Clustering Part 2

EECS 349 Spring 2015

Expectation Maximization

- Learning parameters in Bayes Nets is easy if data is complete
  - Just counting
- But what about missing data?
  - We could use our standard "missing data" techniques (use mean, median, etc.)
  - But when lots of data is missing, we want to infer missing data and parameters simultaneously
    - We can use **Expectation Maximization**

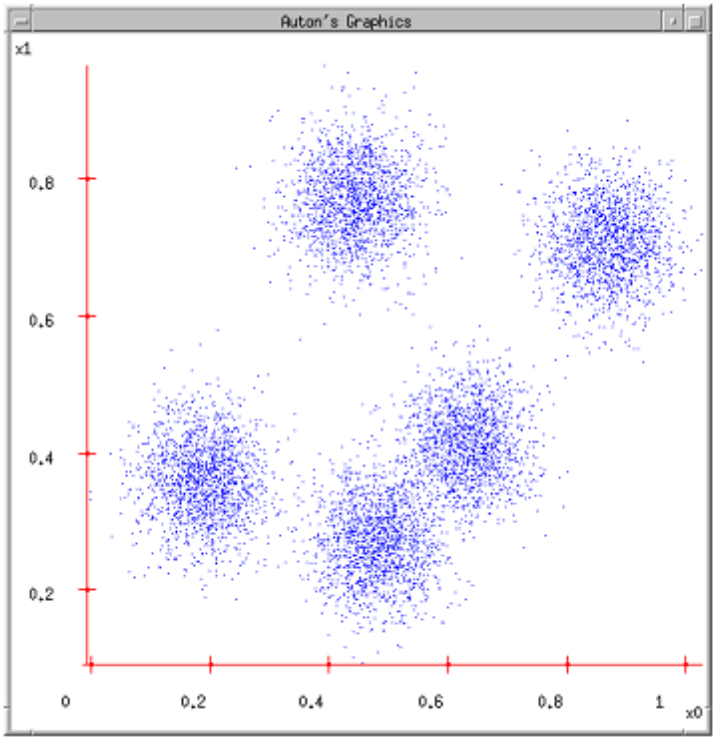
Gaussian Mixtures

- K classes, each class ω_i produces Gaussian observations with mean μ_i with variance $\sigma^2 I$
- Assume $\sigma^2 I$ given (for now), and we have lots of observations
- Task: estimate μ_i
- But, none of the data points are labeled…

Most slides from
http://www.autonlab.org/tutorials/

Gaussian Mixtures

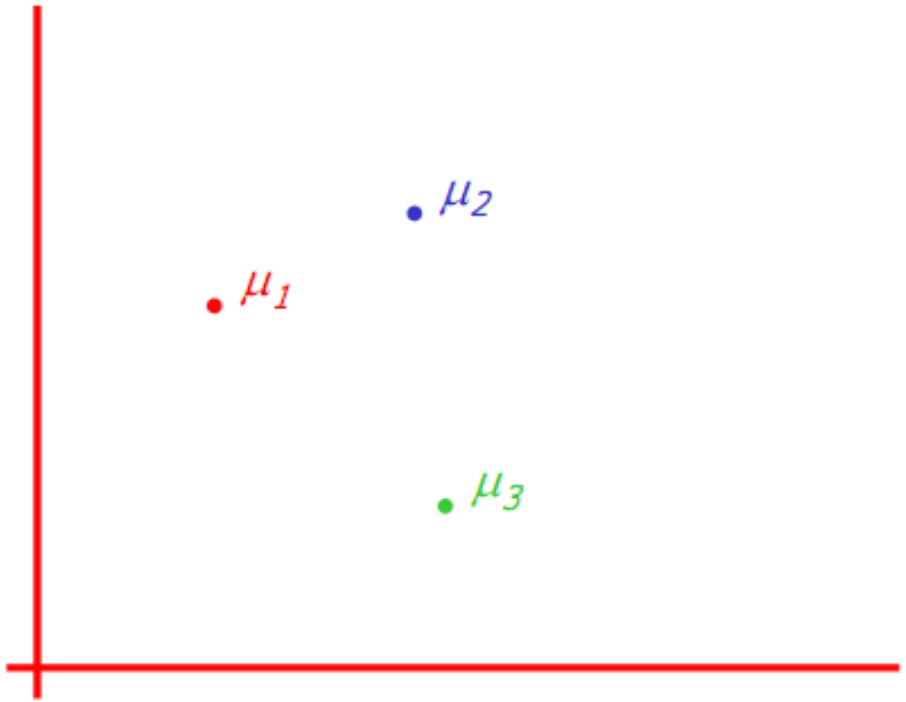
- Know
  - K
  - Data
  - $\sigma^2 I$
  - $P(\omega_i)$
- Don't know
  - Data label
- Objective
  - Estimate the μ_i



Most slides from
http://www.autonlab.org/tutorials/

The GMM assumption

- There are k components. The i'th component is called ω_i
- Component ω_i has an associated mean vector μ_i

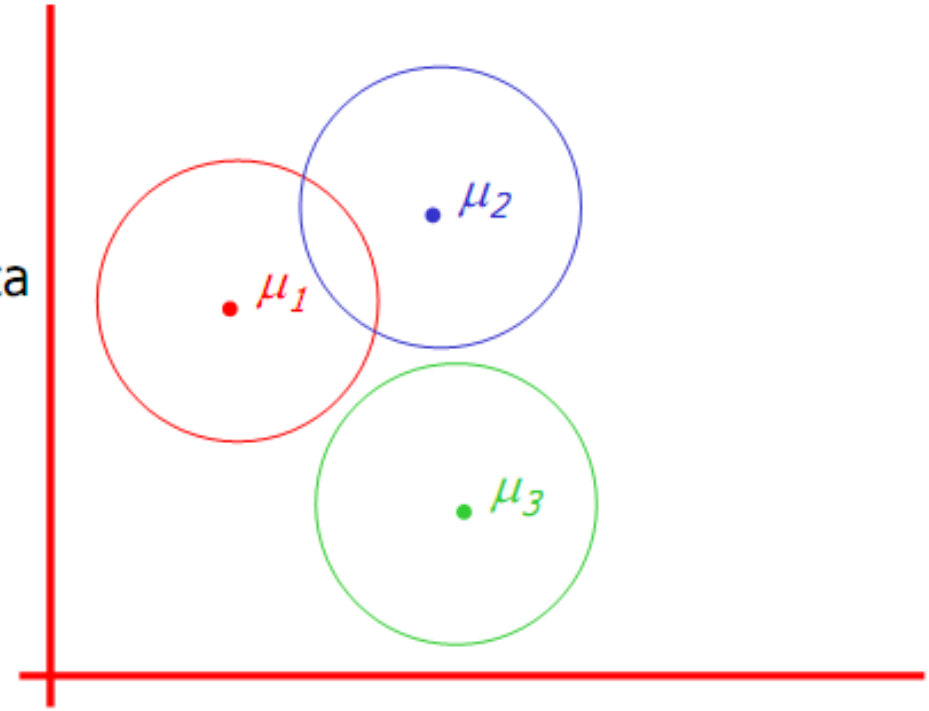


Most slides from
http://www.autonlab.org/tutorials/

The GMM assumption

- There are k components. The i'th component is called ω_i
- Component ω_i has an associated mean vector μ_i
- Each component generates data from a Gaussian with mean μ_i and covariance matrix $\sigma^2 \boldsymbol{I}$

Assume that each datapoint is generated according to the following recipe:



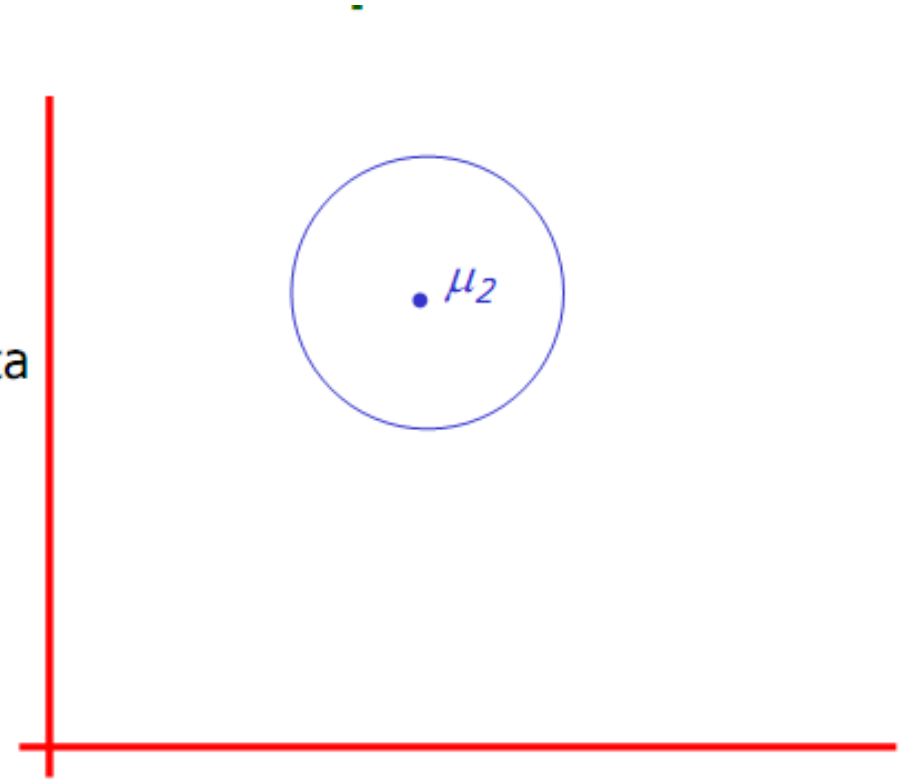
Most slides from
http://www.autonlab.org/tutorials/

The GMM assumption

- There are k components. The i'th component is called ω_i
- Component ω_i has an associated mean vector μ_i
- Each component generates data from a Gaussian with mean μ_i and covariance matrix $\sigma^2 \boldsymbol{I}$

Assume that each datapoint is generated according to the following recipe:

1. Pick a component at random. Choose component i with probability $P(\omega_i)$.



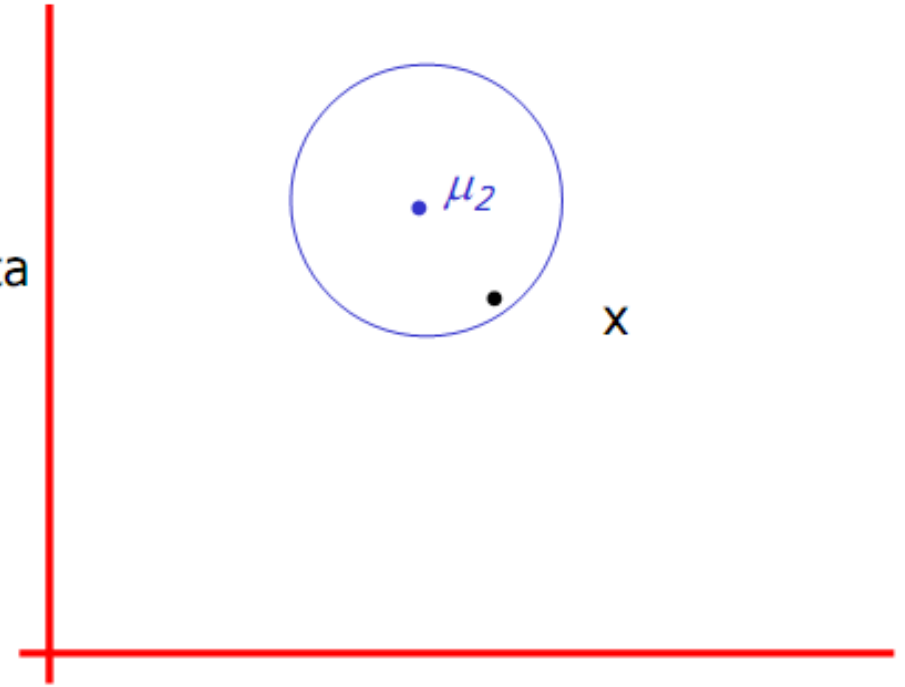
Most slides from
http://www.autonlab.org/tutorials/

The GMM assumption

- There are k components. The i'th component is called ω_i
- Component ω_i has an associated mean vector μ_i
- Each component generates data from a Gaussian with mean μ_i and covariance matrix $\sigma^2\boldsymbol{I}$

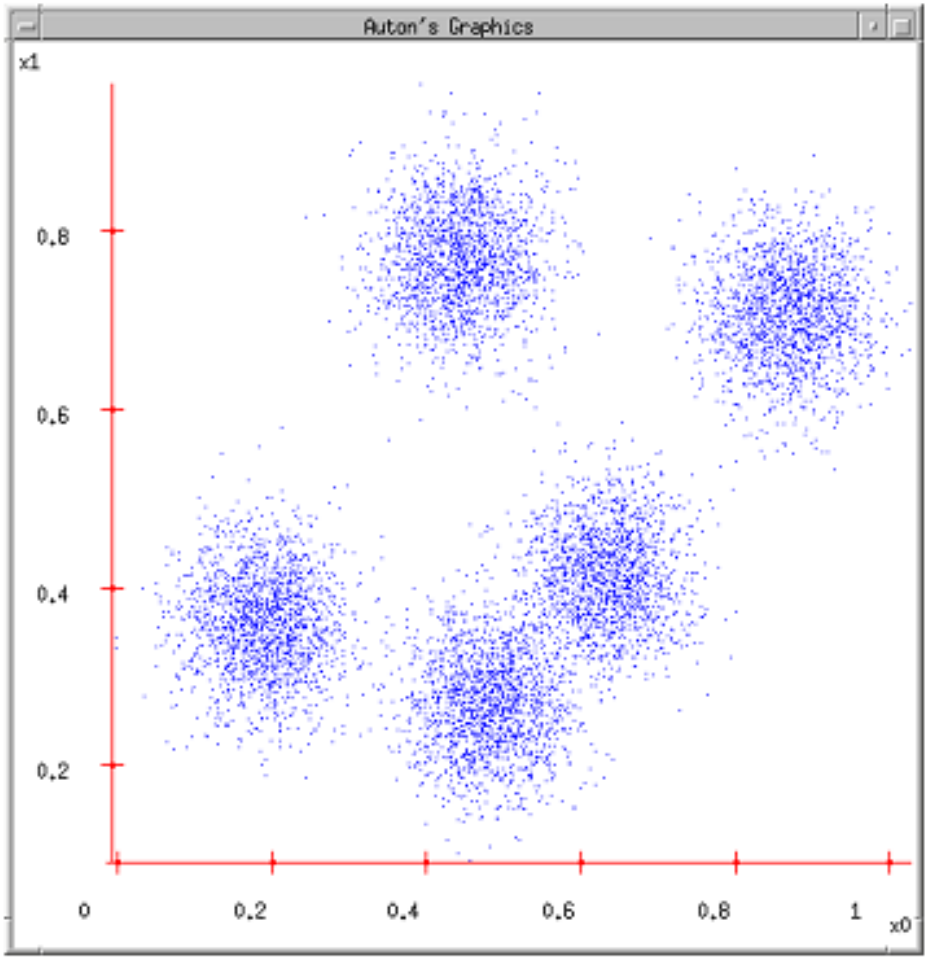
Assume that each datapoint is generated according to the following recipe:

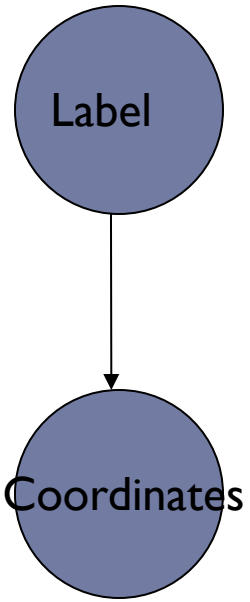
1. Pick a component at random. Choose component i with probability $P(\omega_i)$.
2. Datapoint ~ $N(\mu_i, \sigma^2\boldsymbol{I})$



Most slides from
http://www.autonlab.org/tutorials/

The data generated





Most slides from
http://www.autonlab.org/tutorials/

Computing the likelihood

Remember:

We have unlabeled data $\boldsymbol{x}_1\ \boldsymbol{x}_2\ \ldots\ \boldsymbol{x}_R$

We know there are k classes

We know $P(w_1)\ P(w_2)\ P(w_3)\ \ldots\ P(w_k)$

We don't know $\boldsymbol{\mu}_1\ \boldsymbol{\mu}_2\ \ldots\ \boldsymbol{\mu}_k$

We can write $P(\text{data} \mid \boldsymbol{\mu}_1 \ldots \boldsymbol{\mu}_k)$

$$= p(x_1 \ldots x_R \mid \mu_1 \ldots \mu_k)$$

$$= \prod_{i=1}^{R} p(x_i \mid \mu_1 \ldots \mu_k)$$

$$= \prod_{i=1}^{R} \sum_{j=1}^{k} p(x_i \mid w_j, \mu_1 \ldots \mu_k) P(w_j)$$

$$= \prod_{i=1}^{R} \sum_{j=1}^{k} K \exp\left(-\frac{1}{2\sigma^2}(x_i - \mu_j)^2\right) P(w_j)$$

Most slides from
http://www.autonlab.org/tutorials/

EM for GMMs

For Max likelihood we know $\frac{\partial}{\partial \mu_i} \log \Pr ob(\text{data} | \mu_1 \ldots \mu_k) = 0$

Some wild' n' crazy algebra turns this into : "For Max likelihood, for each j,

$$\mu_j = \frac{\sum_{i=1}^{R} P(w_j | x_i, \mu_1 \ldots \mu_k) x_i}{\sum_{i=1}^{R} P(w_j | x_i, \mu_1 \ldots \mu_k)}$$

This is n nonlinear equations in $\boldsymbol{\mu}_j$'s."

Most slides from
http://www.autonlab.org/tutorials/

EM for GMMs

For Max likelihood we know $\frac{\partial}{\partial \mu_i} \log \Pr ob(\text{data} | \mu_1 \ldots \mu_k) = 0$

Some wild' n' crazy algebra turns this into : "For Max likelihood, for each j,

$$\mu_j = \frac{\sum_{i=1}^{R} P(w_j | x_i, \mu_1 \ldots \mu_k) x_i}{\sum_{i=1}^{R} P(w_j | x_i, \mu_1 \ldots \mu_k)}$$

This is n nonlinear equations in $\boldsymbol{\mu}_j$'s."

If, for each $\mathbf{x}_i$ we knew that for each w_j the prob that $\boldsymbol{\mu}_j$ was in class w_j is $P(w_j | x_i, \mu_1 \ldots \mu_k)$ Then... we would easily compute μ_j.

If we knew each μ_j then we could easily compute $P(w_j | x_i, \mu_1 \ldots \mu_j)$ for each w_j and x_i.

Most slides from
http://www.autonlab.org/tutorials/

EM for GMMs

Iterate. On the *t*'th iteration let our estimates be

$$\lambda_t = \{ \mu_1(t), \mu_2(t) \ldots \mu_c(t) \}$$

$p_i(t)$ is shorthand for estimate of $P(\omega_i)$ on t'th iteration

E-step

Compute "expected" classes of all datapoints for each class

$$P(w_i | x_k, \lambda_t) = \frac{p(x_k | w_i, \lambda_t) P(w_i | \lambda_t)}{p(x_k | \lambda_t)} = \frac{p(x_k | w_i, \mu_i(t), \sigma^2 \mathbf{I}) p_i(t)}{\sum_{j=1}^{c} p(x_k | w_j, \mu_j(t), \sigma^2 \mathbf{I}) p_j(t)}$$

Just evaluate a Gaussian at x_k

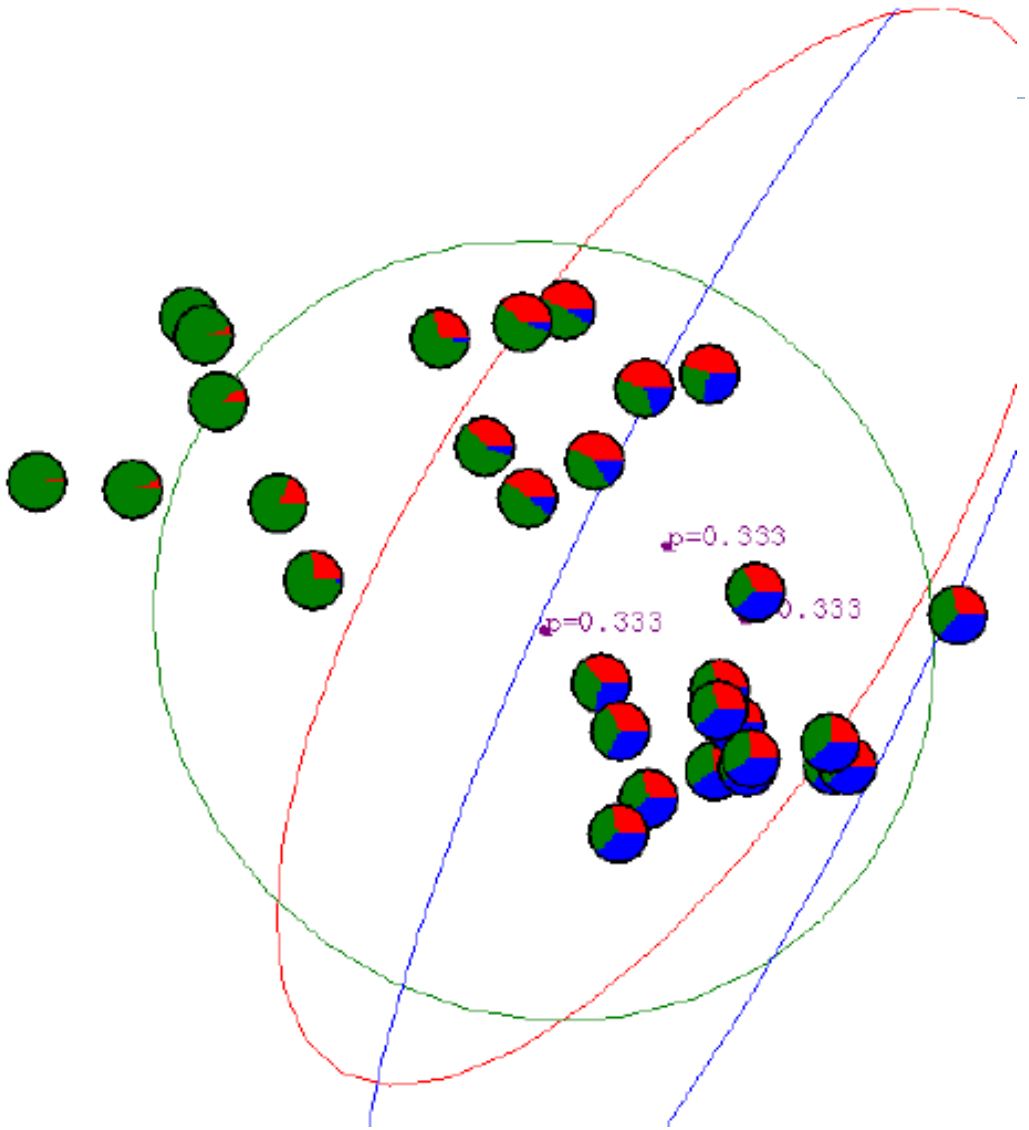
M-step.

Compute Max. like **μ** given our data's class membership distributions

$$\mu_i(t+1) = \frac{\sum_k P(w_i | x_k, \lambda_t) x_k}{\sum_k P(w_i | x_k, \lambda_t)}$$

Most slides from
http://www.autonlab.org/tutorials/

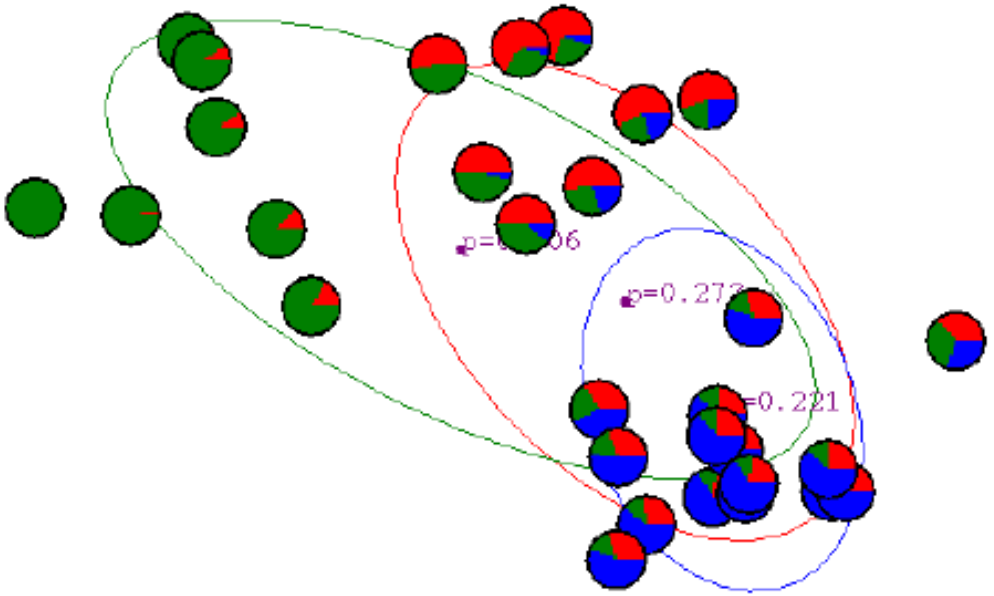
Gaussian Mixture Example: Start



Advance apologies: in Black and White this example will be incomprehensible

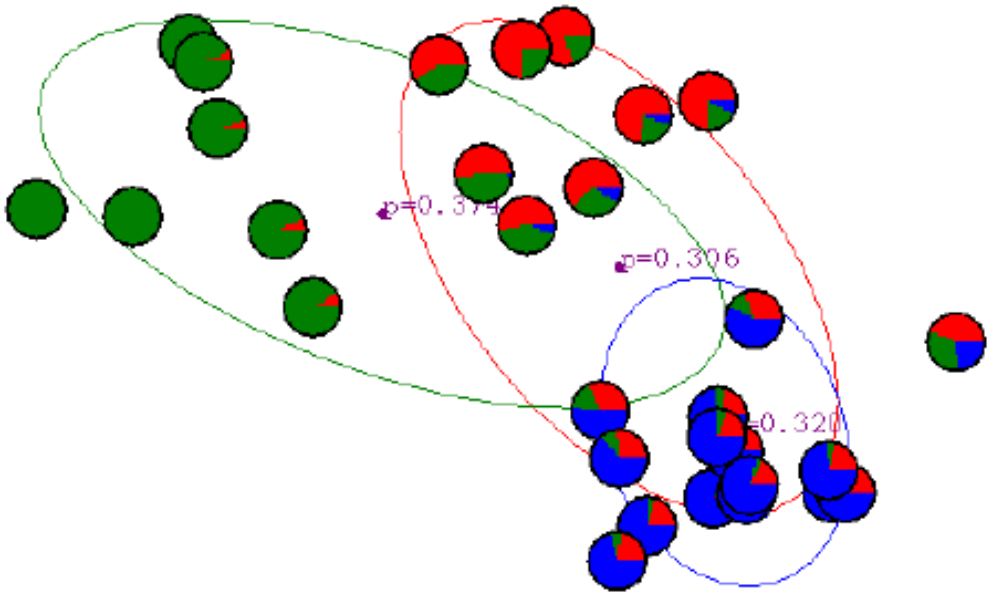
Most slides from
http://www.autonlab.org/tutorials/

After first iteration



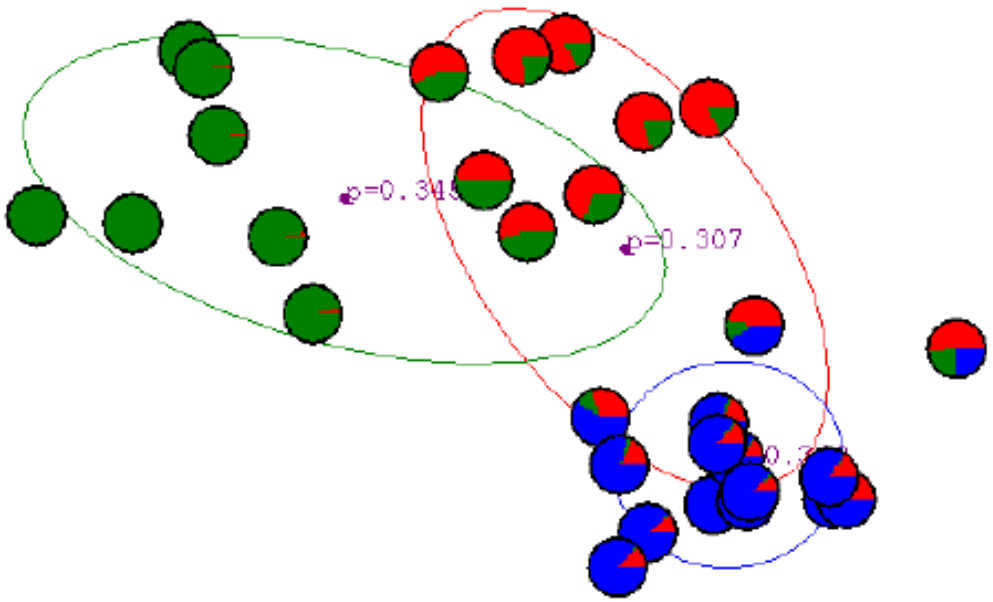
Most slides from
http://www.autonlab.org/tutorials/

After 2nd iteration



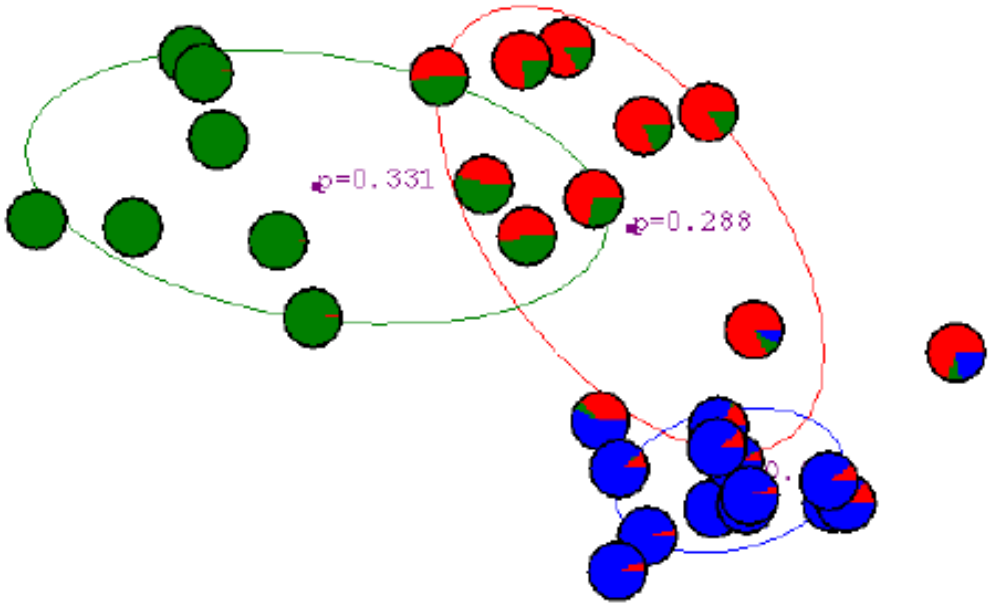
Most slides from
http://www.autonlab.org/tutorials/

After 3rd iteration



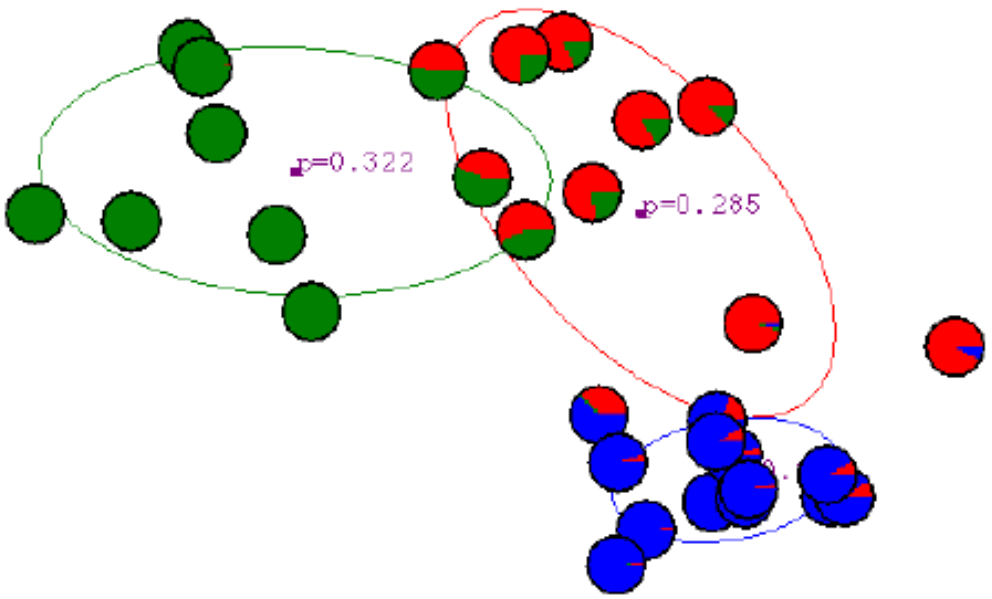
Most slides from
http://www.autonlab.org/tutorials/

After 4th iteration



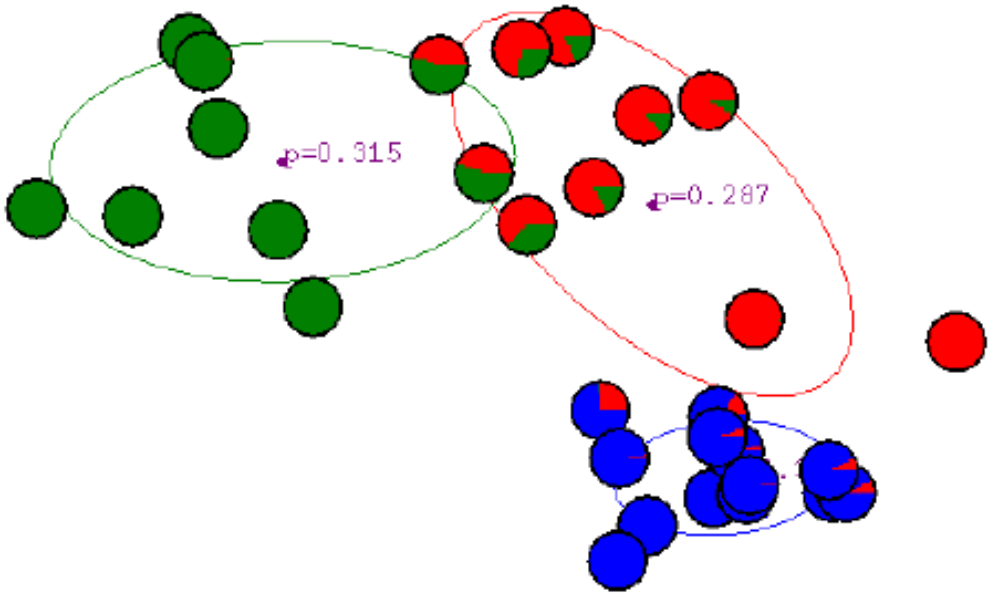
Most slides from
http://www.autonlab.org/tutorials/

After 5th iteration



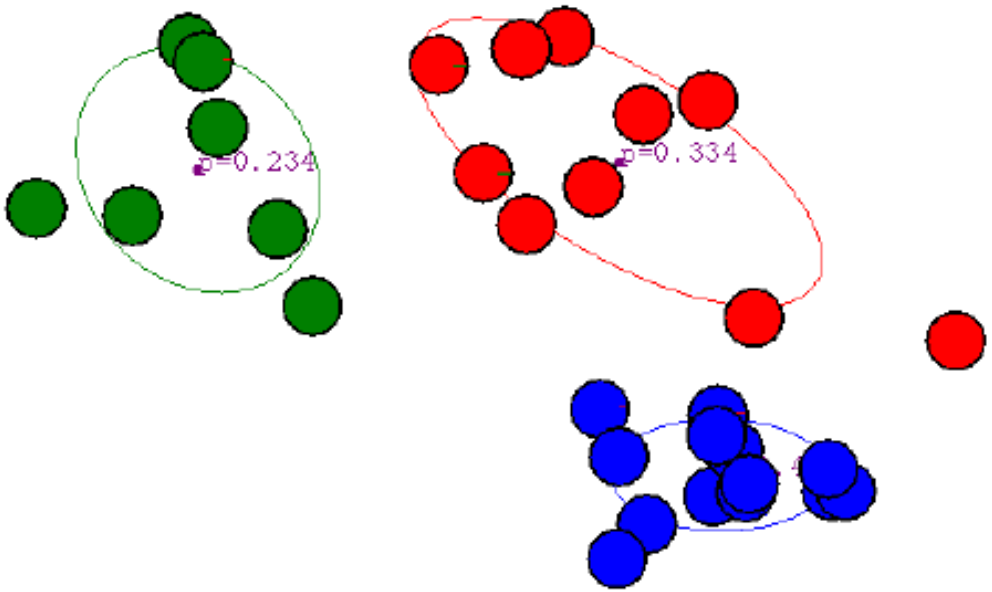
Most slides from
http://www.autonlab.org/tutorials/

After 6th iteration



Most slides from
http://www.autonlab.org/tutorials/

After 20th iteration



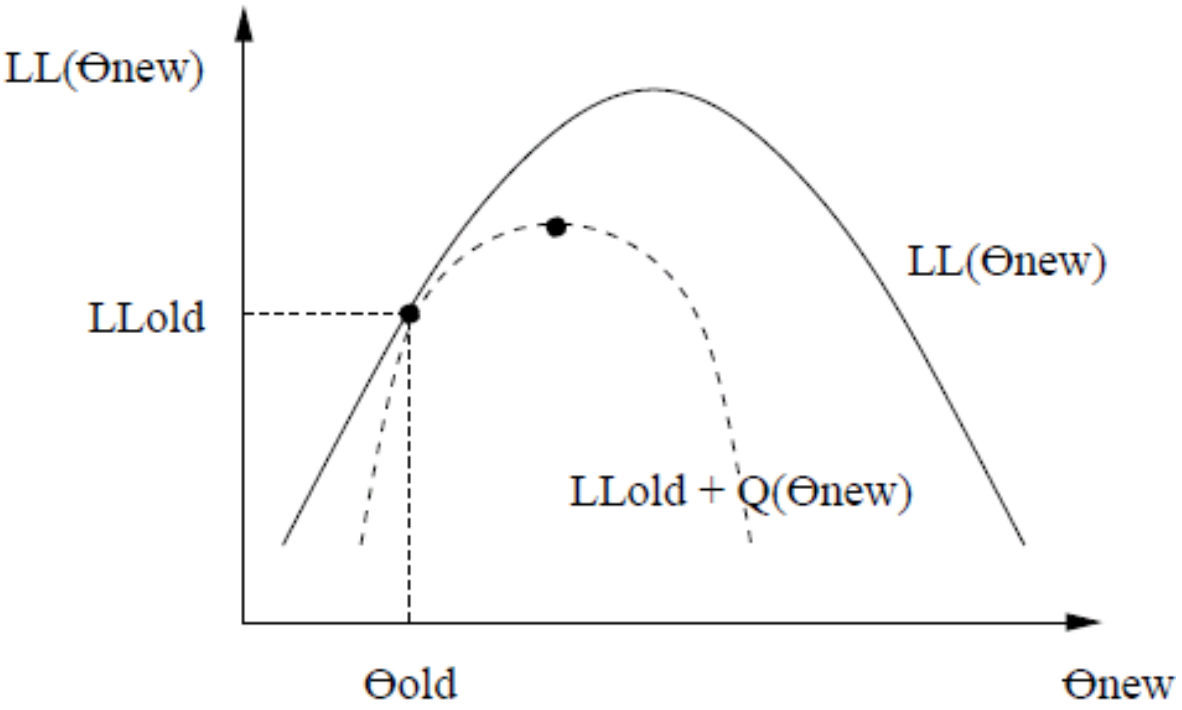
Most slides from
http://www.autonlab.org/tutorials/

EM at the 10,000 foot level

- Guess some parameters, then
  - Use your parameters to get a distribution over hidden variables
  - Re-estimate the parameters as if your distribution over hidden variables is correct
- Seems magical. When/why does this work?

Underlying EM: The basic idea

- EM: Given a guess θ_{old} for θ, improve it
- Idea: construct lower bound that equals the true log likelihood at θ_{old}:



For exponential family

- *E* step:
  - Use θ_n to estimate **expected** sufficient statistics over **complete** data
- *M* step
  - Set θ_{n+1} = ML parameters given sufficient statistics
    - (Or MAP parameters)

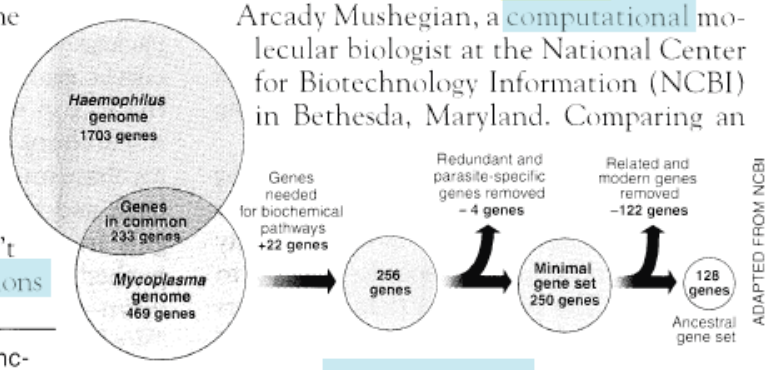
EM in practice

- Local maxima
  - Random re-starts, simulated annealing…
- Variants
  - Hard EM: set Z to most likely value (e.g. k-means)
  - Generalized EM: increase (not nec. maximize) lower bound in each step
  - Approximate E-step (e.g. sampling)

Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK—How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those predictions "are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an



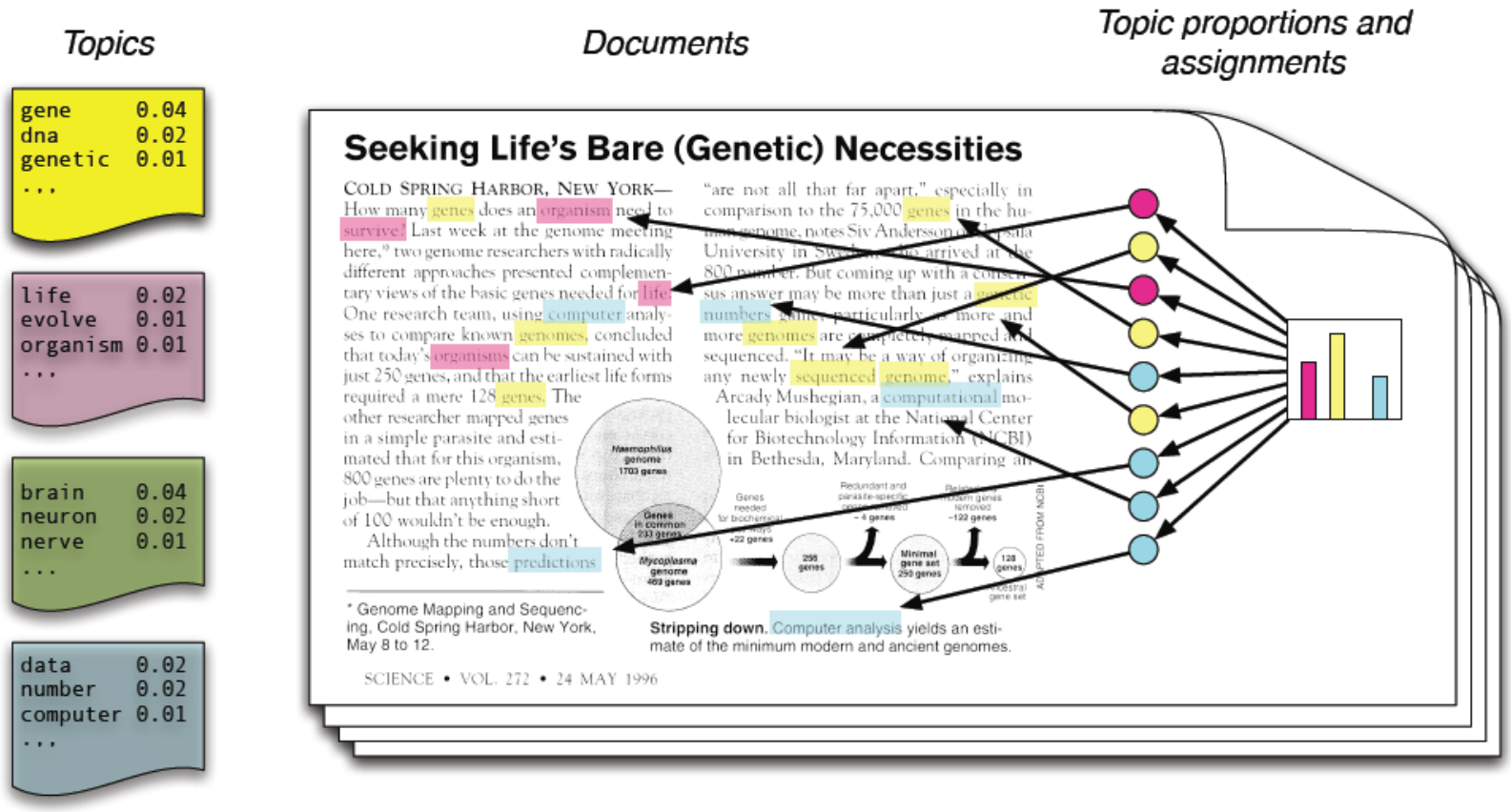
Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

SCIENCE • VOL. 272 • 24 MAY 1996

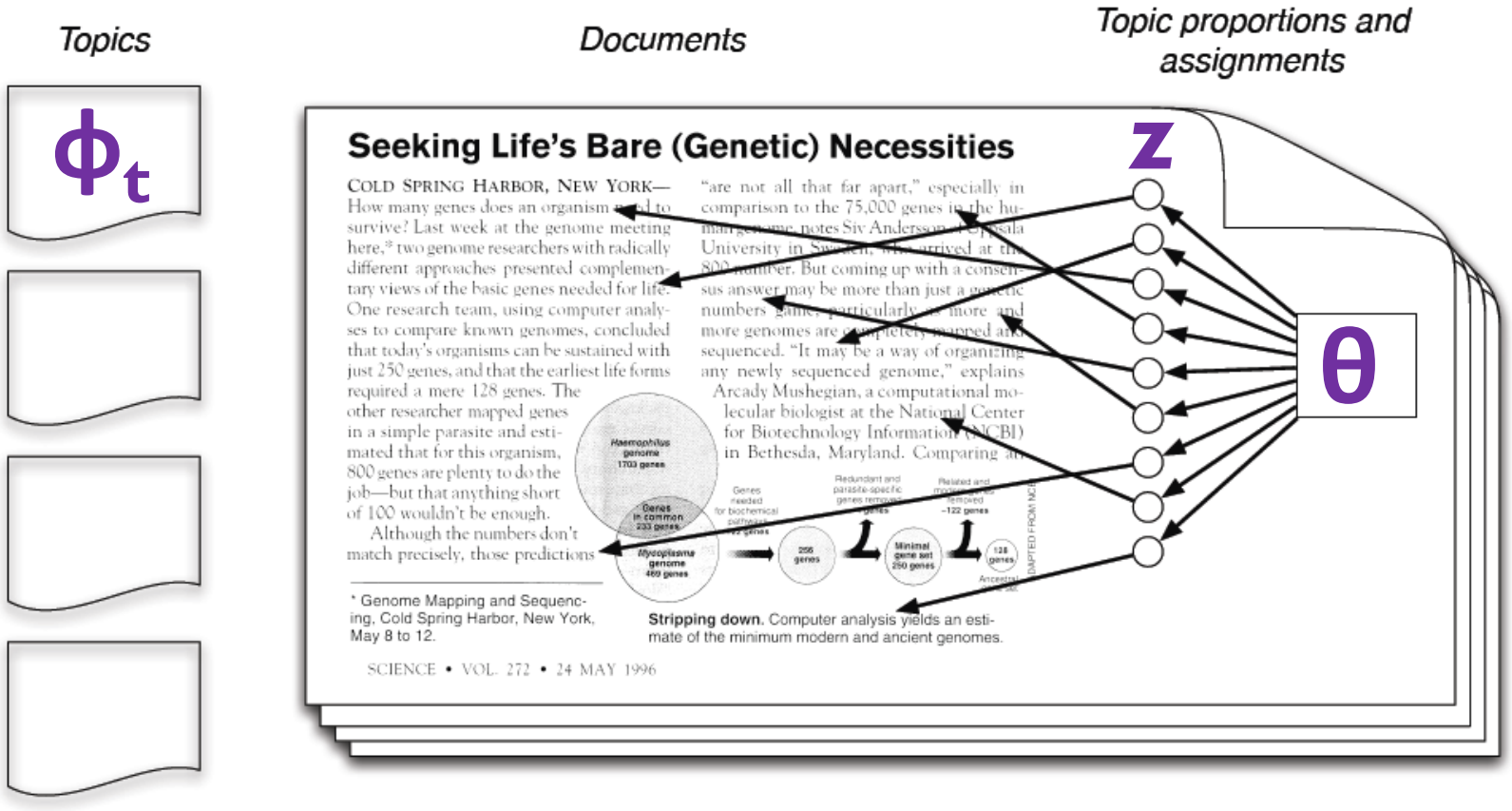
Simple intuition: Documents exhibit multiple topics.

From David Blei's 2012 ICML tutorial



- Each **topic** is a distribution over words
- Each **document** is a mixture of corpus-wide topics
- Each **word** is drawn from one of those topics

From David Blei's 2012 ICML tutorial



- In reality, we only observe the documents
- The other structure are **hidden variables**

From David Blei's 2012 ICML tutorial

LDA: Math Version

- For each topic t
  - Choose distribution $\phi_t \sim \text{Dirichlet}(\beta)$
- For each doc
  - Choose $\theta \sim \text{Dirichlet}(\alpha)$
  - For each token *i*
    - choose topic $z_i \sim \text{Mult}(\theta)$
    - choose word $w_i \sim \text{Mult}(\phi_{z_i})$

- Exact inference is intractable
  - We will use a collapsed sampler that integrates out ϕ and θ

[Griffiths and Steyvers, 2007]

Inference

- Variational and sampling-based methods exist

- Simple collapsed Gibbs sampling approach:
  - Initialize all topic variables z_i randomly to one of K topics
  - For each sampling pass
    - For each token i
      - Sample a new value for z_i given all other topic variable assignments

Sampling Distribution

- P(topic z | word w, doc d) $\propto \frac{n_z^d + \alpha}{n_\cdot^d + \alpha K} \frac{n_z^w + \beta}{n_\cdot^w + \beta V}$

- $n_z^d =$ number of times topic t assigned in doc d
- $n_z^w =$ number of times topic t assigned for word w
- K = number of topics
- V = number of unique words
- α, β : Dirichlet prior hyperparameters

Example Inference

human	evolution	disease	computer
genome	evolutionary	host	models
dna	species	bacteria	information
genetic	organisms	diseases	data
genes	life	resistance	computers
sequence	origin	bacterial	system
gene	biology	new	network
molecular	groups	strains	systems
sequencing	phylogenetic	control	model
map	living	infectious	parallel
information	diversity	malaria	methods
genetics	group	parasite	networks
mapping	new	parasites	software
project	two	united	new
sequences	common	tuberculosis	simulations

From David Blei's 2012 ICML tutorial

From David Blei's 2012 ICML tutorial