MIT 18.700/18.701/6.047/6.878/HST.507
Machine Learning in Computational Biology
Computational Biology: Genomes, Networks, Evolution

Lecture 2: Expression Analysis

Machine Learning
AI-ML-Deep Learning
Gene expression analysis
Clustering and Classification
Supervised/unsupervised learning
Bayesian Inference / Naïve Bayes
Discriminative Learning
Kernels, SVM

Prof. Manolis Kellis – manoli@mit.edu – MIT Computer Science & AI Lab

Course at a Glance

Wk	Date	Lec	Topic
	Introduction: Machine Learning, Deep Learning, Generative AI, and the Unification of Biology		
1	Thu-Sep-05	L1	Course Overview, Machine Learning, Deep Learning, Inference, Genome, Proteins, Chemistry, Imaging
	Module 1: Genomics, Epigenomics, Single-Cell, Networks, Circuitry		
2	Tue-Sep-10	L2	Expression Analysis, Clustering/Classification, Gaussian Mixture Models, K-means, Bayesian Inf, Gen-vs-DiscrML
2	Thu-Sep-12	L3	Sequential Data, Alignment, DynProg, Hidden Markov Models, Parsing, Posterior Decoding, HMM architectures
3	Tue-Sep-17	L4	Regulatory Genomics: Motifs, Information, ChIP, Gibbs Sampling, EM, CNNs for Genome Parsing
3	Thu-Sep-19	L5	Epigenomics: Signal Modeling, Peak calling, Chromatin states, 3D structure, Hi-C, Genome Topology
4	Tue-Sep-24	L6	Single-cell genomics, sc-mutli-omics, non-linear embeddings, spatial transcriptomics, next-gen technologies
4	Thu-Sep-26	L7	Regulatory Networks: Graphs, Linear Algebra, PCA, SVD, Dimentionality Reduction, TF-enhancer-gene circuitry
	Module 2: Protein Structure, Protein Language Models, Geometric Deep Learning		
5	Tue-Oct-01	L8	Intro to structural biology
5	Thu-Oct-03	L9	Protein structure and folding: Diffusion models, Cryo-EM, Protein design
6	Tue-Oct-08	L10	Intro to transformers and Large Language Models LLMs
6	Thu-Oct-10	L11	Protein Language Models PLMs and Transfer Learning
7	Tue-Oct-15	-	-- No Class -- Student holiday
7	Thu-Oct-17	L12	DNA language models: Chromatin Structure
	Module 3: Chemistry, Therapeutics, Graph Neural Networks		
8	Tue-Oct-22	L13	Overview of drug development
8	Thu-Oct-24	L14	Intro to small molecules
9	Tue-Oct-29	L15	Representation of small molecules: Graphs, GNNs, Transformers, RDKit
9	Thu-Oct-31	L16	Docking: Small molecule - proteins docking
10	Tue-Nov-05	L17	Disease Association Mapping, genetics, GWAS, linkage analysis, disease circuitry, variant-to-function
10	Thu-Nov-07	L18	Quantitative trait mapping, molecular traits, eQTLs, mediation analysis, iMWAS, multi-modal QTLs
	Module 4: Electronic Health Records, Imaging, Evolution, Metabolism		
11	Tue-Nov-12	L19	Electronic Health Records, AllOfUs, UKBioBank, Medical Genomics, Pop-Scale Cohorts, Multi-Ancestry [not quiz'd]
11	Thu-Nov-14	L20	**-- In-class Quiz**
12	Tue-Nov-19	L21	Imaging methods for biological applications
12	Thu-Nov-21	L22	Comparative genomics, Conservation, Evolutionary signatures, PhyloCSF, RNA structure, Motif BLS2conf
13	Tue-Nov-26	L23	Evolution, Phylogenetics, Phylogenomics, Duplication, RNA world, RNA folding, lincRNAs, RNA modifications, m6A
13	Thu-Nov-28	-	-- No Class -- Thanksgiving Holiday
14	Tue-Dec-03	L24	Modeling metabolism: Flux balance analysis
14	Thu-Dec-05	L25	Measuring metabolism: Metabolomics and Deep Learning
	Final Projects		
15	Tue-Dec-10	L26	**Project Presentations (6-8 mins/team). Report due Fri@11.59p, Slides due Mon@11.59p, Present Live Tue**

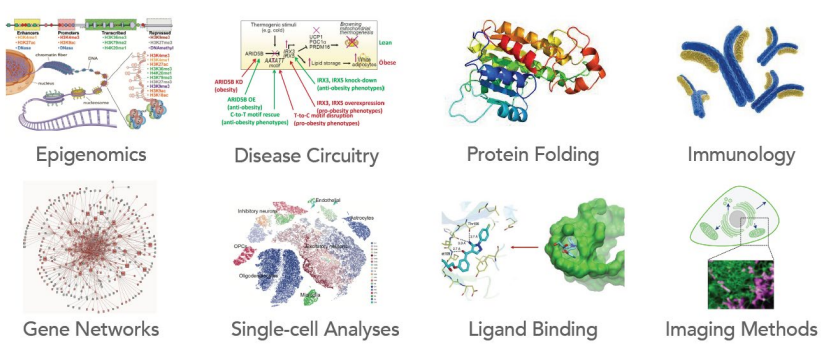
MACHINE LEARNING FOR COMPUTATIONAL BIOLOGY

GENOMES, CIRCUITRY, DISEASE, THERAPEUTICS

6.8700 / 6.8701 / HST.507[J]

NEWLY REVAMPED FOR FALL 2023!

NOW INCLUDES DEEP LEARNING & DRUG DISCOVERY



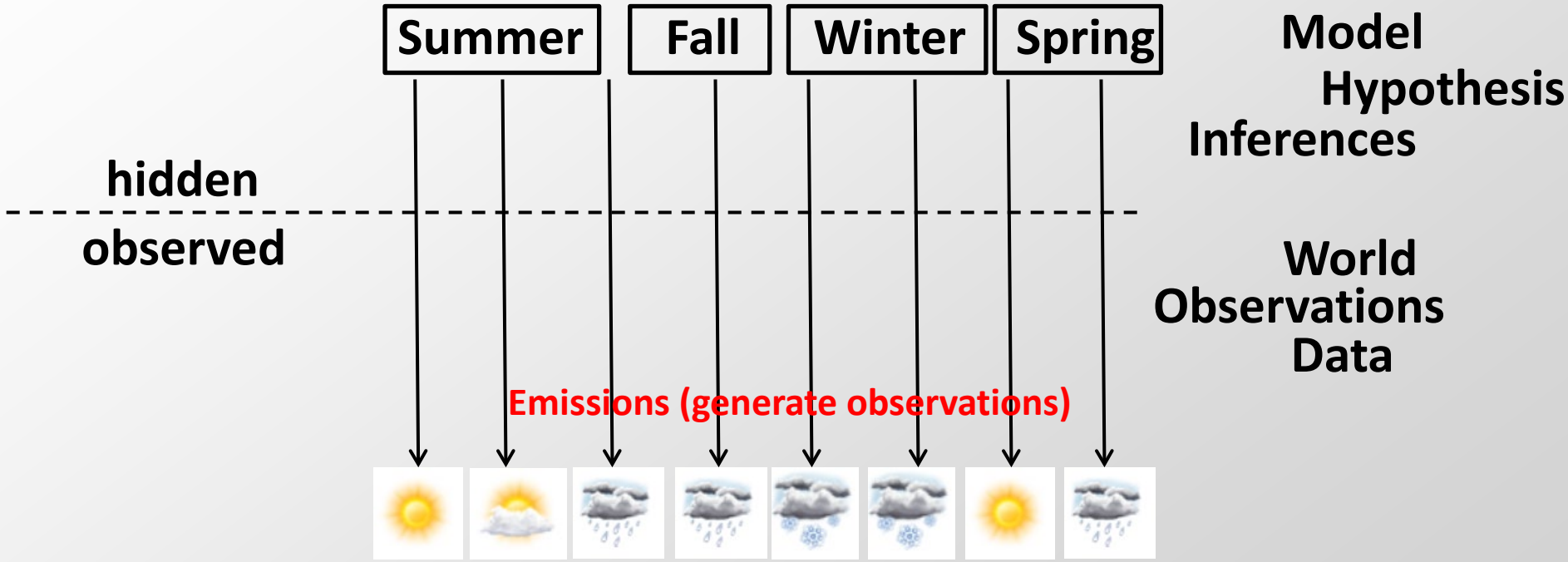
But what is Machine Learning?

Machine Learning is the ability to improve on a task with more training data

- Task T to be performed
  - Classification, Regression, Transcription, Translation, Structured Output, Anomaly Detection, Synthesis, Imputation, Denoising
- Measured by Performance Measure P
- Trained on Experience E (Training Data)

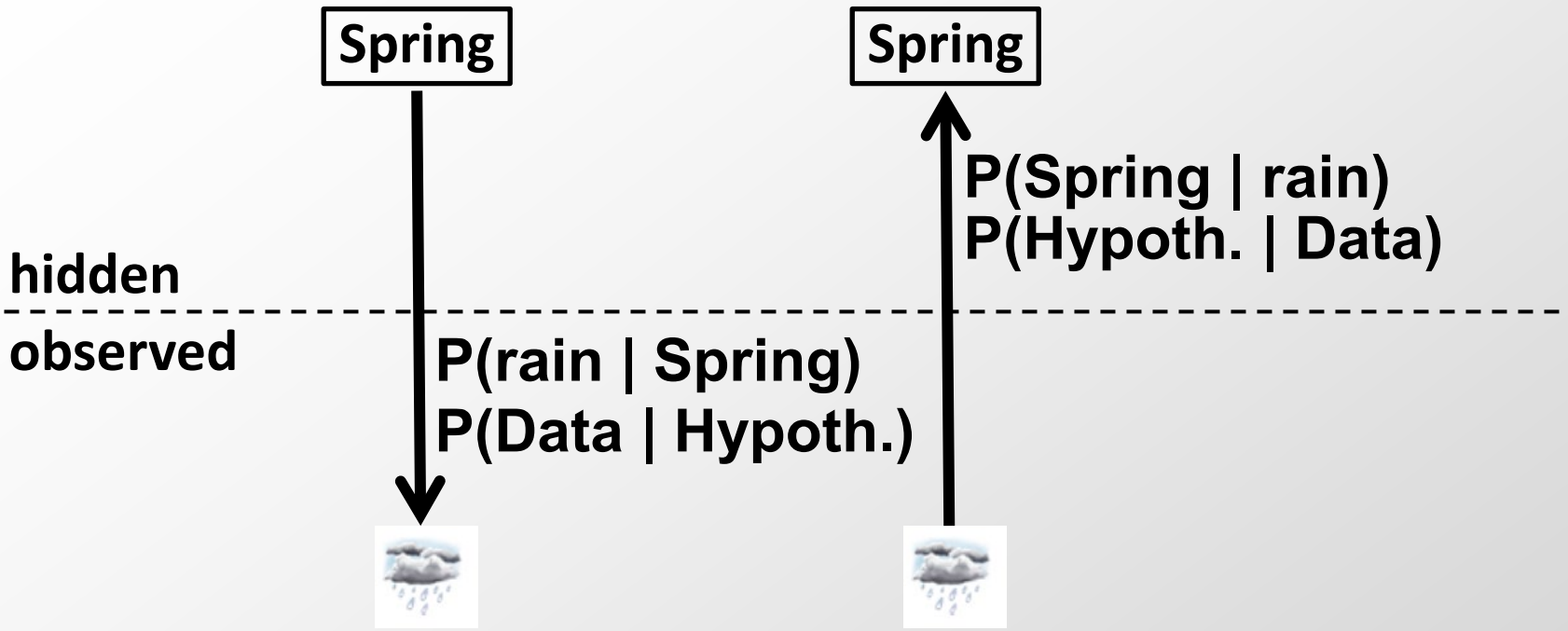
Making inferences about the world

- Generative models:
  Express **forward** probability of an event,
  given the **hidden** state of the world



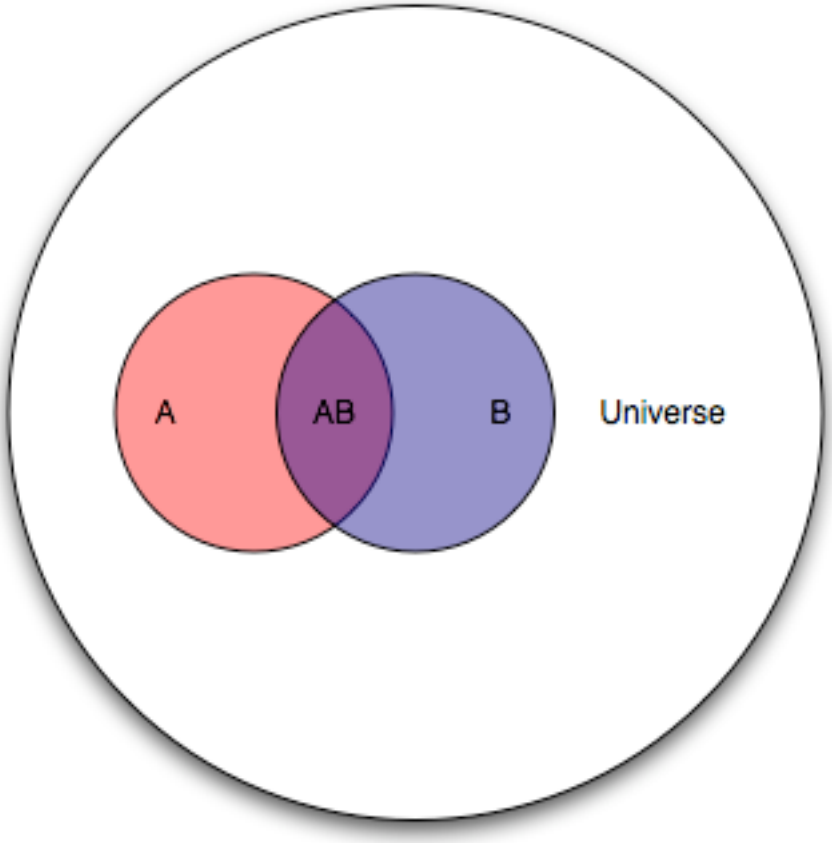
- We can estimate:
  - P(snow | winter), P(observation | season)

"Reversing the arrows"



- Goal: P(D|H) ➔ P(H|D)
- Bayes' Rule allows us to do this:
- $$P(H|D) = \frac{P(D|H) * P(H)}{P(D)}$$

Proving Baye's Rule



$$P(A|B)P(B) = P(B|A)P(A)$$

$$P(A|B) = P(B|A)P(A)/P(B)$$

$$P(B|A) = P(A|B)P(B)/P(A)$$

Bayes' rule

Bayes Theorem

Likelihood
Probability of collecting this data when our hypothesis is true

Prior
The probability of the hypothesis being true before collecting data

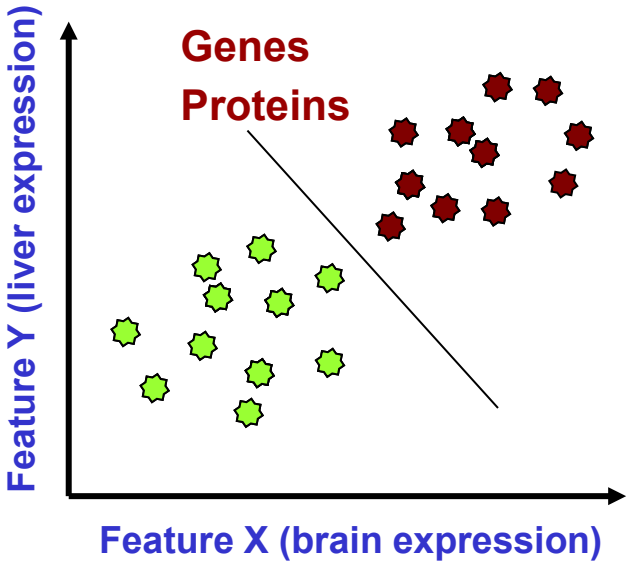
$$P(H|D) = \frac{P(D|H)\,P(H)}{P(D)}$$

Posterior
The probability of our hypothesis being true given the data collected

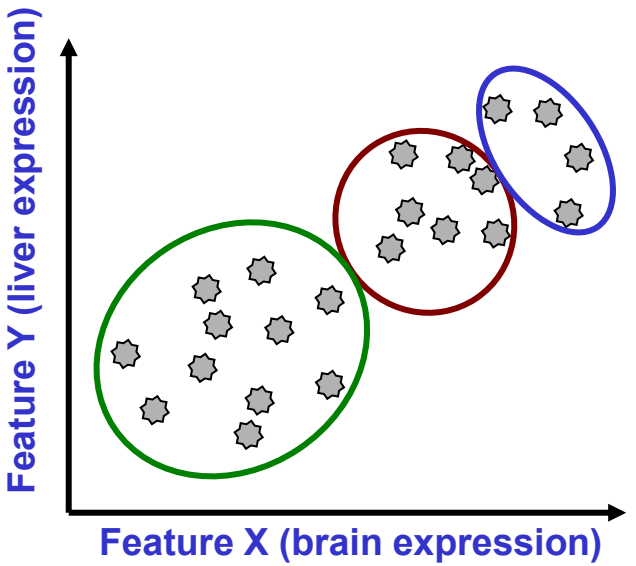
Marginal
What is the probability of collecting this data under all possible hypotheses?

Clustering vs Classification

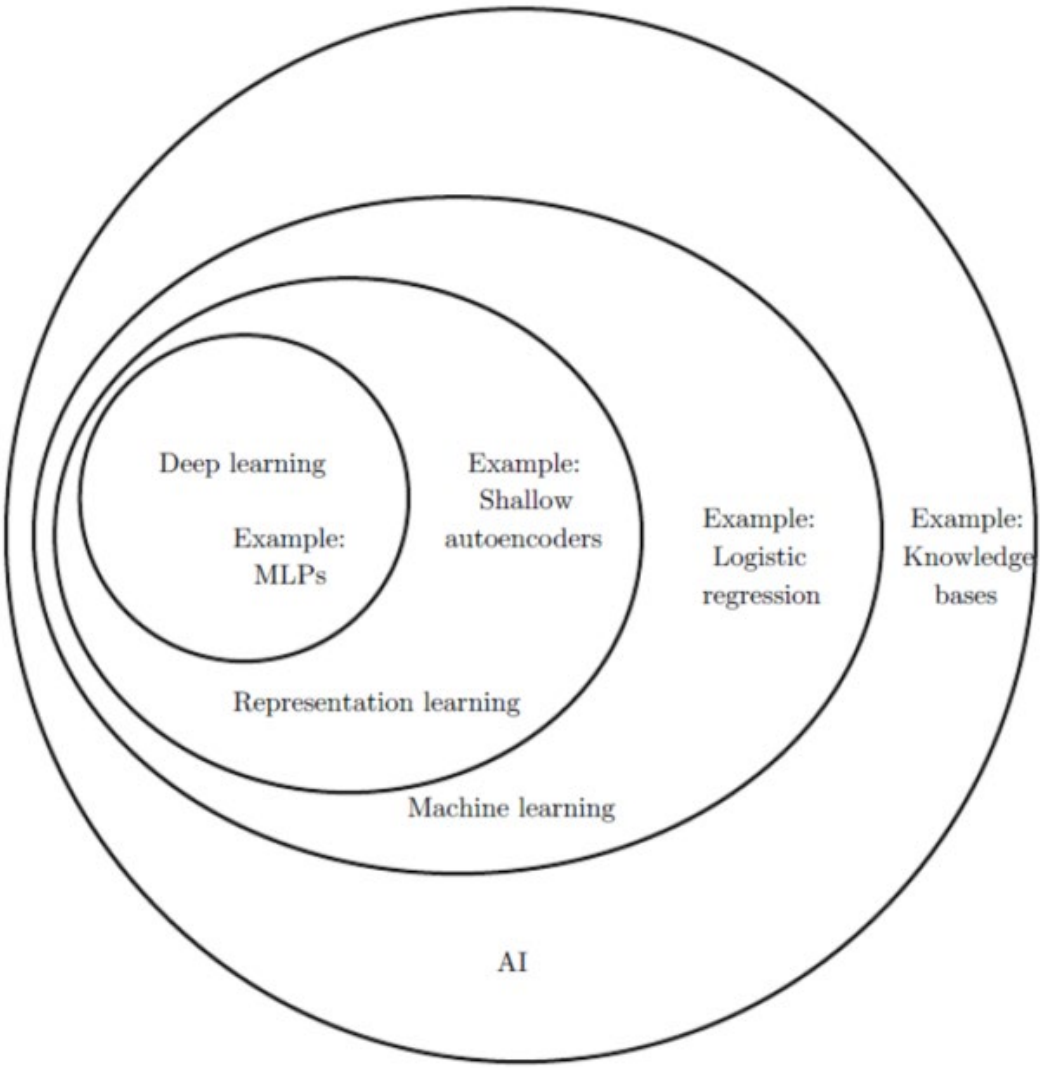
- Objects characterized by one or more features
- **Classification (supervised learning)**
  - Have labels for some points
  - Want a "rule" that will accurately assign labels to new points
  - Sub-problem: Feature selection
  - Metric: Classification accuracy

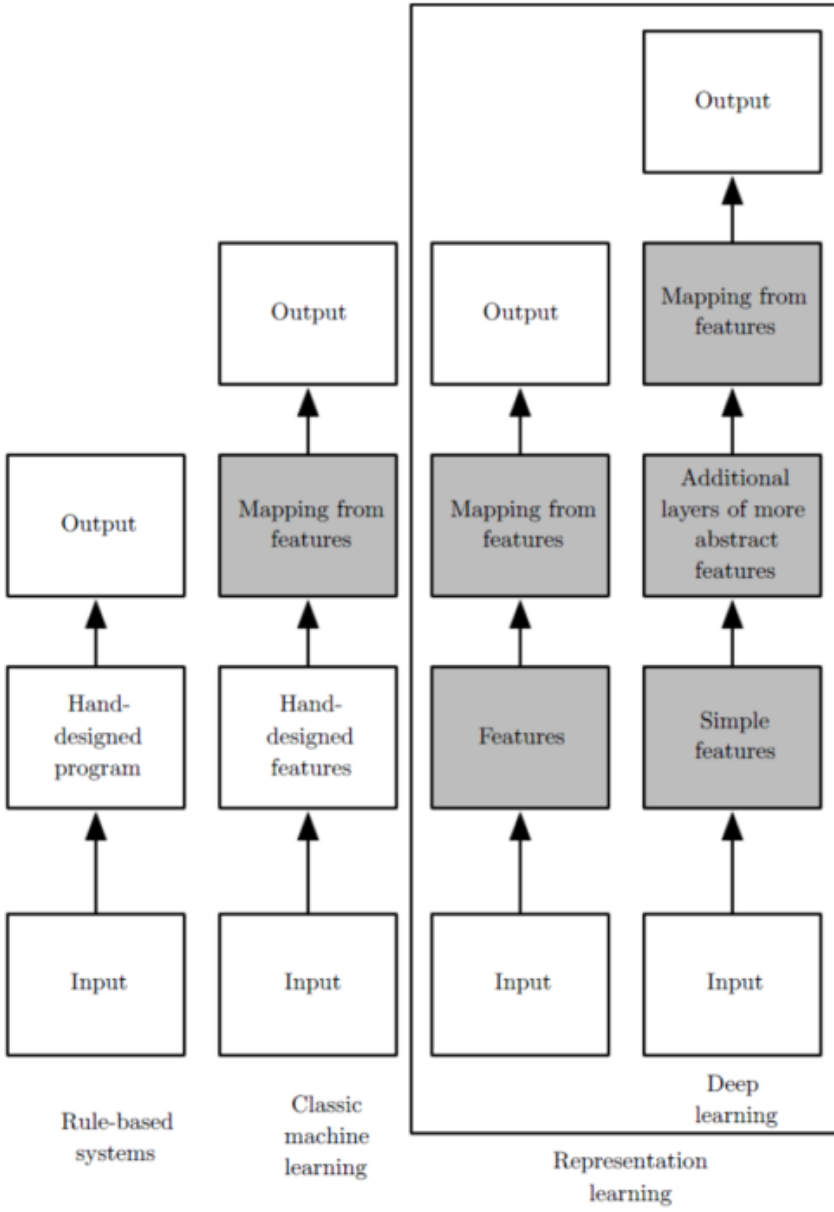


- **Clustering (unsupervised learning)**
  - No labels
  - Group points into clusters based on how "near" they are to one another
  - Identify structure in data
  - Metric: independent validation features

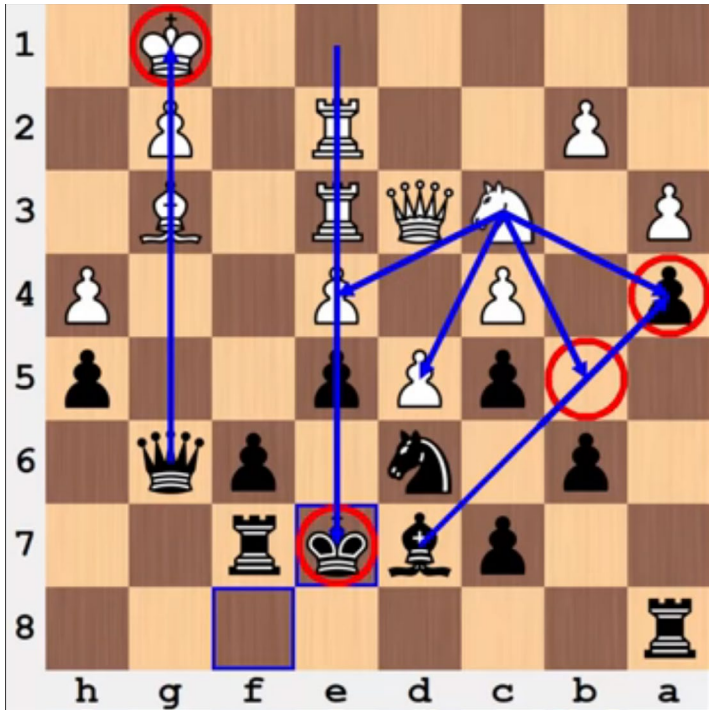


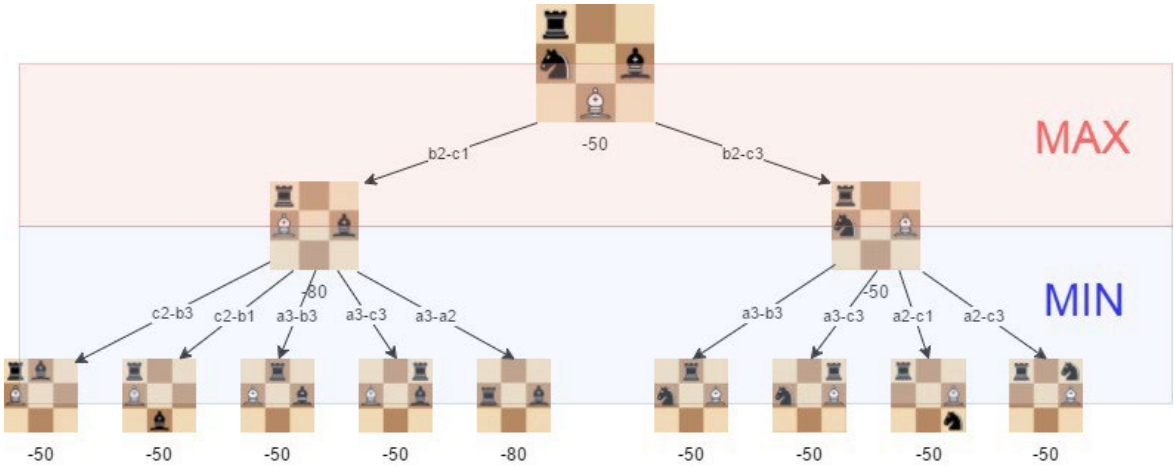
AI vs. ML vs. Deep Learning





Classical AI: How machines play chess?





‘Classical’ AI approach (rule-based, tree search):

1. Human: Program in all the rules of chess
2. Human: Hand-craft a scoring function for each position
3. Search all moves that you can make (max score)
4. Search all moves that opponent can make (min score)
5. Repeat for many iterations
6. Choose move that gives best score

Artificial Intelligence vs. Machine Learning

Hard for Machines

General intelligence

Manipulation

Driving

Images

Faces

Language

Jeopardy!

'Deep' Learning

Prime factorization

Knowledge

Planning

Logic

Reasoning

Remembering

Math

Integration

'Classical' AI

Go

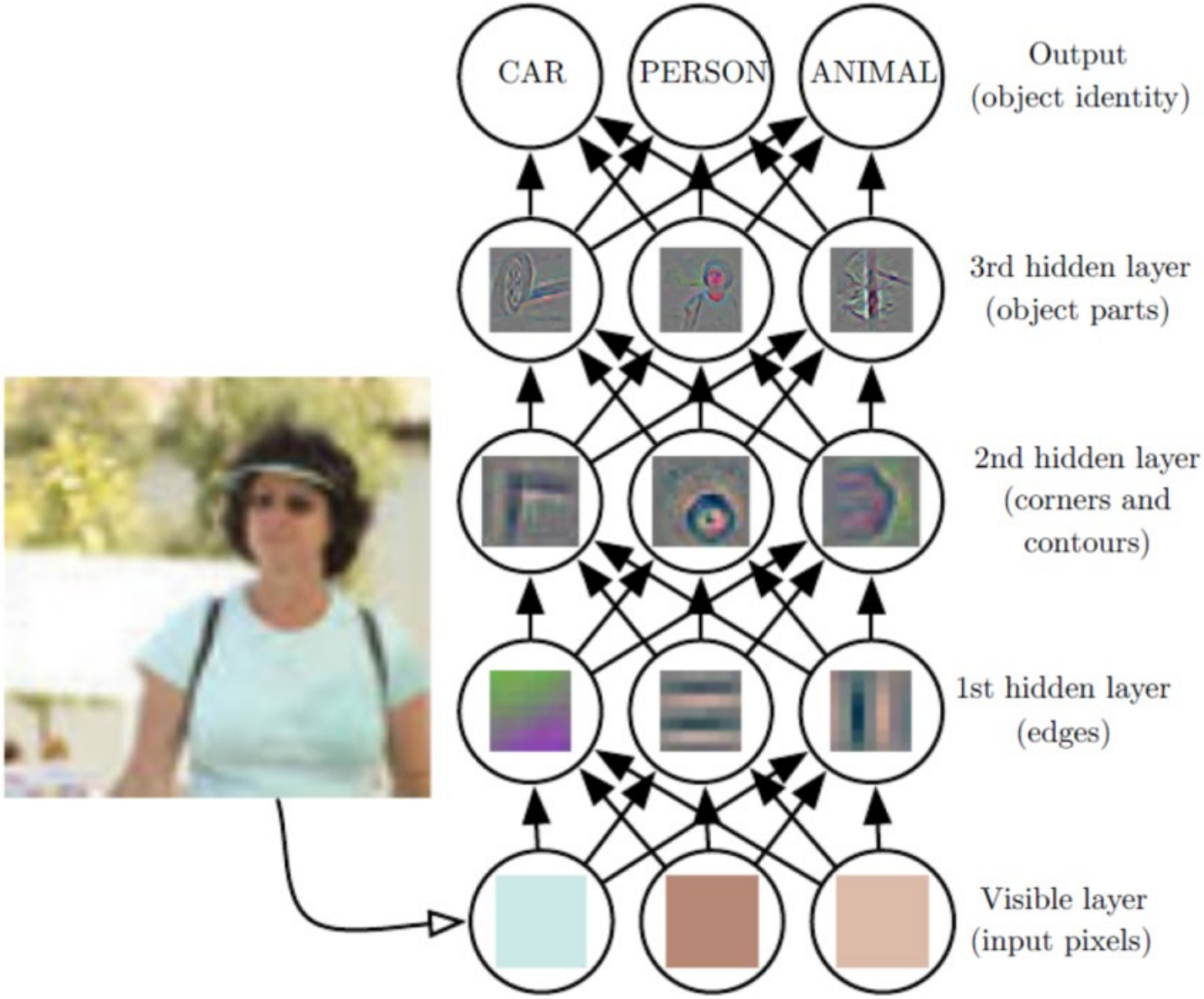
Chess

Easy for Machines

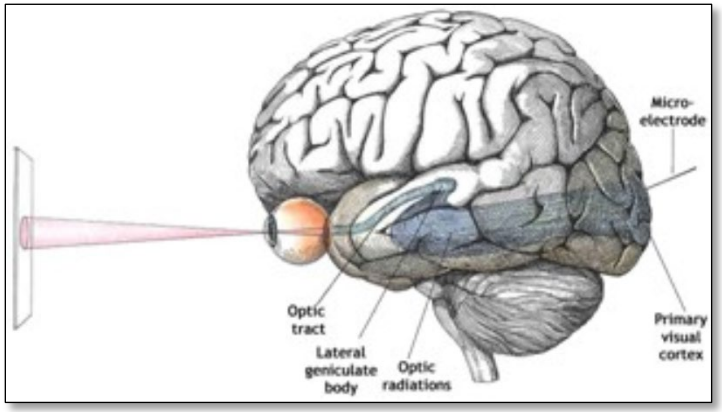
Easy for humans

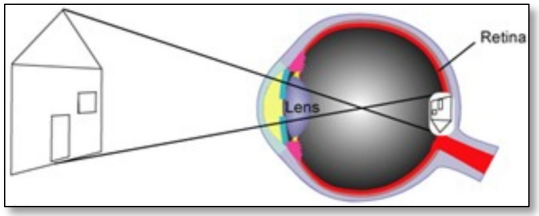
Hard for humans

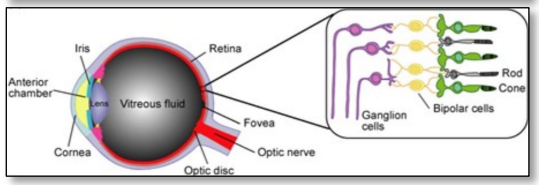
Deep learning ➔ many **layers** of **abstraction**

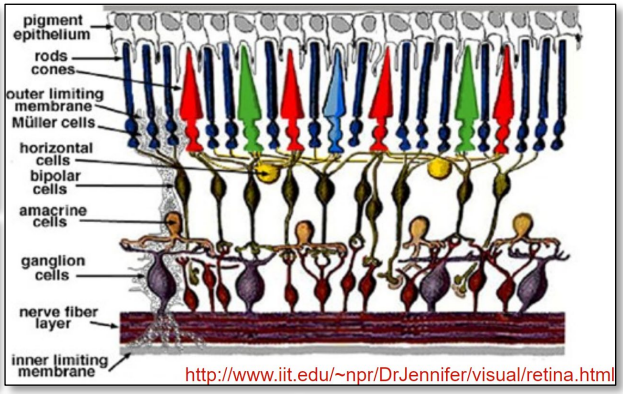


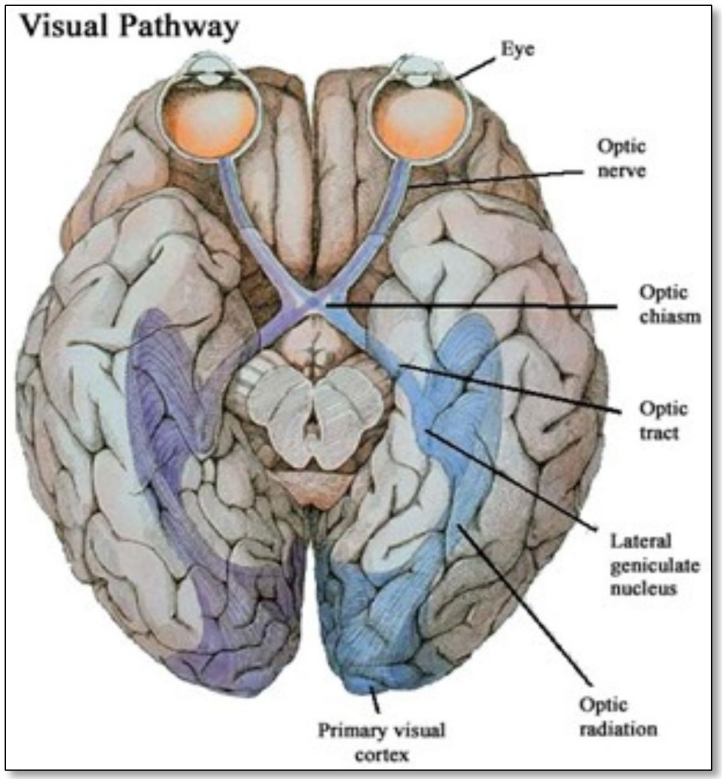
Inspiration: human/animal visual cortex

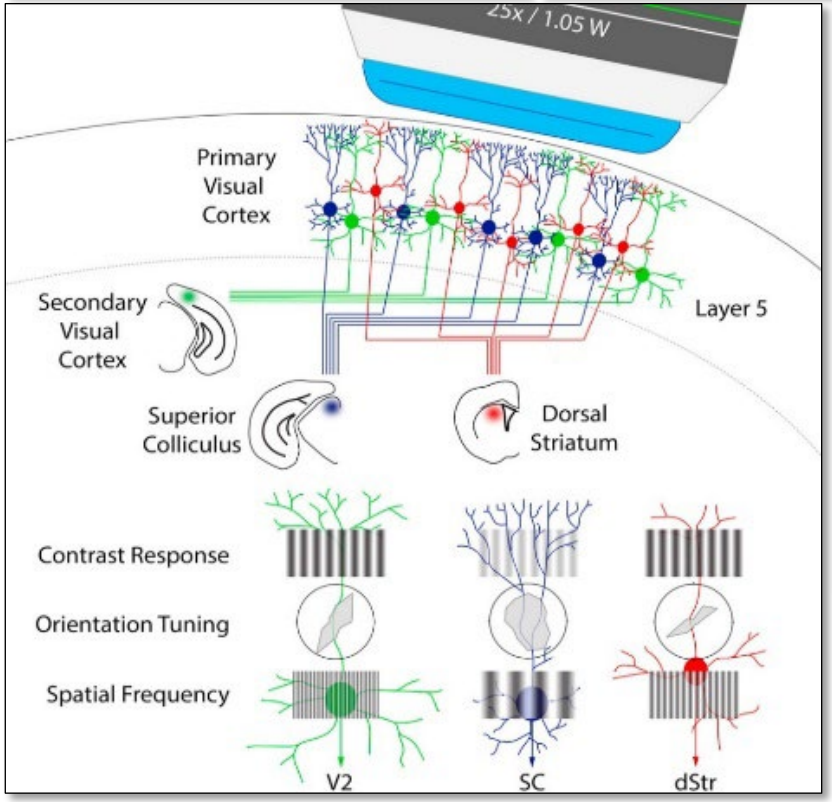


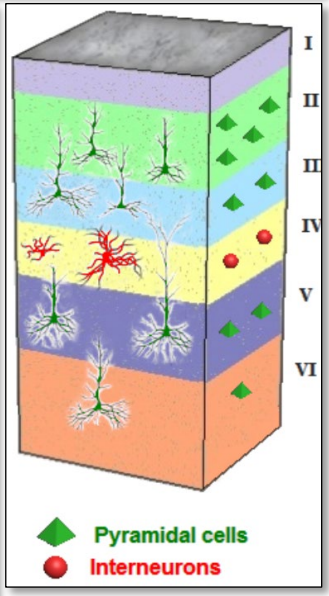






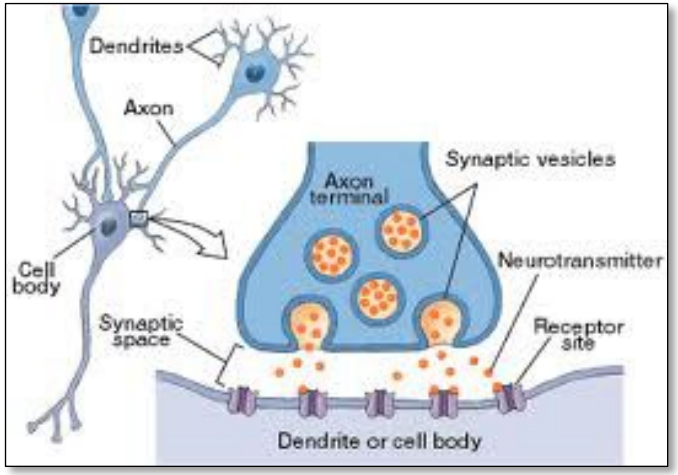


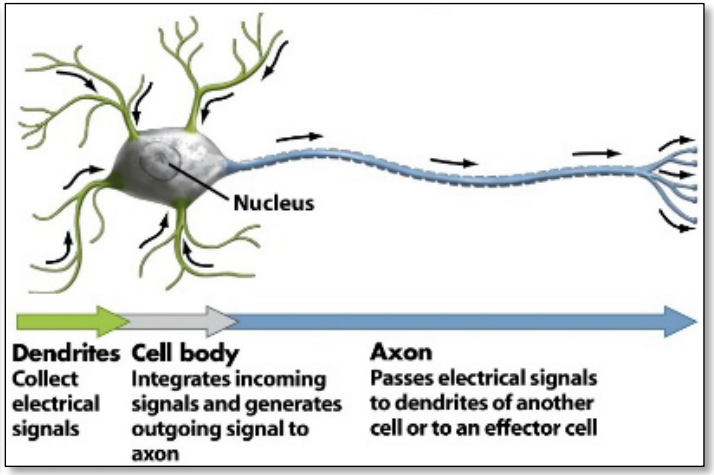


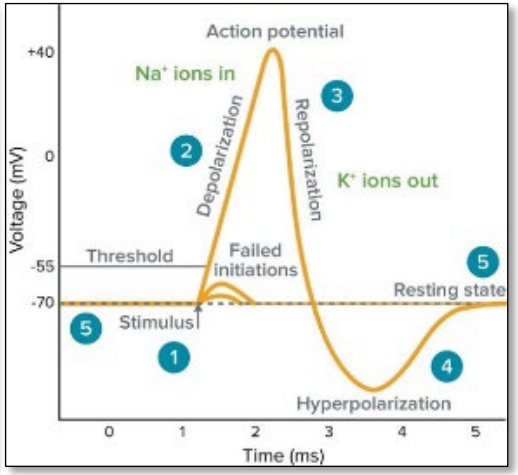


- **Layers of neurons: pixels, edges, shapes, primitives, scenes**
- **E.g. Layer 4 responds to bands w/ given slant, contrasting edges**

Primitives: Neurons & action potentials



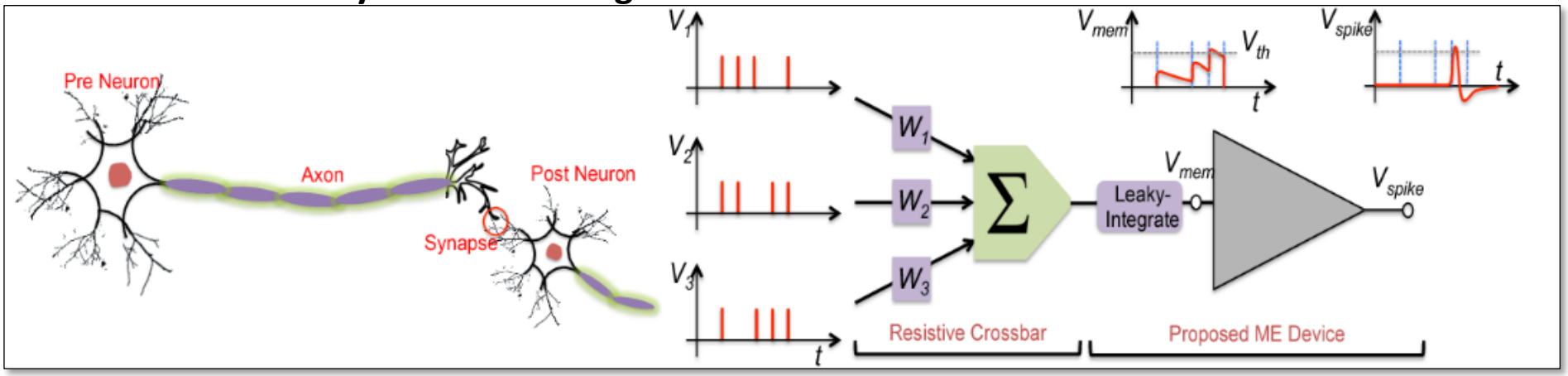




- Chemical accumulation across dendritic connections
- Pre-synaptic axon
  → post-synaptic dendrite
  → neuronal cell body

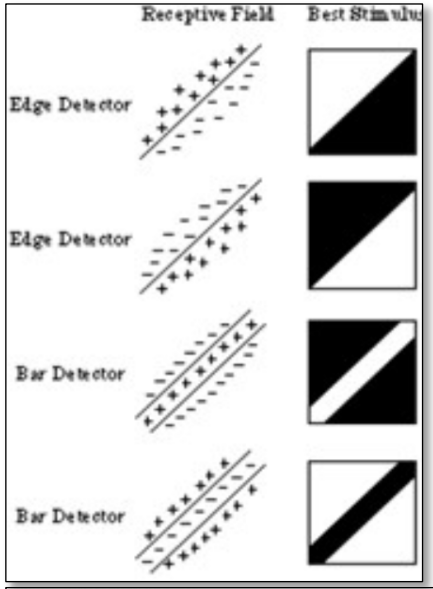
- Each neuron receives multiple signals from its many dendrites
- When threshold crossed, it fires
- Its axon then sends outgoing signal to downstream neurons

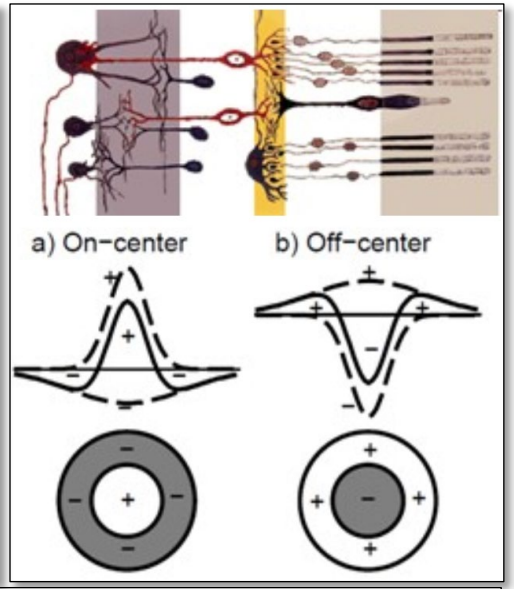
- Weak stimuli ignored
- Sufficiently strong cross activation threshold
- Non-linearity within each neuronal level

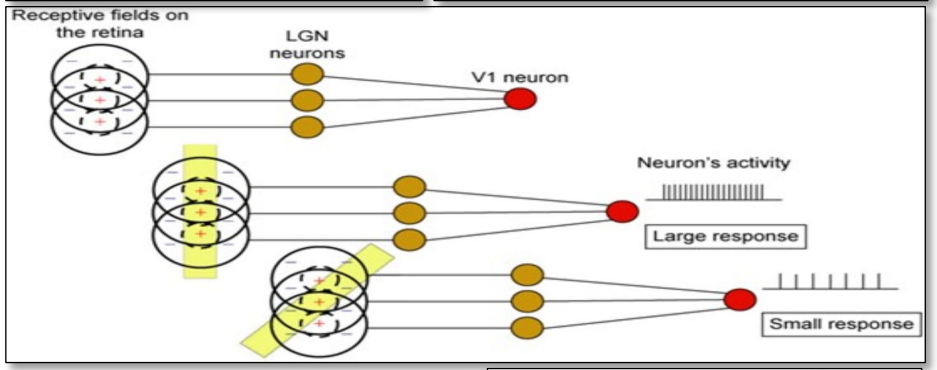


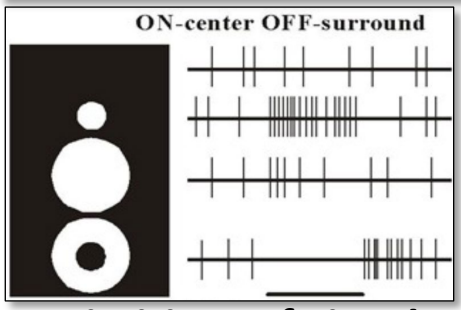
- Neurons connected into circuits (neural networks): emergent properties, learning, memory
- Simple primitives arranged in simple, repetitive, and extremely large networks
- 86 billion neurons, each connects to 10k neurons, 1 quadrillion (10^{12}) connections

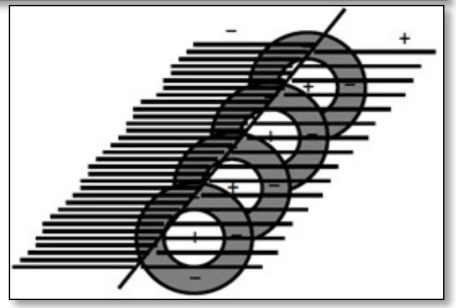
Abstraction layers: edges, bars, dir., shapes, objects, scenes

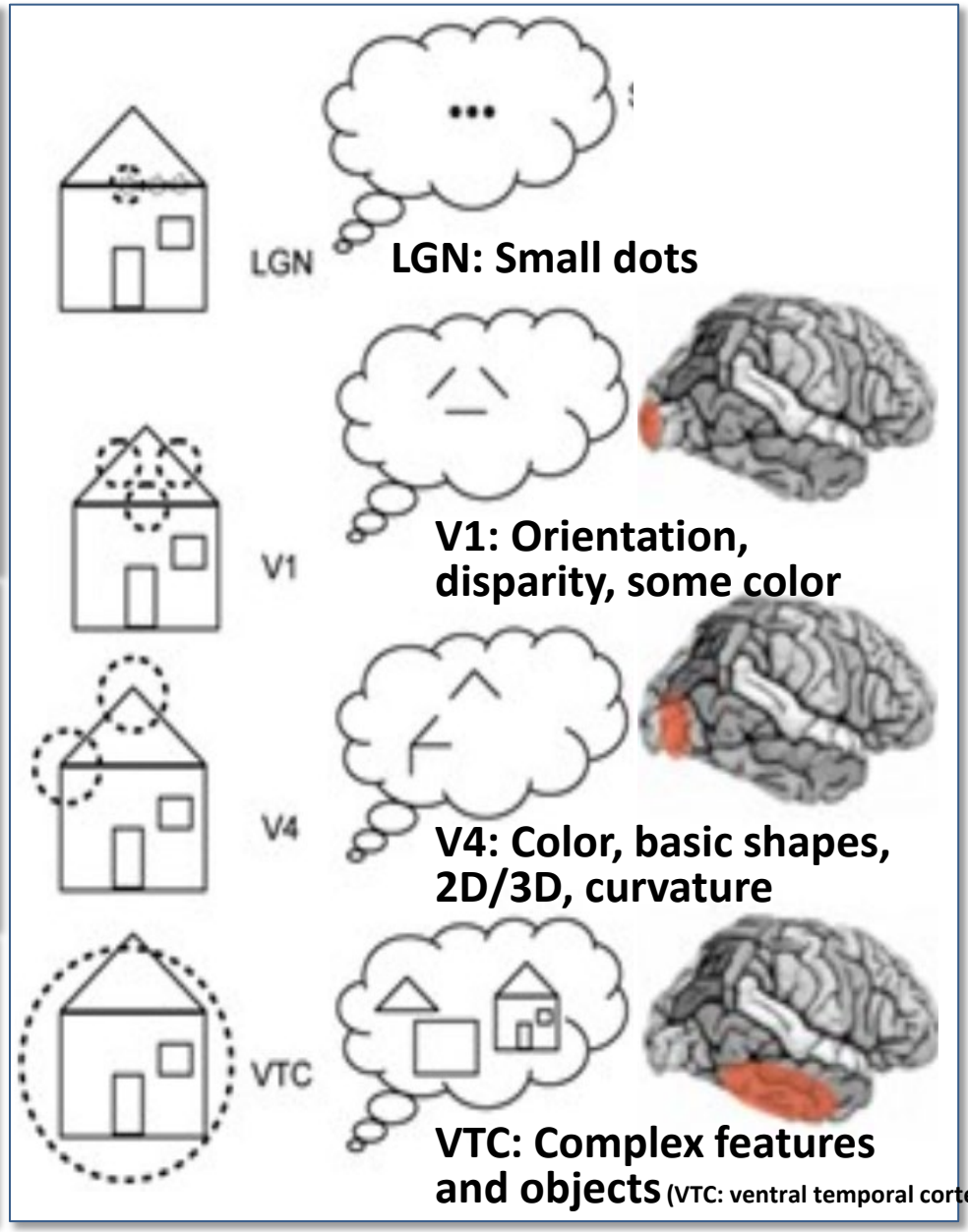






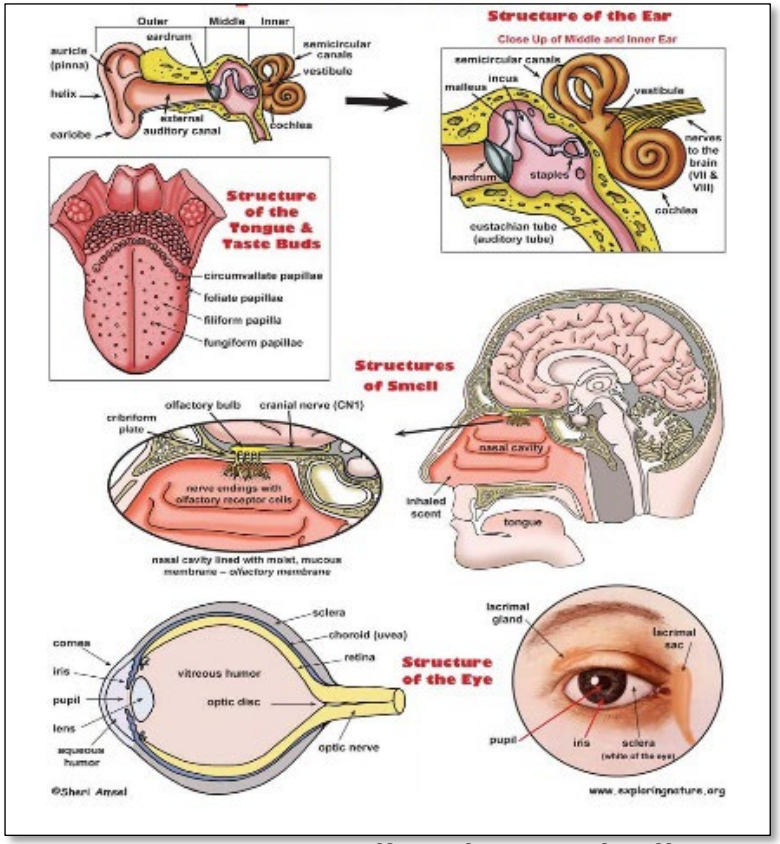




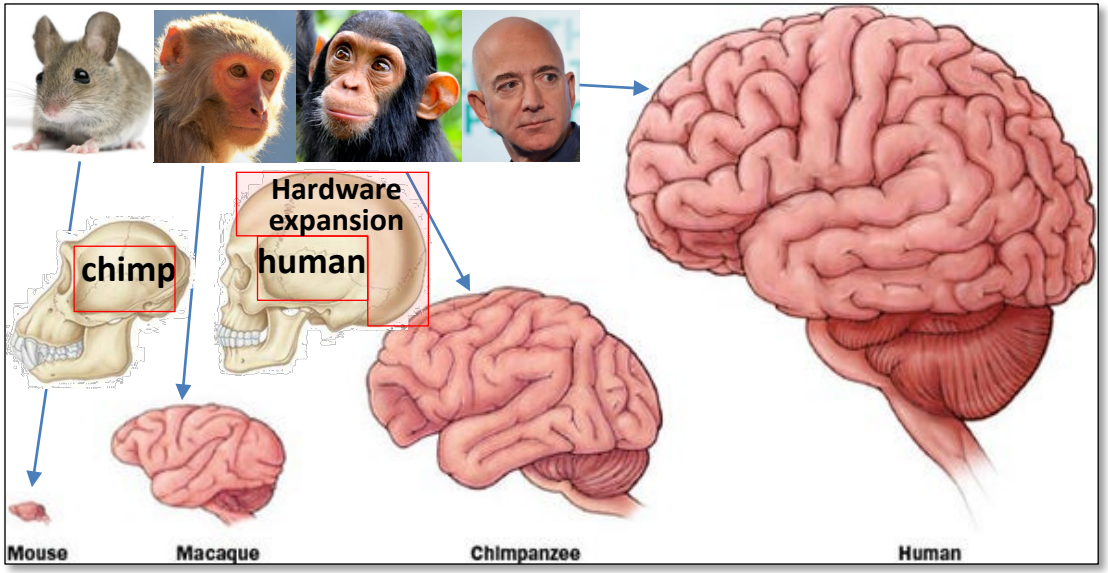


- Primitives of visual concepts encoded in neuronal connection in early cortical layers
- Abstraction layers ⇔ visual cortex layers
- Complex concepts from simple parts, hierarchy

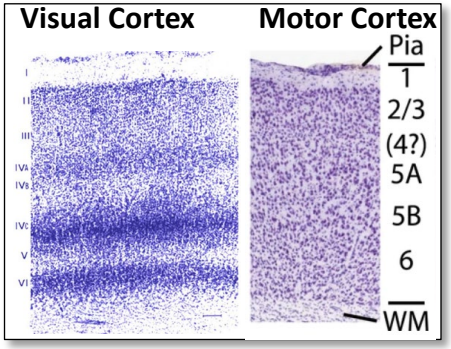
General "learning machine", reused widely



- Hearing, taste, smell, sight, touch all re-use similar learning architecture



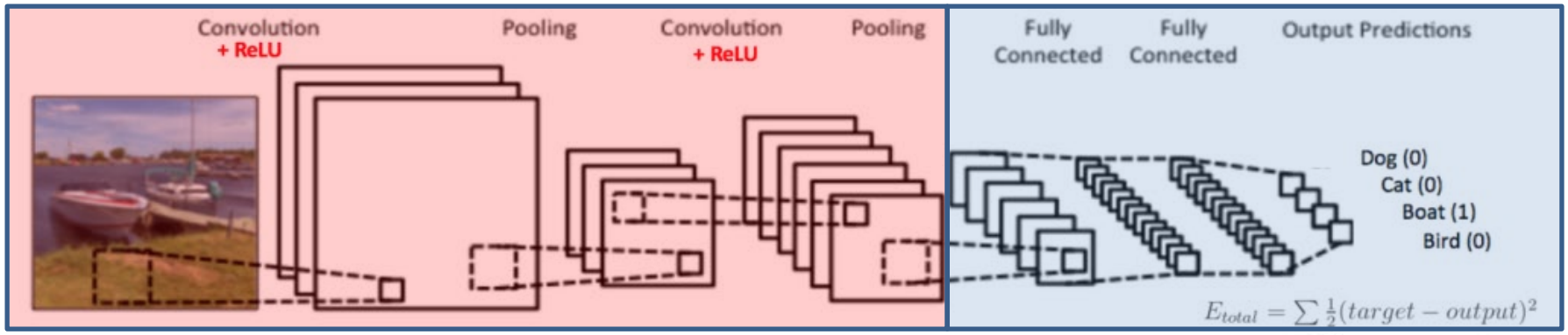
- Massive recent expanse of human brain has re-used a relatively simple but general learning architecture



- Interchangeable circuitry
- Auditory cortex learns to 'see' if sent visual signals
- Injury area tasks shift to uninjured areas

- Not fully-general learning, but well-adapted to our world
- Humans co-opted this circuitry to many new applications
- Modern tasks accessible to any homo sapiens (<70k years)
- ML primitives not too different from animals: more to

Key idea: **Representation** learning



x
data

'Modern' Deep learning:
Hierarchical Representation Learning
Feature extraction

z
Latent space representation

'Classical' Fully-connected
Neural Networks
Classification

y
label

In deep learning, the two tasks are **coupled**:

- the classification task "drives" the feature extraction
- Extremely powerful and general paradigm
  - ➔ **Be creative**! The field is still at its infancy!
  - ➔ New application domains (e.g. beyond images) can have **structure** that current architectures **do not capture/exploit**
  - ➔ Genomics/biology/neuroscience can help drive development of **new architectures**

Course at a Glance

Wk	Date	Lec	Topic
	Introduction: Machine Learning, Deep Learning, Generative AI, and the Unification of Biology		
1	Thu-Sep-05	L1	Course Overview, Machine Learning, Deep Learning, Inference, Genome, Proteins, Chemistry, Imaging
	Module 1: Genomics, Epigenomics, Single-Cell, Networks, Circuitry		
2	Tue-Sep-10	L2	Expression Analysis, Clustering/Classification, Gaussian Mixture Models, K-means, Bayesian Inf, Gen-vs-DiscrML
2	Thu-Sep-12	L3	Sequential Data, Alignment, DynProg, Hidden Markov Models, Parsing, Posterior Decoding, HMM architectures
3	Tue-Sep-17	L4	Regulatory Genomics: Motifs, Information, ChIP, Gibbs Sampling, EM, CNNs for Genome Parsing
3	Thu-Sep-19	L5	Epigenomics: Signal Modeling, Peak calling, Chromatin states, 3D structure, Hi-C, Genome Topology
4	Tue-Sep-24	L6	Single-cell genomics, sc-mutli-omics, non-linear embeddings, spatial transcriptomics, next-gen technologies
4	Thu-Sep-26	L7	Regulatory Networks: Graphs, Linear Algebra, PCA, SVD, Dimentionality Reduction, TF-enhancer-gene circuitry
	Module 2: Protein Structure, Protein Language Models, Geometric Deep Learning		
5	Tue-Oct-01	L8	Intro to structural biology
5	Thu-Oct-03	L9	Protein structure and folding: Diffusion models, Cryo-EM, Protein design
6	Tue-Oct-08	L10	Intro to transformers and Large Language Models LLMs
6	Thu-Oct-10	L11	Protein Language Models PLMs and Transfer Learning
7	Tue-Oct-15	-	-- No Class -- Student holiday
7	Thu-Oct-17	L12	DNA language models: Chromatin Structure
	Module 3: Chemistry, Therapeutics, Graph Neural Networks		
8	Tue-Oct-22	L13	Overview of drug development
8	Thu-Oct-24	L14	Intro to small molecules
9	Tue-Oct-29	L15	Representation of small molecules: Graphs, GNNs, Transformers, RDKit
9	Thu-Oct-31	L16	Docking: Small molecule - proteins docking
10	Tue-Nov-05	L17	Disease Association Mapping, genetics, GWAS, linkage analysis, disease circuitry, variant-to-function
10	Thu-Nov-07	L18	Quantitative trait mapping, molecular traits, eQTLs, mediation analysis, iMWAS, multi-modal QTLs
	Module 4: Electronic Health Records, Imaging, Evolution, Metabolism		
11	Tue-Nov-12	L19	Electronic Health Records, AllOfUs, UKBioBank, Medical Genomics, Pop-Scale Cohorts, Multi-Ancestry [not quiz'd]
11	Thu-Nov-14	L20	**-- In-class Quiz**
12	Tue-Nov-19	L21	Imaging methods for biological applications
12	Thu-Nov-21	L22	Comparative genomics, Conservation, Evolutionary signatures, PhyloCSF, RNA structure, Motif BLS2conf
13	Tue-Nov-26	L23	Evolution, Phylogenetics, Phylogenomics, Duplication, RNA world, RNA folding, lincRNAs, RNA modifications, m6A
13	Thu-Nov-28	-	-- No Class -- Thanksgiving Holiday
14	Tue-Dec-03	L24	Modeling metabolism: Flux balance analysis
14	Thu-Dec-05	L25	Measuring metabolism: Metabolomics and Deep Learning
	Final Projects		
15	Tue-Dec-10	L26	**Project Presentations (6-8 mins/team). Report due Fri@11.59p, Slides due Mon@11.59p, Present Live Tue**

Today: Gene Expression Clustering & Classification

1. **Introduction to gene expression analysis**
   - Technology: microarrays vs. RNAseq. Resulting data matrices
   - Supervised (Classification) vs. unsupervised (clustering) learning

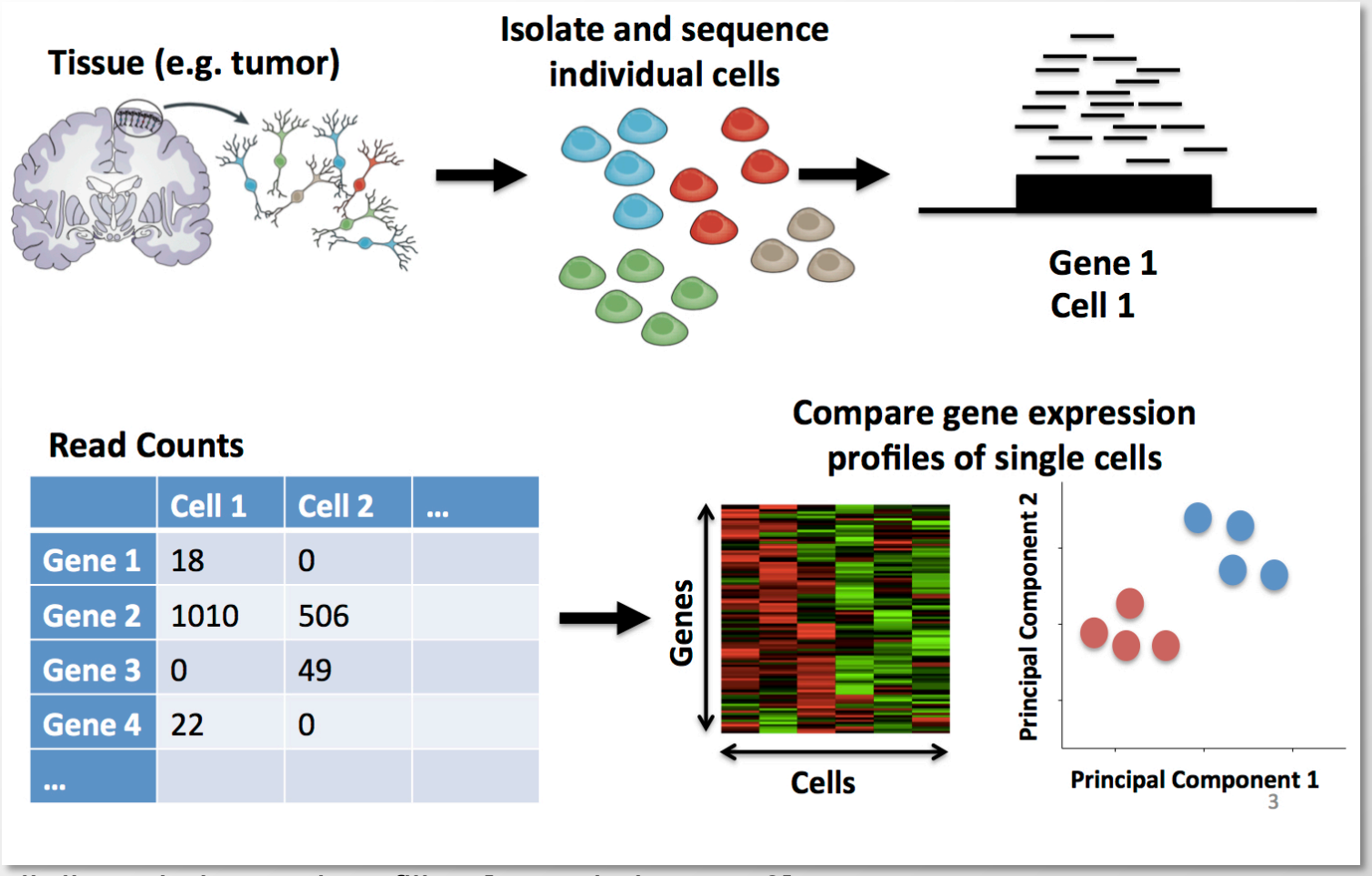
2. **K-means clustering (clustering by partitioning)**
   - Algorithmic formulation: Update rule, optimality criterion. Fuzzy k-means.
   - Machine learning formulation: Generative models, Expectation Maximization.

3. **Hierarchical Clustering (clustering by agglomeration)**
   - Basic algorithm, Distance measures. Evaluating clustering results

4. **Naïve Bayes classification (generative approach to classification)**
   - Discriminant function: class priors, and class-conditional distributions
   - Training and testing, Combine mult features, Classification in practice

5. **(optional) Support Vector Machines (discriminative approach)**
   - SVM formulation, Margin maximization, Finding the support vectors
   - Non-linear discrimination, Kernel functions, SVMs in practice

Single-cell RNA-seq (scRNA) ➔ Expression profiling

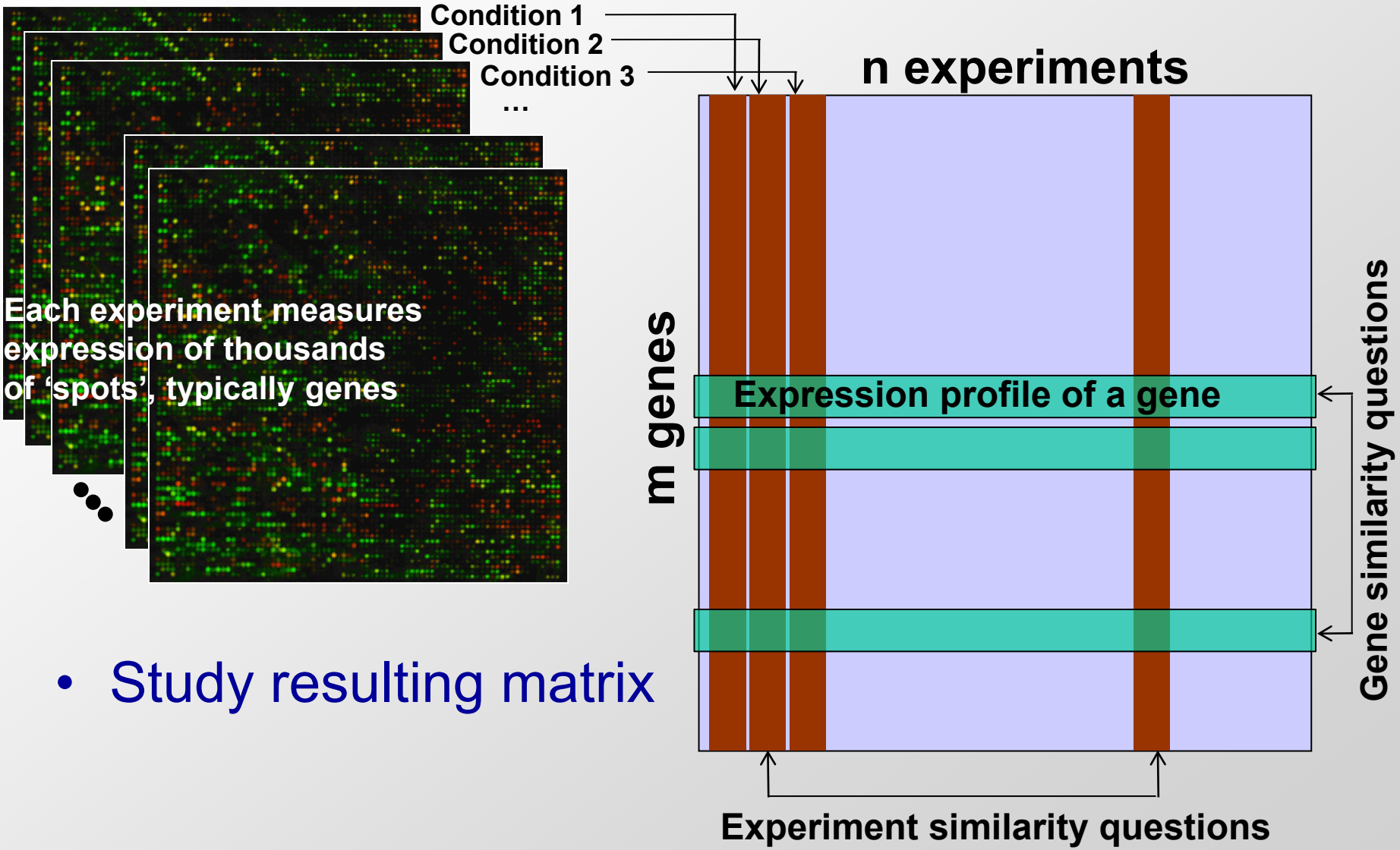


	Cell 1	Cell 2	…
Gene 1	18	0	
Gene 2	1010	506	
Gene 3	0	49	
Gene 4	22	0	
…			

- Single-cell dissociation and profiling [more in lecture 6]
- Sequencing-based counting of RNA reads for each gene
- Counts for every cell, multiple samples/cells from each different condition
- Then:
  - Clustering (unsupervised learning: cell types, patterns)
  - Classification (supervised learning: predict Alzheimer's vs. control, feature selection, etc)

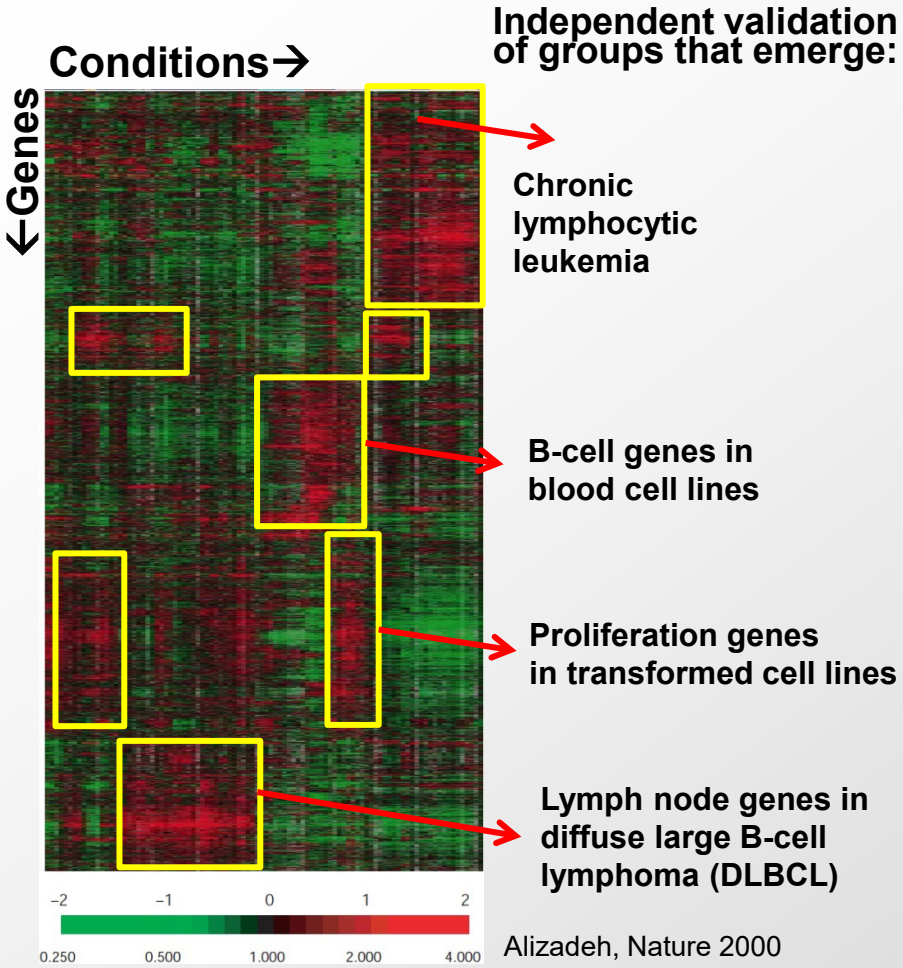
Expression Analysis Data Matrix

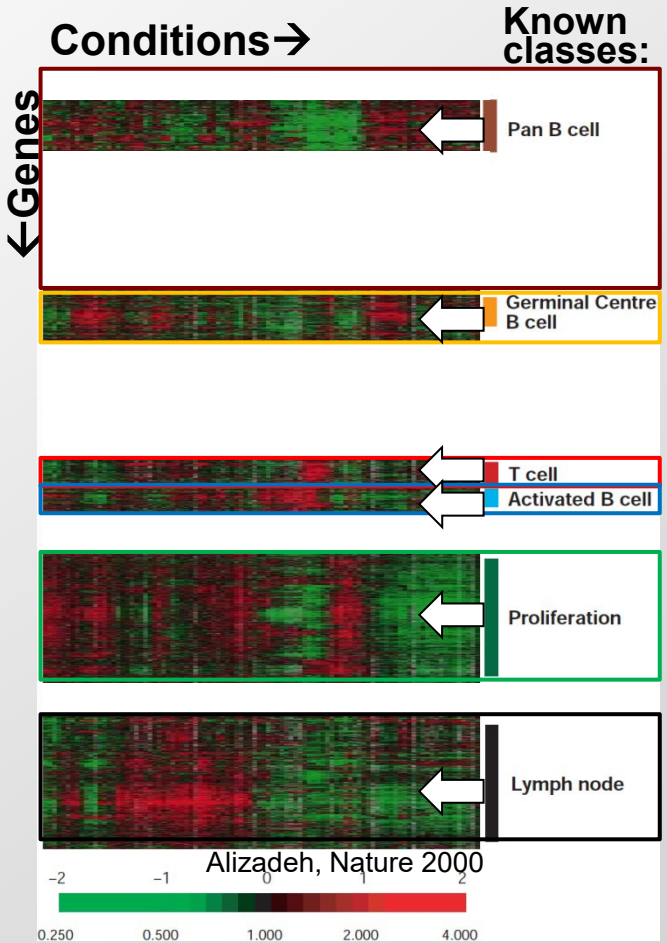
- Measure 20,000 genes in 100s of conditions



- Study resulting matrix

Clustering vs. Classification





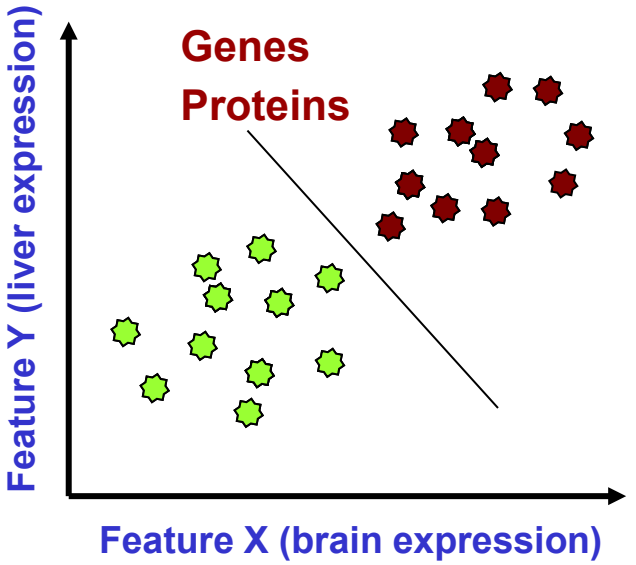
Goal of Clustering: **Group similar items** that likely come from the same category, and in doing so **reveal hidden structure**

Goal of Classification: Extract features from the data that best **assign new elements** to ≥1 of **well-defined classes**

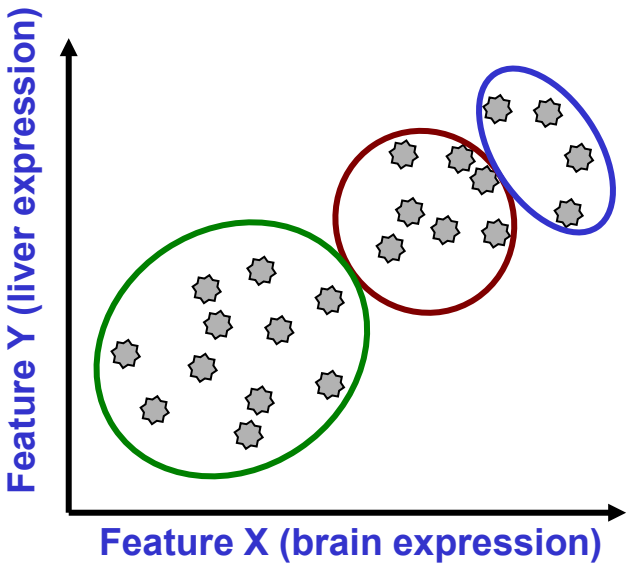
- **Unsupervised learning**
- **Supervised learning**

Clustering vs Classification

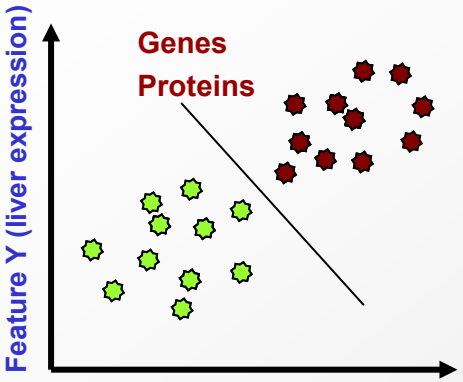
- Objects characterized by one or more features
- **Classification (supervised learning)**
  - Have labels for some points
  - Want a "rule" that will accurately assign labels to new points
  - Sub-problem: Feature selection
  - Metric: Classification accuracy

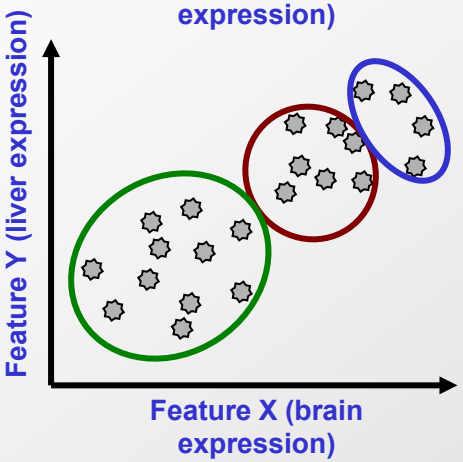


- **Clustering (unsupervised learning)**
  - No labels
  - Group points into clusters based on how "near" they are to one another
  - Identify structure in data
  - Metric: independent validation features

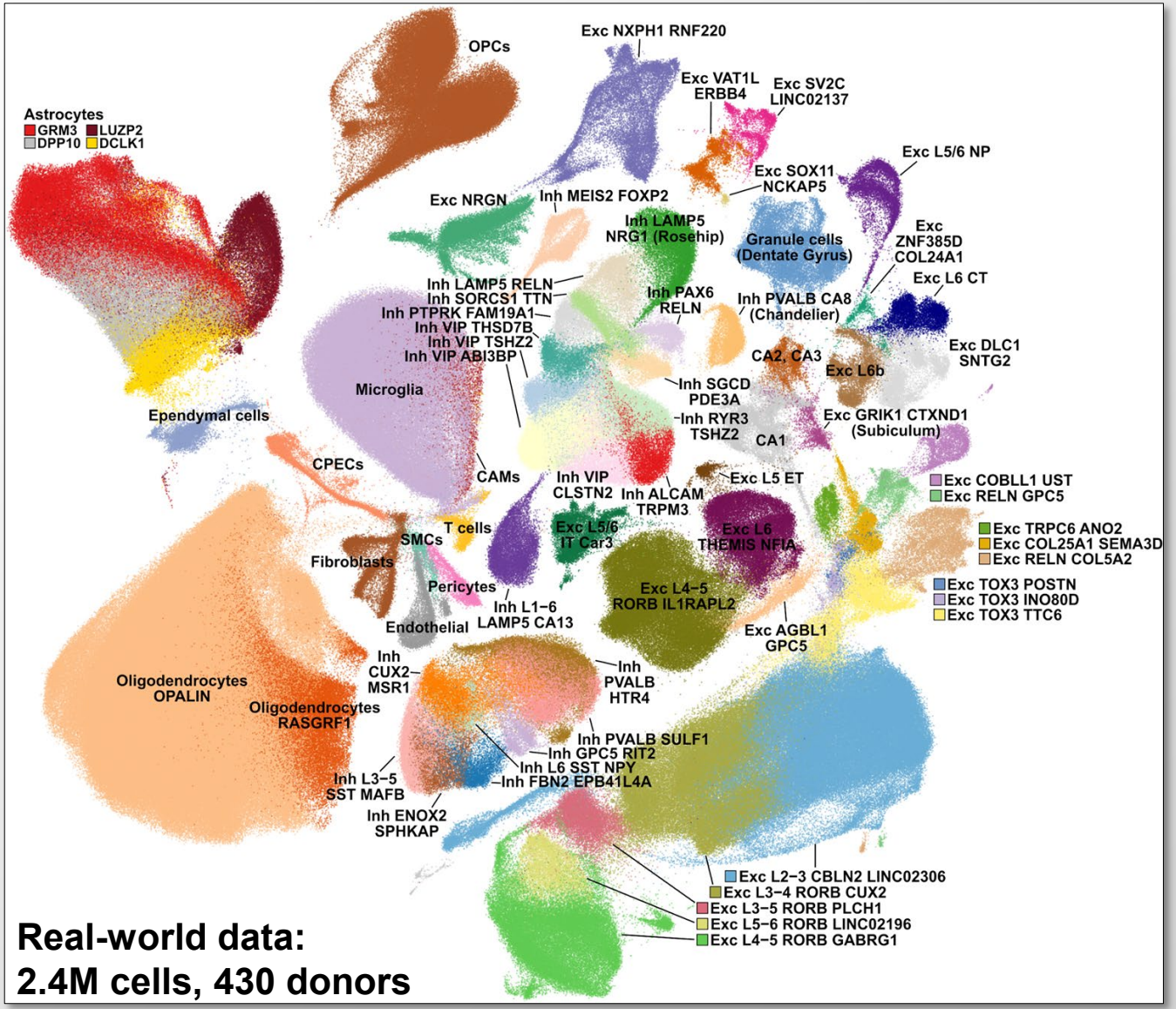


scRNA data is extremely high-dimensional (but same principles apply)





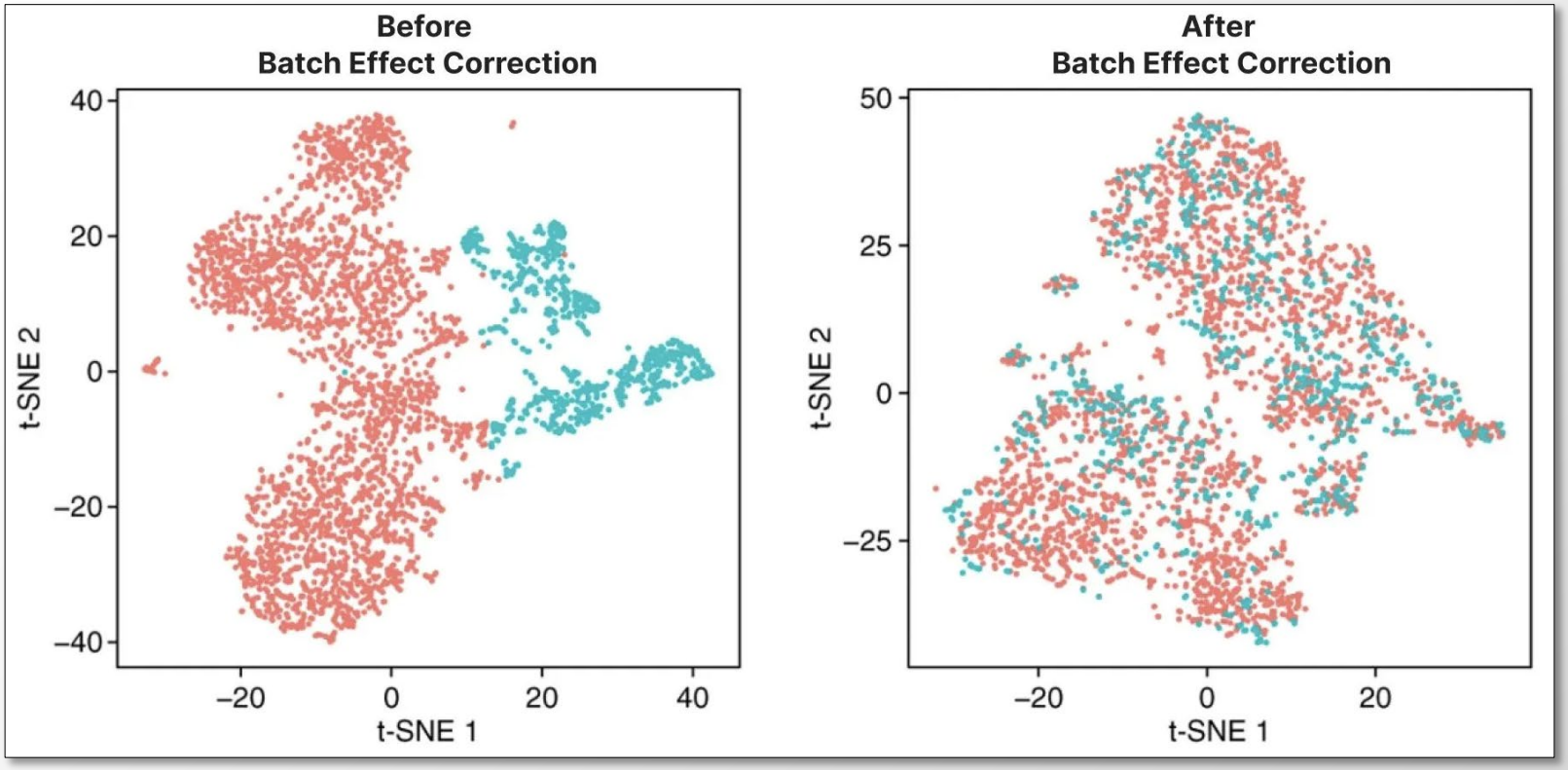
Toy example:
22 cells, 2 measurements each



Real-world data:
2.4M cells, 430 donors

Each dot: one cell ⇔ 20,000-dimensional expression vector
Clusters: *De-novo* discover *one component* of variation: cell type

Single-cell analysis is a lot more complex (Lecture 6)



- **Batch Effects**: Variability from different experimental conditions.
- **Unwanted Variation**: Technical noise often obscures true biological signals.
- **High Dimensionality**: Thousands of genes measured per cell.
- **Dropout Events**: Missing data due to low gene expression.
- **Noise and Sparsity**: Data is inherently noisy (tiny sample) and sparse (zero values).
- **Cell Heterogeneity**: High variability within and between cell types, gradients of transitions.
- **Scalability Issues**: Analyzing millions of cells requires significant computational resources.
- **Alignment and Integration**: Combining data across platforms and experiments is challenging.
- **Doublets/Multiplets**: Multiple cells mistakenly counted as a single cell.
- **Dimensionality reduction**: Linear (PCA), sparse (SPCA), non-linear (t-SNE, UMAP)

Machine Learning: Supervised & Unsupervised

1. **Introduction to gene expression analysis**
   - Technology: microarrays vs. RNAseq. From reads to transcripts, expr levels
   - Expr matrix. Supervised vs. unsupervised learning. Clustering/Classification

2. **K-means clustering (clustering by partitioning)**
   - Algorithmic formulation: Update rule, optimality criterion. Fuzzy k-means.
   - Machine learning formulation: Generative models, Expectation Maximization.

3. **Hierarchical Clustering (clustering by agglomeration)**
   - Basic algorithm, Distance measures. Evaluating clustering results

4. **Naïve Bayes classification (generative approach to classification)**
   - Discriminant function: class priors, and class-conditional distributions
   - Training and testing, Combine mult features, Classification in practice

5. **(optional) Support Vector Machines (discriminative approach)**
   - SVM formulation, Margin maximization, Finding the support vectors
   - Non-linear discrimination, Kernel functions, SVMs in practice

K-Means Clustering

The Basic Idea

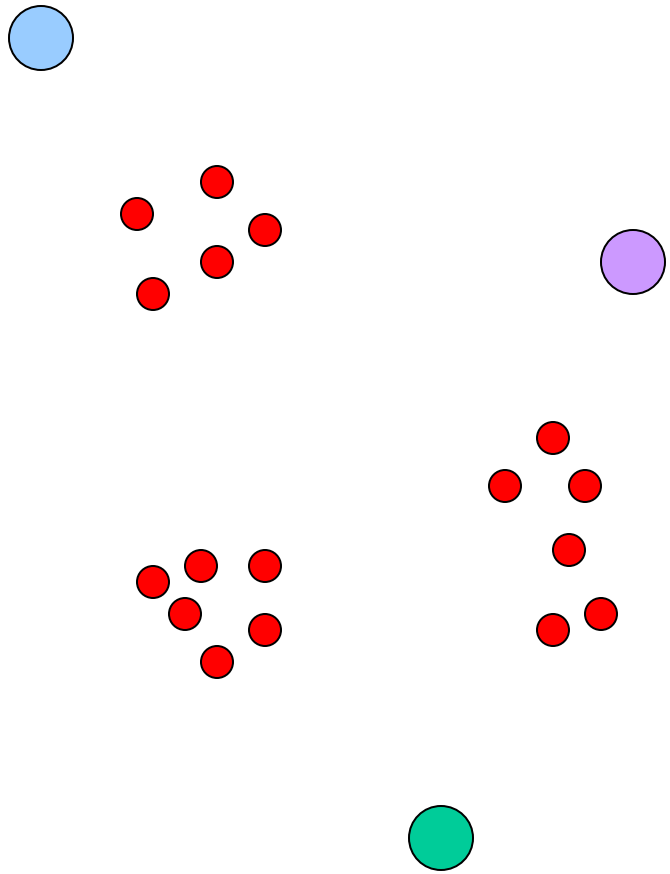
- Assume a fixed number K of clusters
- Partition points into K compact clusters

The Algorithm

- Initialize K cluster centers randomly
- Repeatedly:
  - Assign points to nearest center
  - Move centers to center of gravity of their points
- Stop at convergence (no more reassignments)

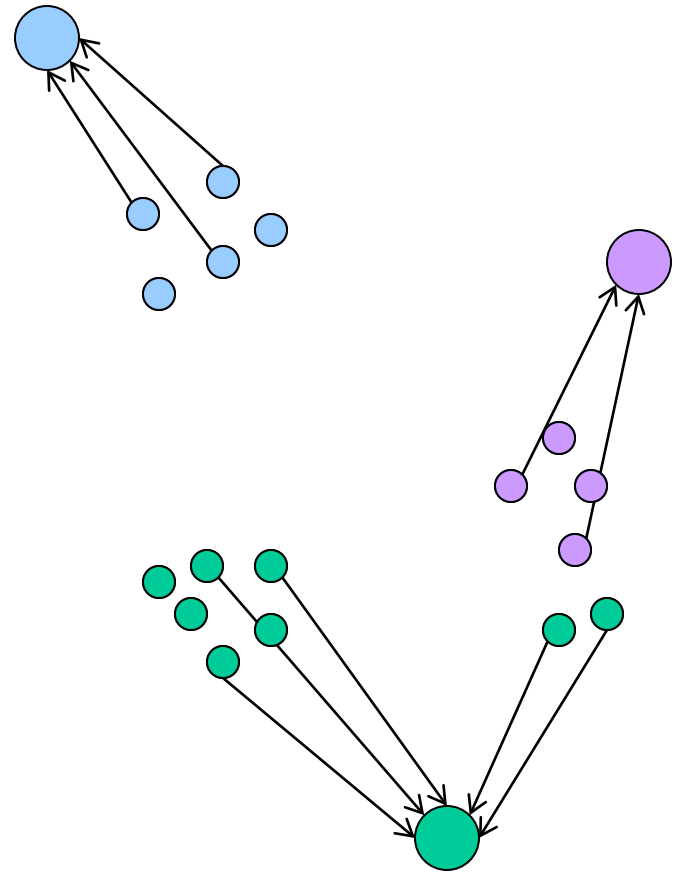
K-Means Algorithm Example

- Randomly Initialize Clusters
- Assign data points to nearest clusters
- Recalculate cluster centers
- Repeat… until convergence



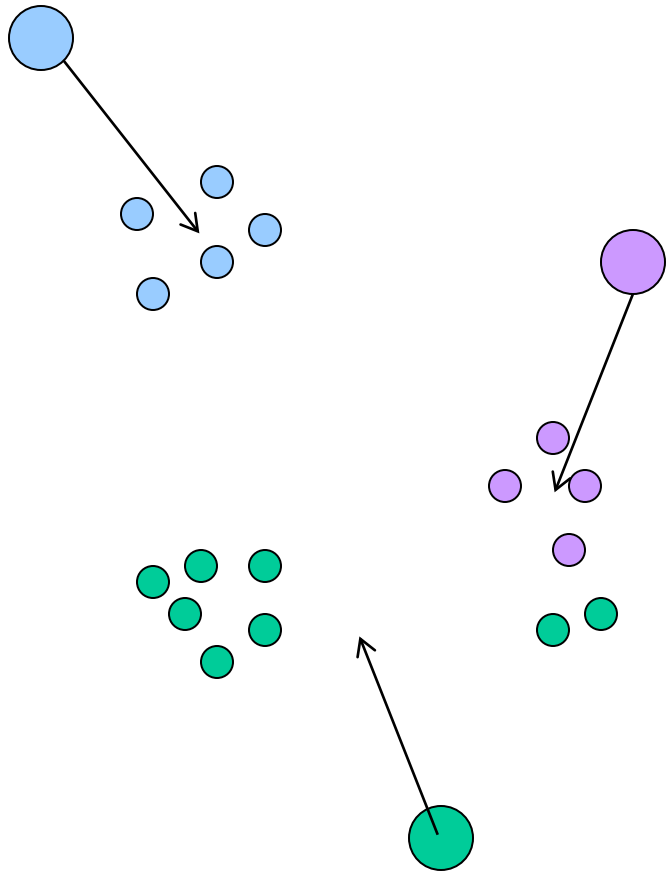
K-Means Algorithm Example

- Randomly Initialize Clusters
- Assign data points to nearest clusters
- Recalculate cluster centers
- Repeat… until convergence



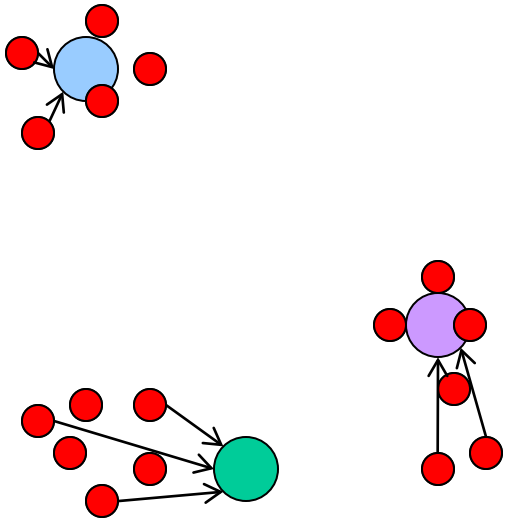
K-Means Algorithm Example

- Randomly Initialize Clusters
- Assign data points to nearest clusters
- Recalculate cluster centers
- Repeat… until convergence



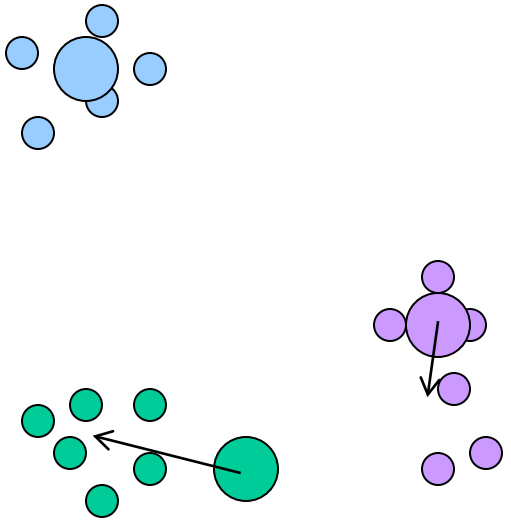
K-Means Algorithm Example

- Randomly Initialize Clusters
- Assign data points to nearest clusters
- Recalculate cluster centers
- Repeat… until convergence



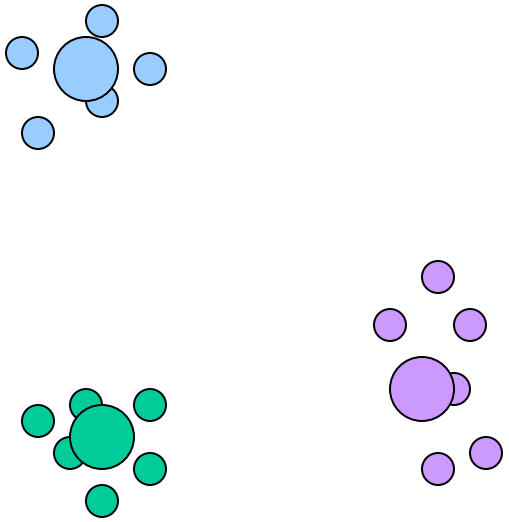
K-Means Algorithm Example

- Randomly Initialize Clusters
- Assign data points to nearest clusters
- Recalculate cluster centers
- Repeat… until convergence

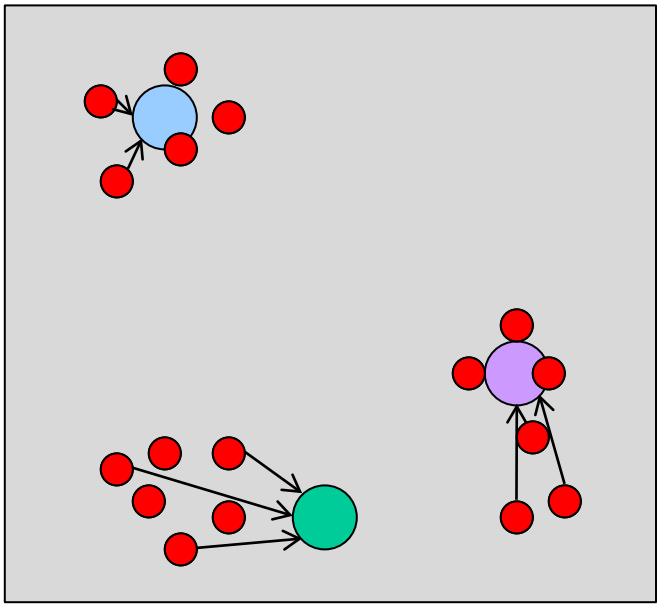


K-Means Algorithm Example

- Randomly Initialize Clusters
- Assign data points to nearest clusters
- Recalculate cluster centers
- Repeat… until convergence

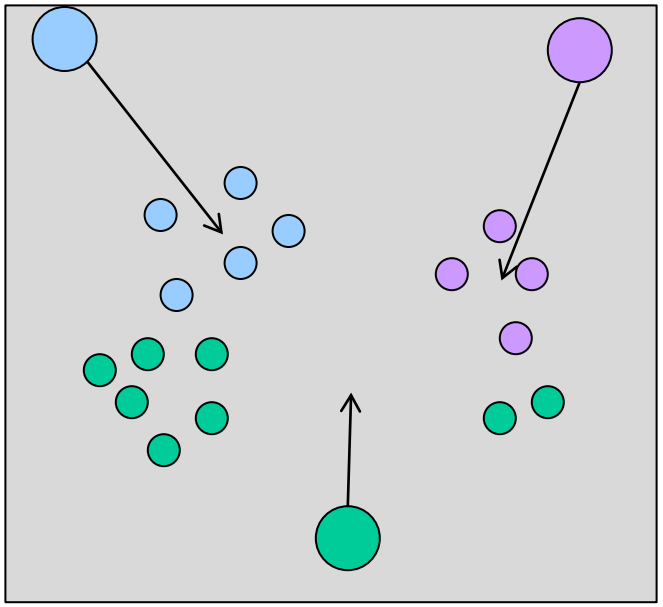


K-means update rules



("M")

("E")



Re-assign each point $\mathbf{x}_i$ to nearest center k

➔ Minimize distance from $\mathbf{x}_i$ to $\boldsymbol{\mu}_k$:

$$d_{i,k} = \left(\mathbf{x}_i - \boldsymbol{\mu}_k\right)^2$$

Update center $\boldsymbol{\mu}_k$ to the mean of the points assigned to it:

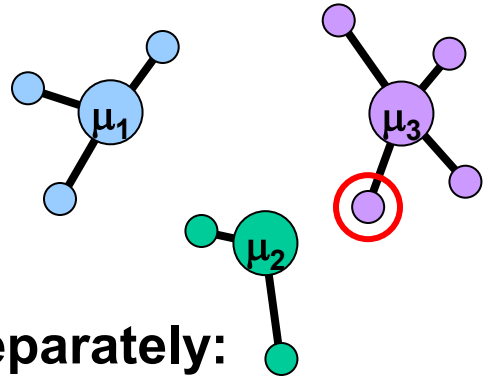
$$\boldsymbol{\mu}_k(n+1) = \sum_{\mathrm{x_i\ with\ label\ j}} \frac{\mathbf{x}_i}{\left|\mathrm{x}^k\right|}$$

where: $\left|\mathrm{x}^k\right| = \#\mathbf{x}_i$ with label k

K-means Optimality Criterion

We can think of K-means as trying to create clusters that minimize a cost criterion associated with the size of the cluster

$$\text{COST}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_n) = \sum_{\boldsymbol{\mu}_k} \sum_{\mathbf{x}_i \text{ with label k}} (\mathbf{x}_i - \boldsymbol{\mu}_k)^2$$



To achieve this, minimize each cluster term separately:

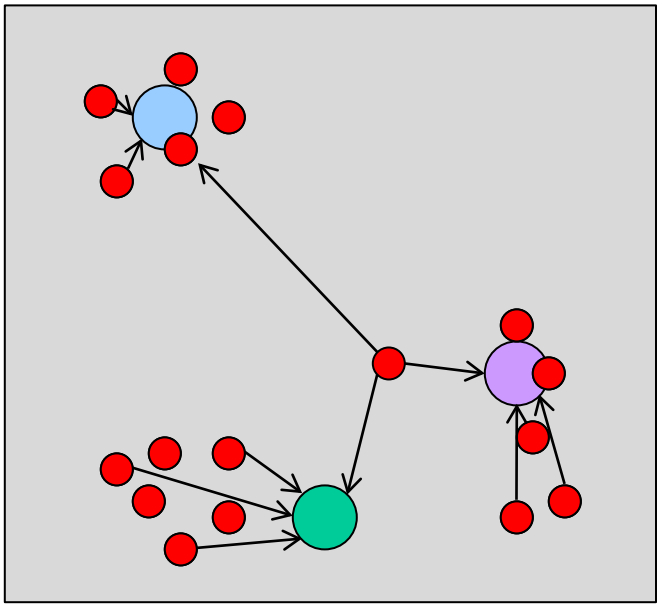
$$\sum_{\mathbf{x}_i \text{ with label k}} (\mathbf{x}_i - \boldsymbol{\mu}_k)^2 = \sum_{\mathbf{x}_i \text{ with label k}} \mathbf{x}_i^2 - 2\mathbf{x}_i\mathbf{u}_k + \boldsymbol{\mu}_k^2 = \sum \mathbf{x}_i^2 - \mathbf{u}_k \sum 2\mathbf{x}_i + \left|\mathbf{x}^k\right| \mathbf{u}_k^2$$

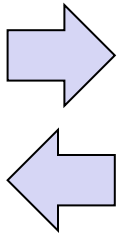
Optimum $\mathbf{u}_k = \sum_{\mathbf{x}_i \text{ with label k}} \frac{\mathbf{x}_i}{\left|\mathbf{x}^k\right|}$, the centroid

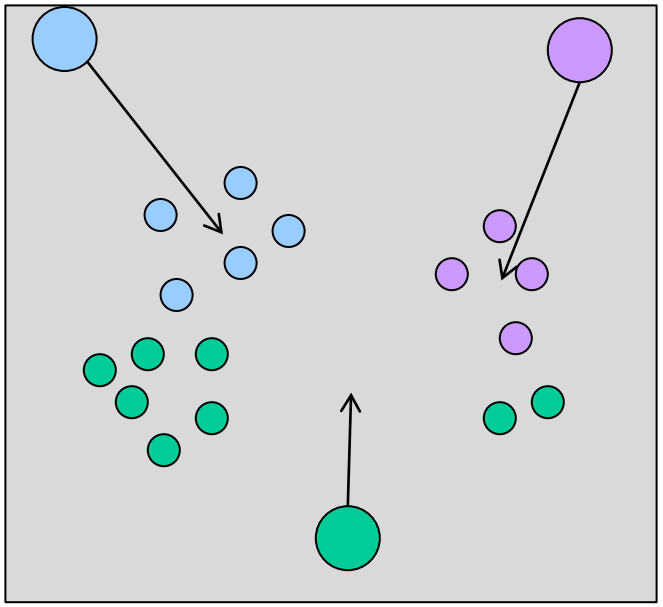
However: Some points can be almost halfway between two centers ➔ Assign partial weights

Fuzzy K-means

Fuzzy K-means update rule







Re-assign each point $\mathbf{x}_i$ to **all** centers, **weighted by distance**

➔ For each point calculate the probability of membership for each category K:

$$P(\text{label K} \mid \mathbf{x}_i, \boldsymbol{\mu}_k)$$

Update center $\boldsymbol{\mu}_k$ to the **weighted mean** of the points assigned to it:

$$\boldsymbol{\mu}_k(n+1) = \sum_{\mathbf{x}_i \text{ with label j}} \mathbf{x}_i \; P(\boldsymbol{\mu}_k \mid \mathbf{x}_i)^b \Big/ \sum_{\mathbf{x}_i \text{ with label j}} P(\boldsymbol{\mu}_k \mid \mathbf{x}_i)^b$$

Regular K-Means is a special case of fuzzy k-means where:

$$P(\text{label K} \mid \mathbf{x}_i, \boldsymbol{\mu}_k) = \begin{cases} 1 & \text{if } \mathbf{x}_i \text{ is closest to } \boldsymbol{\mu}_k \\ 0 & \textbf{otherwise} \end{cases}$$

Today: Gene Expression Clustering & Classification

1. **Introduction to gene expression analysis**
   - Technology: microarrays vs. RNAseq. From reads to transcripts, expr levels
   - Expr matrix. Supervised vs. unsupervised learning. Clustering/Classification

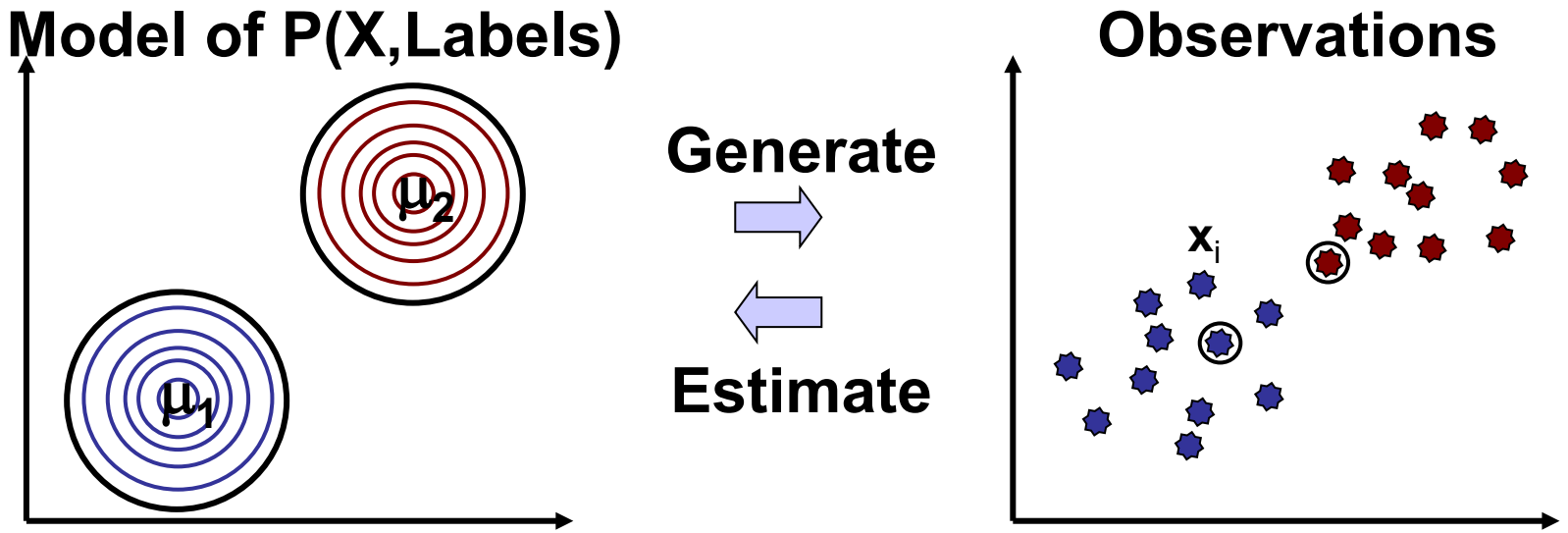
2. **K-means clustering (clustering by partitioning)**
   - Algorithmic formulation: Update rule, optimality criterion. Fuzzy k-means.
   - Machine learning formulation: Generative models, Expectation Maximization.

3. **Hierarchical Clustering (clustering by agglomeration)**
   - Basic algorithm, Distance measures. Evaluating clustering results

4. **Naïve Bayes classification (generative approach to classification)**
   - Discriminant function: class priors, and class-conditional distributions
   - Training and testing, Combine mult features, Classification in practice

5. **(optional) Support Vector Machines (discriminative approach)**
   - SVM formulation, Margin maximization, Finding the support vectors
   - Non-linear discrimination, Kernel functions, SVMs in practice

K-Means as a Generative Model

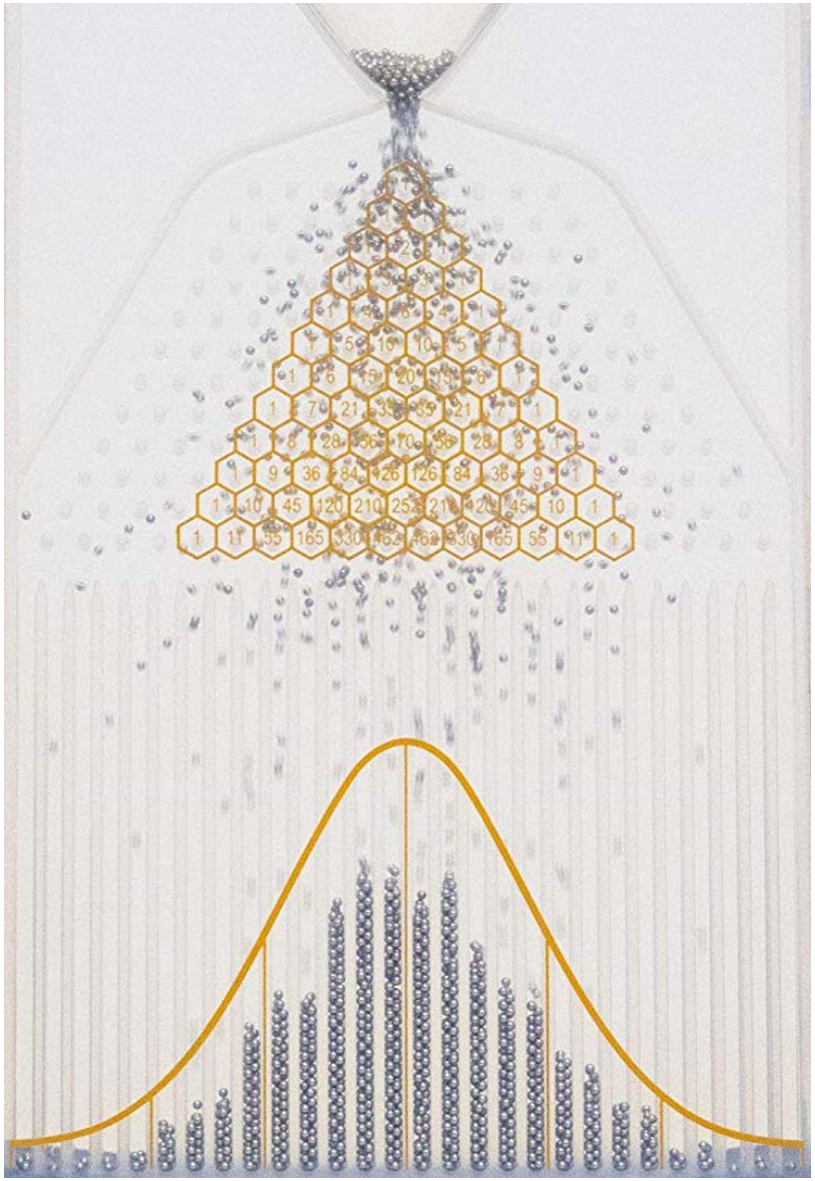


Samples drawn from normal distributions with unit variance - a *Gaussian Mixture Model*

$$P\left(\mathbf{x}_i \mid \mathbf{u}_j\right) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{\left(\mathbf{x}_i - \mathbf{u}_j\right)^2}{2}\right\}$$

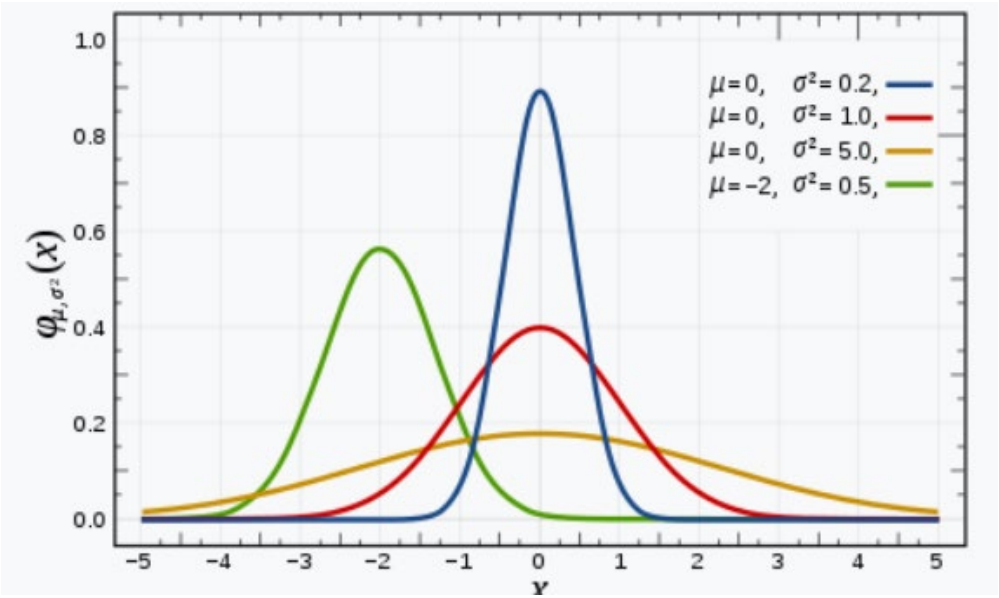
Given only samples, how do we estimate max lik model params: (1) centroid definitions, (2) point assignments?

One-dimensional Gaussian



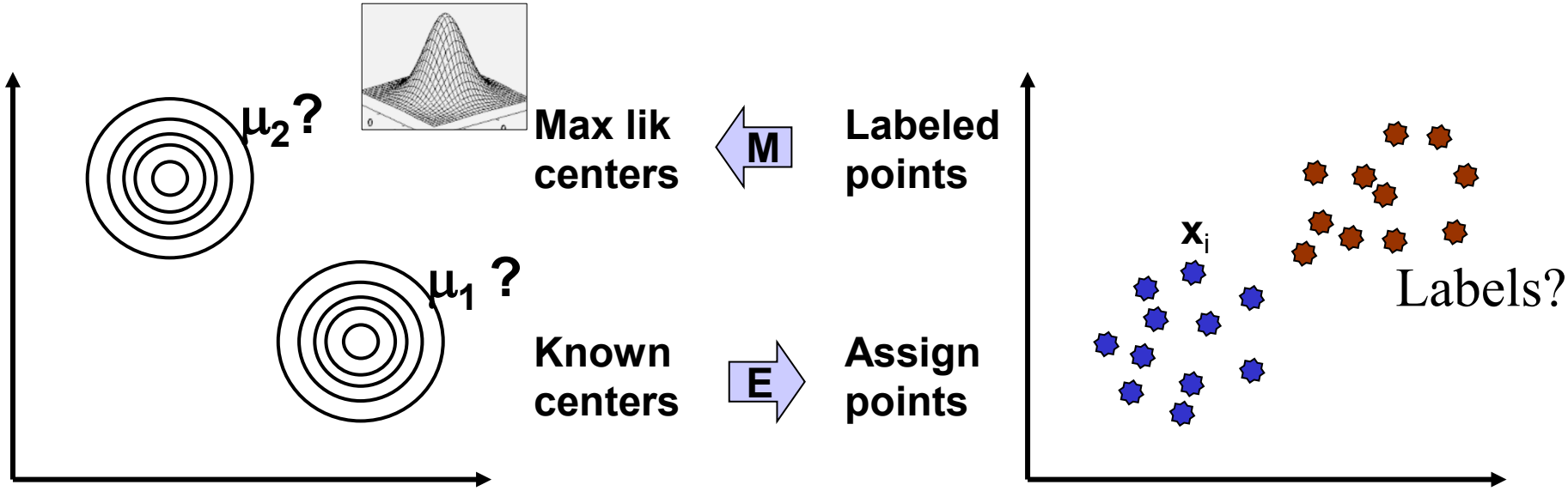
Balls falling at random hit nails push left or right. Equal chance of falling left or right at every row. The number of same-side falls drops geometrically. In the limit, the balls distribution approximates a normal distribution

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



EM solution: iteratively estimate one from the other

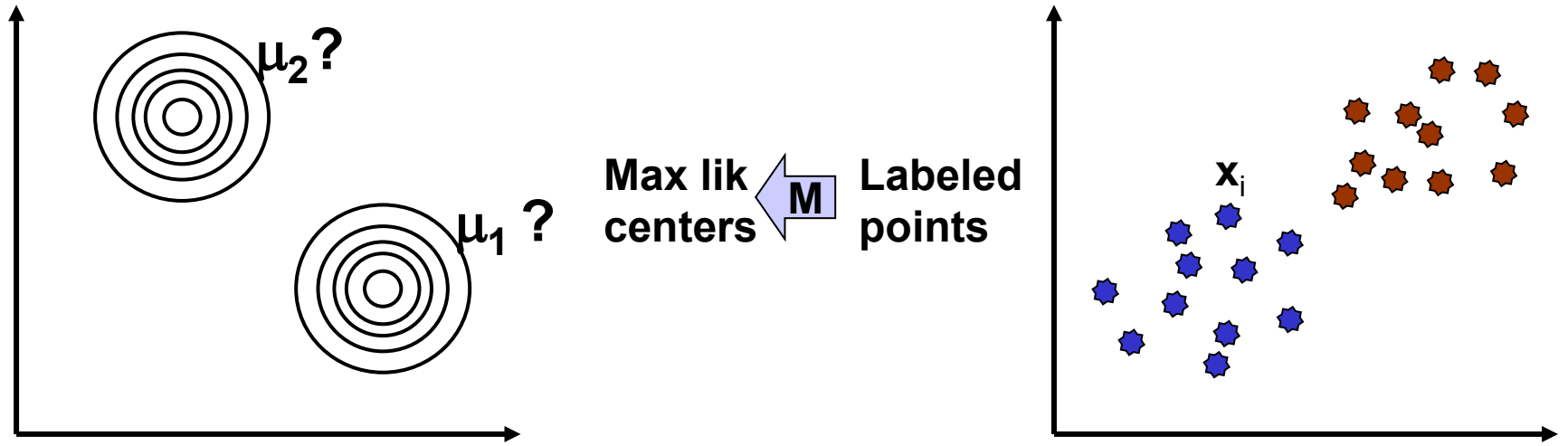
E step: If centers are known ➔ Estimate memberships
M step: If assignments known ➔ Compute centroids



Choose μ_k and labels that maximize P(data|model)

Solution is exactly the k-means algorithm!

M step: assignments known ➔ compute centroids



Choose μ_k and labels that maximize P(data|model)

$$\arg\max_{\mu}\left\{\log\prod_i P\left(\mathbf{x}_i \mid \boldsymbol{\mu}\right)\right\} = \arg\max_{\mu}\sum_i\left\{-\frac{1}{2}\left(\mathbf{x}_i-\mathbf{u}\right)^2+\log\left(\frac{1}{\sqrt{2\pi}}\right)\right\}$$

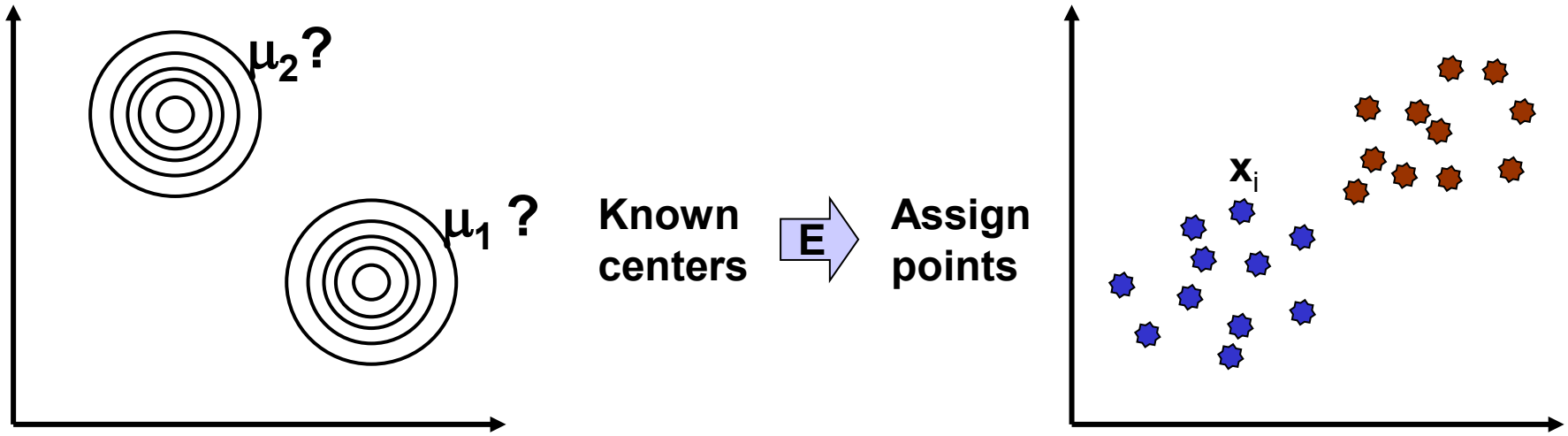
Seeking the max likelihood estimate of the cluster mean

$$= \arg\min_{\mu}\sum_i\left(\mathbf{x}_i-\mathbf{u}\right)^2$$

Solution is the centroid of the x_i

EM solution ⟺ **Equivalent** ⟺ **K-means solution**

E step: centers known ➔ Estimate memberships



Choose μ_k and labels that maximize P(data|model)

$$\arg\max_k P_k(\mathbf{x}_i \mid \boldsymbol{\mu}_i) = \arg\max_k \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{(\mathbf{x}_i - \mathbf{u}_k)^2}{2}\right\} = \arg\min_k (\mathbf{x}_i - \mathbf{u}_k)^2$$

Seeking the label k that maximizes likelihood of point

Solution is the nearest center

EM solution ⟺ **K-means solution**

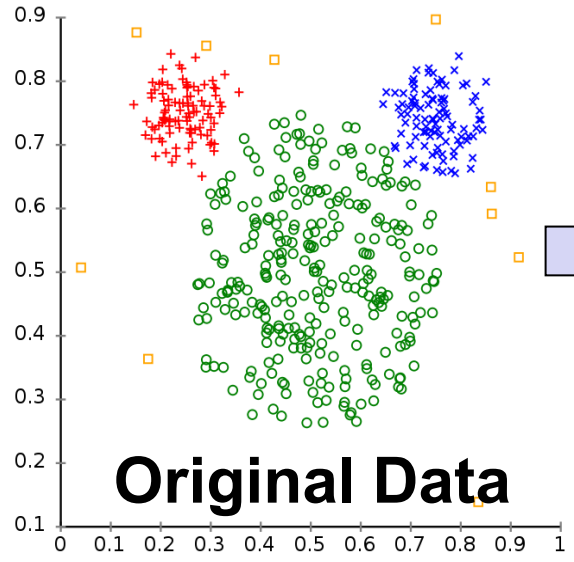
Equivalent

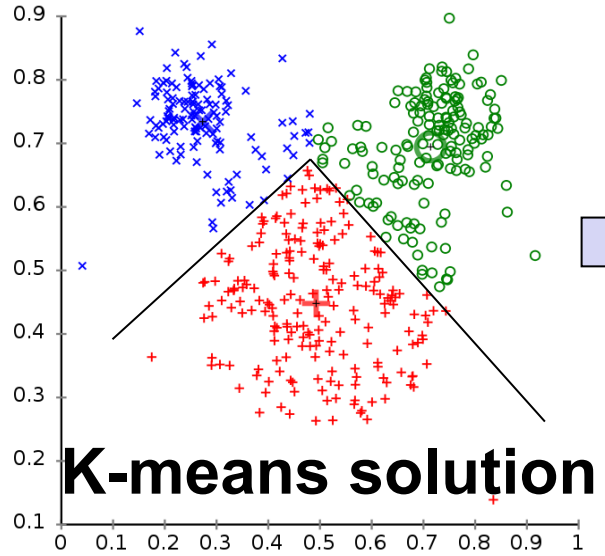
Algorithmic vs. machine learning formulations

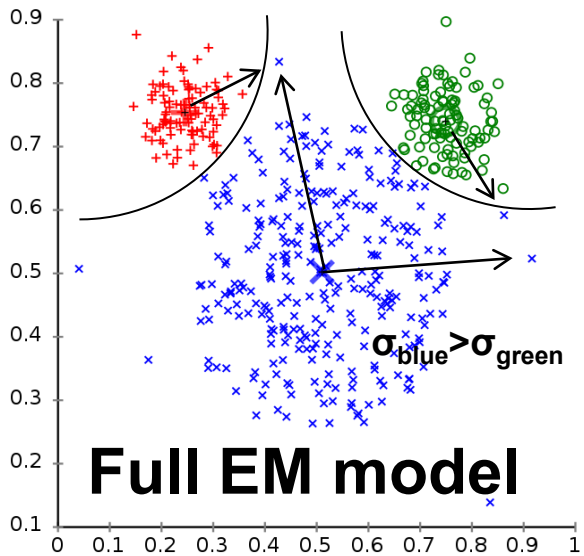
	K-means		Fuzzy K-means	
	algorithmic formulation	probabilistic interpretation	algorithmic formulation	probabilistic interpretation
Initialization	Initialize K centers $\boldsymbol{\mu}_k$	Initialize model parameters	Initialize K centers $\boldsymbol{\mu}_k$	Initialize model parameters
E-step: Estimate prob of hidden labels (point assignments to classes)	Assign $\mathbf{x}_i$ label of nearest center distance $d_{i,k} = (\mathbf{x}_i - \boldsymbol{\mu}_k)^2$	Estimate most likely missing label given previous parameters	Calculate probability of membership for each point to each class $P(\text{label K} \mid \mathbf{x}_i, \boldsymbol{\mu}_k)$	Estimate probability over missing labels given previous parameters
M-step: Update params to max likelihood estimates given assignments	Move $\boldsymbol{\mu}_k$ to centroid of all points with that label	Choose new max likelihood params given points in label	Move $\boldsymbol{\mu}_k$ to weighted centroid of all points, each weighted by P(label)	Choose new params to maximize expected likelihood given label estimates
Iteration	Iterate	Iterate	Iterate	Iterate

P(x|Model) *guaranteed* to increase each iteration of EM algo

EM is much more general than fuzzy K-means







	K-means solution	EM generalization
Cluster sizes	**Uniform** priors	Class priors $P(class_i)$
Spread of points	**Unit** distance function	$Gaussian\ (\mu_i, \sigma_i)$
Cluster shape	**Symmetric**, x-y indpt	Co-variance matrix $q_{jk} = \frac{1}{N}\sum_{i=1}^{N}(x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k)$
Label assignment	K-means: Pick **max** Fuzzy: Full **density**	EM: Full **density** Gibbs: **sample** posterior

Three options for assigning points, and their parallels across K-means, HMMs, Motifs

Update rule	Update assignments (E step) ➔ Estimate hidden labels	Algorithm implementing E step in each of the three settings			Update model parameters (M step) ➔ max likelihood
		Expression clustering	**HMM learning**	**Motif discovery**	
The hidden label is:		Cluster labels	State path π	Motif positions	
Pick a best	Assign each point to best label	**K-means:** Assign each point to nearest cluster	**Viterbi training:** label sequence with best path	**Greedy:** Find best motif match in each sequence	Average of those points assigned to label
Average all	Assign each point to all labels, probabilistically	**Fuzzy K-means:** Assign to all clusters, weighted by proximity	**Baum-Welch training:** label sequence w all paths (posterior decoding)	**MEME:** Use all positions as a motif occurrence weighed by motif match score	Average of all points, weighted by membership
Sample one	Pick one label at random, based on their relative probability	**N/A:** Assign to a random cluster, sample by proximity	**N/A:** Sample a single label for each position, according to posterior prob.	**Gibbs sampling:** Use one position for the motif, by sampling from the match scores	Average of those points assigned to label(a sample)

Today: Gene Expression Clustering & Classification

1. **Introduction to gene expression analysis**
   - Technology: microarrays vs. RNAseq. From reads to transcripts, expr levels
   - Expr matrix. Supervised vs. unsupervised learning. Clustering/Classification

2. **K-means clustering (clustering by partitioning)**
   - Algorithmic formulation: Update rule, optimality criterion. Fuzzy k-means.
   - Machine learning formulation: Generative models, Expectation Maximization.

3. **Hierarchical Clustering (clustering by agglomeration)**
   - Basic algorithm, Distance measures. Evaluating clustering results

4. **Naïve Bayes classification (generative approach to classification)**
   - Discriminant function: class priors, and class-conditional distributions
   - Training and testing, Combine mult features, Classification in practice

5. **(optional) Support Vector Machines (discriminative approach)**
   - SVM formulation, Margin maximization, Finding the support vectors
   - Non-linear discrimination, Kernel functions, SVMs in practice

Challenge of K-means: picking K

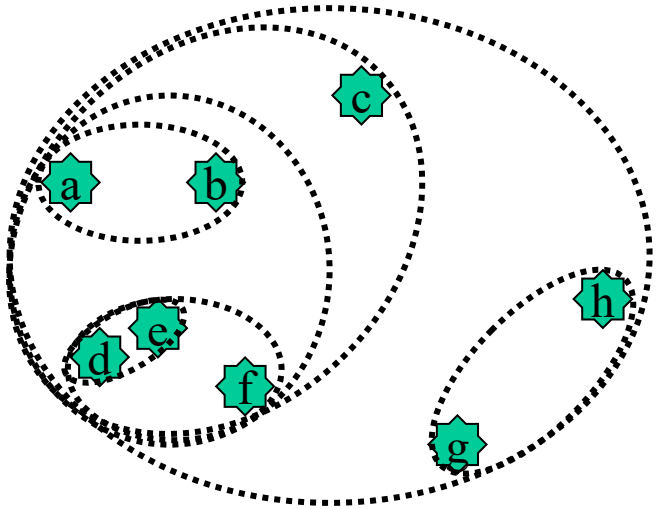
- How do we select K?
  - We can always make clusters “more compact” by increasing K
  - e.g. What happens is if K=number of data points?
  - What is a meaningful improvement?
- Hierarchical clustering side-steps this issue

Two approaches to clustering

- Partitioning (e.g. k-means)
  - Divides objects into non-overlapping clusters such that each data object is in exactly one subset
- Agglomerative (e.g. hierarchical clustering)
  - A set of nested clusters organized as a hierarchy

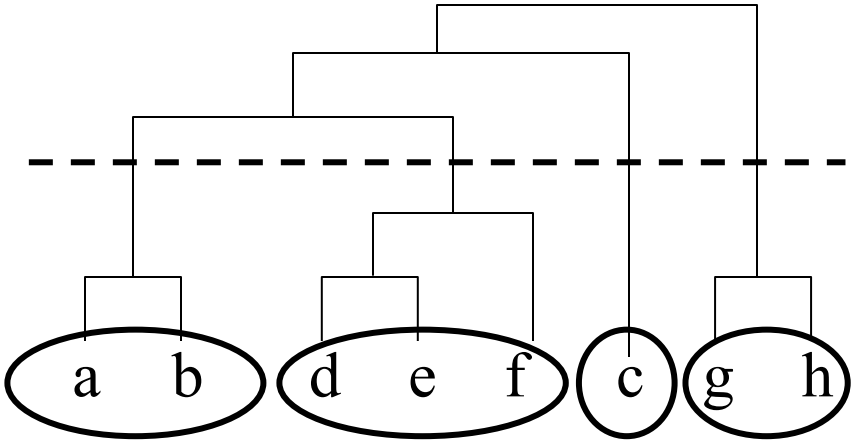
Hierarchical clustering

Most widely used algorithm for expression data



- Start with each point in a separate cluster
- At each step:
  - Choose the pair of **closest clusters**
  - Merge

→ Phylogeny (UPGMA)

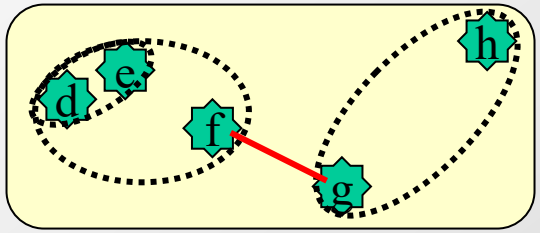


Unweighted **P**air **G**roup **M**ethod with **A**rithmetic-mean

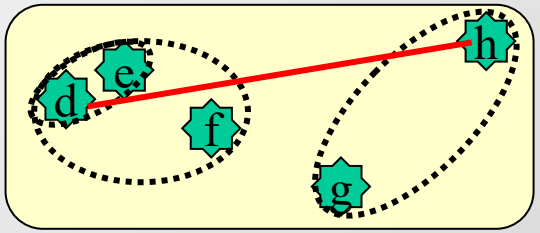
Select a “cut level” to create disjoint clusters

Distance between clusters

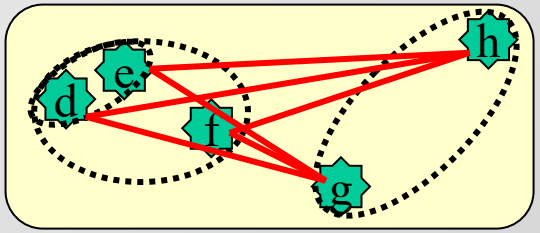
- $CD(X,Y)=\min_{x \in X,\, y \in Y} D(x,y)$
  Single-link method



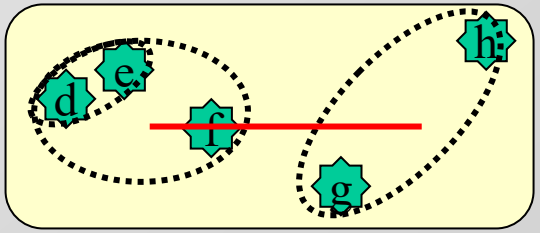
- $CD(X,Y)=\max_{x \in X,\, y \in Y} D(x,y)$
  Complete-link method



- $CD(X,Y)=\text{avg}_{x \in X,\, y \in Y} D(x,y)$
  Average-link method



- $CD(X,Y)=D(\text{avg}(X), \text{avg}(Y))$
  Centroid method



Cluster distance affects both results and runtime

Point-to-point (Dis)Similarity Measures

Table 1 Gene expression similarity measures

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

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

D'haeseleer (2005) Nat Biotech

Cluster-to-cluster distance as a function of point-to-point

Today: Gene Expression Clustering & Classification

1. **Introduction to gene expression analysis**
   - Technology: microarrays vs. RNAseq. From reads to transcripts, expr levels
   - Expr matrix. Supervised vs. unsupervised learning. Clustering/Classification

2. **K-means clustering (clustering by partitioning)**
   - Algorithmic formulation: Update rule, optimality criterion. Fuzzy k-means.
   - Machine learning formulation: Generative models, Expectation Maximization.

3. **Hierarchical Clustering (clustering by agglomeration)**
   - Basic algorithm, Distance measures. Evaluating clustering results

4. **Naïve Bayes classification (generative approach to classification)**
   - Discriminant function: class priors, and class-conditional distributions
   - Training and testing, Combine mult features, Classification in practice

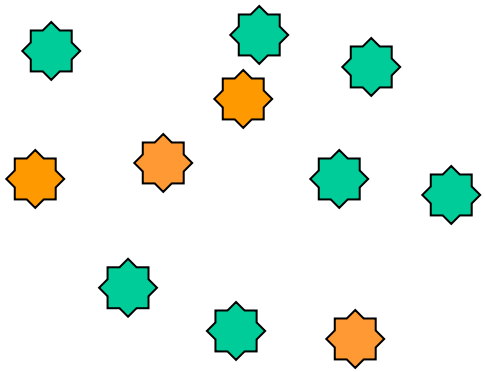
5. **(optional) Support Vector Machines (discriminative approach)**
   - SVM formulation, Margin maximization, Finding the support vectors
   - Non-linear discrimination, Kernel functions, SVMs in practice

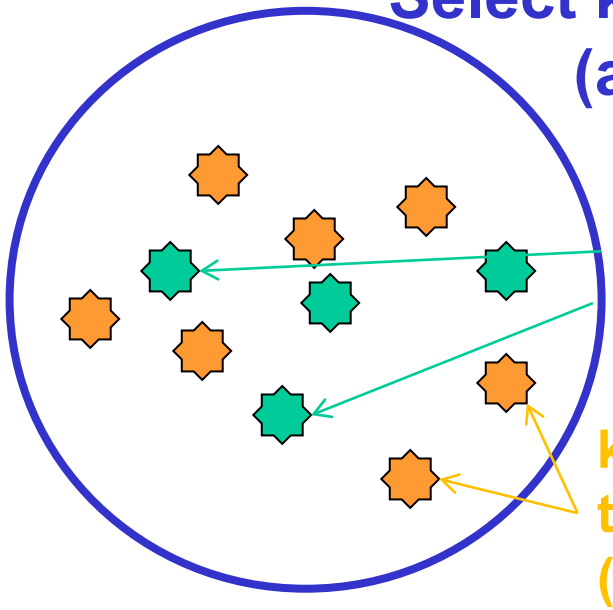
Evaluating Cluster Performance

In general, it depends on your goals in clustering

- Robustness
  - Select random samples from data set and cluster
  - Repeat
  - Robust clusters show up in all clusters
- Category Enrichment
  - Look for categories of genes “over-represented” in particular clusters
  - Also used in Motif Discovery

Evaluating clusters – Hypergeometric Distribution





$$P(pos \geq r) = \sum_{m \geq r} \frac{\binom{p}{m}\binom{N-p}{k-m}}{\binom{N}{k}}$$

P-value of uniformity in computed cluster

Prob that a randomly chosen set of k experiments would result in m positive and k-m negative

- N experiments, **p labeled +**, **(N-p) –**
- **Cluster: k elements**, **m labeled +**, **k-m labeled -**
- P-value of *single* cluster containing k elements of which at least r are **+**

Today: Gene Expression Clustering & Classification

1. **Introduction to gene expression analysis**
   - Technology: microarrays vs. RNAseq. From reads to transcripts, expr levels
   - Expr matrix. Supervised vs. unsupervised learning. Clustering/Classification

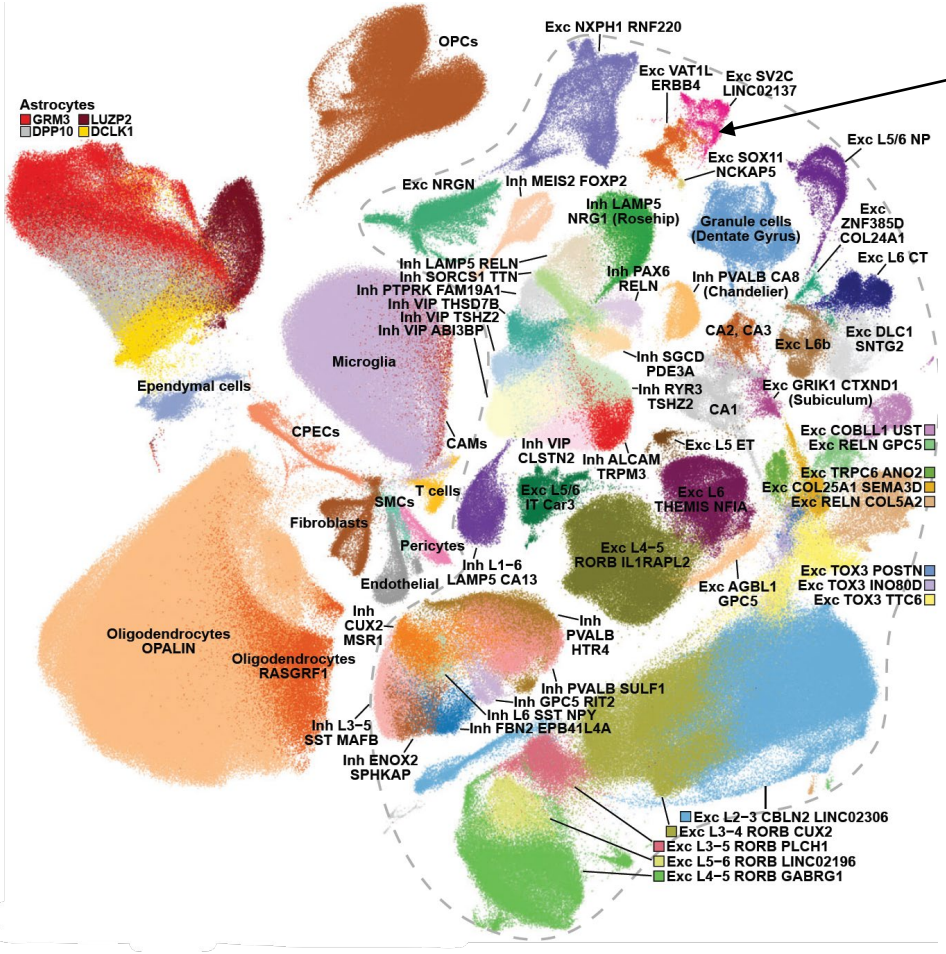
2. **K-means clustering (clustering by partitioning)**
   - Algorithmic formulation: Update rule, optimality criterion. Fuzzy k-means.
   - Machine learning formulation: Generative models, Expectation Maximization.

3. **Hierarchical Clustering (clustering by agglomeration)**
   - Basic algorithm, Distance measures. Evaluating clustering results
   - Clustering beyond numbers: Text, Word2Vec, Latent Embeddings

4. **Naïve Bayes classification (generative approach to classification)**
   - Discriminant function: class priors, and class-conditional distributions
   - Training and testing, Combine mult features, Classification in practice

5. **(optional) Support Vector Machines (discriminative approach)**
   - SVM formulation, Margin maximization, Finding the support vectors
   - Non-linear discrimination, Kernel functions, SVMs in practice

Multi-modal Embeddings of 2.4 million Human Cells

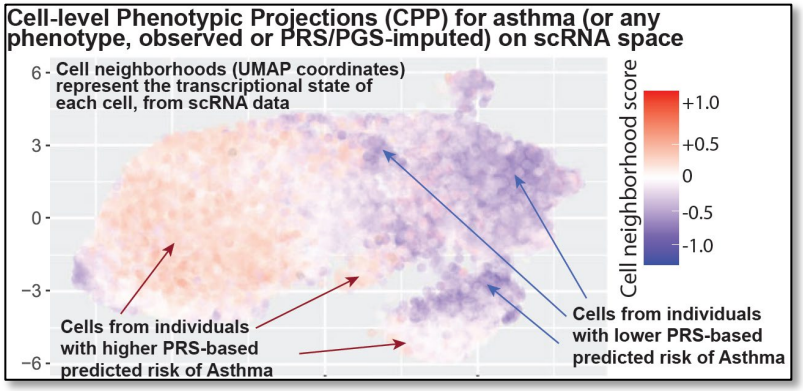


Every dot is a 20,000-dimensional vector

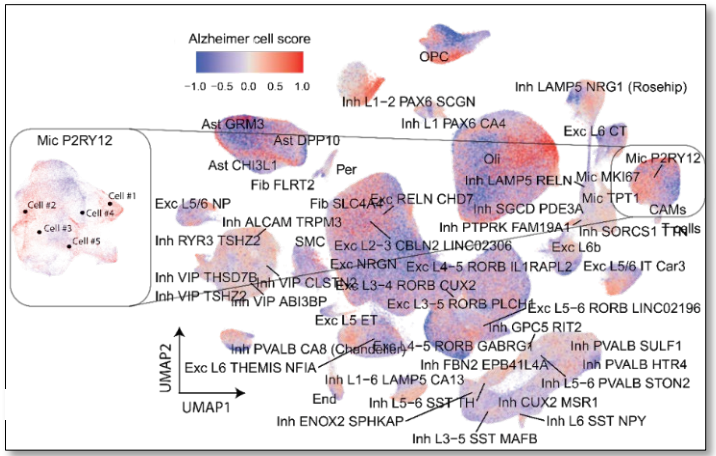
Integrate 2.4 million 'documents'

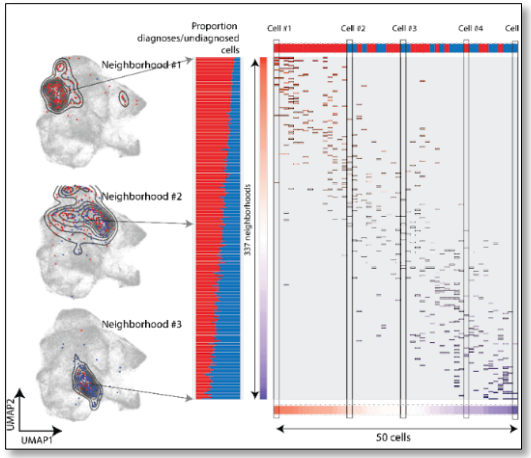
Impact:

- **Understand gene relationships**
- **Understand impact of phenotype**
- **Understand impact of age, sex**
- **Understand pathway correlations**
- **Understand gene co-variation**
- **Map phenotype to cell space**



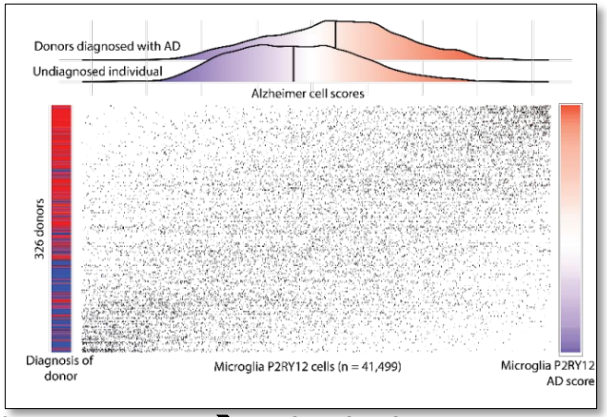
Cell-Projected Phenotypes: heterogeneity, cell-specificity, gradations

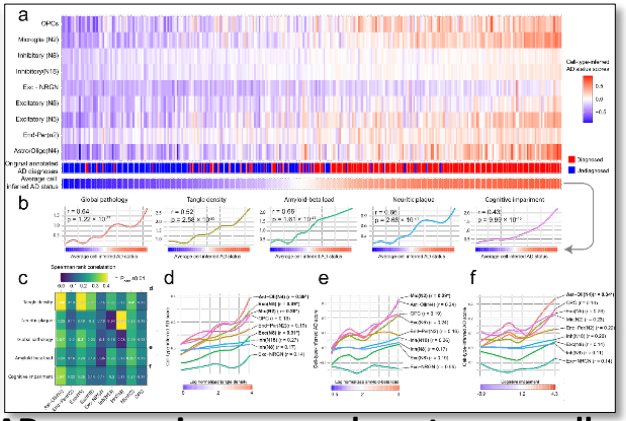


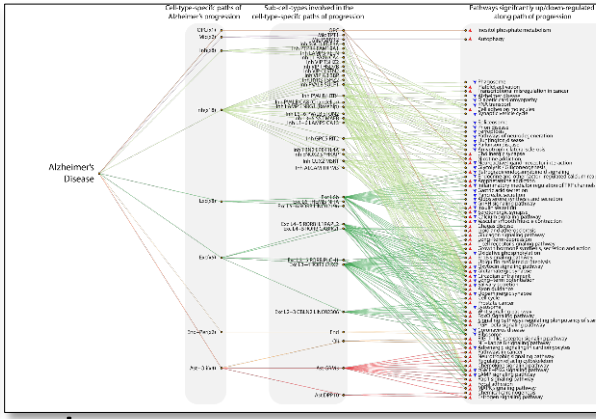


Cell Transcriptional neighborhoods: phenotype enrichments
For each phenotype, different neighborhoods manifest

Neighborhood memberships ➔ Cell annotation



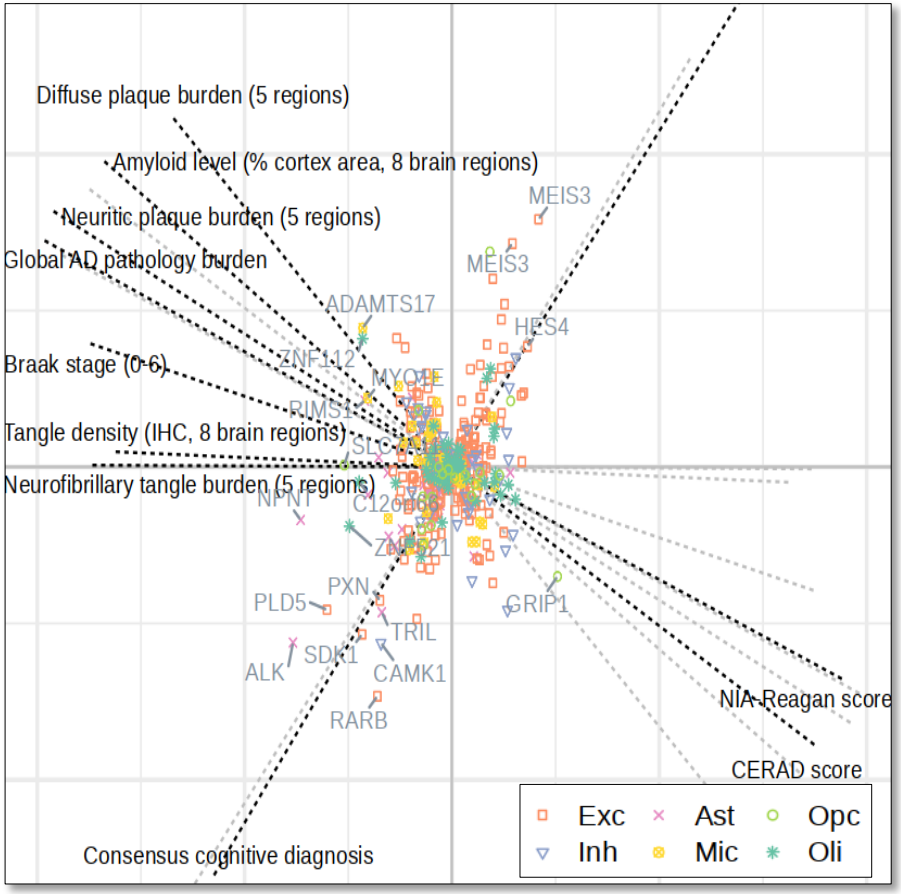




Cell annotations ➔ Individual annotations

AD progression across phenotypes + cells + pathways

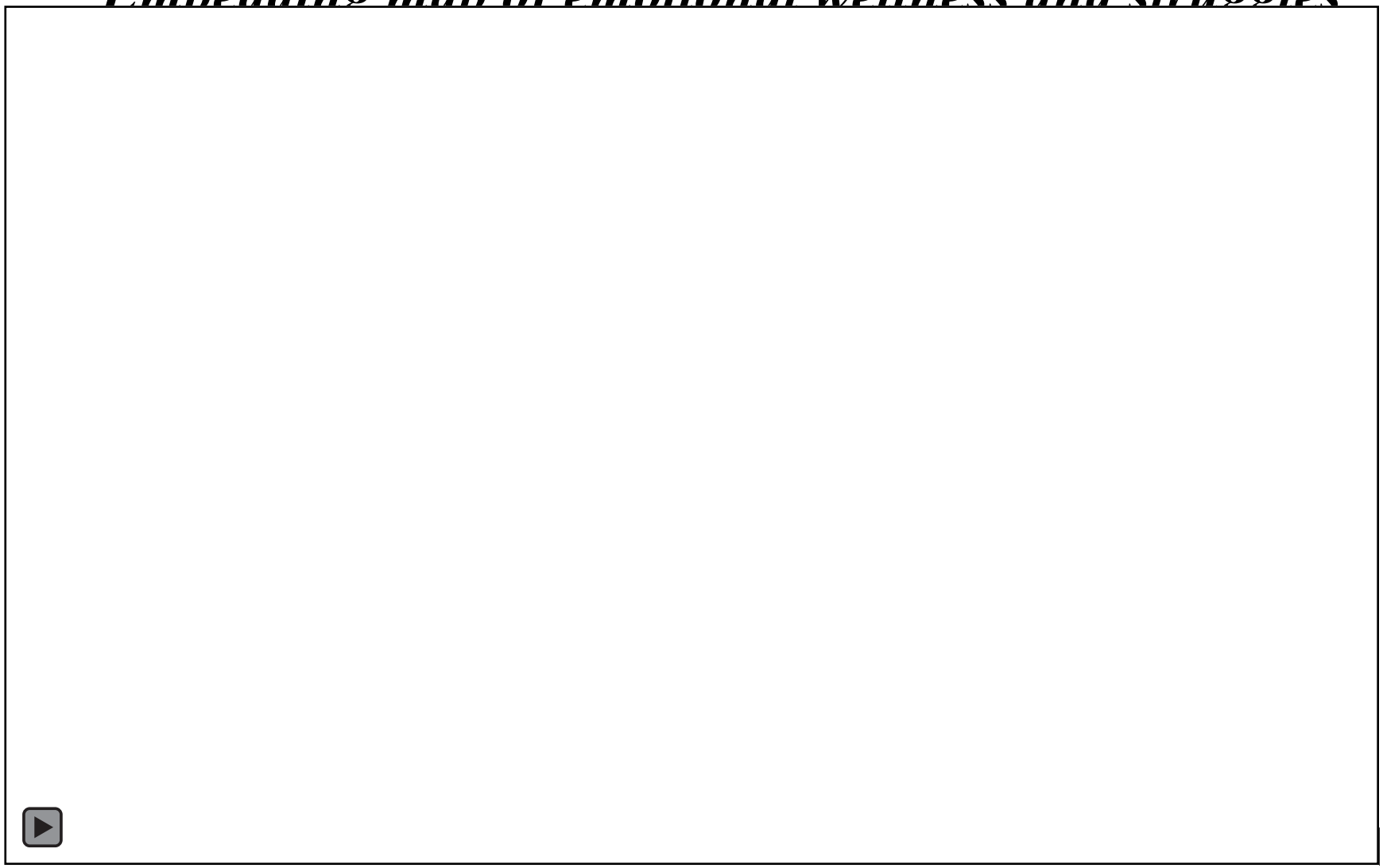
Multi-modal embeddings of 430 Alzheimer's patients



Every person is a point, based on their cellular expression patterns
Reveal impact of gene expression, phenotype, genotype

Use 'cartography' from gene expression to map phenotype
Common foundational map ➔ reason about health impact

Embedding map of emotional wellness and struggles

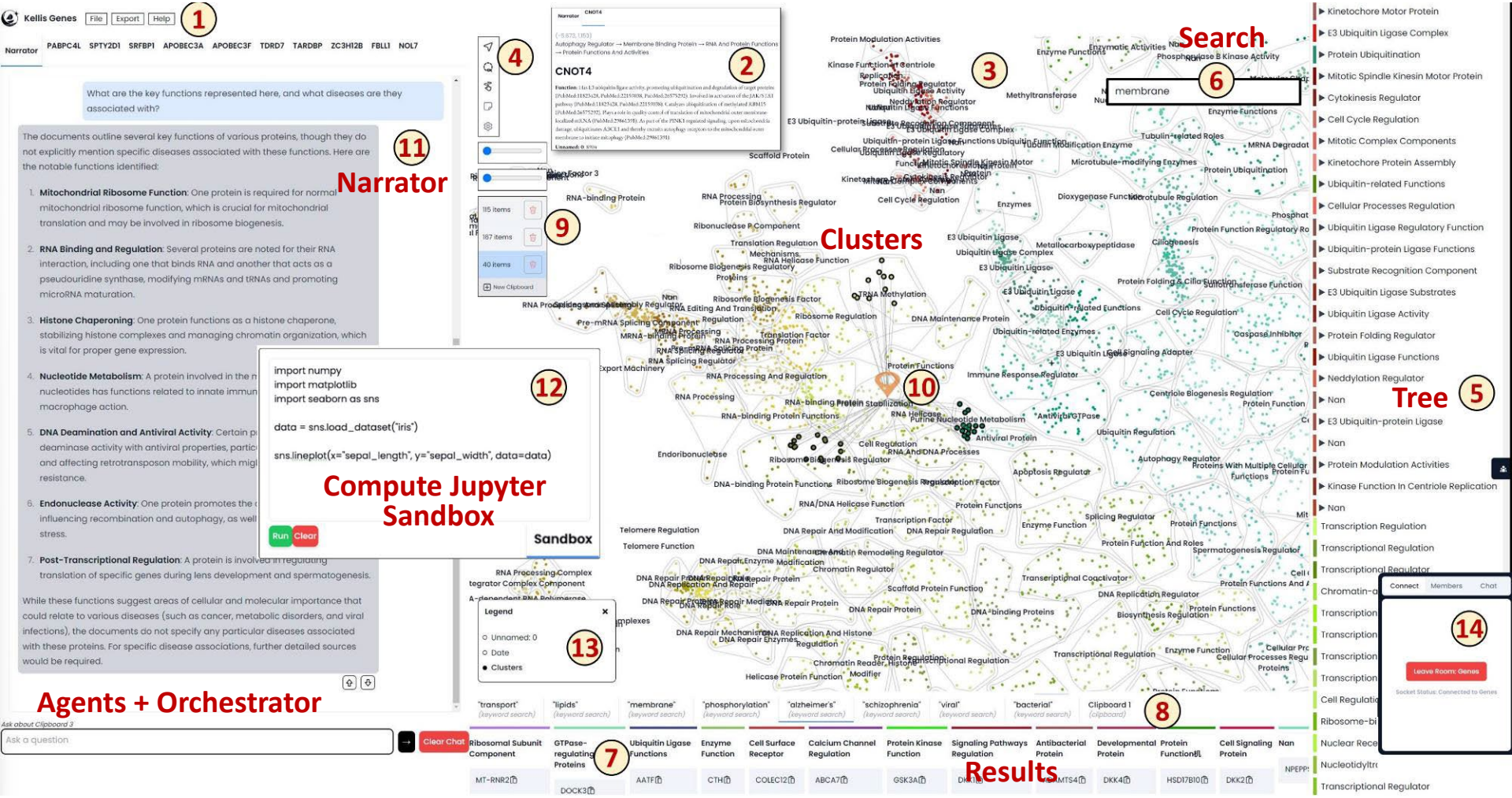


https://alpha.withmantis.com/space/91df84be-1ee5-4f03-85e9-fa1855375a18/

Interactive embedding map of human gene function

https://alpha.withmantis.com/space/807acd2d-001f-467d-ad77-e21e437ed221/

Mantis: AI-Powered Interactive Visual Data Science Workbench



Lex Clips: https://alpha.withmantis.com/space/382814c1-45c3-4d0d-9101-ad52b7ce7a27/

Today: Gene Expression Clustering & Classification

1. **Introduction to gene expression analysis**
   - Technology: microarrays vs. RNAseq. From reads to transcripts, expr levels
   - Expr matrix. Supervised vs. unsupervised learning. Clustering/Classification

2. **K-means clustering (clustering by partitioning)**
   - Algorithmic formulation: Update rule, optimality criterion. Fuzzy k-means.
   - Machine learning formulation: Generative models, Expectation Maximization.

3. **Hierarchical Clustering (clustering by agglomeration)**
   - Basic algorithm, Distance measures. Evaluating clustering results

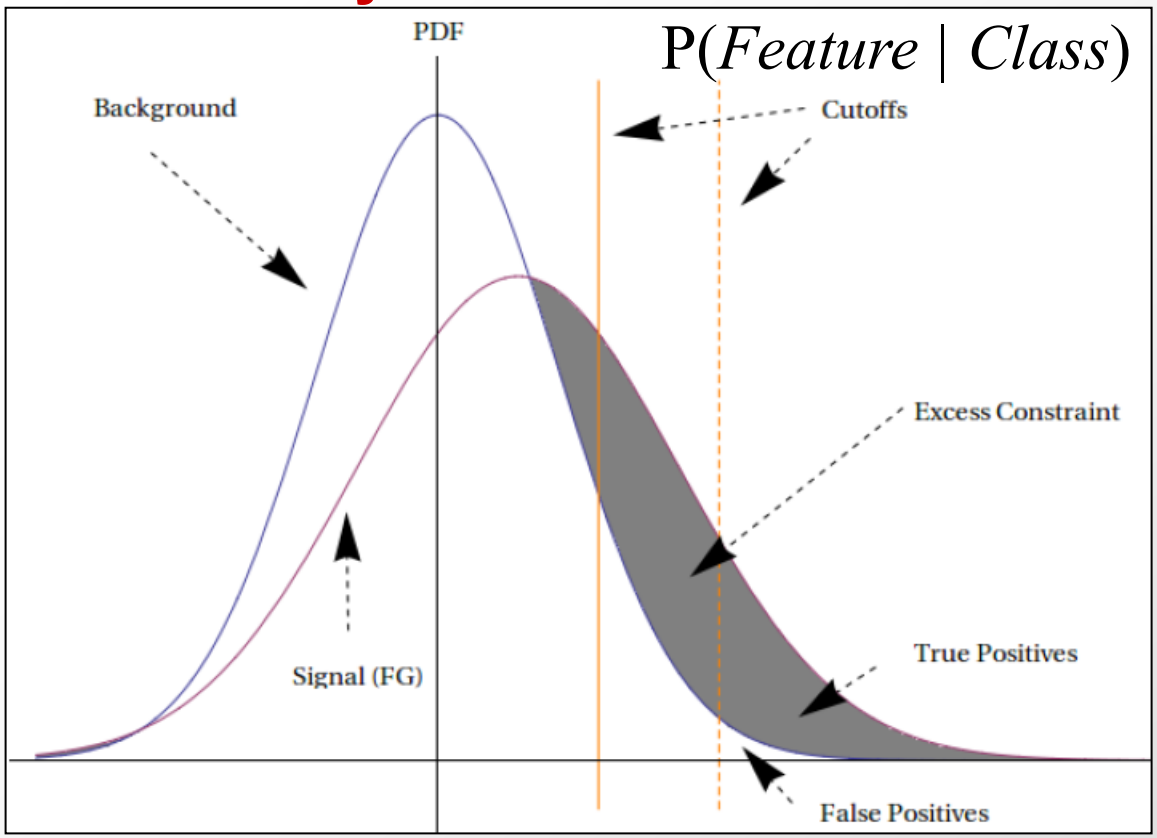
4. **Naïve Bayes classification (generative approach to classification)**
   - Discriminant function: class priors, and class-conditional distributions
   - Training and testing, Combine mult features, Classification in practice

5. **(optional) Support Vector Machines (discriminative approach)**
   - SVM formulation, Margin maximization, Finding the support vectors
   - Non-linear discrimination, Kernel functions, SVMs in practice

Two Approaches to Classification

- Generative
  - Bayesian Classification (e.g. Naïve Bayes)
  - Pose classification problem in prob terms
  - Model feature distribution in different classes
  - Use probability calculus for making decisions
- Discriminative
  - E.g. Support Vector Machines
  - No modeling of underlying distributions
  - Make decisions using distance from boundary
- Example: Gene finding: HMMs vs. CRFs

Bayesian classification with a single feature



Ex 1: DNA repair genes show higher expression during stress

Ex 2: Protein-coding regions show higher conservation levels

Ex 3: Regulatory regions show higher GC-content

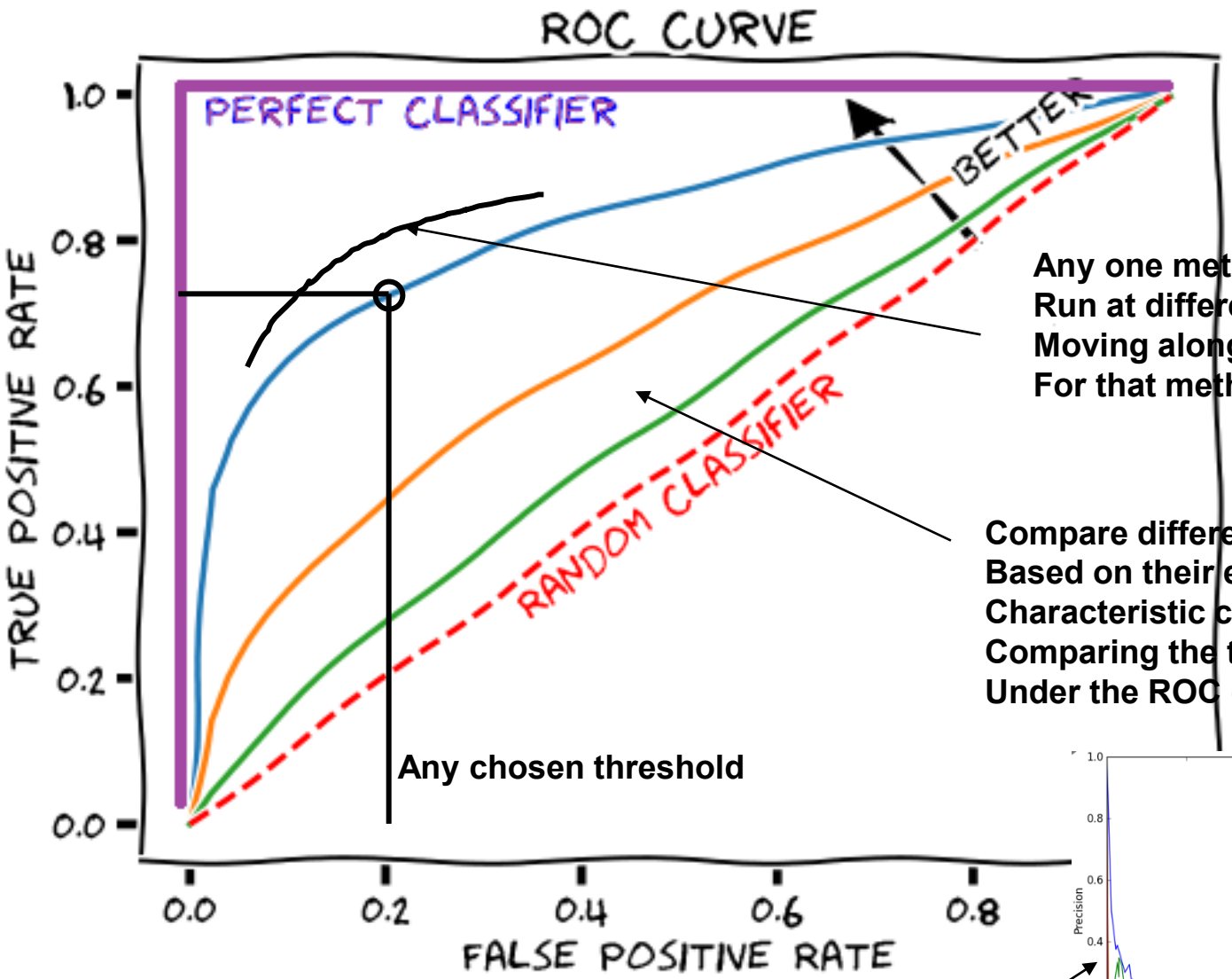
In general: foreground signal vs. background

1. If you know both distributions, how to classify a new example
   - Picking a cutoff. Minimizing classification error. Maximizing posterior prob.
2. If you have many classified examples, how to estimate model params.
   - Parametric vs. non-parametric models. Class-conditional distributions. Priors
3. Bayes' Rule:
   - P(C|F) from P(F|C)
   - Take probability ratios

$$\underbrace{P(Class \mid Feature)}_{\textbf{Posterior}} = \frac{\overbrace{P(Feature \mid Class)}^{\textbf{Likelihood}}\overbrace{P(Class)}^{\textbf{Prior}}}{\underbrace{P(Feature)}_{\textbf{Evidence}}}$$

Classification performance at different thresholds

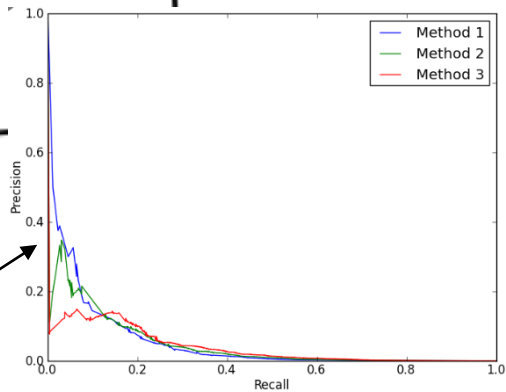
Receiver Operating Characteristic (ROC) Curve



Any one method can be Run at different thresholds, Moving along the ROC line For that method

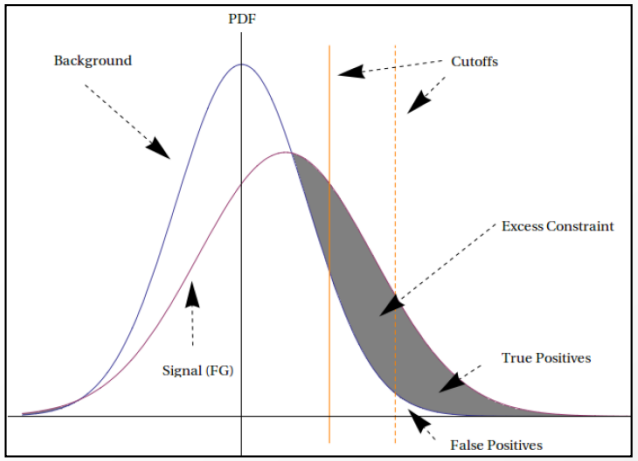
Compare different methods Based on their entire operating Characteristic curve, by Comparing the total Area Under the ROC curve (AUROC)

Any chosen threshold



When datasets are imbalanced, instead use a Precision Recall Curve

Classification problem: Max Probability Class



Select the class that maximizes posterior:

$$\underbrace{P(Class \mid Feature)}_{\textbf{Posterior}} = \frac{\overbrace{P(Feature \mid Class)}^{\textbf{Likelihood}}\overbrace{P(Class)}^{\textbf{Prior}}}{\underbrace{P(Feature)}_{\textbf{Evidence}}}$$

Maximum-A-Posteriori (MAP) estimates

$$BestClass = argmax_C\, P(Class|Feature)$$

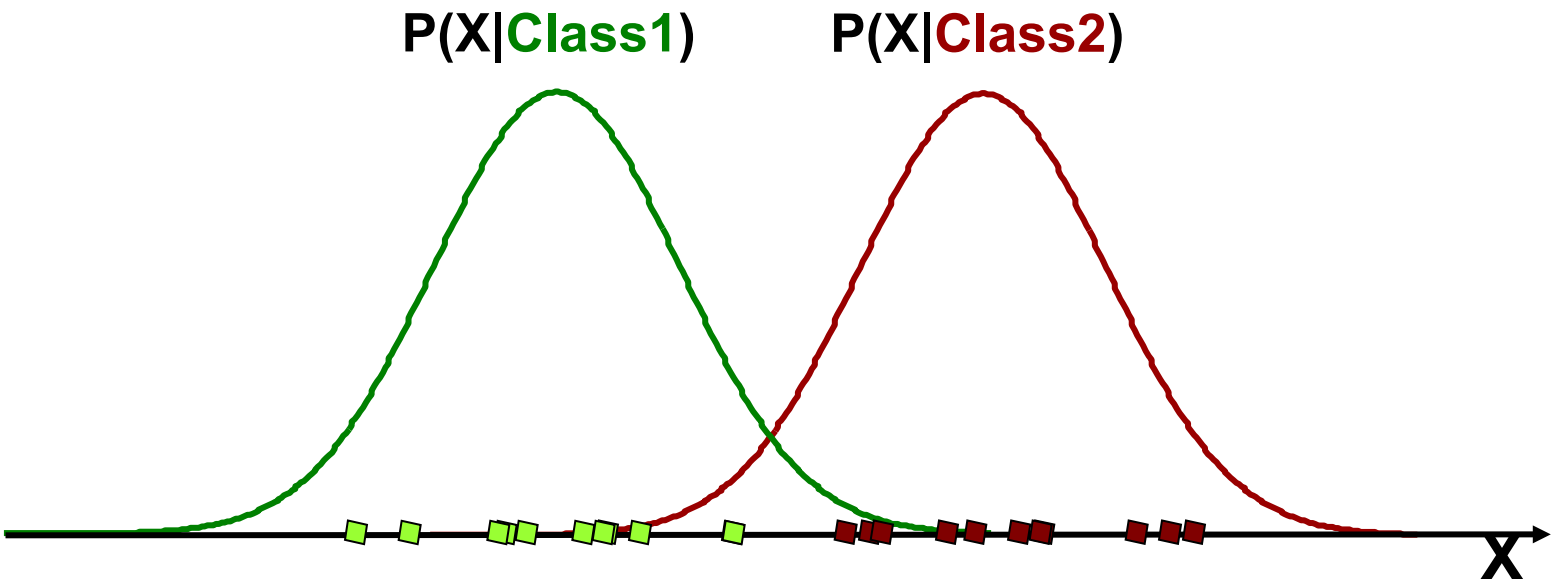
$$= argmax_C\, P(Feature|Class)\; P(Class)$$

Scaling the above distribution based on class priors

Likelihood:

$$P(Class \mid Feature) = \frac{P(Feature \mid Class)P(Class)}{P(Feature)}$$

Features for each class drawn from conditional probability distributions (conditional on the class)



Our first goal will be to *model* these class-conditional probability distributions (CCPD)

Class Priors:

$$P(Class \mid Feature) = \frac{P(Feature \mid Class)P(Class)}{P(Feature)}$$

We model prior probabilities to quantify the expected *a priori* chance of seeing a class

P(Class2) & P(Class1)

P(mito) = how likely is the next protein to be a mitochondrial protein *before I see any features to help me decide*

We expect ~1500 mitochondrial genes out of ~21000 total, so

P(mito)=1500/21000

P(~mito)=19500/21000

Evidence

$$P(Class \mid Feature) = \frac{P(Feature \mid Class)P(Class)}{P(Feature)}$$

Total evidence is $P(Feature)=\Sigma_i P(Feature|Class_i)P(Class_i)$
But it does not need to be known for classification

If we observe an object with feature X, how do decide if the object is from Class 1?

The Bayes Decision Rule is simply choose Class1 if:

$$P(Class1 \mid X) > P(Class2 \mid X)$$

$$\frac{P(X \mid Class1)P(L1)}{P(X)} > \frac{P(X \mid Class2)P(L2)}{P(X)}$$

same

$$P(X \mid Class1)P(Class1) > P(X \mid Class2)P(Class2)$$

➔ **P(Feature) does not need to be computed for classification**

Discriminant Function for selecting Class1

We can create a convenient representation of the Bayes Decision Rule

$$P(X \mid Class1)P(Class1) > P(X \mid Class2)P(Class2)$$

$$\frac{P(X \mid Class1)P(Class1)}{P(X \mid Class2)P(Class2)} > 1$$

$$G(X) = \log \frac{P(X \mid Class1)}{P(X \mid Class2)} \frac{P(Class1)}{P(Class2)} > 0$$

If G(X) > 0, we classify as Class 1

Today: Gene Expression Clustering & Classification

1. **Introduction to gene expression analysis**
   - Technology: microarrays vs. RNAseq. From reads to transcripts, expr levels
   - Expr matrix. Supervised vs. unsupervised learning. Clustering/Classification

2. **K-means clustering (clustering by partitioning)**
   - Algorithmic formulation: Update rule, optimality criterion. Fuzzy k-means.
   - Machine learning formulation: Generative models, Expectation Maximization.

3. **Hierarchical Clustering (clustering by agglomeration)**
   - Basic algorithm, Distance measures. Evaluating clustering results

4. **Naïve Bayes classification (generative approach to classification)**
   - Discriminant function: class priors, and class-conditional distributions
   - Training and testing, Combine mult features, Classification in practice

5. **(optional) Support Vector Machines (discriminative approach)**
   - SVM formulation, Margin maximization, Finding the support vectors
   - Non-linear discrimination, Kernel functions, SVMs in practice

Training and Testing Datasets

The Rule

We *must* test our classifier on a different set from the training set: the labeled test set

The Task

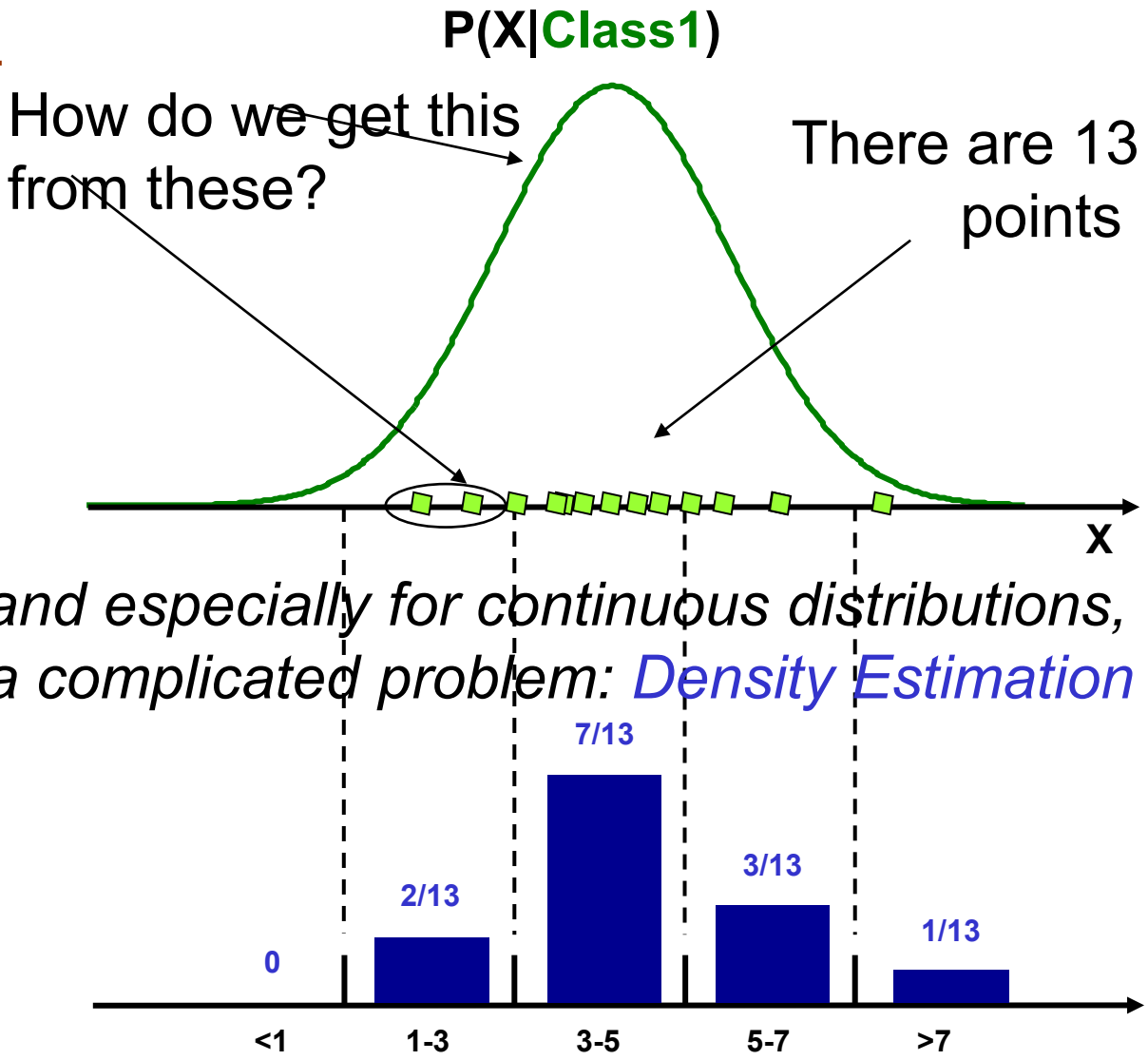
We will classify each object in the test set and count the number of each type of error

Getting P(X|Class) from Training Set

One Simple Approach

Divide X values into bins

And then we simply count frequencies

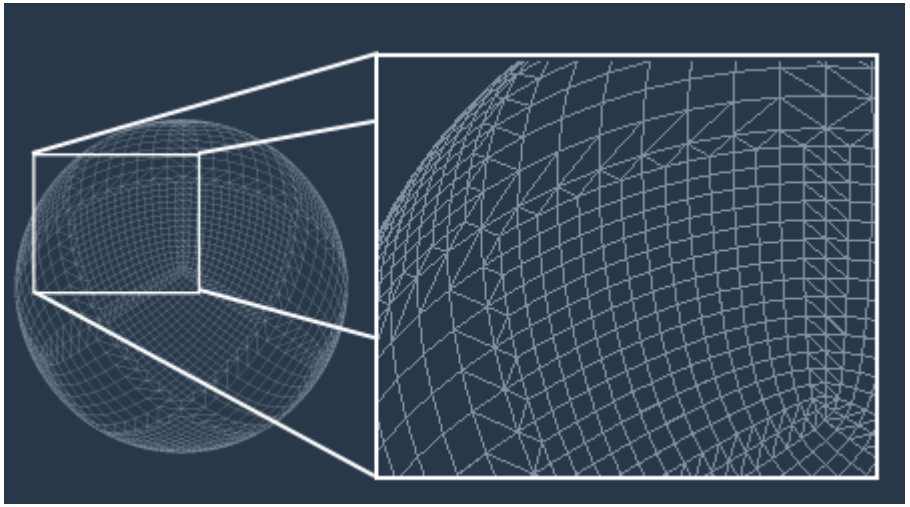


In general, and especially for continuous distributions, this can be a complicated problem: Density Estimation

Distributions Over Many Features

Estimating P(X1,X2,X3,…,X8|Class1) can be difficult

- Assume each feature binned into 5 possible values
- We have 5^8 combinations of values we need to count the frequency for



- Generally will not have enough data
  - We will have lots of nasty zeros

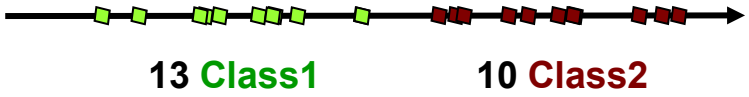
Getting Priors

Three general approaches

1. Estimate priors by counting fraction of classes in training set

P(Class1)=13/23

P(Class2)=10/23



13 Class1 10 Class2

But sometimes fractions in training set are not representative of world

2. Estimate from "expert" knowledge

Example
P(mito)=1500/21000
P(~mito)=19500/21000

3. We have no idea – use equal (uninformative) priors

P(Class1)=P(Class2)

Today: Gene Expression Clustering & Classification

1. **Introduction to gene expression analysis**
   - Technology: microarrays vs. RNAseq. From reads to transcripts, expr levels
   - Expr matrix. Supervised vs. unsupervised learning. Clustering/Classification

2. **K-means clustering (clustering by partitioning)**
   - Algorithmic formulation: Update rule, optimality criterion. Fuzzy k-means.
   - Machine learning formulation: Generative models, Expectation Maximization.

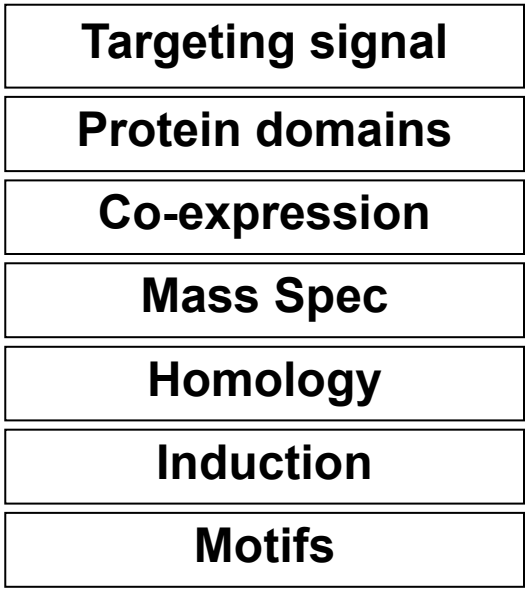
3. **Hierarchical Clustering (clustering by agglomeration)**
   - Basic algorithm, Distance measures. Evaluating clustering results

4. **Naïve Bayes classification (generative approach to classification)**
   - Discriminant function: class priors, and class-conditional distributions
   - Training and testing, Combine mult features, Classification in practice

5. **(optional) Support Vector Machines (discriminative approach)**
   - SVM formulation, Margin maximization, Finding the support vectors
   - Non-linear discrimination, Kernel functions, SVMs in practice

Combining Multiple Features

- We have focused on a single feature for an object
- But mitochondrial protein prediction (for example) has 7 features



So P(X|Class) become P(X1,X2,X3,…,X8|Class) and our discriminant function becomes

$$G(X) = \log \frac{P(X_1, X_2, ..., X_7 \mid Class1)}{P(X_1, X_2, ..., X_7 \mid Class2)} \frac{P(Class1)}{P(Class2)} > 0$$

Naïve Bayes Classifier

We are going to make the following assumption:

All features are independent given the class

$$P(X_1, X_2, ..., X_n \mid Class) = P(X_1 \mid Class)P(X_2 \mid Class)...P(X_n \mid Class)$$

$$= \prod_{i=1}^{n} P(X_i \mid Class)$$

We can thus estimate individual distributions for each feature and just multiply them together!

Naïve Bayes Discriminant Function

Thus, with the Naïve Bayes assumption, we can now rewrite, this:

$$G(X_1,...,X_7) = \log \frac{P(X_1, X_2, ..., X_7 \mid Class1)}{P(X_1, X_2, ..., X_7 \mid Class2)} \frac{P(Class1)}{P(Class2)} > 0$$

As this:

$$G(X_1,...,X_7) = \log \frac{\prod P(X_i \mid Class1)}{\prod P(X_i \mid Class2)} \frac{P(Class1)}{P(Class2)} > 0$$

Which can be simply computed as the sum of log scores

Binary Classification Errors

	True (Mito)	False (~Mito)
Predicted True	TP	FP
Predicted False	FN	TN

Sensitivity = TP/(TP+FN) Specificity = TN/(TN+FP)

- Sensitivity
  - Fraction of all Class1 (True) that we correctly predicted at Class 1
  - *How good are we at finding what we are looking for*

- Specificity
  - Fraction of all Class 2 (False) called Class 2
  - *How many of the Class 2 do we filter out of our Class 1 predictions*

Today: Gene Expression Clustering & Classification

1. **Introduction to gene expression analysis**
   - Technology: microarrays vs. RNAseq. From reads to transcripts, expr levels
   - Expr matrix. Supervised vs. unsupervised learning. Clustering/Classification

2. **K-means clustering (clustering by partitioning)**
   - Algorithmic formulation: Update rule, optimality criterion. Fuzzy k-means.
   - Machine learning formulation: Generative models, Expectation Maximization.

3. **Hierarchical Clustering (clustering by agglomeration)**
   - Basic algorithm, Distance measures. Evaluating clustering results

4. **Naïve Bayes classification (generative approach to classification)**
   - Discriminant function: class priors, and class-conditional distributions
   - Training and testing, Combine mult features, Classification in practice

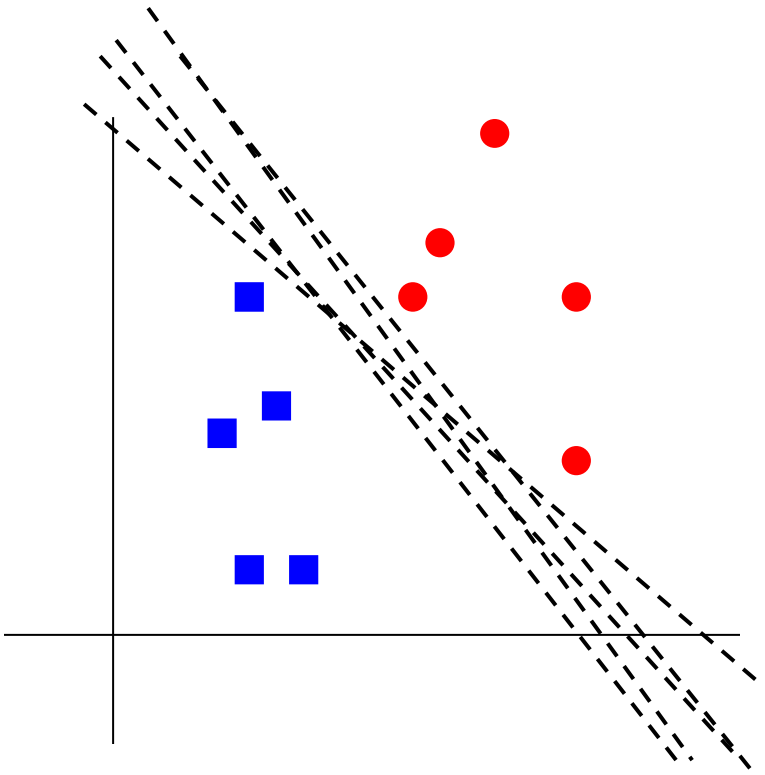
5. **(optional) Support Vector Machines (discriminative approach)**
   - SVM formulation, Margin maximization, Finding the support vectors
   - Non-linear discrimination, Kernel functions, SVMs in practice

Support Vector Machines (SVMs)

Easy to select a line

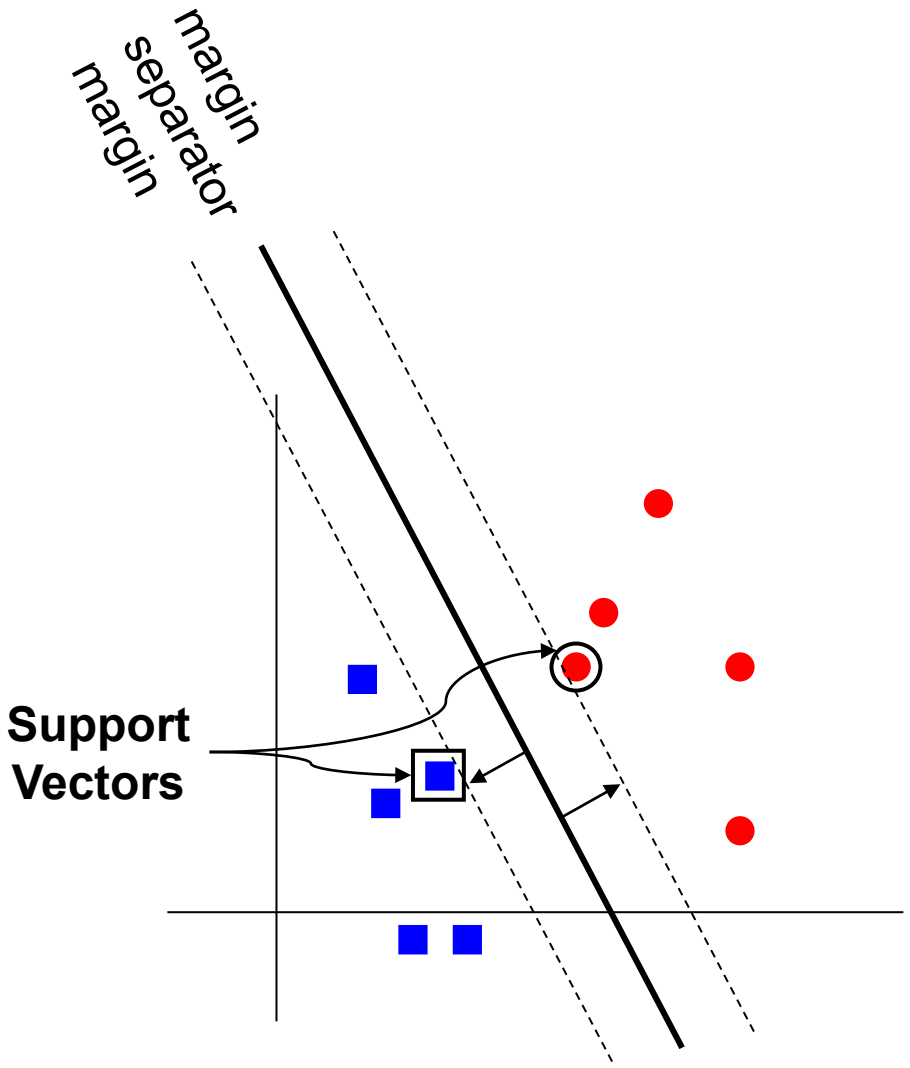
But many lines will separate these training data

What line should we choose?



Support Vector Machines (SVMs)

A sensible choice is to select a line that maximizes the *margin* between classes



SVM Formulation

We define a vector **w** normal to the separating line

Assume all data satisfy the following:

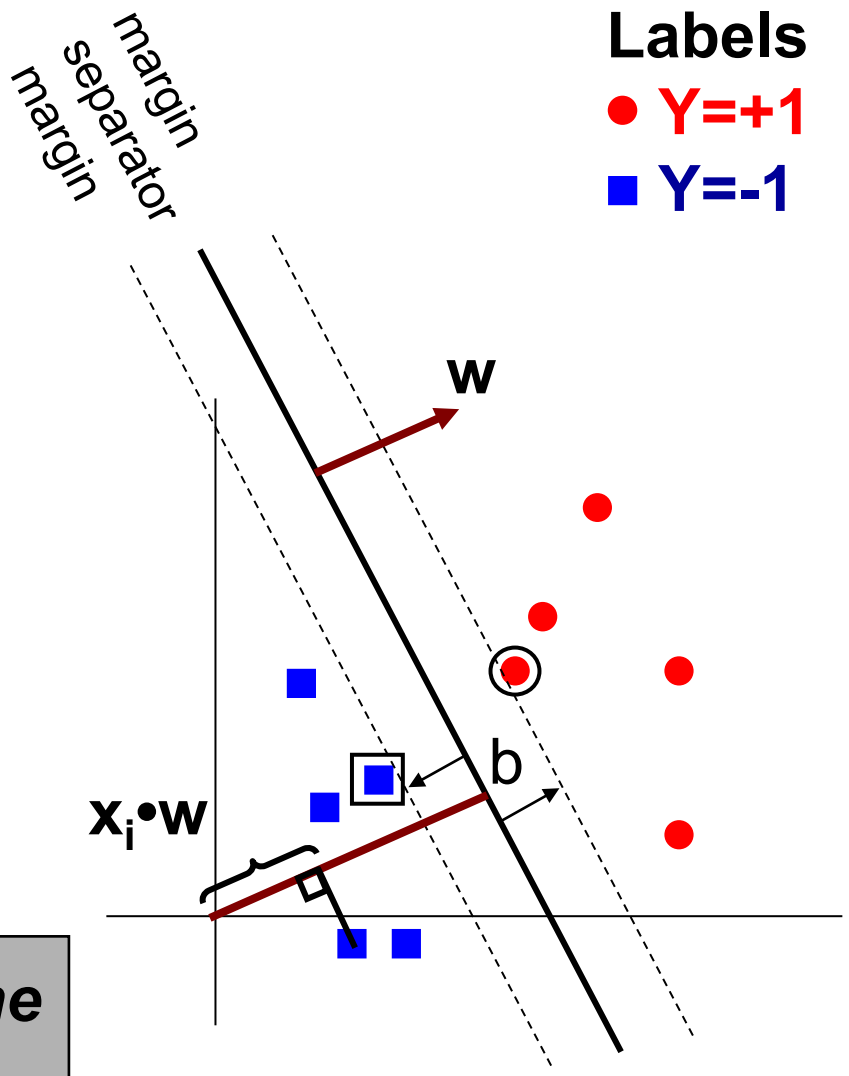
$$\mathbf{x_i} \bullet \mathbf{w} - b \geq +1 \quad \text{for } y_i = +1$$

$$\mathbf{x_i} \bullet \mathbf{w} - b \leq -1 \quad \text{for } y_i = -1$$

$$\Downarrow$$

$$y_i \left(\mathbf{x_i} \bullet \mathbf{w} - b \right) \geq 1$$

Find the separator with the largest margin



An Optimization Problem

For full derivation, see Burges (1998)

Only need dot product of input data!

$$\text{Minimize } L_D = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x_i} \bullet \mathbf{x_j}$$

Quadratic Programming

$$\text{subject to } \sum_i \alpha_i y_i = 0 \text{ and } \alpha_j > 0$$

Solving for

$$\alpha_i \left(y_i \left(\mathbf{x_i} \bullet \mathbf{w} - b \right) - 1 \right) = 0$$

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x_i}$$

Only some α_i are non-zero

$\mathbf{x}_i$ with $a_i > 0$ are the *support vectors*

w is *determined by these data points!*

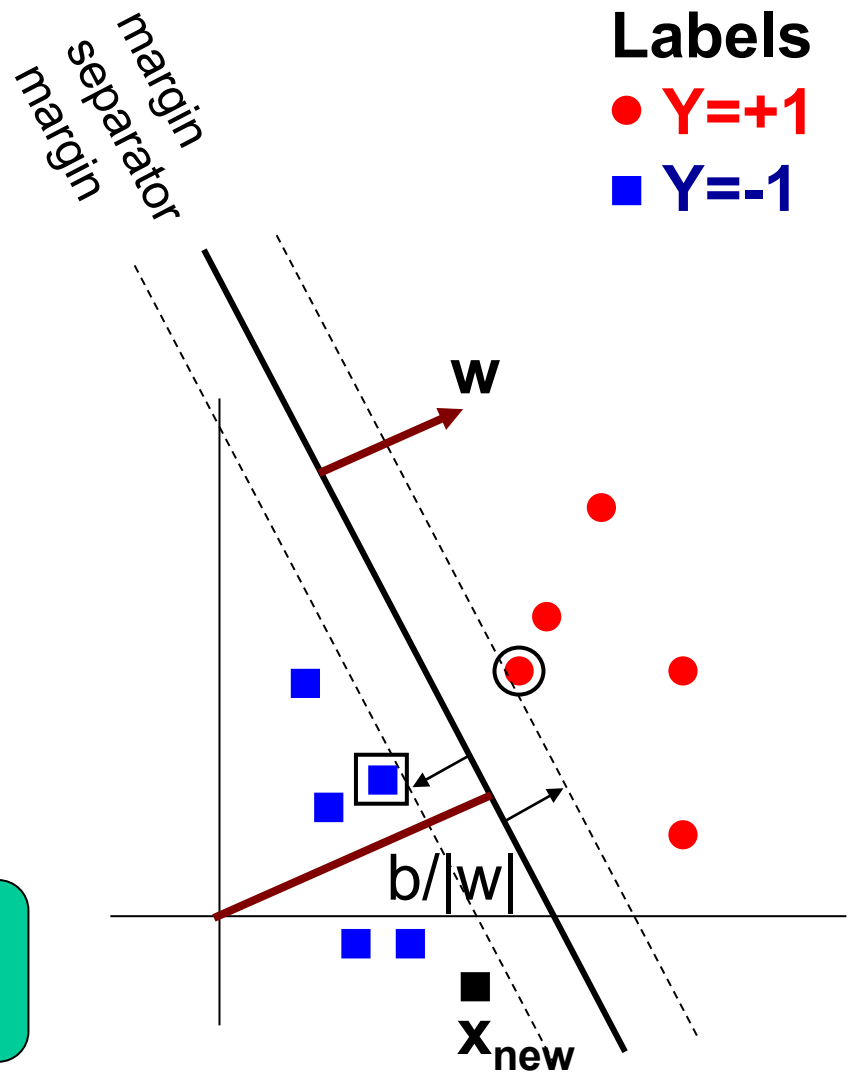
Using an SVM

Given a new data point we simply assign it the label:

$$y_i = \text{sign}\left(\mathbf{w} \bullet \mathbf{x}_{\text{new}} - b\right)$$

$$= \text{sign}\left(\sum_i \alpha_i y_i \boxed{\mathbf{x}_i \bullet \mathbf{x}_{\text{new}}} - b\right)$$

Again, only dot product of input data!



Today: Gene Expression Clustering & Classification

1. **Introduction to gene expression analysis**
   - Technology: microarrays vs. RNAseq. From reads to transcripts, expr levels
   - Expr matrix. Supervised vs. unsupervised learning. Clustering/Classification

2. **K-means clustering (clustering by partitioning)**
   - Algorithmic formulation: Update rule, optimality criterion. Fuzzy k-means.
   - Machine learning formulation: Generative models, Expectation Maximization.

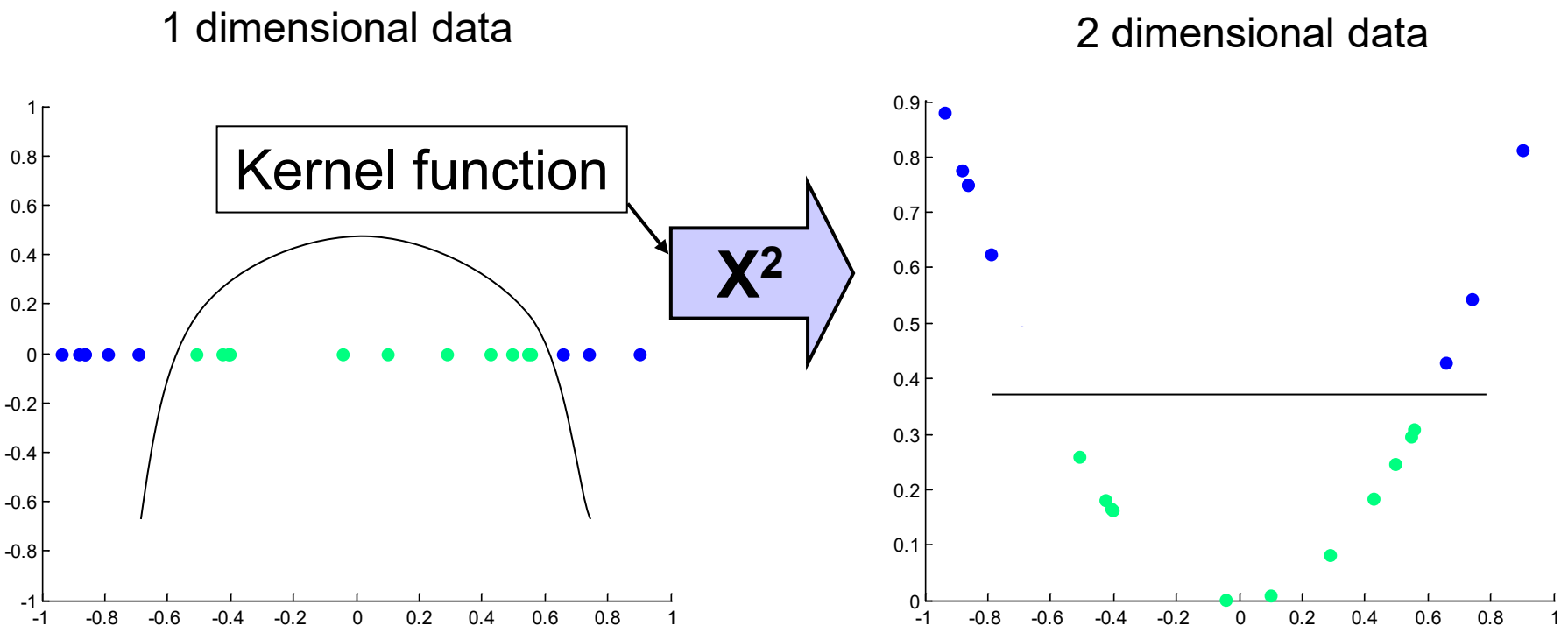
3. **Hierarchical Clustering (clustering by agglomeration)**
   - Basic algorithm, Distance measures. Evaluating clustering results

4. **Naïve Bayes classification (generative approach to classification)**
   - Discriminant function: class priors, and class-conditional distributions
   - Training and testing, Combine mult features, Classification in practice

5. **(optional) Support Vector Machines (discriminative approach)**
   - SVM formulation, Margin maximization, Finding the support vectors
   - Non-linear discrimination, Kernel functions, SVMs in practice

Non-linear Classifier

- **Some data not linearly separable in low dimensions**
- **What if we transform it to a higher dimension?**



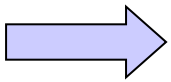
Noble, 2006. NATURE BIOTECHNOLOGY 24:1565.

Kernel Mapping

Want a **mapping** from input space, R^d, to other euclidean space, H

$\Phi(x)$: R^d -> H

But $\Phi(X)$ can be a mapping to an infinite dimensional space
i.e. d points become an infinite number of points

$\mathbf{X=(x_1,x_2)}$  $\mathbf{\Phi(X)=(\phi_1,\phi_2,\phi_3,\ldots\phi_\infty)}$

Rather difficult to work with!

Kernel Mapping

Want a **mapping** from input space, R^d, to other euclidean space, H

From previous slide, SVMs *only depend* on **dot product**

$\Phi(x)$: R^d -> H

$\mathbf{X_i \bullet X_j}$ **becomes** $\Phi(\mathbf{X_i}) \bullet \Phi(\mathbf{X_j})$

Here is trick: if we have a kernel function such that

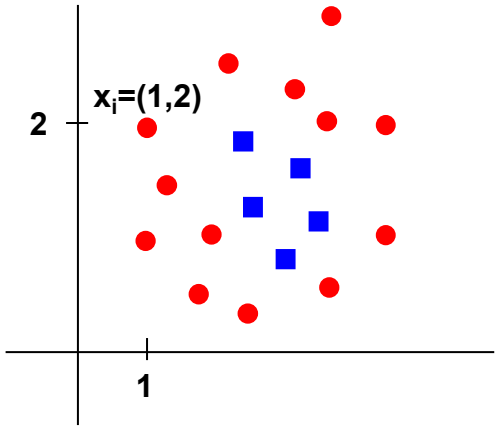
$$\mathbf{K(X_i,X_j) = \Phi(X_i) \bullet \Phi(X_j)}$$

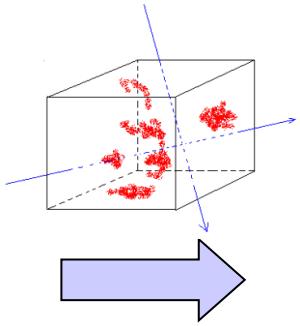
We can just use K and never know Φ(x) explicitly!

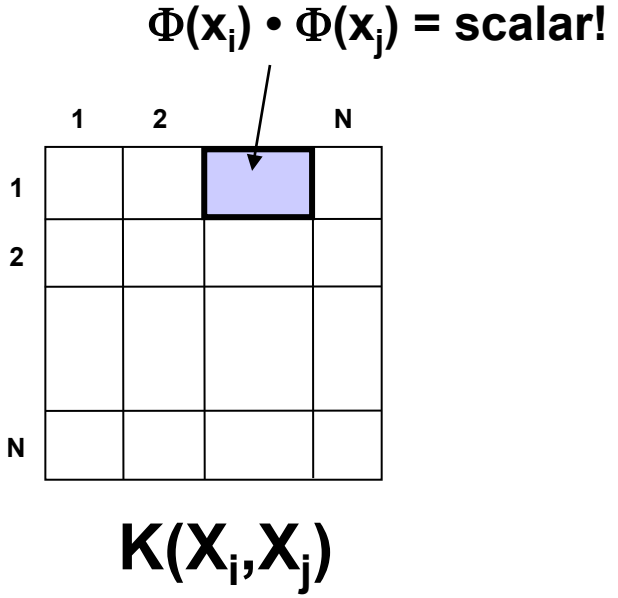
Φ(X) is high dimensional
K is a scalar

Kernels

So the key step is to take your input data and transform it into a kernel matrix







We have then done two very useful things:

1. Transformed X into a high (possibly infinite) dimensional space (where we hope are data are separable)
2. Taken dot products in this space to create scalars

Example Kernels

$$K\left(\mathbf{x}_i,\mathbf{x}_j\right)=\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j$$ **Linear**

$$K\left(\mathbf{x}_i,\mathbf{x}_j\right)=\left(\gamma\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j+\mathrm{r}\right)^d$$ **Polynomial**

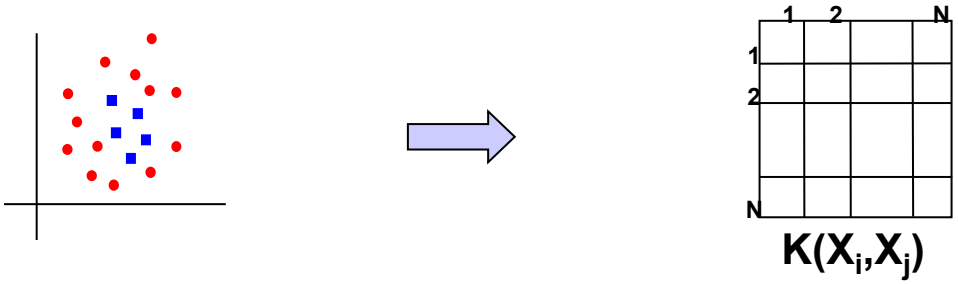
$$K\left(\mathbf{x}_i,\mathbf{x}_j\right)=\exp\left(-\gamma\left\|\mathbf{x}_i-\mathbf{x}_j\right\|^2\right)$$ **Radial Basis Function**

$$K\left(\mathbf{x}_i,\mathbf{x}_j\right)=\tanh\left(\gamma\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j+\mathrm{r}\right)$$ **Sigmoid**

What $K(X_i,X_j)$ are valid kernels?
Answer given by Mercer's Condition (see Burgess 1998)

Using (Non-Linear) SVMs

Step 1 – Transform data to Kernel Matrix K



Step 2 – Train SVM on transformed data – get support vectors

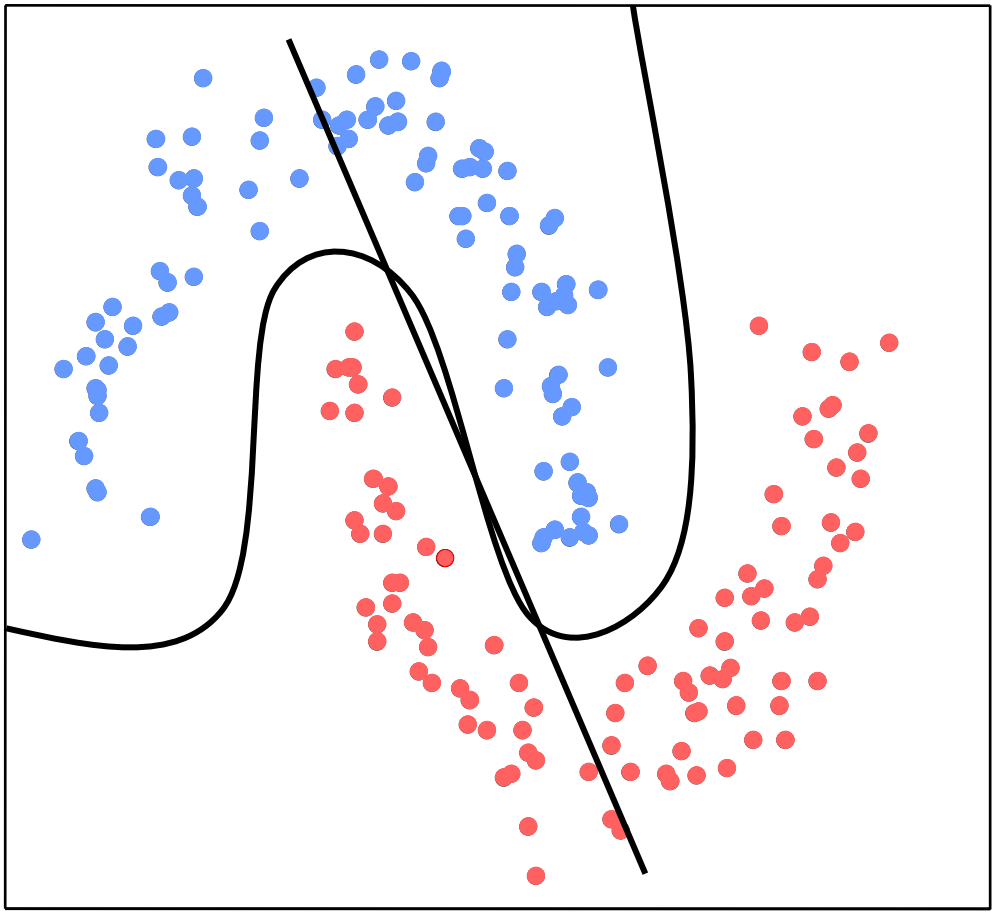
$$\text{Minimize } L_D = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x_i} \bullet \mathbf{x_j} = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j K\left(\mathbf{x_i}, \mathbf{x_j}\right)$$

Step 2 – Test/Classify on new samples

$$y_{new} = \text{sign}\left(\mathbf{w} \bullet \mathbf{x}_{\text{new}}\right) = \text{sign}\left(\sum_i \alpha_i y_i \mathbf{x_i} \bullet \mathbf{x}_{\text{new}}\right) = \text{sign}\left(\sum_i \alpha_i y_i K\left(\mathbf{x_i}, \mathbf{x}_{\text{new}}\right)\right)$$

Bringing Clustering and Classification Together

Semi-Supervised Learning



Common Scenario

- Few labeled
- Many unlabeled
- Structured data

What if we cluster first?

Then clusters can help us classify

Today: Gene Expression Clustering & Classification

1. **Introduction to gene expression analysis**
   - Technology: microarrays vs. RNAseq. From reads to transcripts, expr levels
   - Expr matrix. Supervised vs. unsupervised learning. Clustering/Classification

2. **K-means clustering (clustering by partitioning)**
   - Algorithmic formulation: Update rule, optimality criterion. Fuzzy k-means.
   - Machine learning formulation: Generative models, Expectation Maximization.

3. **Hierarchical Clustering (clustering by agglomeration)**
   - Basic algorithm, Distance measures. Evaluating clustering results

4. **Naïve Bayes classification (generative approach to classification)**
   - Discriminant function: class priors, and class-conditional distributions
   - Training and testing, Combine mult features, Classification in practice

5. **(optional) Support Vector Machines (discriminative approach)**
   - SVM formulation, Margin maximization, Finding the support vectors
   - Non-linear discrimination, Kernel functions, SVMs in practice