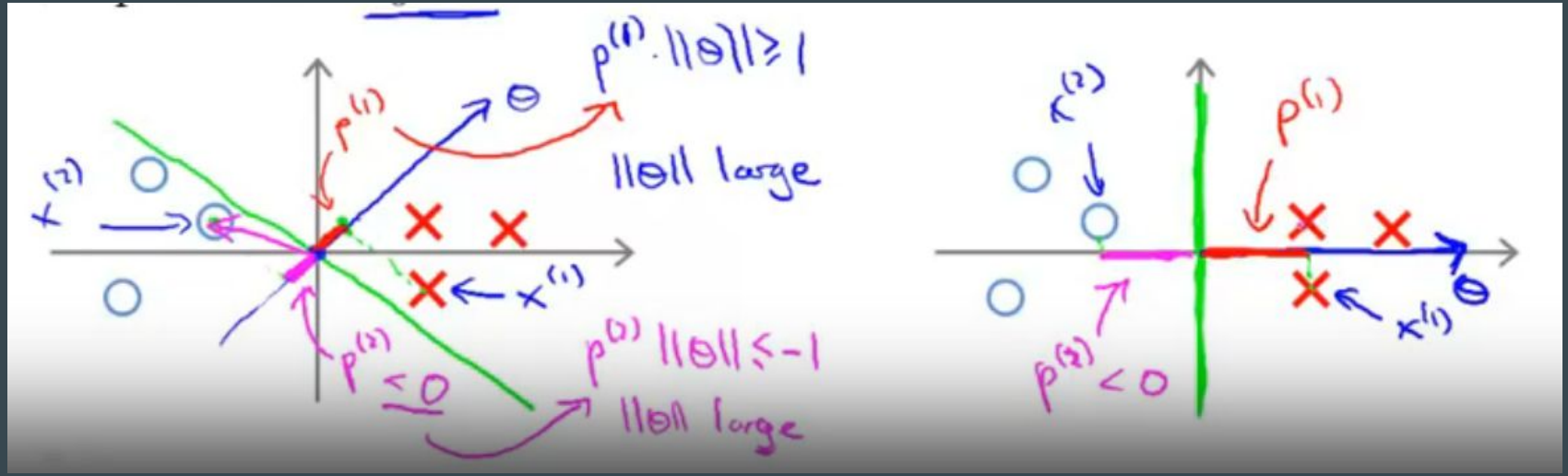
Support Vector Machine: Mathematical Intuition

- We will also make a modification to the restraint conditions that must be met, noting that $\theta^T x^{(i)} = p^{(i)}||\theta||$, we now have

$$J(\theta) = \frac{1}{2}||\theta||^2 \quad \text{s.t.} \quad \begin{aligned} p^{(i)}||\theta|| &\geq 1 \text{ if } y^{(i)} = 1 \\ p^{(i)}||\theta|| &\leq -1 \text{ if } y^{(i)} = 0 \end{aligned}$$

- With this alternate view of the cost function, we can see that the optimization objection is really to minimize the squared norm of the parameter vector $\boldsymbol{\theta}$, which can only be done by maximizing the projection of each training example onto $\boldsymbol{\theta}$

Support Vector Machine: Mathematical Intuition



[2]

Briefly: Random Forests

- **Random forests** have been developed to combat many common issues in decision trees, primarily overfitting
  - They reduce overfitting without substantially increasing error due to bias (which is usually introduced when limiting depth to prevent overfitting)
- To produce a more robust classifier, random forests build many standard decision trees on random subsets of the data, and only use randomly selected features of each of these subsets
  - For final classification of new samples, each tree is allotted a vote on the true class, and the predicted class of the model is the class with the largest overall sum of votes

Applications

- All of the discussed supervised classifiers can be easily implemented in R, with the following results for the first 5 principle components of SAM selected differentially expressed genes with a median fdr of .05

```
             knn_output
test_labels ALL AML
        ALL  19   1
        AML   2  12
```

```
              test_labels
tree_output ALL AML
        ALL  19   2
        AML   1  12
```

```
              test_labels
forest_output ALL AML
        ALL  20   2
        AML   0  12
```

```
             svm_output
test_labels ALL AML
        ALL  19   1
        AML   1  13
```

```
              test_labels
log_output ALL AML
        ALL  19   1
        AML   1  13
```

References

[1] Sharma, Aditya. “Understanding Activation Functions in Deep Learning.” Learn OpenCV, Learnopencv, 30 Oct. 2017, www.learnopencv.com/understanding-activation-functions-in-deep-learning/.

[2] “Machine Learning.” Coursera, Standford University, 2018, www.coursera.org/learn/machine-learning.

[3] Raval, Siraj. “Random Forests - The Math of Intelligence (Week 6).” YouTube, YouTube, 26 July 2017, www.youtube.com/watch?v=QHOazyP-YlM.

Question: Gradient Descent

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$