Maximum Likelihood Estimation and Expectation Maximization – P2

Bhiksha Raj

Agenda

- Generative Models
- Fitting models to data
- Where'd the closed forms go?
- Dealing with missing information
- How expectation maximization solves all our problems

What is a generative model

- A model for the probability distribution of a data x
  - E.g. a multinomial, Gaussian etc.

- Computational equivalent: a model that can be used to “generate” data with a distribution similar to the given data x

Some "simple" generative models

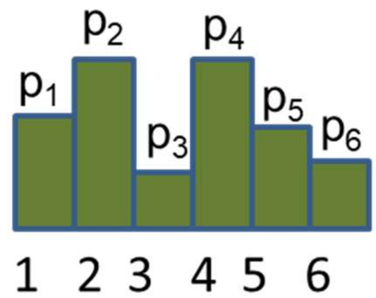
- The multinomial PMF

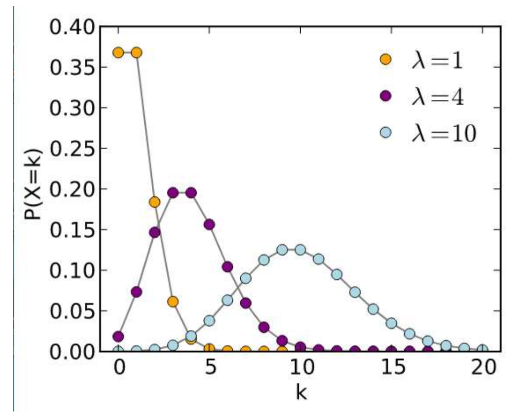
$$P(x = v) \equiv P(v)$$

  - For discrete data
    - v belongs to a discrete set
  - Can be expressed as a table of probabilities if the set of possible vs is finite
  - Else, requires a parametric form, e.g. Poisson

$$P(x = k) = \frac{\lambda^k e^{-\lambda}}{k!} \quad for\ k \geq 0$$

    - λ is the Poisson parameter



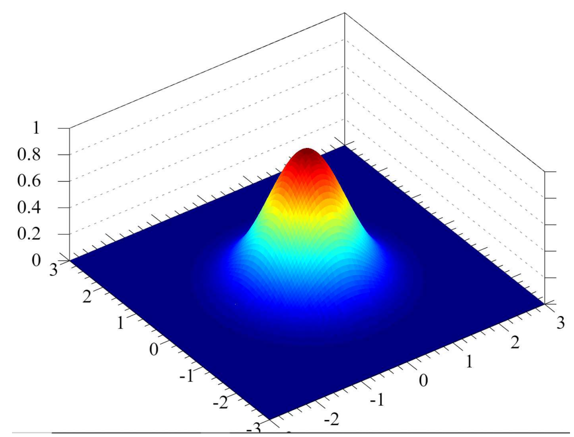


- The Gaussian PDF

$$P(x = v)$$

$$= \frac{1}{\sqrt{2\pi|\Sigma|}^D} \exp\left(-0.5(x-\mu)^T \Sigma^{-1} (x-\mu)\right)$$

  - For continuous-valued data
  - μ is the mean of the distribution
  - Σ is the Covariance matrix



Learning a generative model for data

- You are given some set of observed data $X = \{x\}$.
- You choose a model $P(x; \theta)$ for the distribution of x
  - θ are the parameters of the model

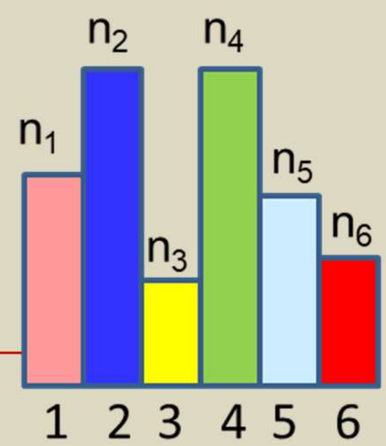
- Estimate the theta such that $P(x; \theta)$ best "fits" the observations $X = \{x\}$
  - Hoping it will also represent data outside the training set.

Defining “Best Fit”: Maximum likelihood

- Assumption: The world is a boring place
  - The data you have observed are very typical of the process
- Consequent assumption: The distribution has a high probability of generating the observed data
  - Not necessarily true
- Select the distribution that has the *highest* probability of generating the data

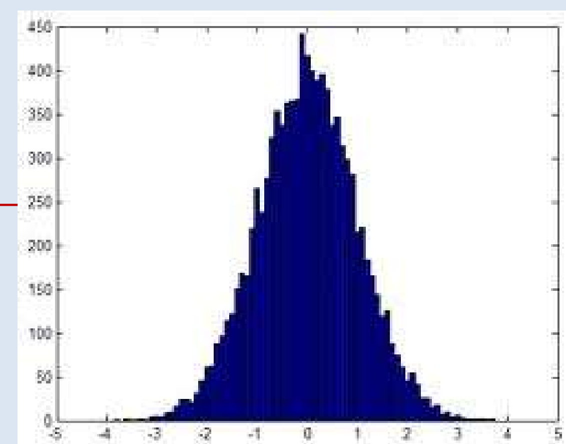
Maximum likelihood

- The maximum likelihood principle:
  - $\underset{\theta}{\operatorname{argmax}} P(X;\theta) = \underset{\theta}{\operatorname{argmax}} \log(P(X;\theta))$
- For the histogram
  - $\underset{\{p_1,p_2,p_3,p_4,p_5,p_6\}}{\operatorname{argmax}} \sum_i n_i \log(p_i)$



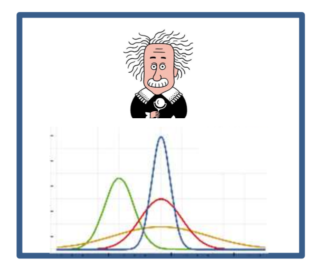
  $\Rightarrow p_i = \frac{n_i}{N}$ (N is the total number of observations)

- For the Gaussian
  - $\underset{\mu,\sigma^2}{\operatorname{argmax}} \sum_{x\in X} \log Gaussian\left(x;\mu,\sigma^2\right)$



  $\Rightarrow \mu = \frac{1}{N}\sum_{x\in X} x$; $\qquad \sigma^2 = \frac{1}{N}\sum_{x\in X}(x-\mu)^2$

The missing-info challenge



- In some estimation problems there is often some information missing

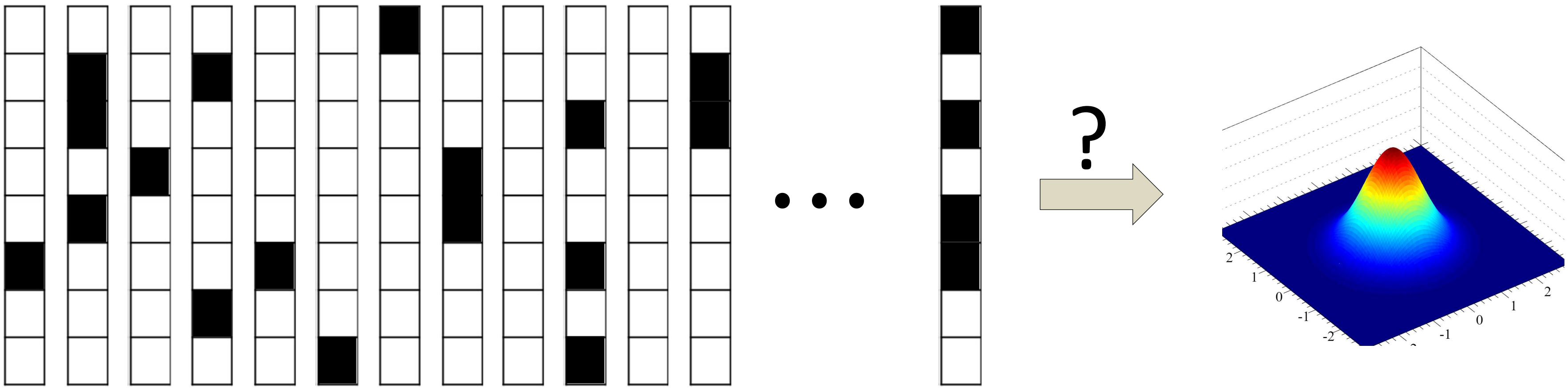
- If this information were available, estimation would’ve been trivial

Let's Look at Missing Information

Missing Information
about **Underlying Data**

Missing Information
about **Underlying Process**

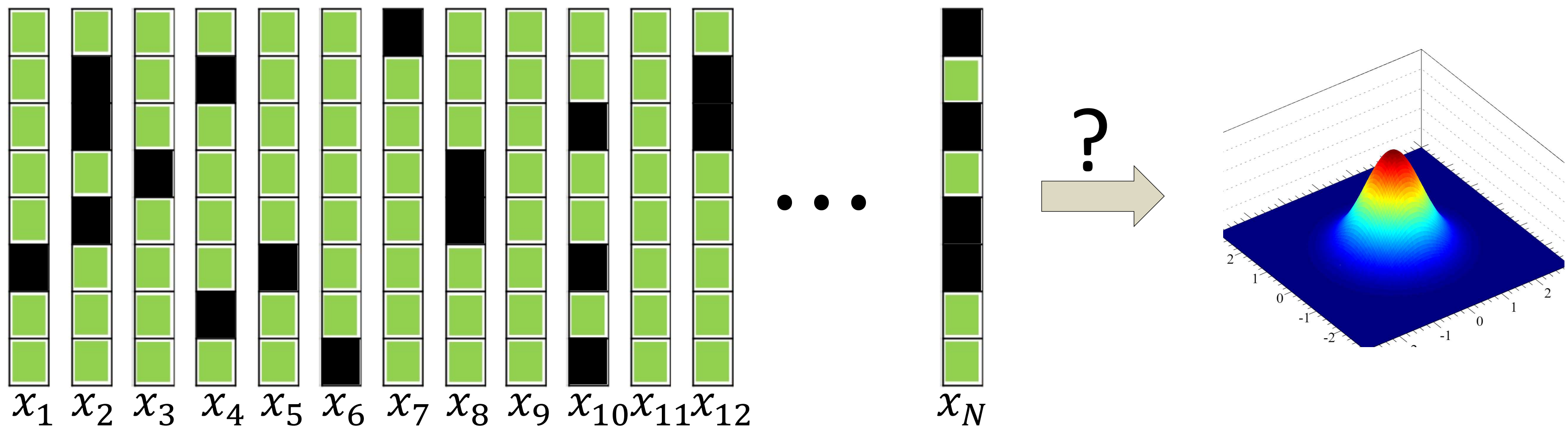
Examples of incomplete data: missing data



Blacked-out components are missing from data

- Objective: Estimate a Gaussian distribution from a collection of vectors
- Problem: Several of the vector components are missing
- Must estimate the mean and covariance of the Gaussian with these incomplete data
  - What would be a good way of doing this?

Maximum likelihood estimation with incomplete data



- Maximum likelihood estimation: Maximize the likelihood of the *observed* data

$$\underset{\mu,\Sigma}{\operatorname{argmax}} \log(P(O)) = \underset{\mu,\Sigma}{\operatorname{argmax}} \sum_{o \in O} \log \int_{-\infty}^{\infty} P(o,m)\,dm$$

- This requires the maximization of the log of an integral!
  - No closed form
  - Challenging on a good day, impossible on a bad one

Let’s Look at Missing Information

Missing Information
about **Underlying Data**

Missing Information
about **Underlying Process**

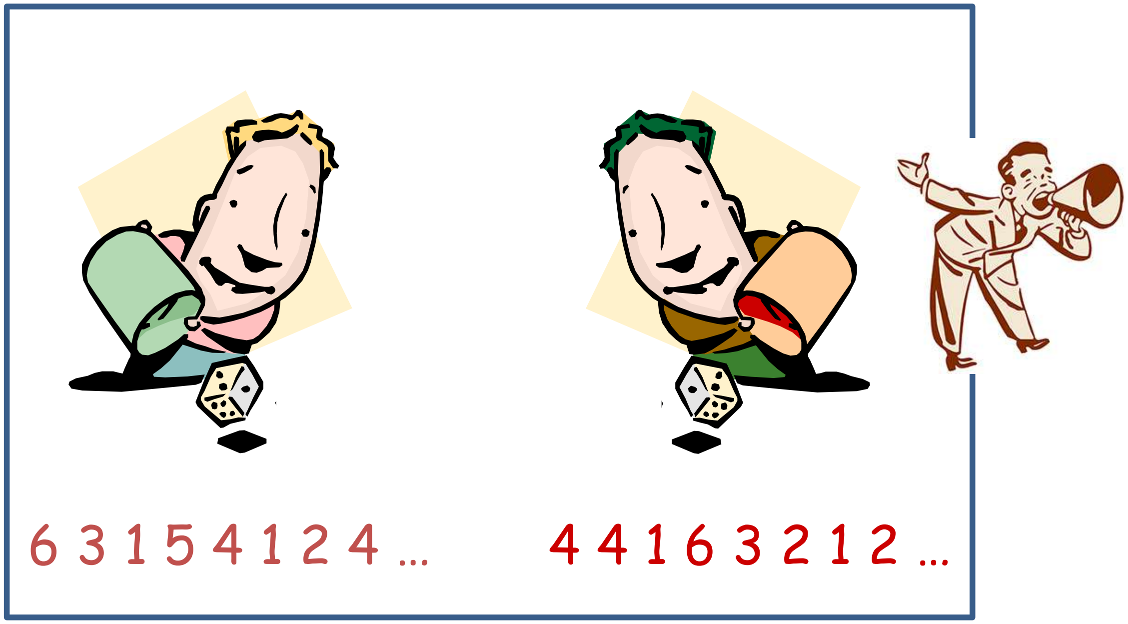
Let's Look at Missing Information

Missing Information
about **Underlying Data**

Missing Information
about **Underlying Process**

Shooting Dice **General Mixtures**

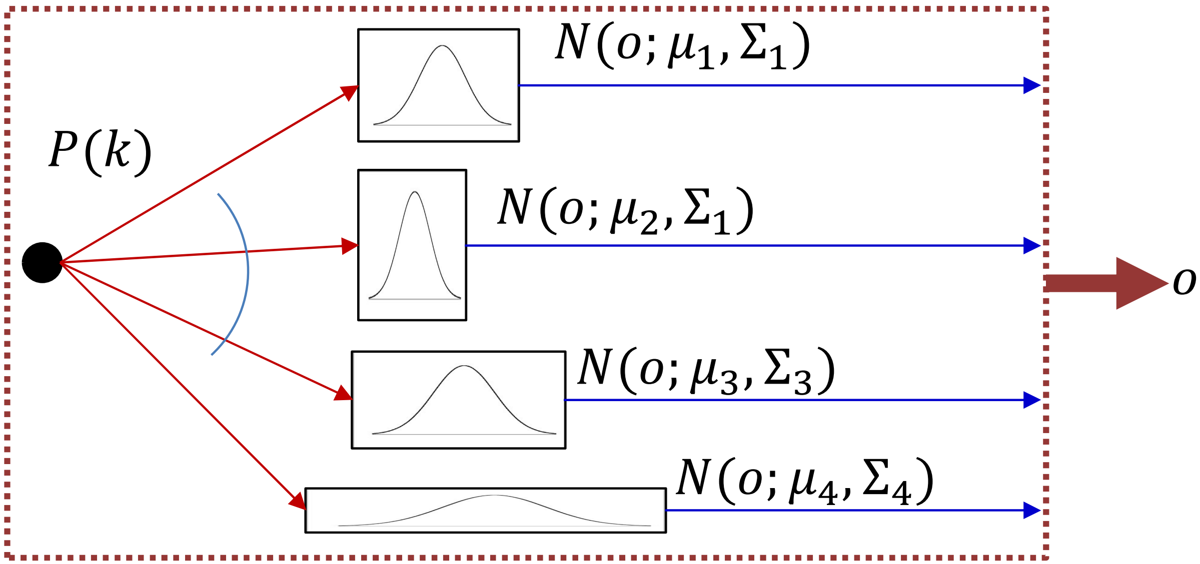
Our dice rolling gamblers

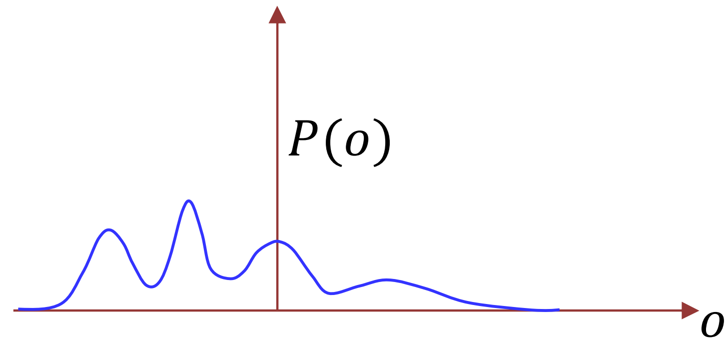


- Two persons shoot loaded dice repeatedly
  - The dice are differently loaded for the two of them
- We observe the series of outcomes for both persons

- **How to determine the probability distributions of the two dice?**

The Mixture Gaussian





$$P(o) = \sum_k P(k)N(o;\mu_k,\Sigma_k)$$

Mixture Gaussian

- The generative model randomly selects a Gaussian
- Then it draws an observation from the selected Gaussian
- Given only a collection of observations, how to estimate the parameters of the individual Gaussians, and the probability of selecting Gaussians?

The general form of the problem

- The "presence" of missing data or variables requires them to be marginalized out of your probability
  - By summation or integration

- This results in a maximum likelihood estimate of the form

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \sum_{o} \log \sum_{h} P\left(h, o; \theta\right)$$

  - The inner summation may also be an integral in some problems
  - Explicitly introducing θ in the RHS to show that the probability is computed by a model with parameter θ which must be estimated

- The log of a sum (or integral) makes estimation challenging
  - No closed form solution
  - Need efficient iterative algorithms

Expectation Maximization for Maximum Likelihood Estimation

- Objective: Estimate

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \sum_{o \in O} \log \sum_h P(h, o; \theta)$$

- Solution: Iteratively perform the following optimization instead

$$\theta^{k+1} \leftarrow \underset{\theta}{\operatorname{argmax}} \sum_{o \in O} \sum_h P(h|o; \theta^k) \log P(h, o; \theta)$$

- This maximizes an Empirical Lower Bound (ELBO) and guarantees increasing log likelihood with iterations
  - Giving you a *local maximum log likelihood* estimate for θ^*

Expectation Maximization for Maximum Likelihood Estimation

- Objective: Estimate

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \sum_{o \in O} \log \sum_h P(h, o; \theta)$$

- Solution: Iteratively perform the following optimization instead

$$\theta^{k+1} \leftarrow \underset{\theta}{\operatorname{argmax}} \sum_{o \in O} \sum_h P(h|o; \theta^k) \log P(h, o; \theta)$$

- This maximizes an Empirical Lower Bound (ELBO) and guarantees increasing log likelihood with iterations
  – Giving you a *local maximum log likelihood* estimate for θ^*

Expectation Maximization

- Initialize θ^0
- $k = 0$
- Iterate (over k) until $\log P(O;\theta)$ converges:
  - **Expectation Step**

    Compute $P(h|o;\theta^k)$ for all $o \in O$ for all h

  - **Maximization step**

$$\theta^{k+1} \leftarrow \underset{\theta}{\operatorname{argmax}} \sum_{o \in O} \sum_{h} P(h|o;\theta^k) \log P(h,o;\theta)$$

Expectation Maximization

Let's put this to work

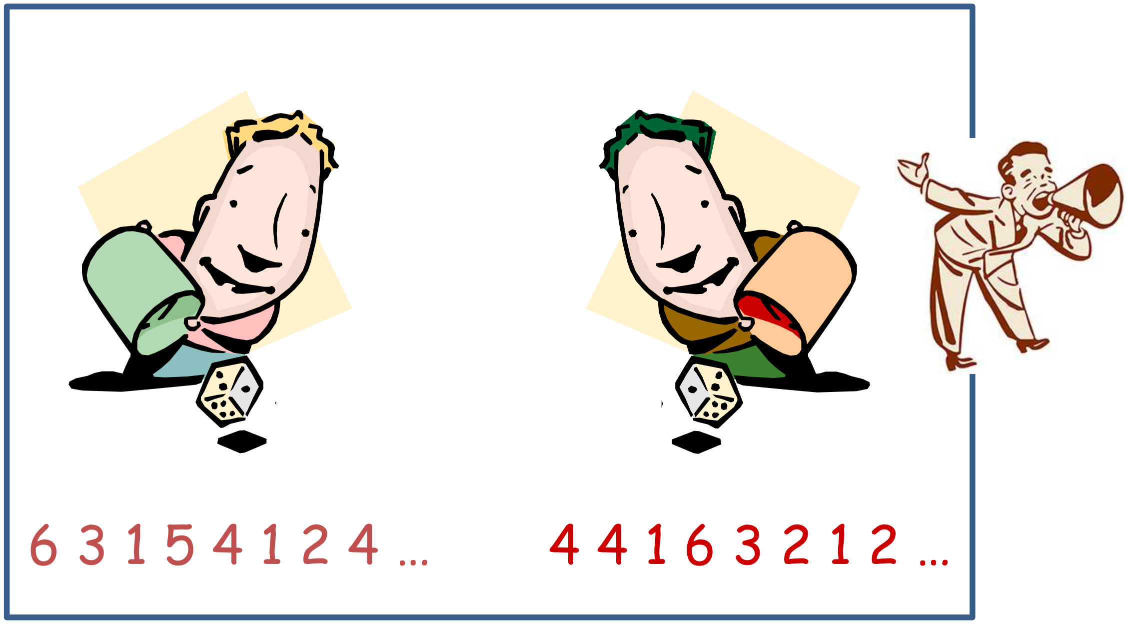
- Initialize θ^0
- $k = 0$
- Iterate (over k) until $\log P(O; \theta)$ converges:
  - **Expectation Step**

    Compute $P(h|o; \theta^k)$ for all $o \in O$ for all h

  - **Maximization step**

$$\theta^{k+1} \leftarrow \underset{\theta}{\operatorname{argmax}} \sum_{o \in O} \sum_{h} P(h|o; \theta^k) \log P(h, o; \theta)$$

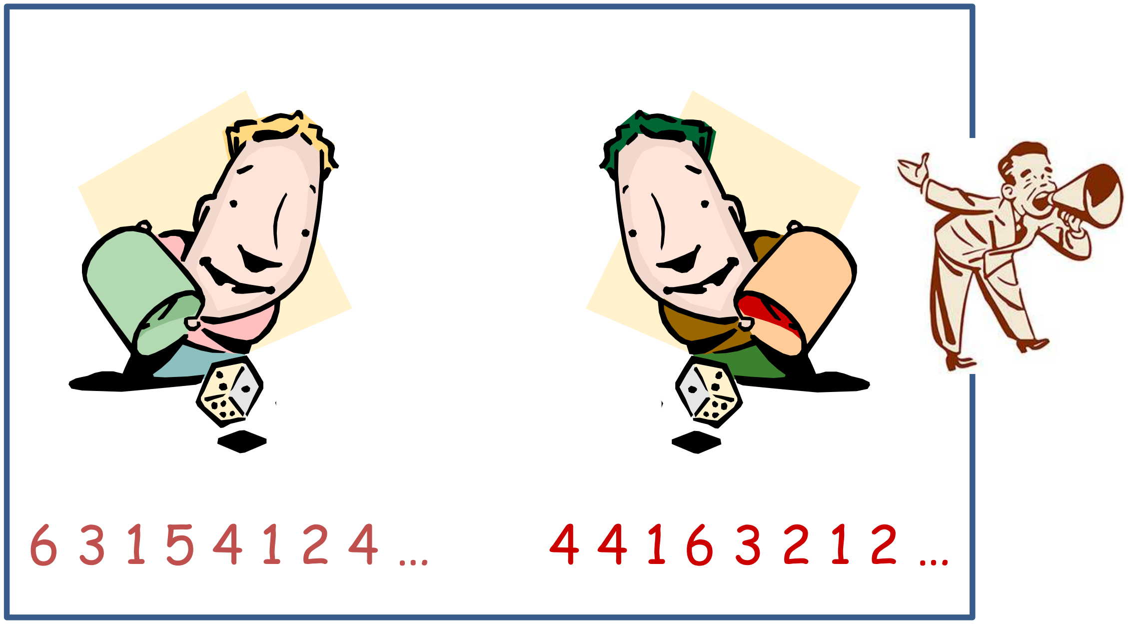
Our dice rolling gamblers



$$P(k,o) = P(k)P_k(o)$$

$$\boxed{P(o) = \sum_k P(k)P_k(o)}$$

Our dice rolling gamblers



$$P(k,o) = P(k)P_k(o)$$

$$P(o) = \sum_k P(k)P_k(o)$$

$$P(k|o) = \frac{P(k)P(o|k)}{P(o)}$$

$$P(k|o) = \frac{P(k)P_k(o)}{\sum_{k'} P(k')P_{k'}(o)}$$

Expectation Maximization

- Initialize θ^0
- $l = 0$
- Iterate (over l) until $\log P(O;\theta)$ converges:
  - **Expectation Step**

Compute $P(k|o;\theta^l)$ for all $o \in O$ for all k

$$P_{cur}(k|o) = \frac{P(k)P_k(o)}{\sum_{k'} P(k')P_{k'}(o)}$$

Using the current set of estimated parameters

Let's put this to work

Expectation Maximization

Let's put this to work

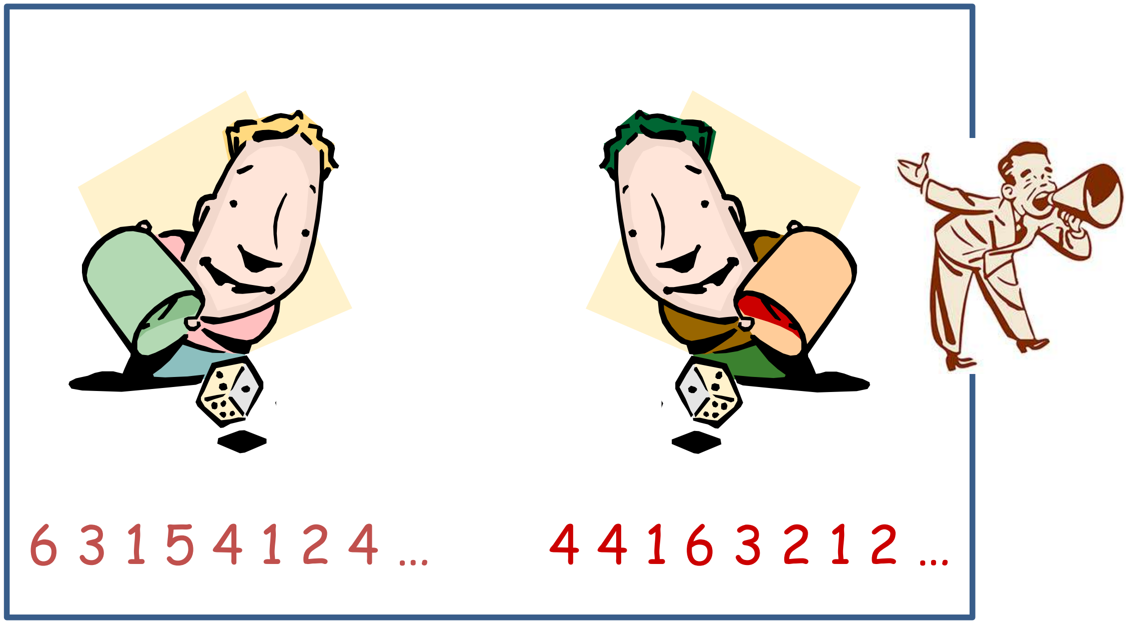
- Initialize θ^0
- $l = 0$
- Iterate (over l) until $\log P(O;\theta)$ converges:
  - **Expectation Step**

    Compute $P(k|o;\theta^l)$ for all $o \in O$ for all k

  - **Maximization step**

$$\theta^{l+1} \leftarrow \underset{\theta}{\operatorname{argmax}} \sum_{o \in O} \sum_{h} P(h|o;\theta^l) \log P(h,o;\theta)$$

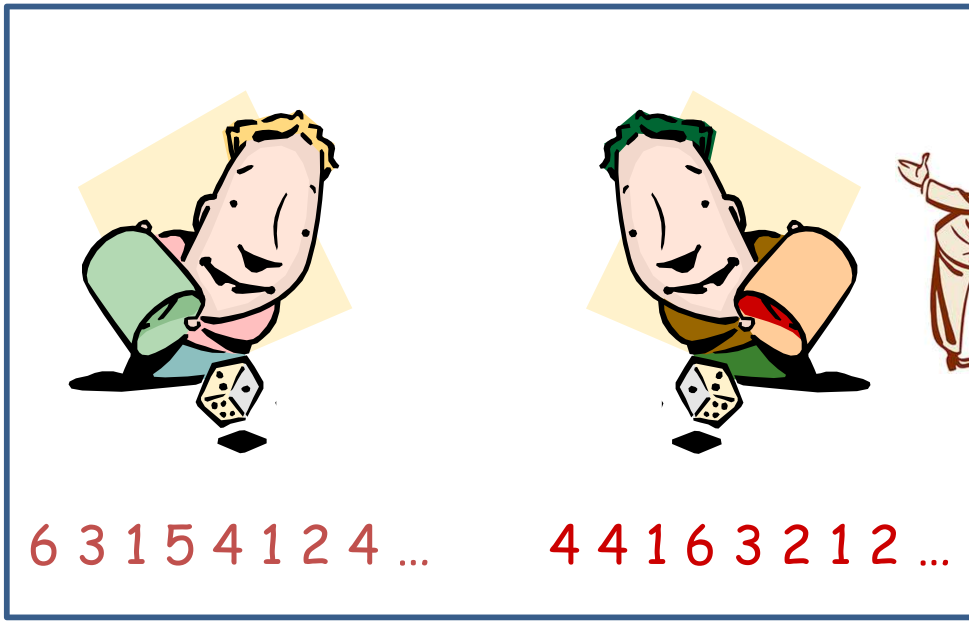
Our dice rolling gamblers



$$\underset{\theta}{\operatorname{argmax}} \sum_{o \in O} \sum_{h} P(h|o;\theta^k) \log P(h,o;\theta)$$

$$\underset{\theta}{\operatorname{argmax}} \sum_{o \in O} \sum_{k} P_{cur}(k|o) \log P(k) P_k(o)$$

Our dice rolling gamblers

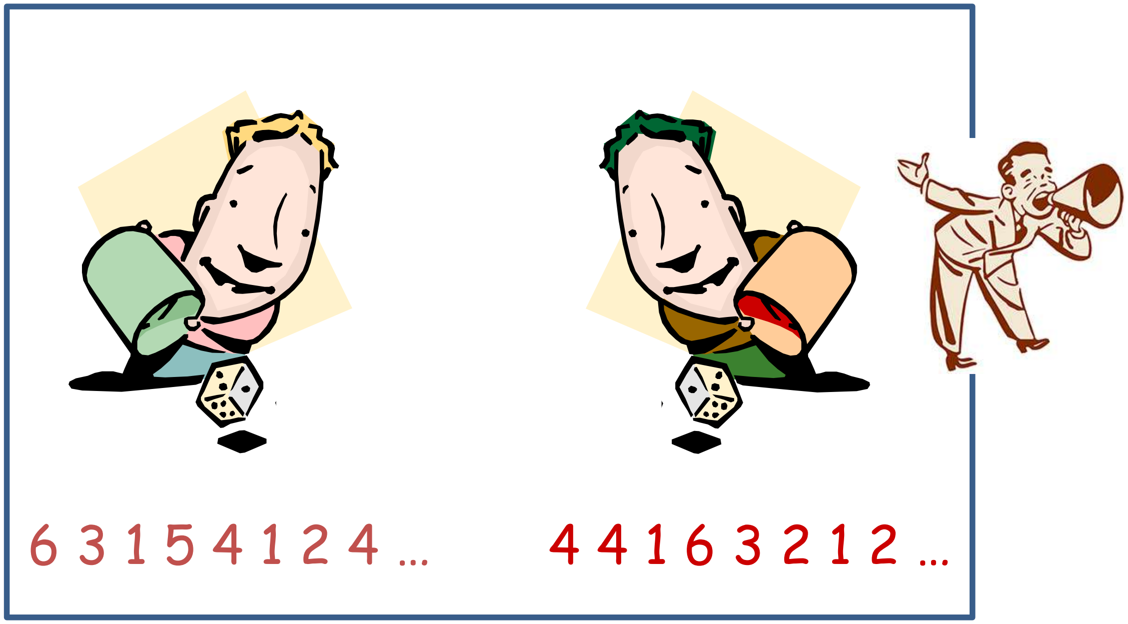


$$\underset{\theta}{\operatorname{argmax}} \sum_{o \in O} \sum_{h} P(h|o;\theta^k) \log P(h,o;\theta)$$

$$\underset{\theta}{\operatorname{argmax}} \sum_{o \in O} \sum_{k} P_{cur}(k|o) \log P(k) P_k(o) \; + \lambda \left(\sum_{k} P(k) - 1 \right) + \sum_{k} \lambda_k \left(\sum_{o} P_k(o) - 1 \right)$$

Differentiate and equate to 0

Our dice rolling gamblers

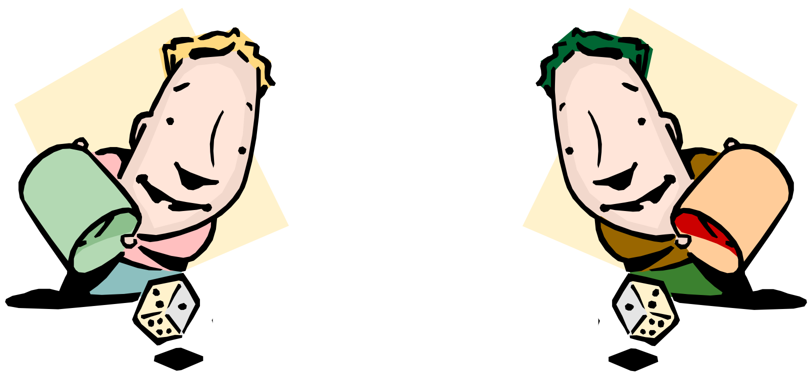


$$P_k(o) = \frac{N_o P_{cur}(k|o)}{\sum_{o'} N_{o'} P_{cur}(k|o')}$$

$$P_{cur}(k|o) = \frac{P(k)P_k(o)}{\sum_{k'} P(k')P_{k'}(o)}$$

$$P(k) = \frac{\sum_o N_o P_{cur}(k|o)}{\sum_{k'} \sum_o N_o P_{cur}(k'|o)}$$

Our dice rolling gamblers



6 3 1 5 4 1 2 4 …   4 4 1 6 3 2 1 2 …

$$P_{cur}(k|o) = \frac{P(k)P_k(o)}{\sum_{k'} P(k')P_{k'}(o)}$$

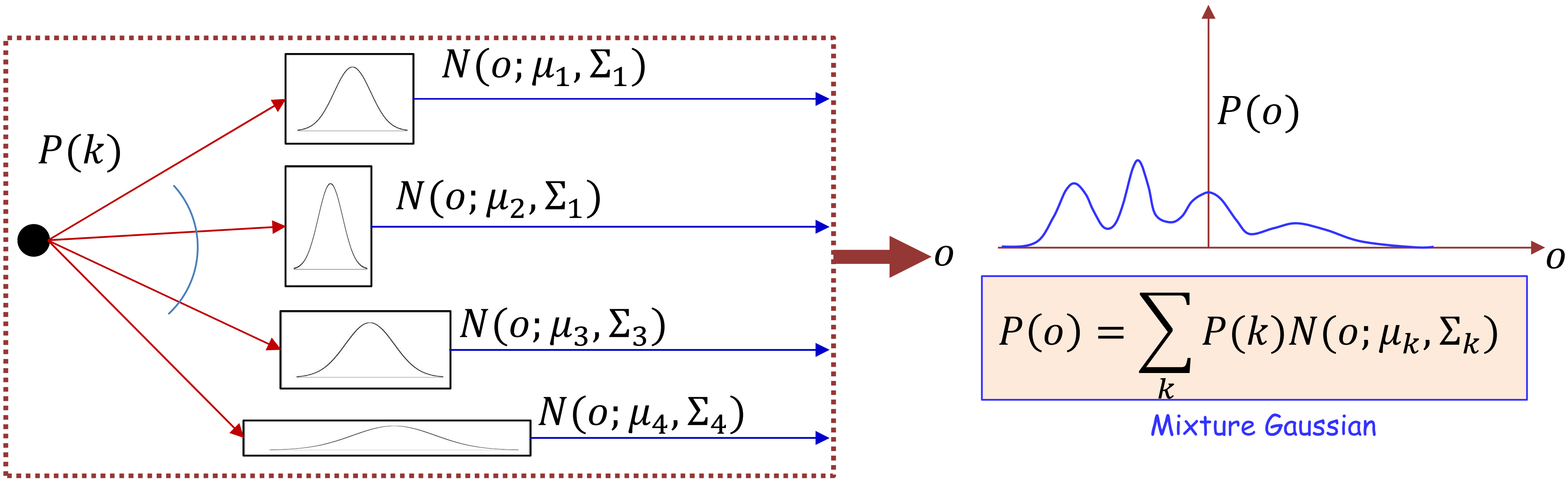
E

$$P_k(o) = \frac{N_o P_{cur}(k|o)}{\sum_{o'} N_{o'} P_{cur}(k|o')}$$

$$P(k) = \frac{\sum_o N_o P_{cur}(k|o)}{\sum_{k'} \sum_o N_o P_{cur}(k'|o)}$$

M

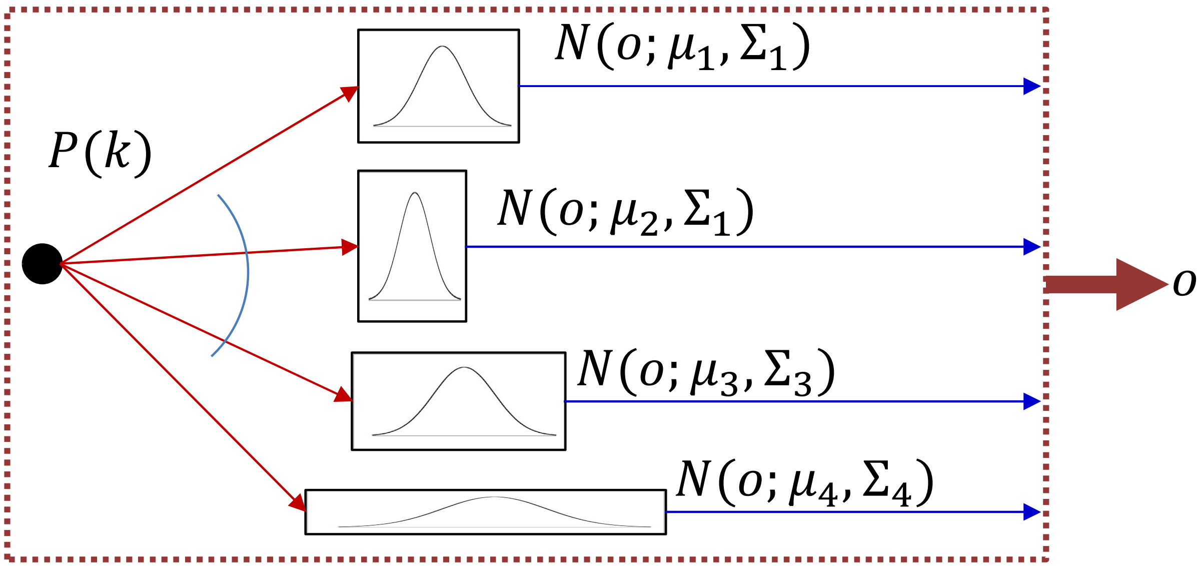
Examples of incomplete data: missing information in Gaussian mixtures

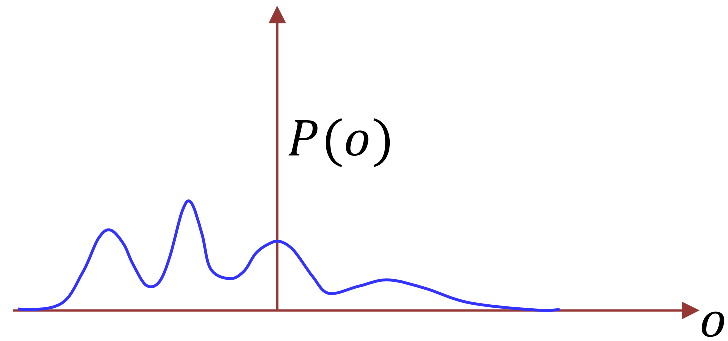


$$P(o) = \sum_k P(k)N(o;\mu_k,\Sigma_k)$$

Mixture Gaussian

Examples of incomplete data:
missing information in Gaussian mixtures





$$P(o) = \sum_k P(k)N(o;\mu_k,\Sigma_k)$$

Mixture Gaussian

$$P(k,o) = P(k)N(o;\mu_k,\Sigma_k)$$

$$P(k|o) = \frac{P(k)N(o;\mu_k,\Sigma_k)}{\sum_{k'} P(k')N(o;\mu_{k'},\Sigma_{k'})}$$

Expectation Maximization

Let's put this to work

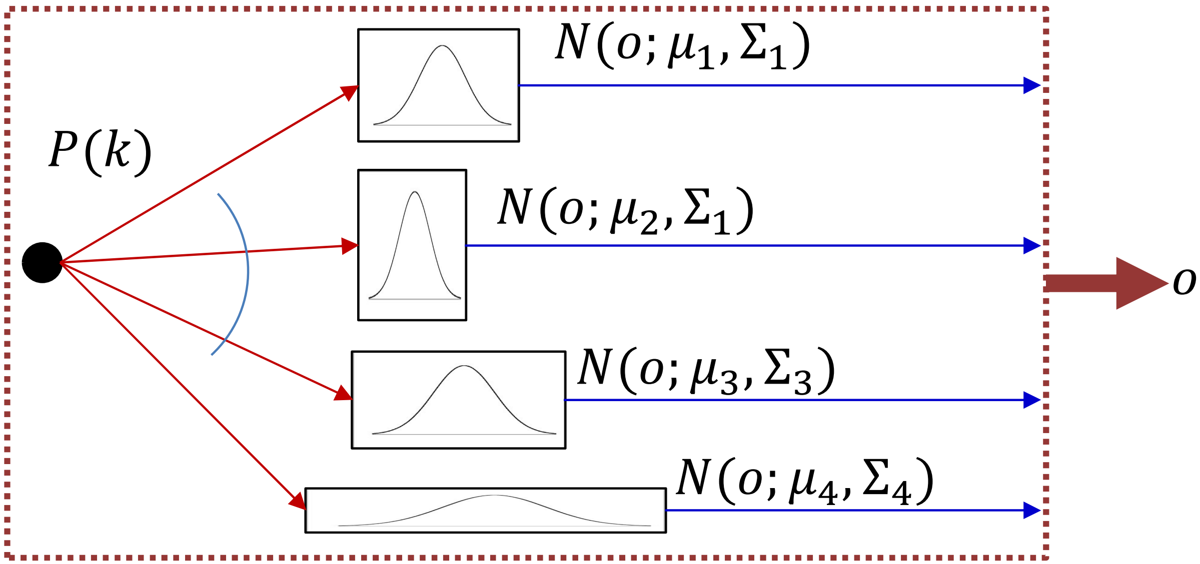
- Initialize θ^0
- $l = 0$
- Iterate (over l) until $\log P(O;\theta)$ converges:
  - **Expectation Step**

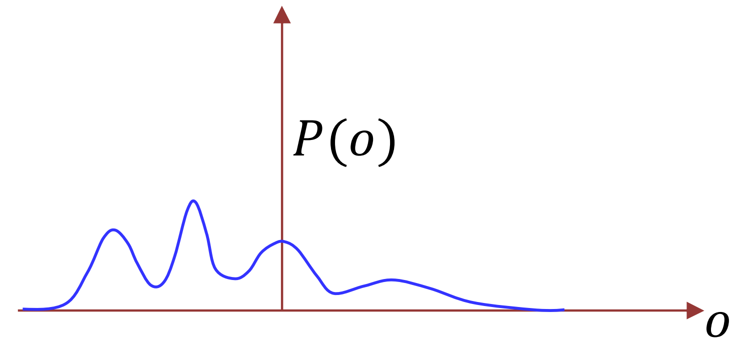
    Compute $P(k|o;\theta^l)$ for all $o \in O$ for all k

$$P(k|o;\theta^l) = \frac{P^l(k)N\left(o;\mu_k^l,\Sigma_k^{\mathrm{l}}\right)}{\sum_{k'} P^l(k')N(o;\mu_{k'}^l,\Sigma_{k'}^l)}$$

Using the current set of estimated parameters

The Mixture Gaussian





$$P(o) = \sum_k P(k) N(o;\mu_k,\Sigma_k)$$

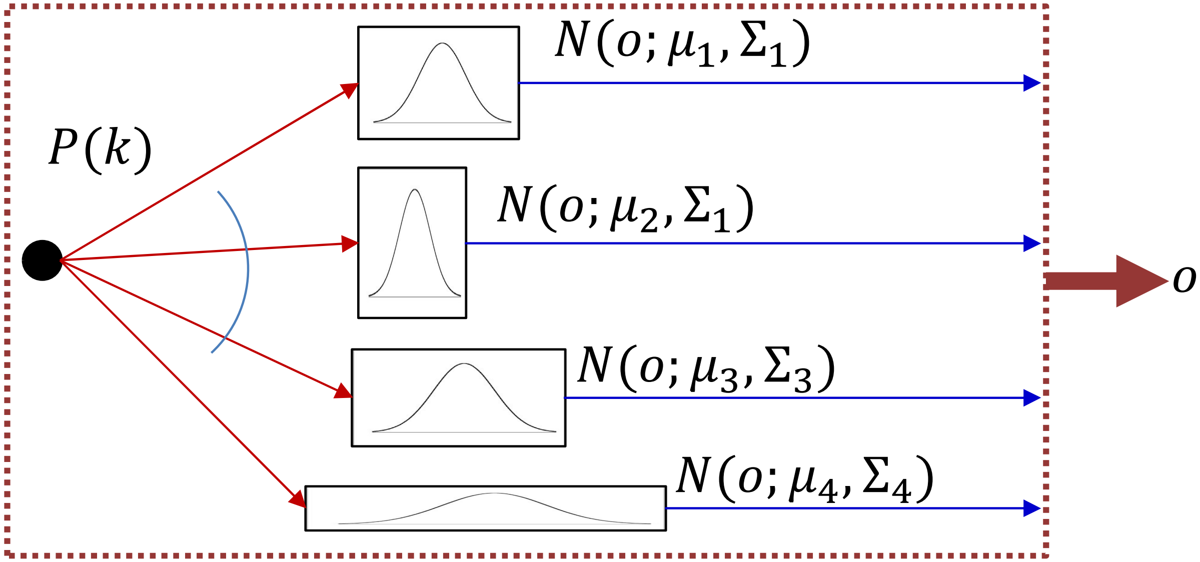
Mixture Gaussian

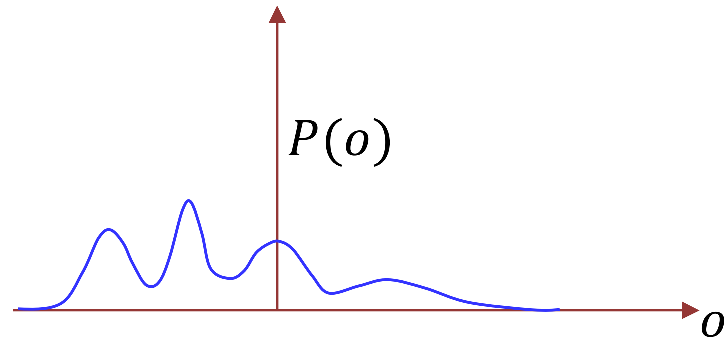
$$\underset{\theta}{\operatorname{argmax}} \sum_{o\in O} \sum_h P(h|o;\theta^l) \log P(h,o;\theta)$$

$$\underset{\{P(k),\mu_k,\Sigma_k\}}{\operatorname{argmax}} \sum_{o\in O} \sum_k P\left(k\middle|o;\theta^l\right)(\log P(k) + \log N(o;\mu_k,\Sigma_k)) + \lambda\left(\sum_k P(k) - 1\right)$$

Differentiate and equate to 0

The Mixture Gaussian





$$P(o) = \sum_k P(k) N(o;\mu_k,\Sigma_k)$$

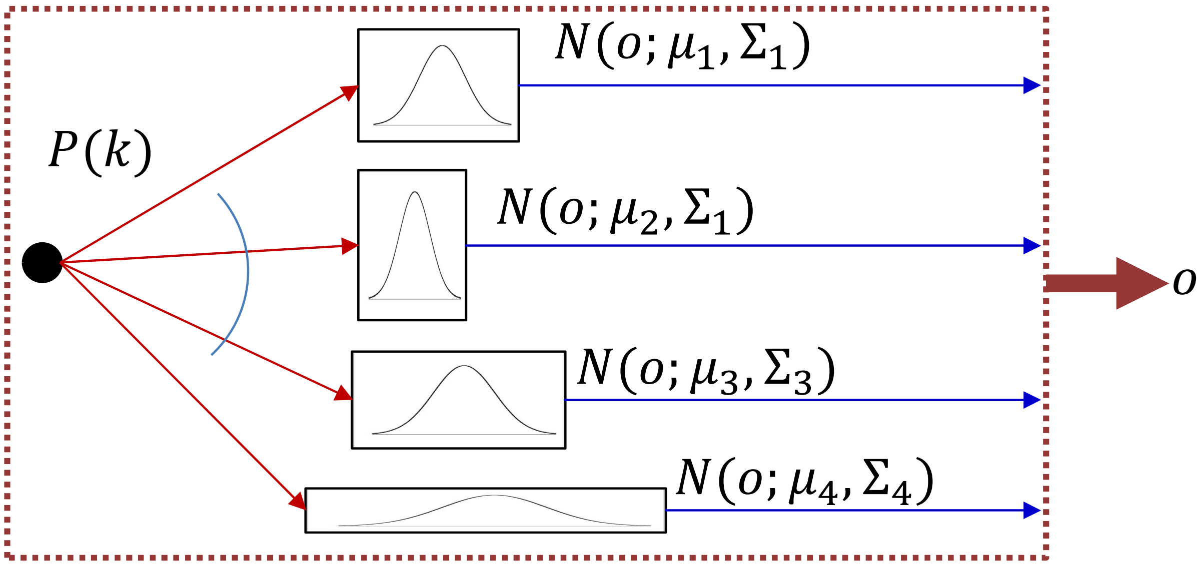
Mixture Gaussian

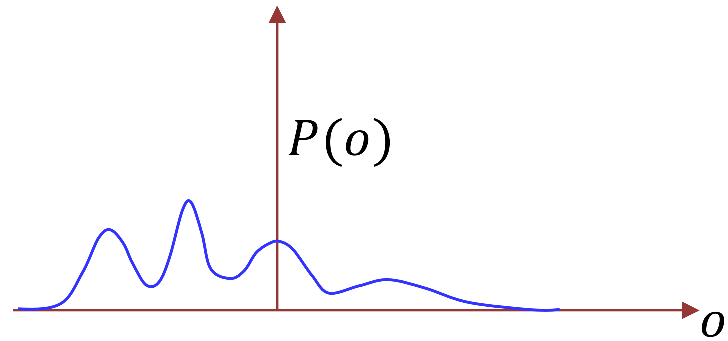
$$P^{l+1}(k) = \frac{1}{N}\sum_o P\left(k \middle| o;\theta^l\right)$$

$$\mu_k^{l+1} = \frac{1}{\sum_o P\left(k \middle| o;\theta^l\right)} \sum_o P\left(k \middle| o;\theta^l\right) o$$

$$\Sigma_k^{l+1} = \frac{1}{\sum_o P\left(k \middle| o;\theta^l\right)} \sum_o P\left(k \middle| o;\theta^l\right)(o-\mu_k^{l+1})(o-\mu_k^{l+1})^T$$

The Mixture Gaussian





$$P(o) = \sum_k P(k)N(o;\mu_k,\Sigma_k)$$

Mixture Gaussian

$$P(k|o;\theta^l) = \frac{P^l(k)N(o;\mu_k^l,\Sigma_k^l)}{\sum_{k'} P^l(k')N(o;\mu_{k'}^l,\Sigma_{k'}^l)}$$

$$P^{l+1}(k) = \frac{1}{N}\sum_o P(k|o;\theta^l)$$

$$\mu_k^{l+1} = \frac{1}{\sum_o P(k|o;\theta^l)}\sum_o P(k|o;\theta^l)o$$

$$\Sigma_k^{l+1} = \frac{1}{\sum_o P(k|o;\theta^l)}\sum_o P(k|o;\theta^l)(o-\mu_k^{l+1})(o-\mu_k^{l+1})^T$$

E **M**

Poll 1: tinyurl.com/mlsp23-20231109-1

- Select all true statements
  - The E step in the EM algorithm computes the a posteriori probability distribution of missing variables
  - The E step in EM maximizes the expectation over missing variables of the log of the probability of the complete data
  - The M step in the EM algorithm computes the a posteriori probability distribution of missing variables
  - The M step in EM maximizes the expectation over missing variables of the log of the probability of the complete data

Poll 1

- Select all true statements
  - **The E step in the EM algorithm computes the a posteriori probability distribution of missing variables**
  - The E step in EM maximizes the expectation over missing variables of the log of the probability of the complete data
  - The M step in the EM algorithm computes the a posteriori probability distribution of missing variables
  - **The M step in EM maximizes the expectation over missing variables of the log of the probability of the complete data**

That's so much math, but what does it really do?

- What does EM practically do when we have missing data?
  - What is the intuition behind how it resolves the problem?

Let's Look at Missing Information *again*

Let's Look at Missing Information *again*

Missing Information
about **Underlying Data**

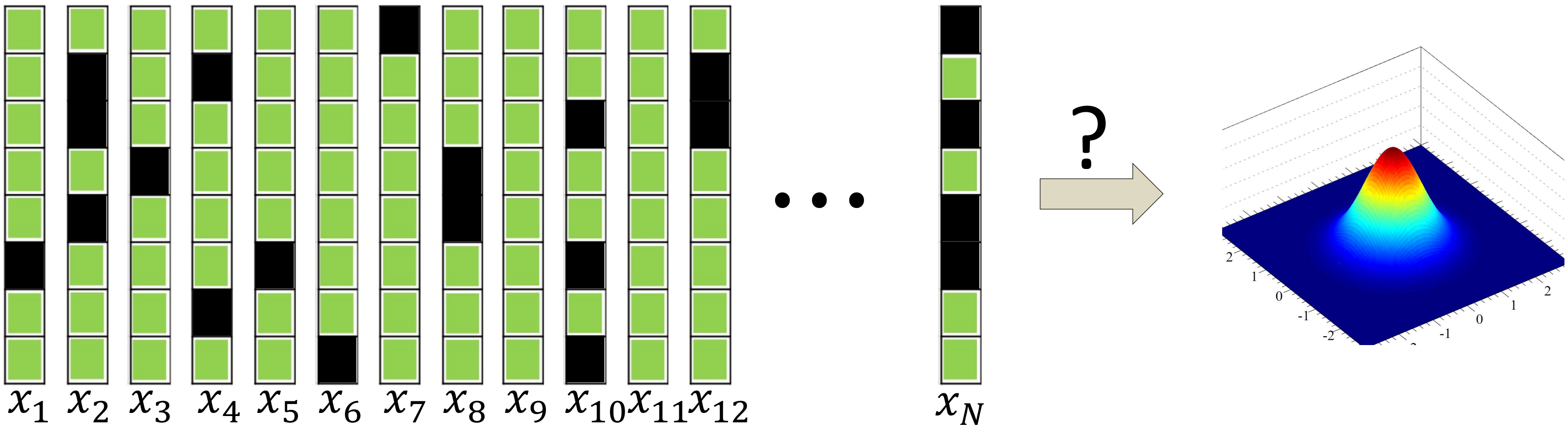
Missing Information
about **Underlying Process**

Let’s Look at Missing Information *again*

Missing Information
about **Underlying Data**

Missing Information
about **Underlying Process**

Recall this: Gaussian estimation with incomplete vectors

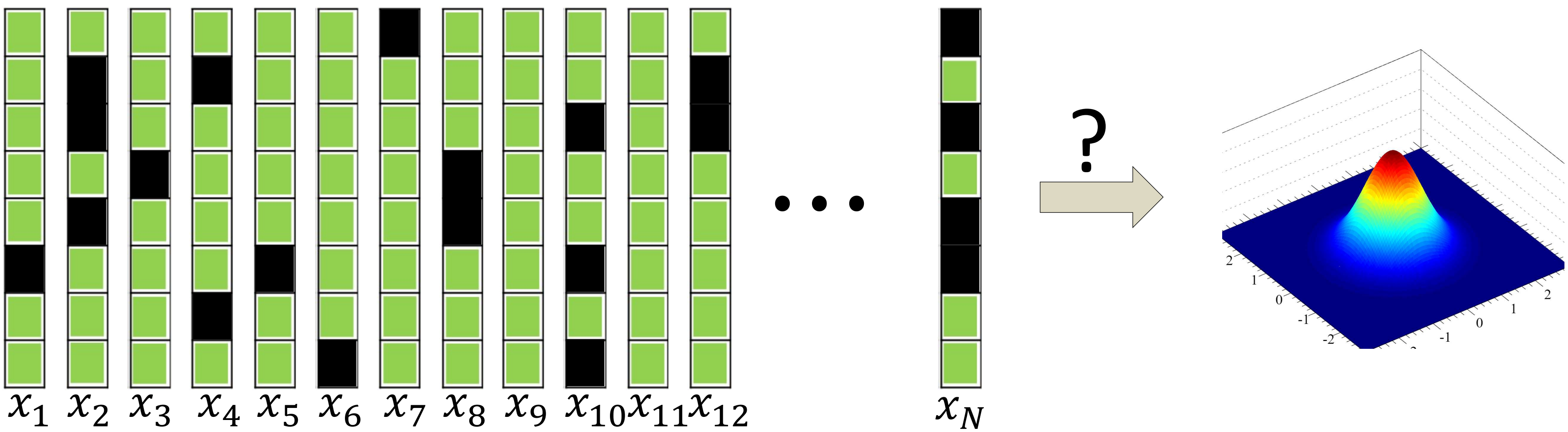


- These are the actual data we have: A set $O = \{o_1, \ldots, o_N\}$ of *incomplete* vectors
  - Comprising only the *observed* components of the data
- We are *missing* the data $M = \{m_1, \ldots, m_N\}$
  - Comprising the *missing* components of the data
- The *complete* data includes both the observed and missing components

$$X = \{x_1, \ldots, x_N\}, \qquad x_i = (o_i, m_i)$$

  - Keep in mind that at the complete data are *not* available (the missing components are missing)

Let's look at a single vector

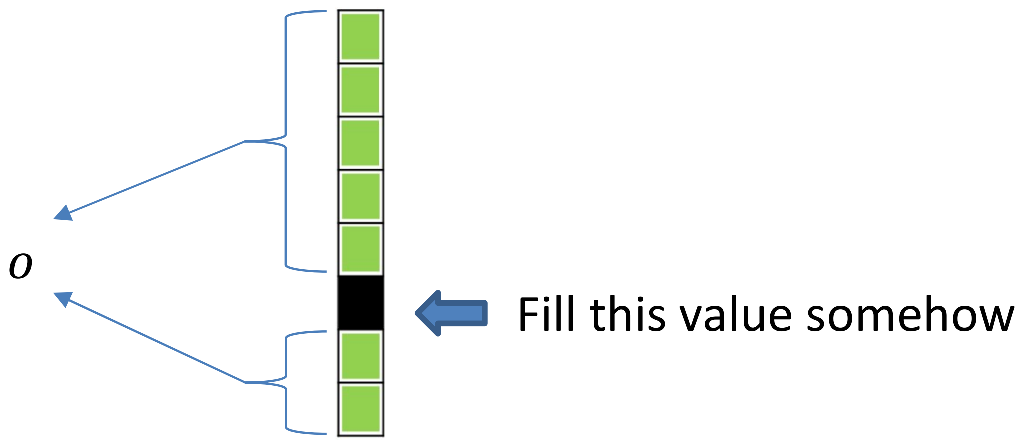


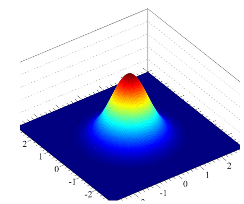
- These are the actual data we have: A set $O = \{o_1, \dots, o_N\}$ of *incomplete* vectors
  - Comprising only the *observed* components of the data
- We are *missing* the data $M = \{m_1, \dots, m_N\}$
  - Comprising the *missing* components of the data
- The *complete* data includes both the observed and missing components

$$X = \{x_1, \dots, x_N\}, \qquad x_i = (o_i, m_i)$$

  - Keep in mind that at the complete data are *not* available (the missing components are missing)

Let's look at a single vector

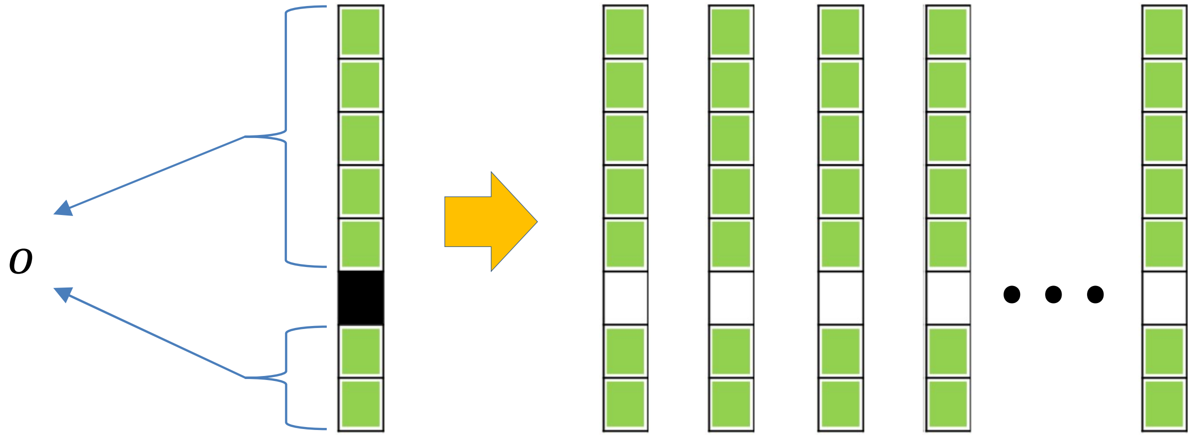


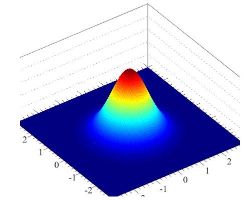


- We will try to complete the vector by filling in the missing value with *plausible* values that match the observed components

- Plausible: Values that "go with" the observed values, according to the distribution of the data

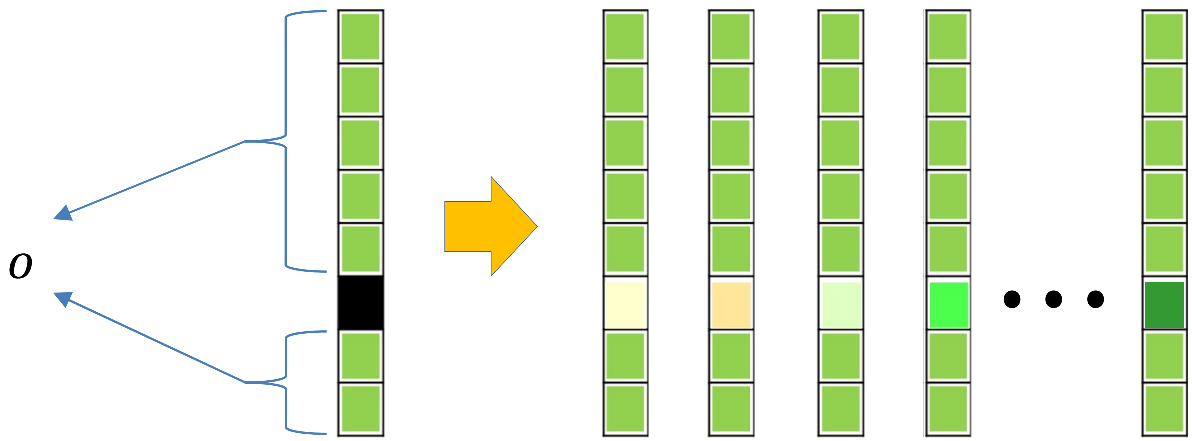
Let’s look at a single vector

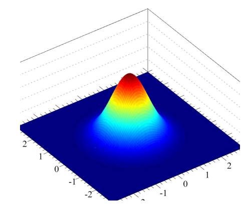




- Question: If we have a very large number of vectors from the Gaussian, all with the same observed components o, what would their missing components be?

Let’s look at a single vector

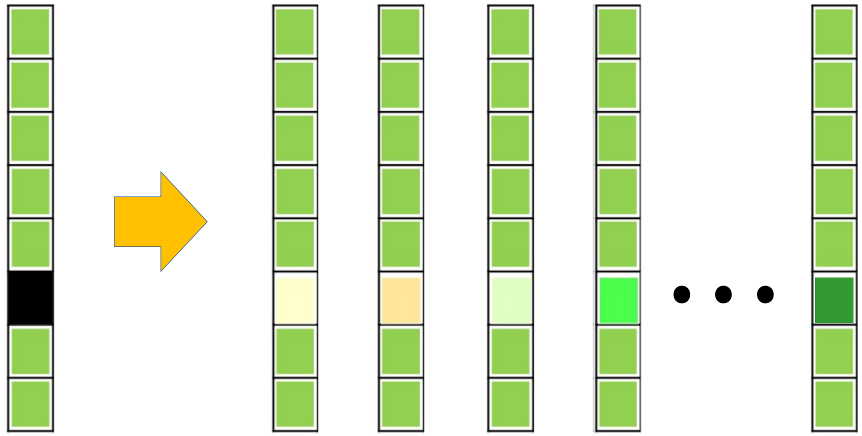




- Question: If we have a very large number of vectors from the Gaussian, all with the same observed components o, what would their missing components be?

- We would see every possible value, but in proportion to their probability: $P(m|o)$ (conditioned on the observations)

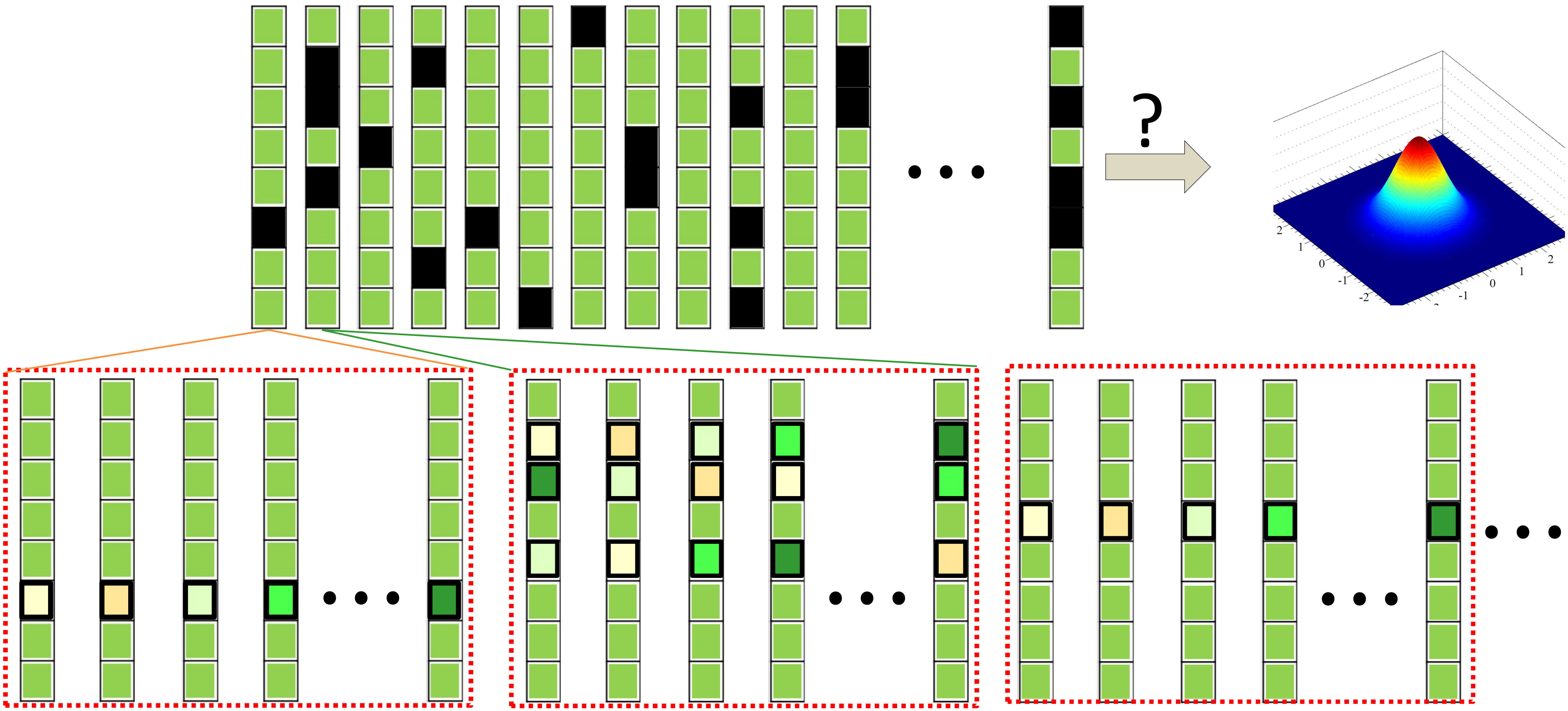
Completing incomplete vectors



in proportion: $P(*\,|o)$

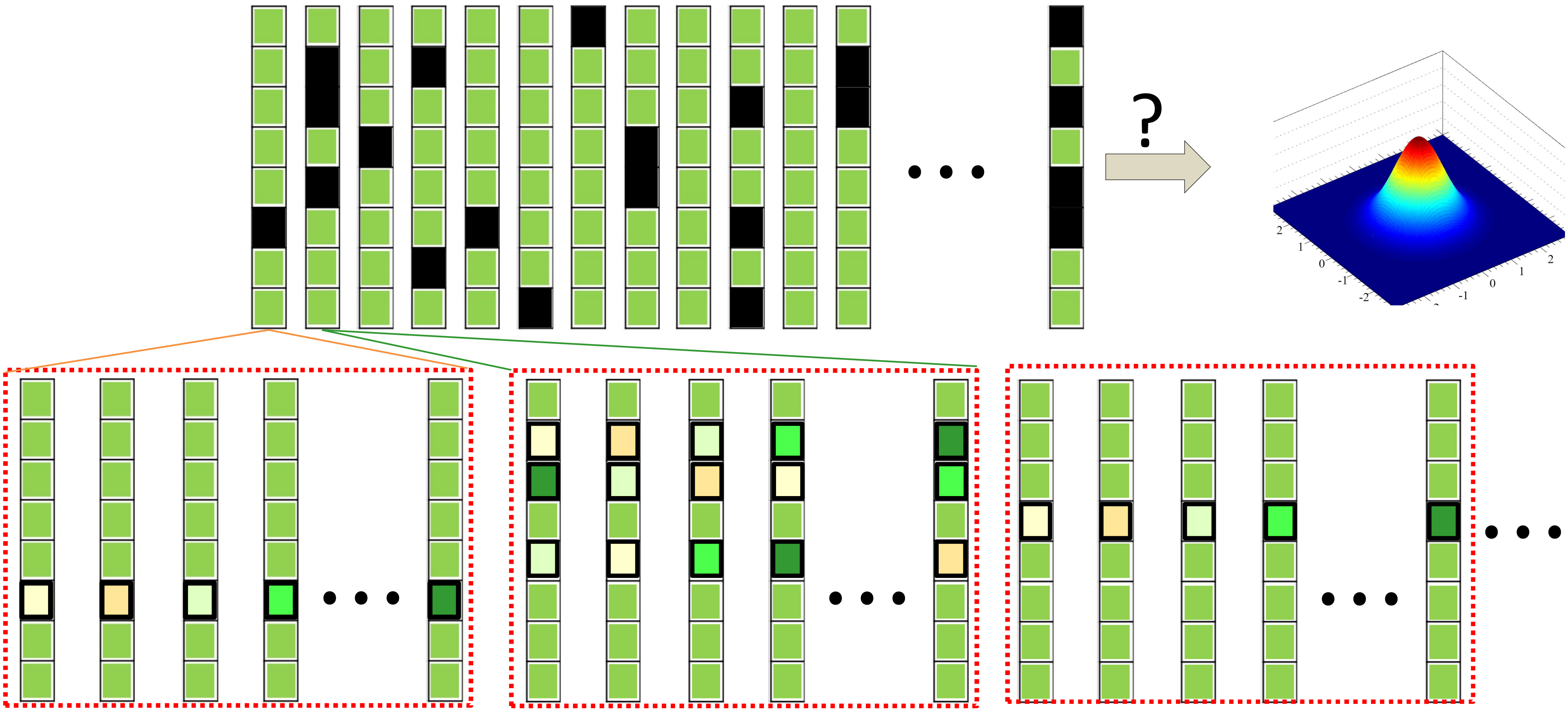
- Complete vector by filling up the missing components with *every possible value*
  - I.e. make many complete “clones” of the incomplete vector
- But assign a *proportion* to each value
  - Proportion is $P(m|o)$
    - Which can be computed if we know $P(x) = P(o,m)$

Gaussian estimation with incomplete vectors



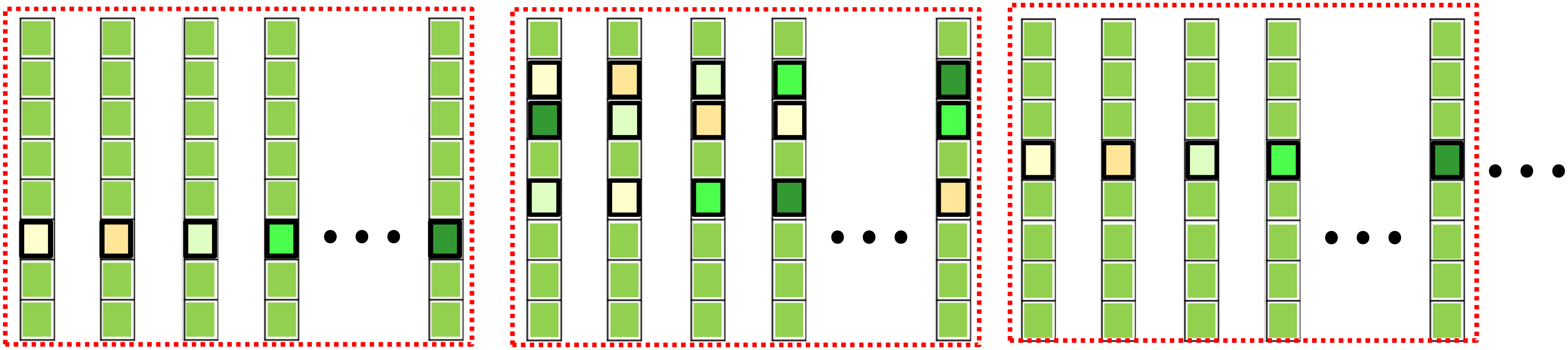
- "Expand" every incomplete vector out into all possibilities
  - In appropriate proportions $P(m|o)$
  - For already complete observations, there is no expansion
- Estimate the statistics from the expanded data

Gaussian estimation with incomplete vectors



- “Expand” every incomplete vector out into all possibilities
  - In appropriate proportions $P(m|o)$ From a previous estimate of the model
  - For already complete observations, there is no expansion
- Estimate the statistics from the expanded data

Estimating the Gaussian Parameters



- Compute the statistics from the (proportionately) expanded set
- Let $x_i(m)$ be the "completed" version of the observation o_i, when the missing components are filled with value m

$$x_i(m) = (m, o_i)$$

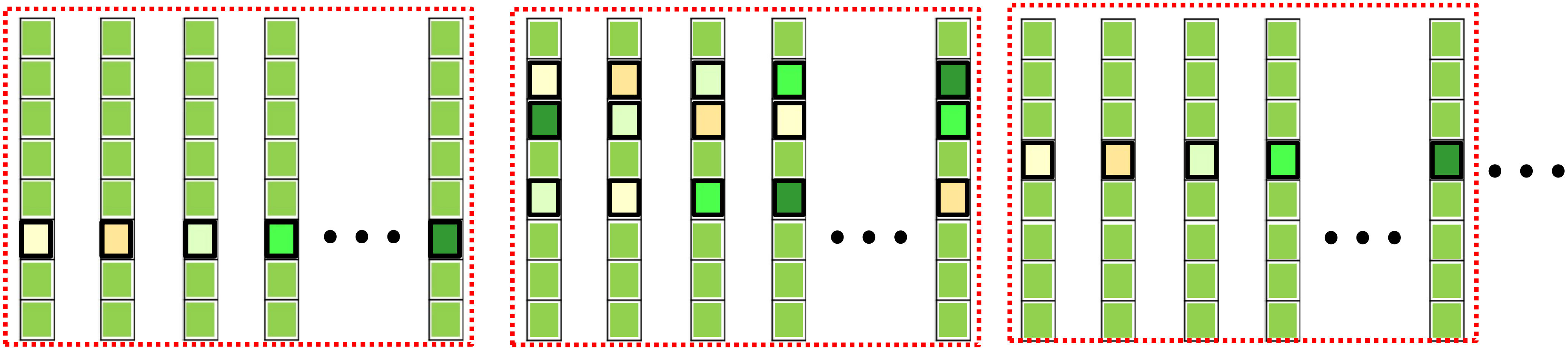
  - There will be one such vector for every value of m

- Estimate the statistics from the expanded data

$$\mu^{k+1} = \frac{1}{N} \sum_{o \in O} \int_{-\infty}^{\infty} P(m|o;\theta^k) x_i(m) dm$$

$$\Sigma^{k+1} = \frac{1}{N} \sum_{o \in O} \int_{-\infty}^{\infty} P(m|o;\theta^k)(x_i(m) - \mu^{k+1})(x_i(m) - \mu^{k+1})^T dm$$

EM for computing the Gaussian Parameters



- Initial $\theta^0 = (\mu^0, \Sigma^0)$
- Until $P(O;\theta)$ converges:

$$\mu^{k+1} = \frac{1}{N}\sum_{o \in O}\int_{-\infty}^{\infty} P\left(m \middle| o;\theta^k\right) x_i(m)\, dm$$

$$\Sigma^{k+1} = \frac{1}{N}\sum_{o \in O}\int_{-\infty}^{\infty} P\left(m \middle| o;\theta^k\right)\left(x_i(m) - \mu^{k+1}\right)\left(x_i(m) - \mu^{k+1}\right)^T dm$$

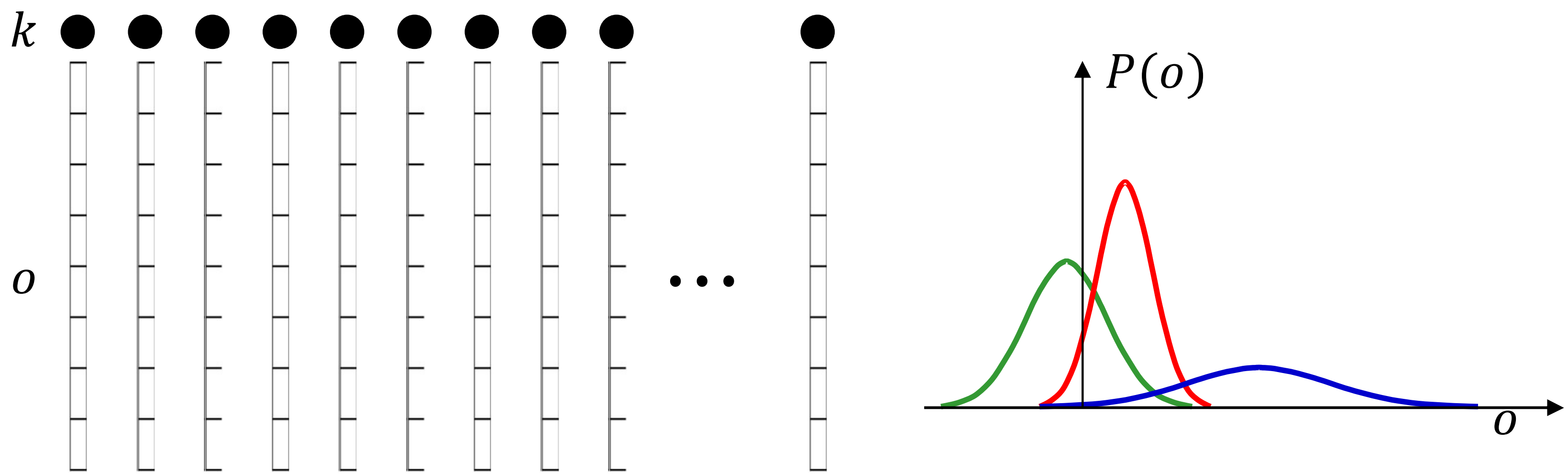
Where $x_i(m) = (m, o_i)$ and the parameters of $P\left(m \middle| o;\theta^k\right)$ are derived from the $P\left(x;\theta^k\right) = Gaussian(x;\mu^k,\Sigma^k)$

Let's Look at Missing Information *again*

Missing Information
about **Underlying Data**

Missing Information
about **Underlying Process**

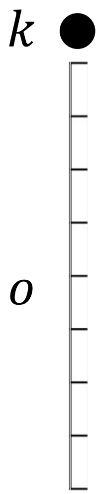
The GMM problem of incomplete data: missing information

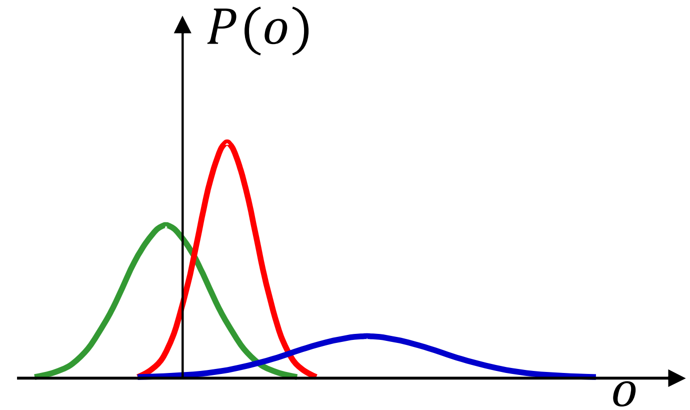


- Problem : We are not given the actual Gaussian for each observation
  - Our data are incomplete

- What we want : $(o_1, k_1), (o_2, k_2), (o_3, k_3)$ ...
- What we have: o_1, o_2, o_3 ...

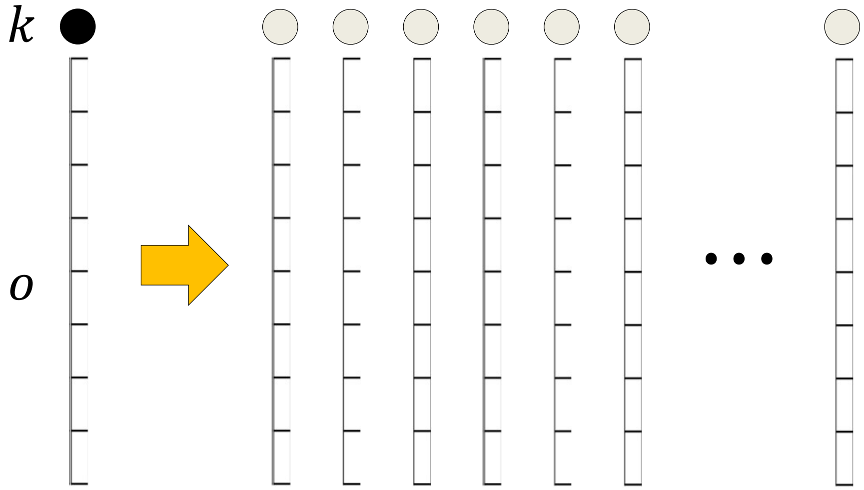
Consider a single vector

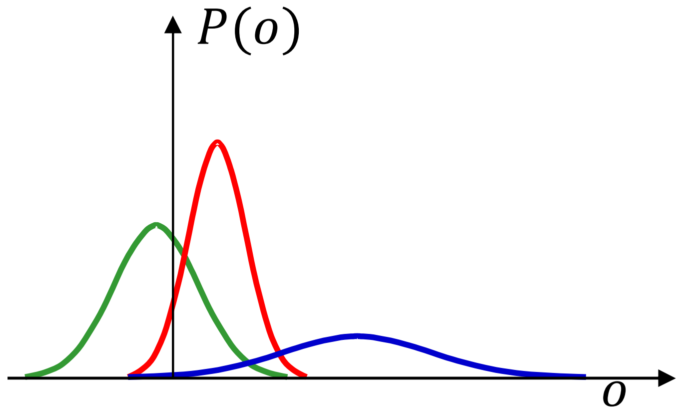




- *Every* Gaussian is capable of generating this vector
  – With different probabilities

Consider a single vector

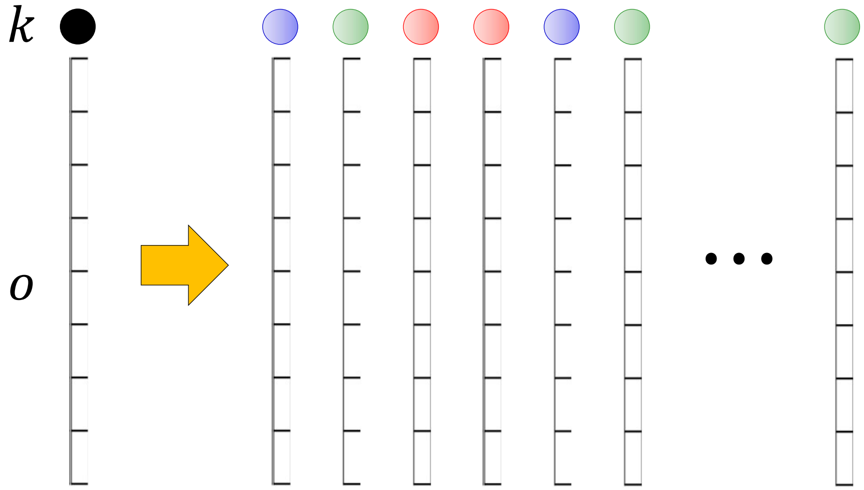


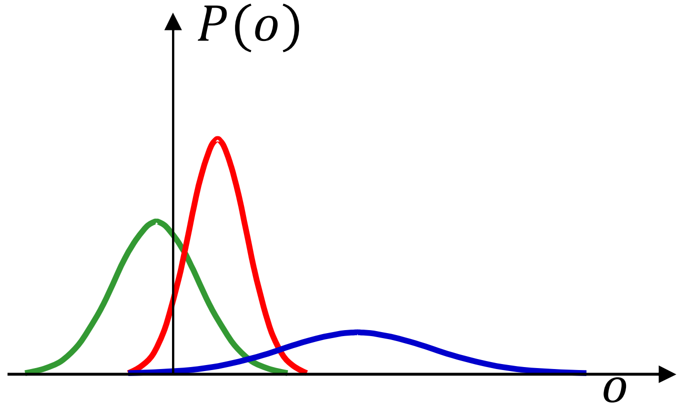


- *Every* Gaussian is capable of generating this vector
  – With different probabilities

- If we saw a large number of these vectors, how many of these would have come from each Gaussian?

Consider a single vector



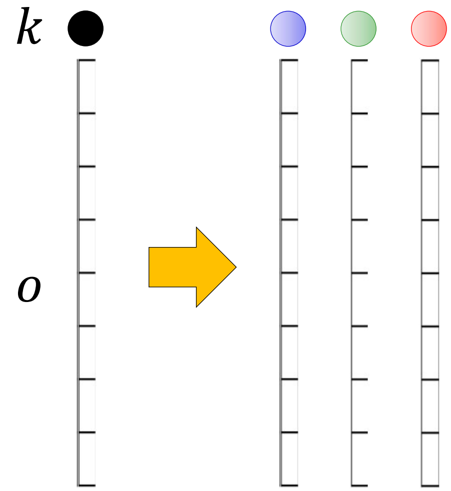


- *Every* Gaussian is capable of generating this vector
  - With different probabilities

- If we saw a large number of these vectors, how many of these would have come from each Gaussian

- All of them, but in proportion to $P(k|o)$

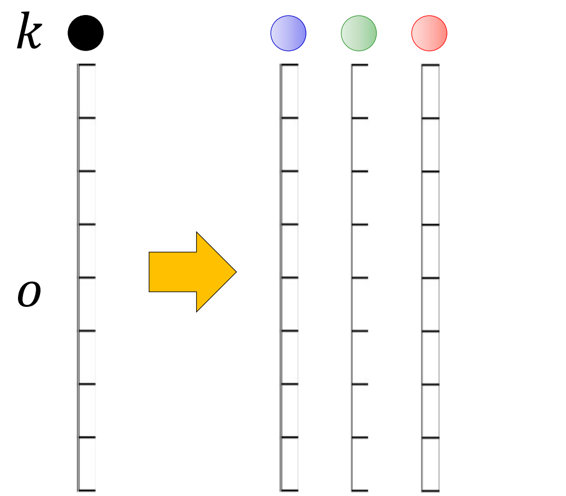
Completing incomplete vectors



$in\ proportion\colon\ P(*\,|o)$

- Complete the data by attributing to *every Gaussian*
  - I.e. make many complete “clones” of the data
- But assign a *proportion* to each completed vector
  - Proportion is $P(k|o)$
    - Which can be computed if we know $P(k)$ and $P(o|k)$
- Then estimate the parameters using the complete data

Completing incomplete vectors



$in\ proportion\colon\ P(*\,|o)$

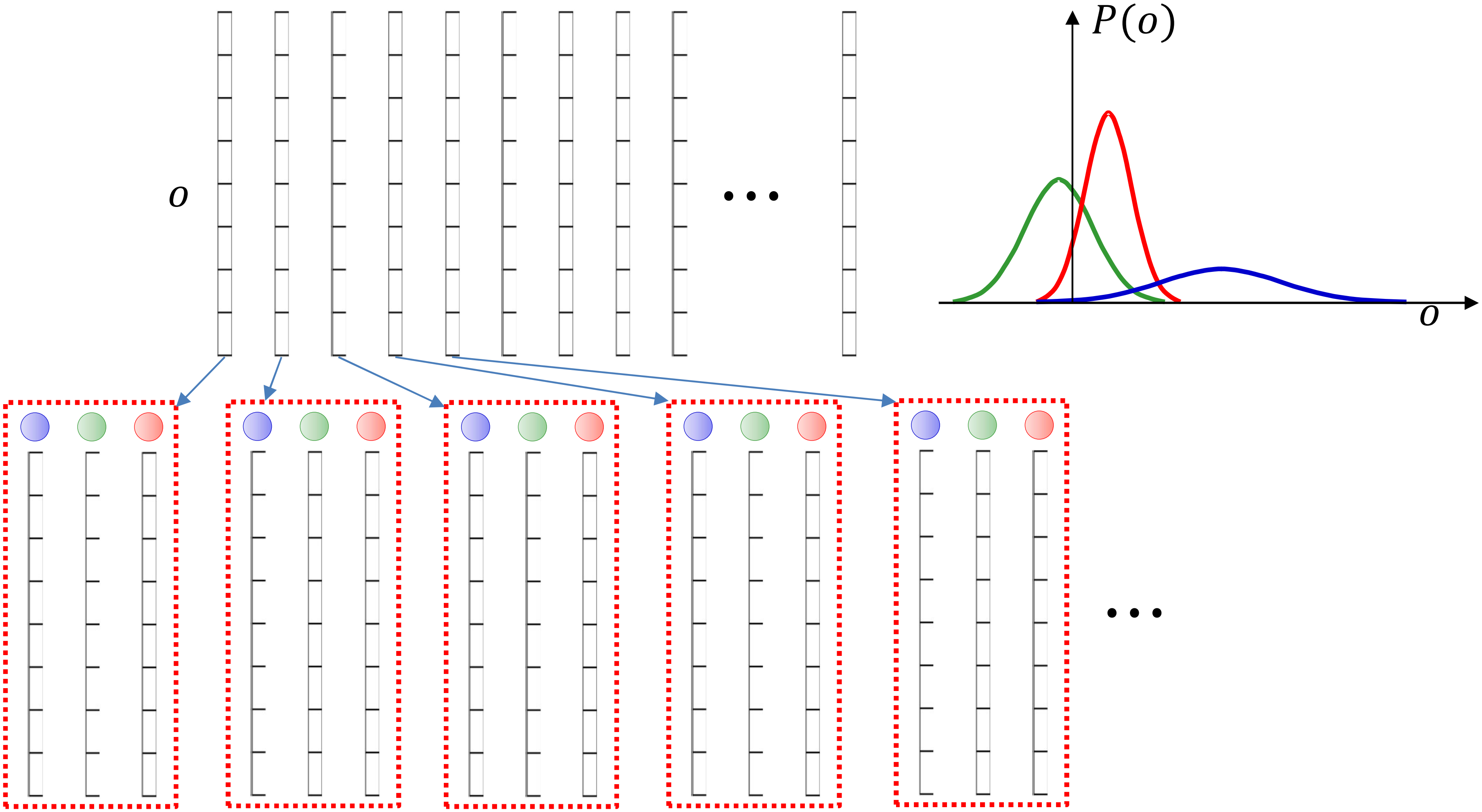
- Complete the data by attributing to *every Gaussian*
  - I.e. make many complete “clones” of the data

From previous estimate of model

- But assign a *proportion* to each completed vector
  - Proportion is $P(k|o)$
    - Which can be computed if we know $P(k)$ and $P(o|k)$

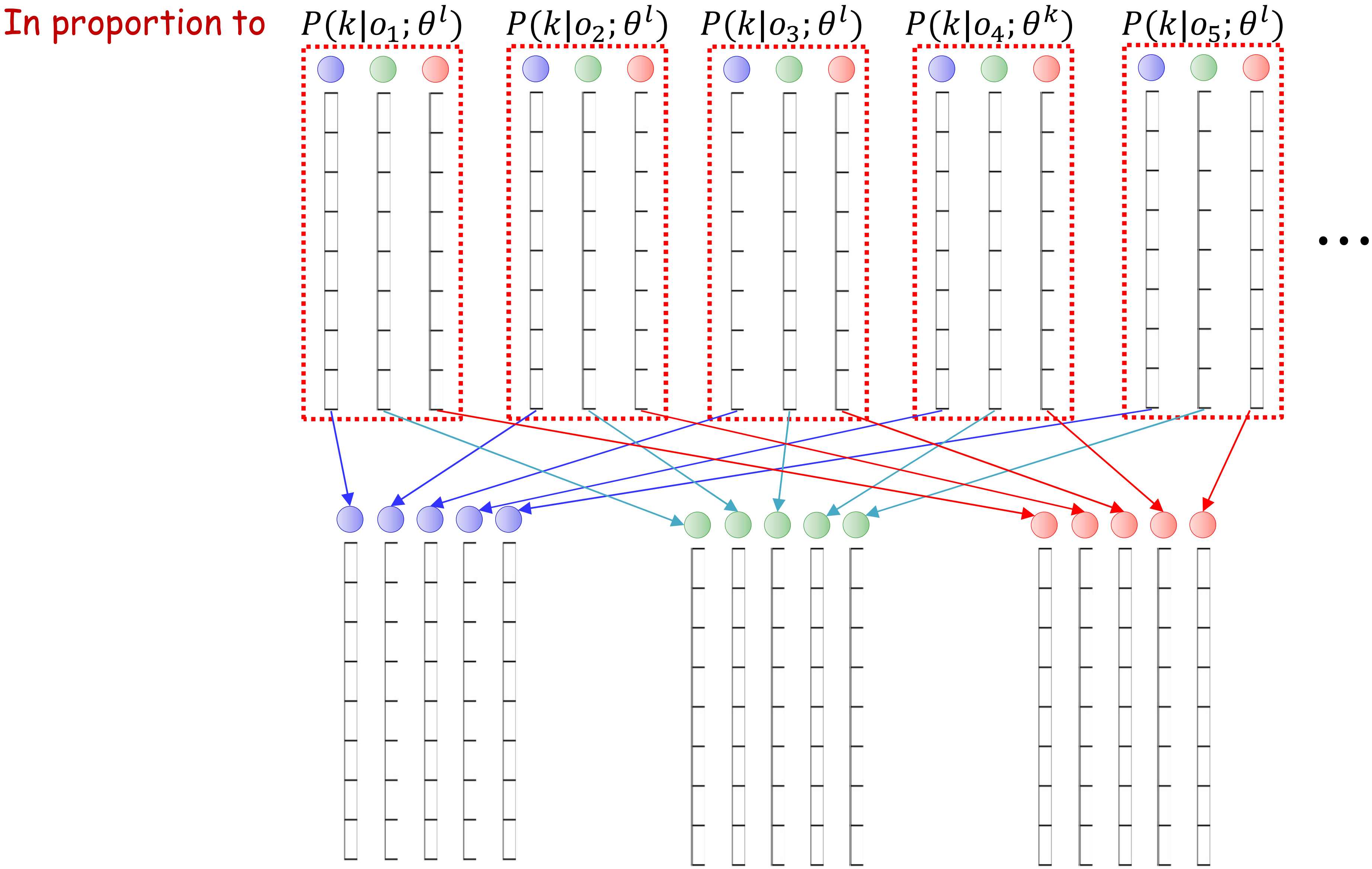
- Then estimate the parameters using the complete data

EM for GMMs



- "Complete" each vector in every possible way:
  - assign each vector to every Gaussian
  - In proportion $P(k|o;\theta^l)$ (computed from current model estimate)
- Compute statistics from "completed" data

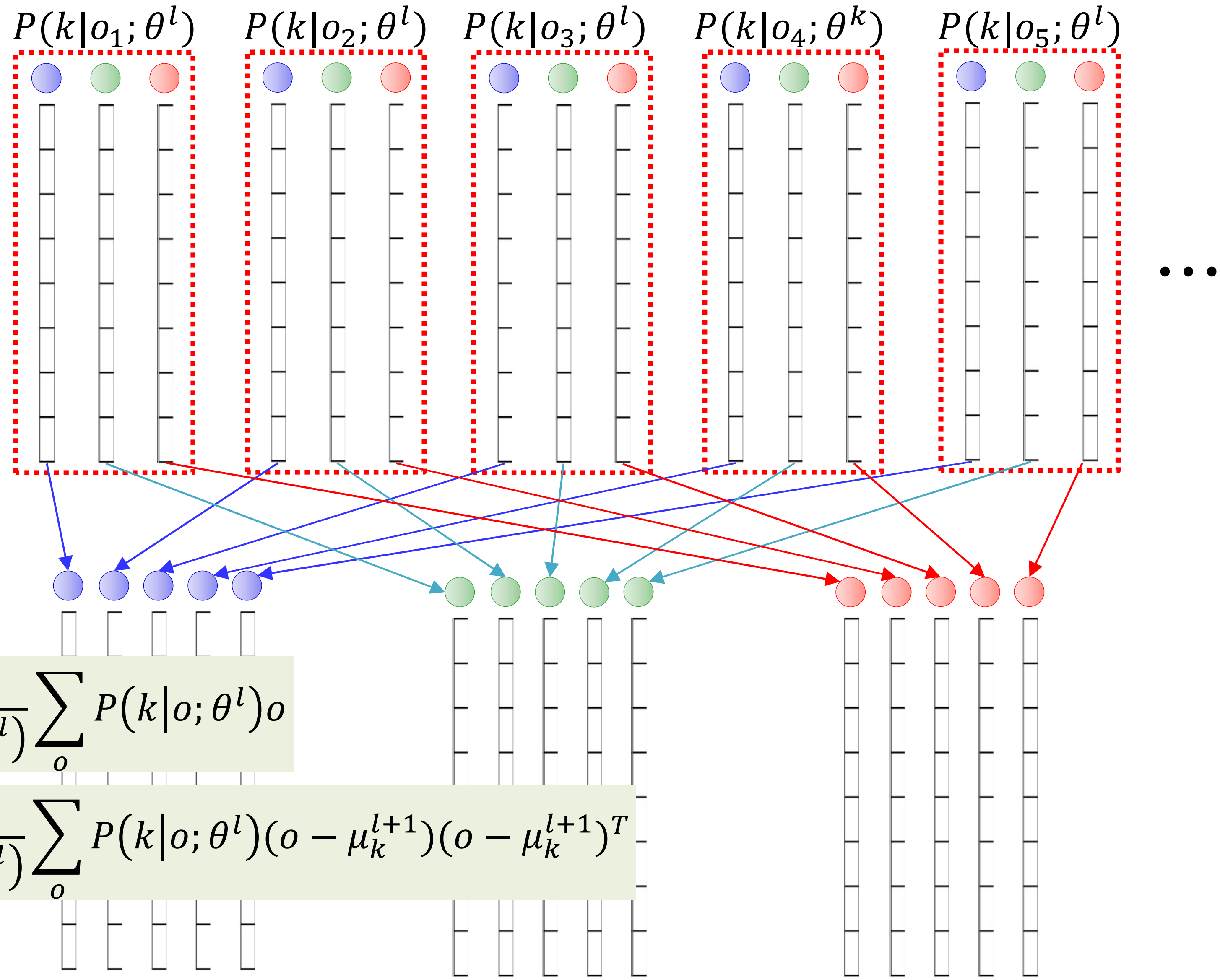
EM for GMMs



- *Now* you can segregate the vectors by Gaussian
  - The number of segregated complete vectors from each observation will be in proportion to $P(k|o;\theta^l)$

EM for GMMs

In proportion to

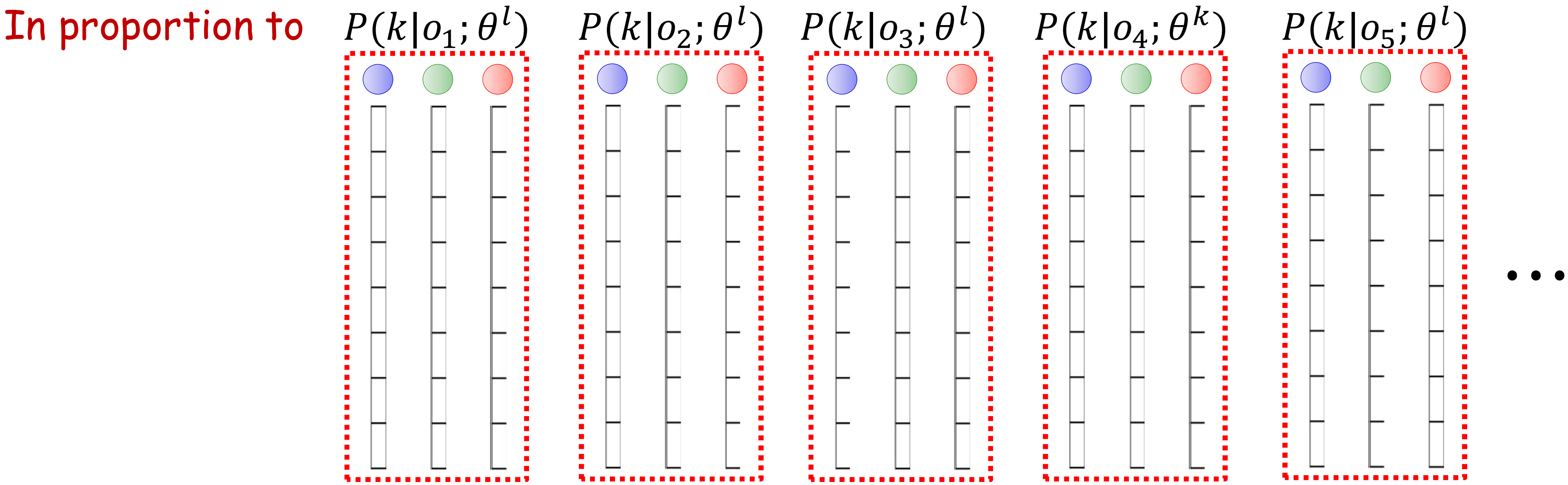


$$\mu_k^{l+1} = \frac{1}{\sum_o P\left(k\middle|o;\theta^l\right)} \sum_o P\left(k\middle|o;\theta^l\right) o$$

$$\Sigma_k^{l+1} = \frac{1}{\sum_o P\left(k\middle|o;\theta^l\right)} \sum_o P\left(k\middle|o;\theta^l\right)(o-\mu_k^{l+1})(o-\mu_k^{l+1})^T$$

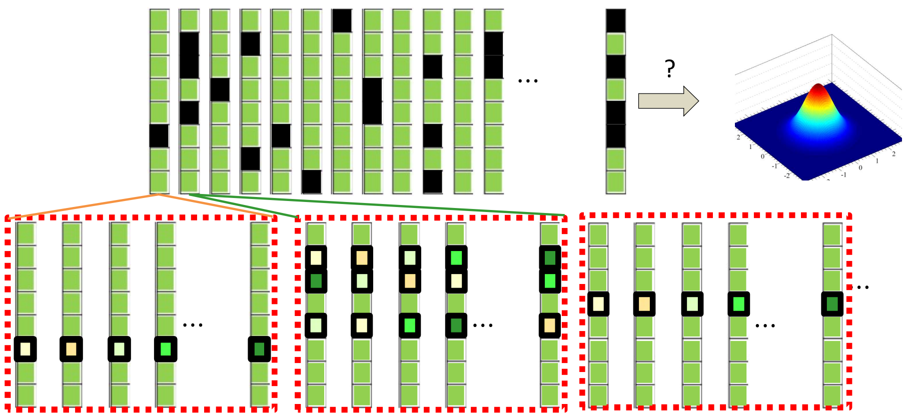
- *Now* you can segregate the vectors by Gaussian
  - The number of segregated complete vectors from each observation will be in proportion to $P(k|o;\theta^l)$

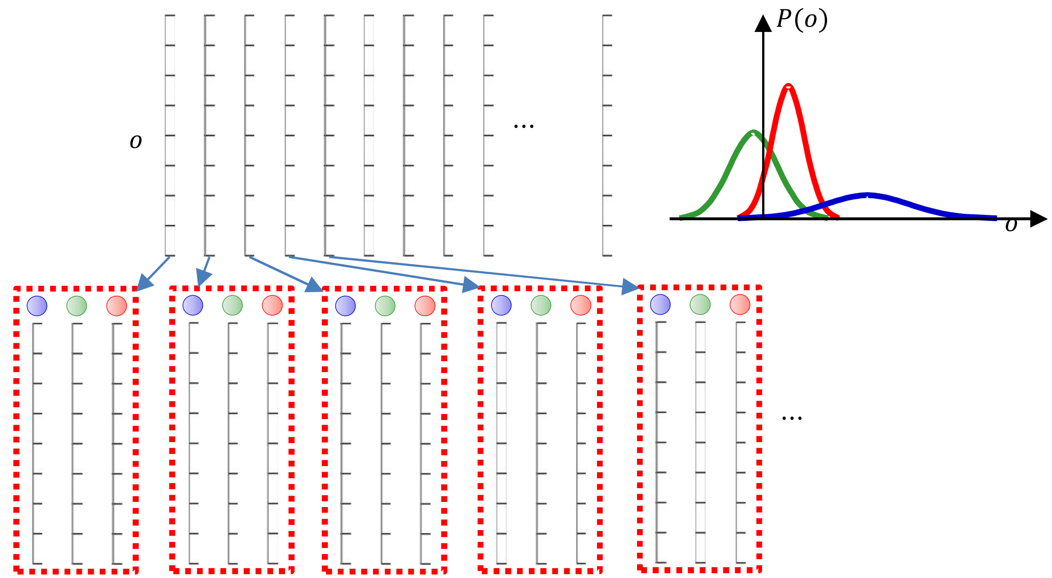
EM for GMMs



- Initialize μ_k^0 and Σ_k^0 for all k
- Iterate (over l):
  - Compute $P(k|o;\theta^l)$ for all o
    - Compute the proportions by which o is assigned to all Gaussians
  - Update:
  - $\mu_k^{l+1} = \frac{1}{\sum_o P\left(k\middle|o;\theta^l\right)} \sum_o P\left(k\middle|o;\theta^l\right) o$
  - $\Sigma_k^{l+1} = \frac{1}{\sum_o P\left(k\middle|o;\theta^l\right)} \sum_o P\left(k\middle|o;\theta^l\right)(o - \mu_k^{l+1})(o - \mu_k^{l+1})^T$

General EM principle

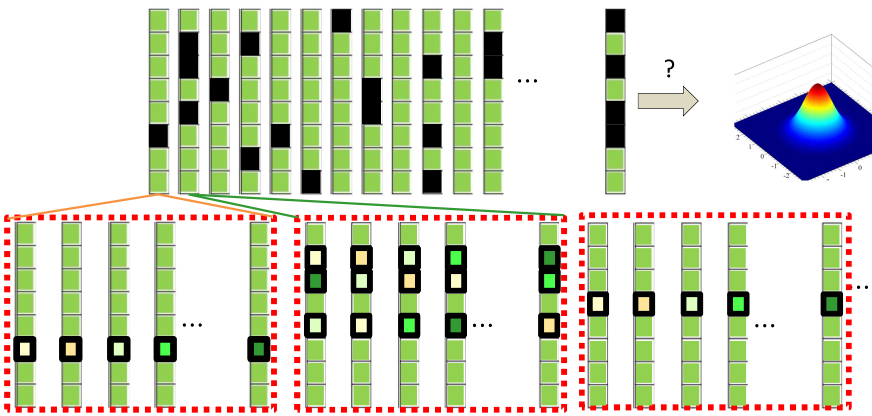


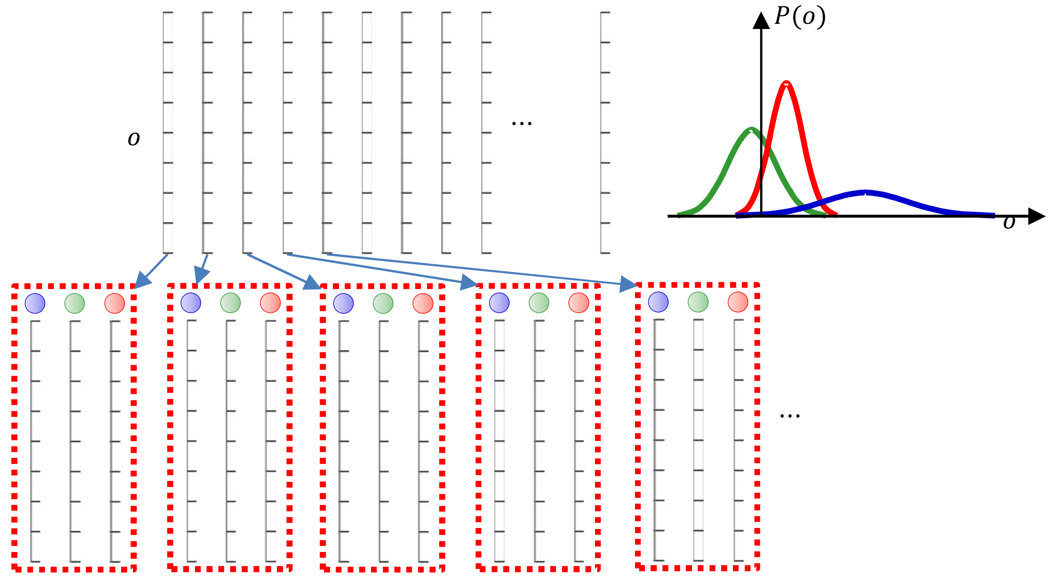


- "Complete" the data by considering *every* possible value for missing data/variables
  - In proportion to their posterior probability, given the observation, $P(m|o)$ (or $P(k|o)$)

- Reestimate parameters from the "completed" data

General EM principle

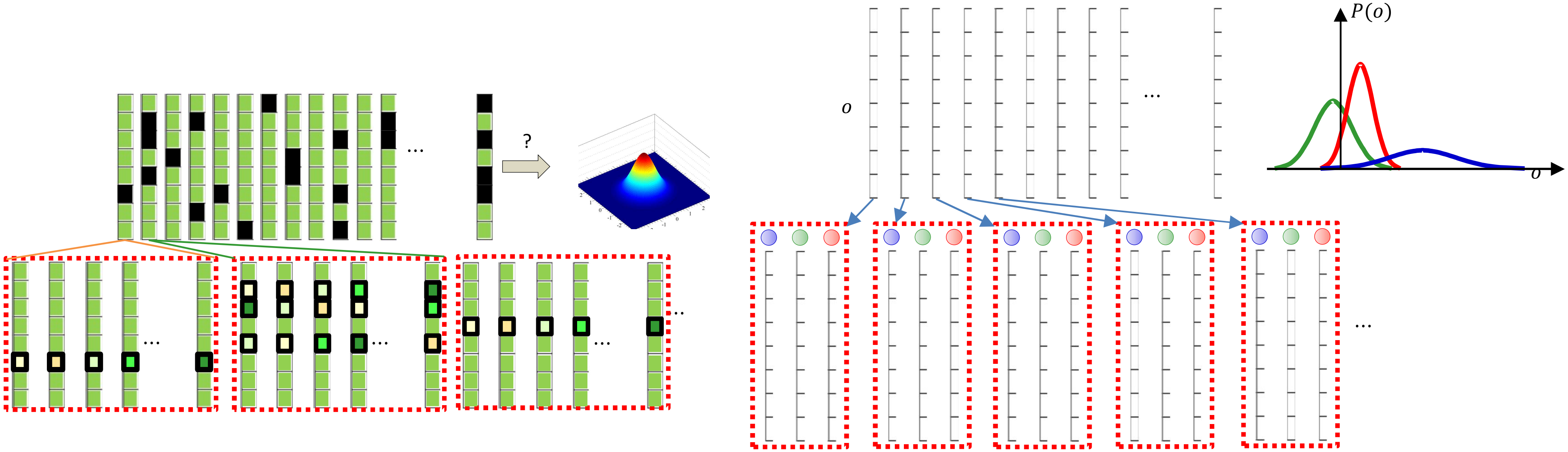




- "Complete" the data by considering *every* possible value for missing data/variables
  - In proportion to their posterior probability, given the observation, $P(m|o)$ (or $P(k|o)$)

- Reestimate parameters from the "completed" data

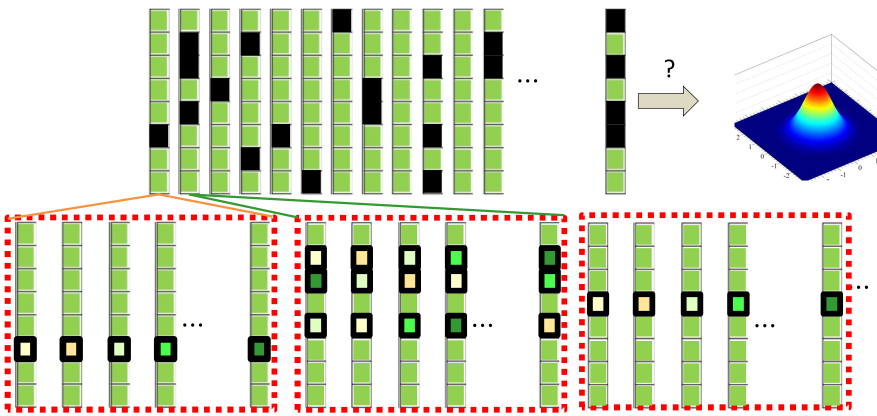
General EM principle

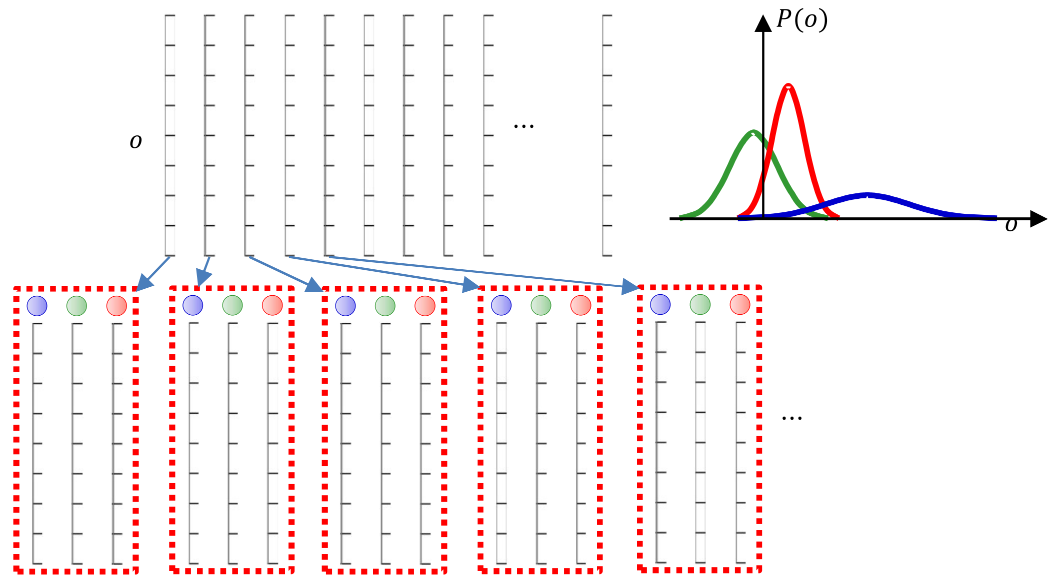


- "Complete" the data by considering *every* possible value for missing data/variables
  - In proportion to their posterior probability, given the observation, $P(m|o)$ (or $P(k|o)$)

Sufficient to "complete" the data by *sampling* missing values from the posterior $P(m|o)$ (or $P(k|o)$) instead

Alternate EM principle

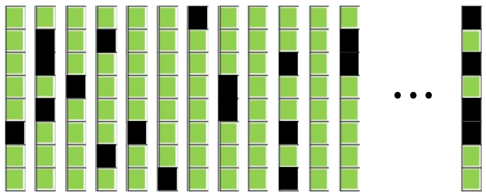


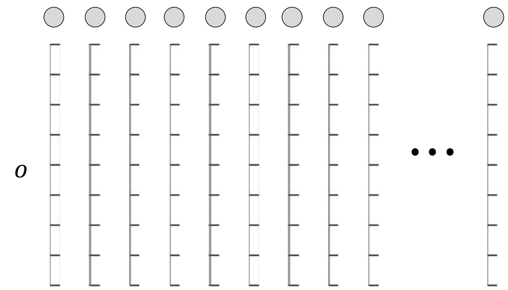


- “Complete” the data by *sampling* possible value for missing data/variables from $P(m|o)$ (or $P(k|o)$)

- Reestimate parameters from the “completed” data

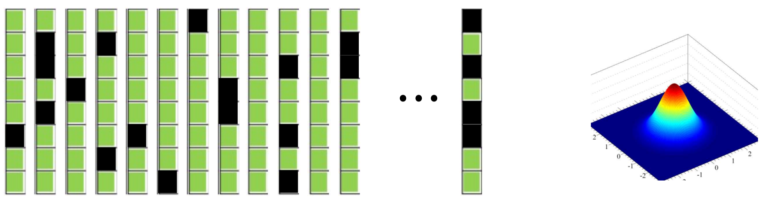
Overall EM principle: Remember this

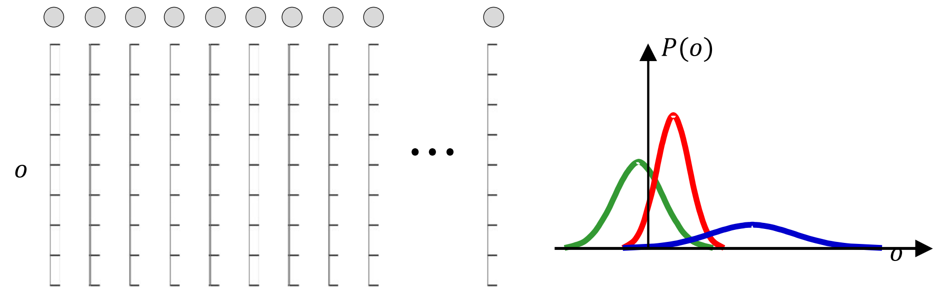




- Initially, some data/information are missing

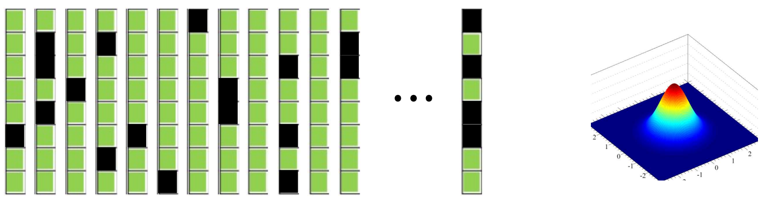
Overall EM principle: Remember this

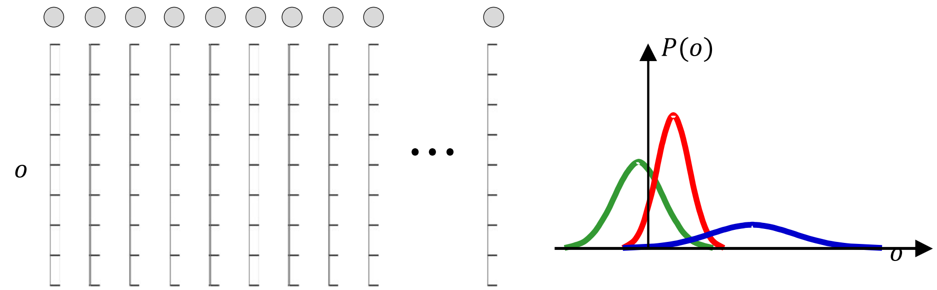




- Initially, some data/information are missing
- ***Initialize model parameters***

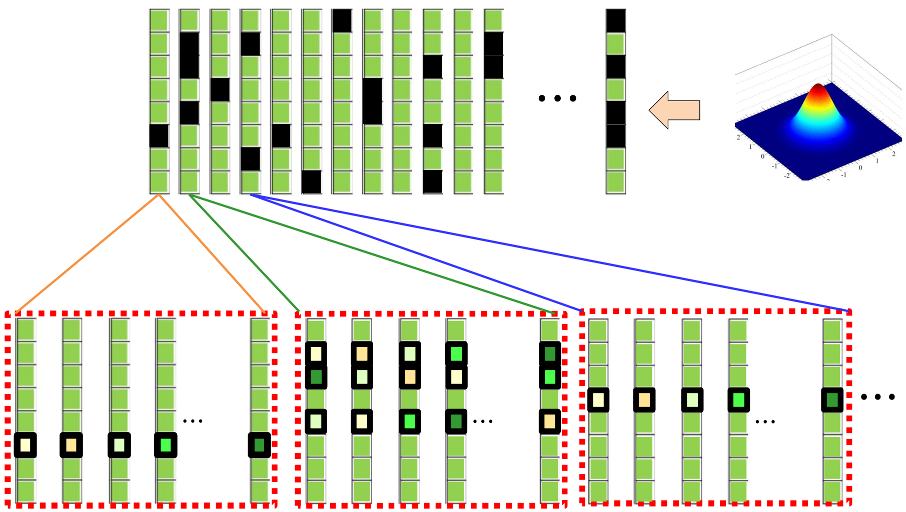
Overall EM principle: Remember this

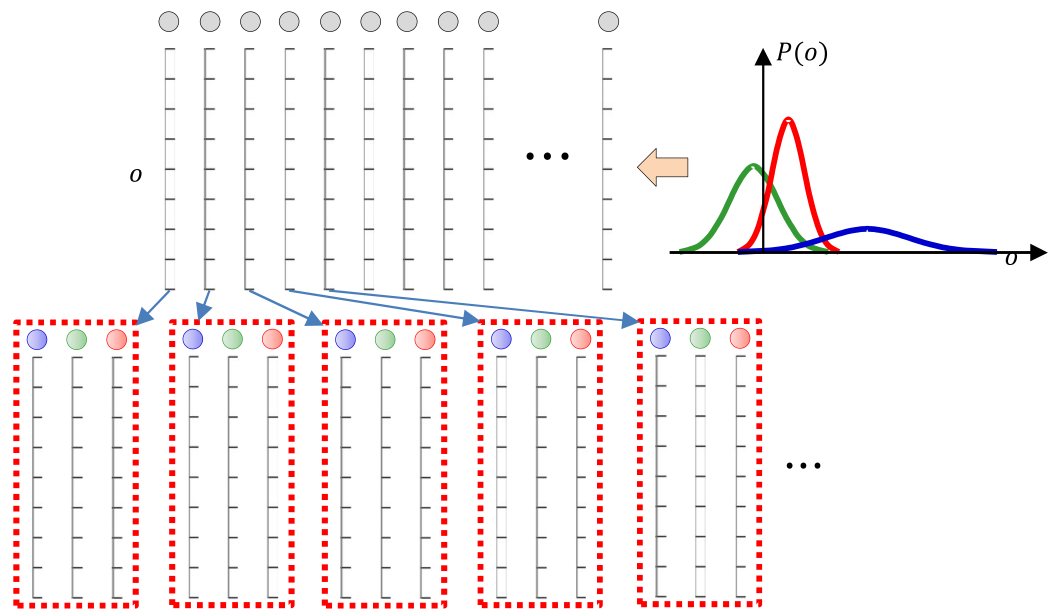




- Initially, some data/information are missing
- Initialize model parameters
- **Iterate:**

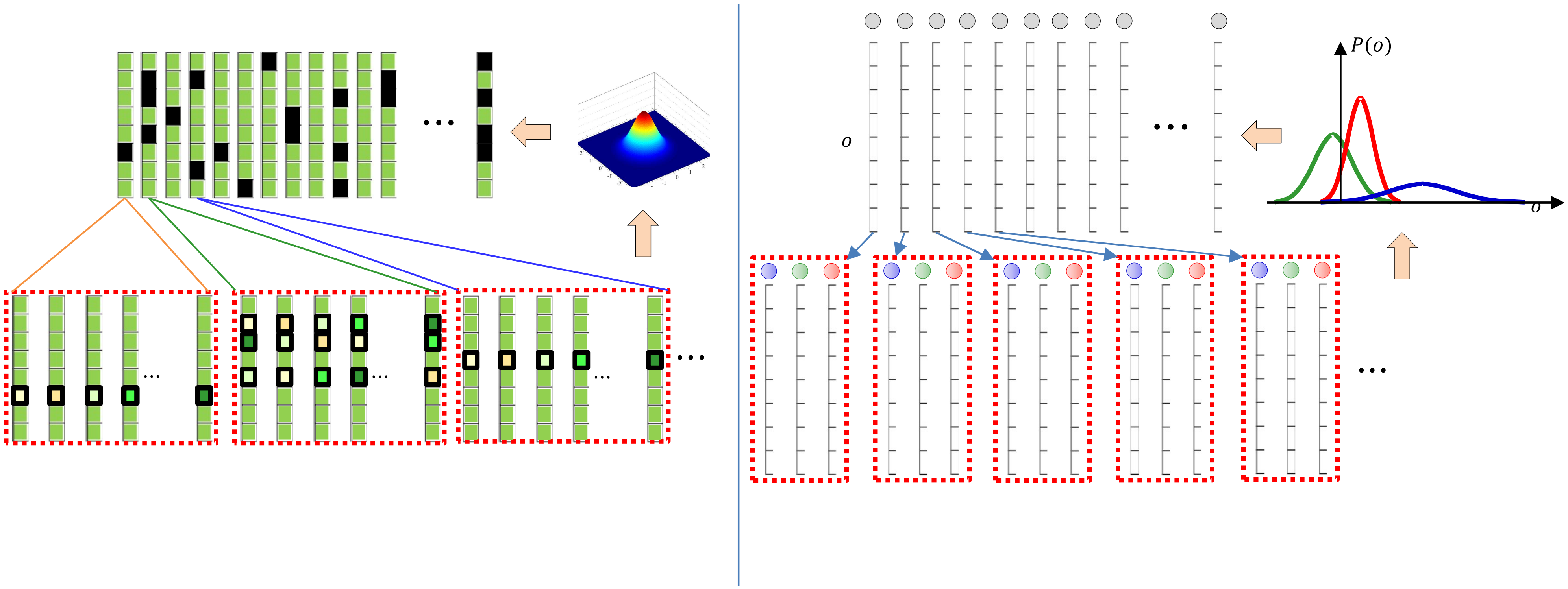
Overall EM principle: Remember this





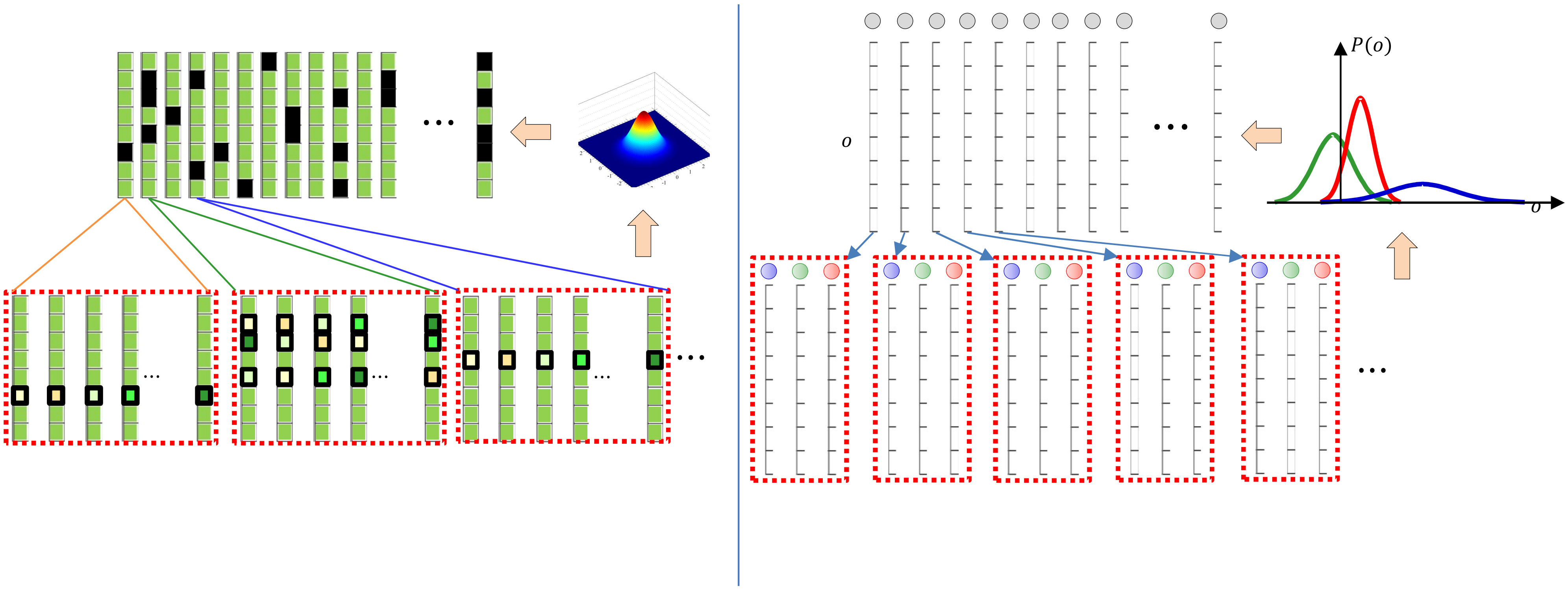
- Initially, some data/information are missing
- Initialize model parameters
- Iterate
  - **Complete the data according to the posterior probabilities $P(m|o)$ computed by the current model**
    - By explicitly considering every possible value, with its posterior-based proportionality
    - Or by sampling the posterior probability distribution $P(m|o)$

Overall EM principle: Remember this



- Initially, some data/information are missing
- Initialize model parameters
- Iterate
  - Complete the data according to the posterior probabilities $P(m|o)$ computed by the current model
    - By explicitly considering every possible value, with its posterior-based proportionality
    - Or by sampling the posterior probability distribution $P(m|o)$
  - **Reestimate the model**

Overall EM principle: Remember this



- Initially, some data/information are missing
- Initialize model parameters
- Iterate
  - Complete the data according to the posterior probabilities $P(m|o)$ computed by the current model
    - By explicitly considering every possible value, with its posterior-based proportionality
    - Or by sampling the posterior probability distribution $P(m|o)$
  - Reestimate the model

Poll 2: tinyurl.com/mlsp23-20231109-2

- EM attempts to "complete" the data, and estimate the model parameters with the now completed data
  - True
  - False

- It completes the data by drawing missing values in proportion to P(m|o), where o are the observed data
  - True
  - False

- Instead of attempting to complete the data with every possible value of the missing variables, we can complete them by sampling P(m|o) and reestimate the parameters with the completed data
  - True
  - False

Poll 2

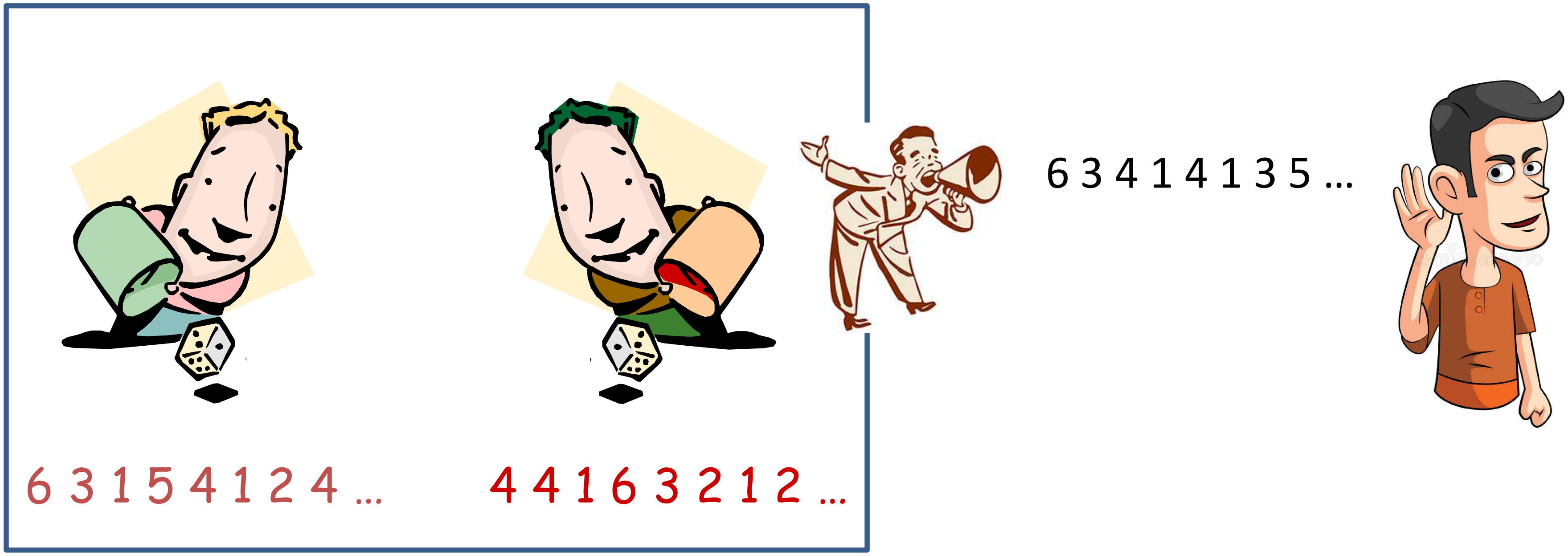
- EM attempts to “complete” the data, and estimate the model parameters with the now completed data
  - **True**
  - False

- It completes the data by drawing missing values in proportion to P(m|o), where o are the observed data
  - **True**
  - False

- Instead of attempting to complete the data with every possible value of the missing variables, we can complete them by sampling P(m|o) and reestimate the parameters with the completed data
  - **True**
  - False

Lets try it out…

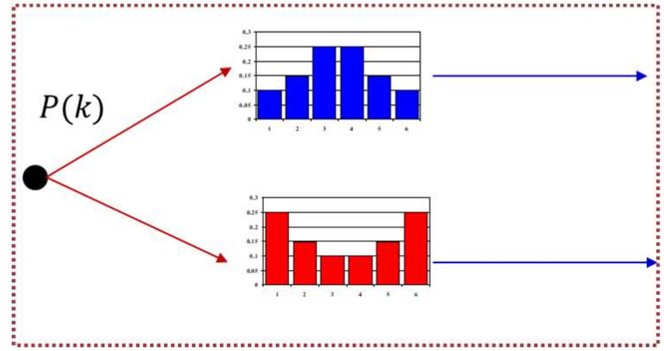
Your friendly neighborhood gamblers



- Two gamblers shoot dice in a closed room
  - The dice are differently loaded for the two of them
- A crazy crier randomly select one of the them and calls out his number
  - But doesn't mention whose number he chose
- You only see the numbers
  - But do not know which of them rolled the number
- **How to determine the probability distributions of the two dice?**

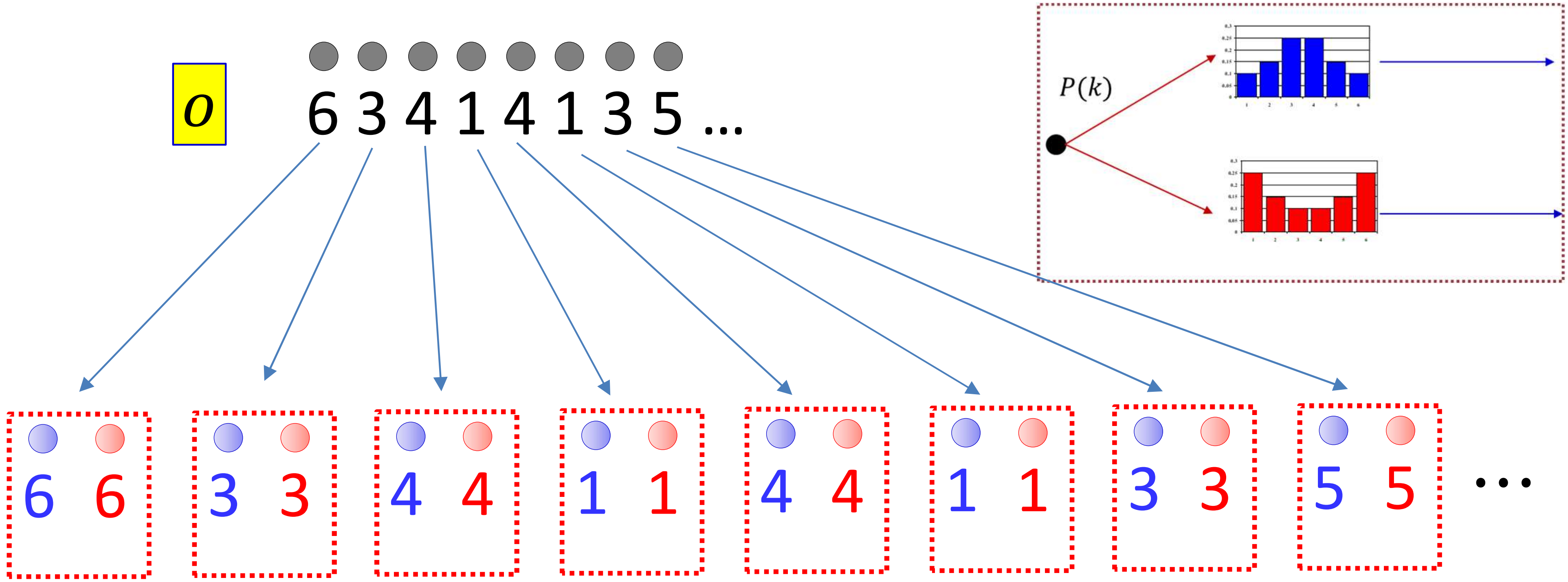
EM for multinomial mixture

o 6 3 4 1 4 1 3 5 …



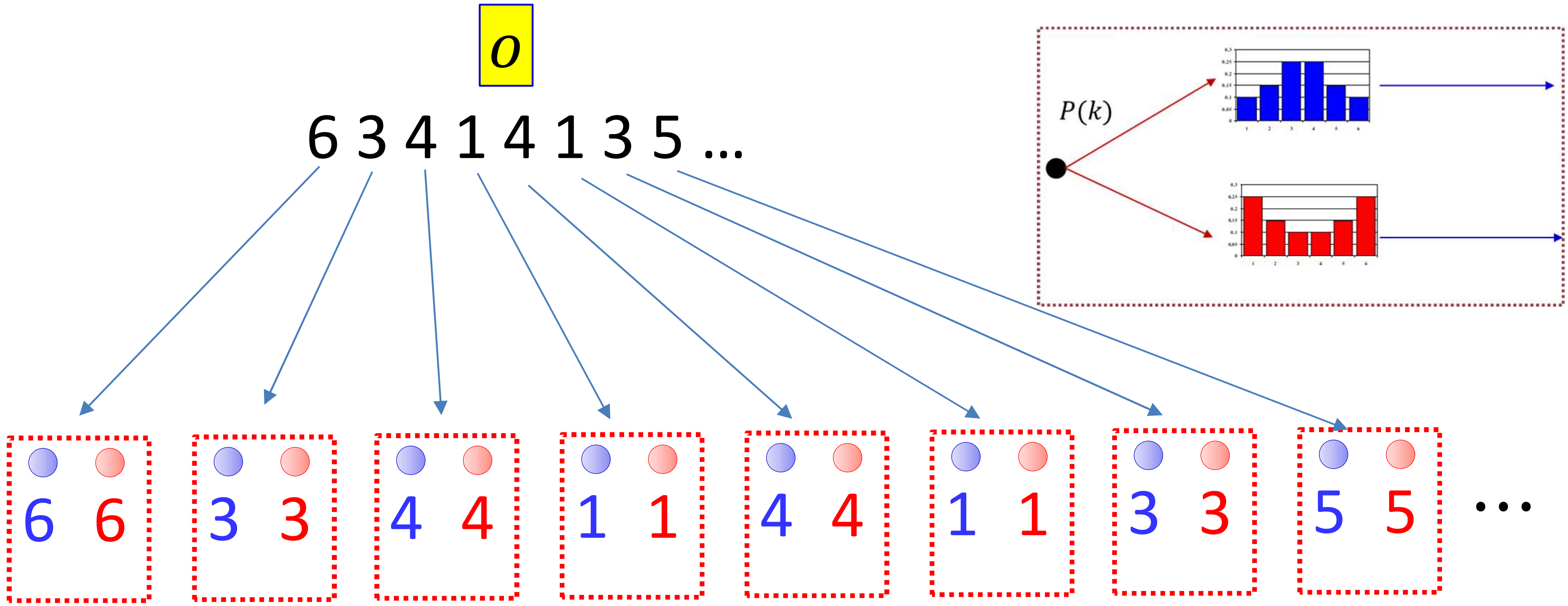
- The “color” of the dice (multinomial) is missing

EM for multinomial mixture



- The "color" of the dice (multinomial) is missing
- "Complete" each observation in every possible way:
  - assign each vector to every multinomial
  - In proportion $P(k|o;\theta^l)$ (computed from current model estimate)
- Compute statistics from "completed" data

EM for multinomial mixture

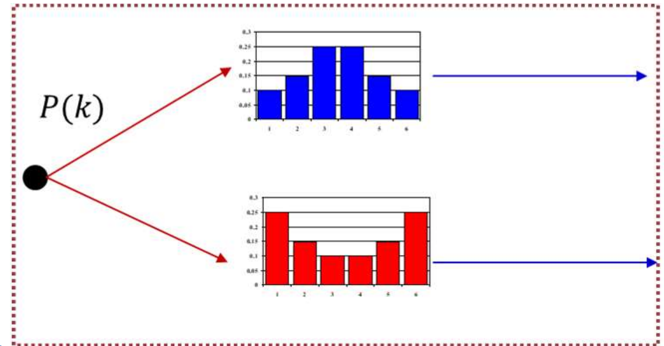


$$P(k|o) = \frac{P(k)P_k(o)}{\sum_{k'} P(k')P_{k'}(o)}$$

EM for multinomial mixture

o

6 3 4 1 4 1 3 5 …



6 6 | 3 3 | 4 4 | 1 1 | 4 4 | 1 1 | 3 3 | 5 5 | …

$$P(k|o) = \frac{P(k)P_k(o)}{\sum_{k'} P(k')P_{k'}(o)}$$

$$\boxed{P_k(o) = \frac{N_o P(k|o)}{\sum_{o'} N_{o'} P(k|o')}}$$

$$\boxed{P(k) = \frac{\sum_o N_o P(k|o)}{\sum_{k'} \sum_o N_o P(k'|o)}}$$

But now for something somewhat different

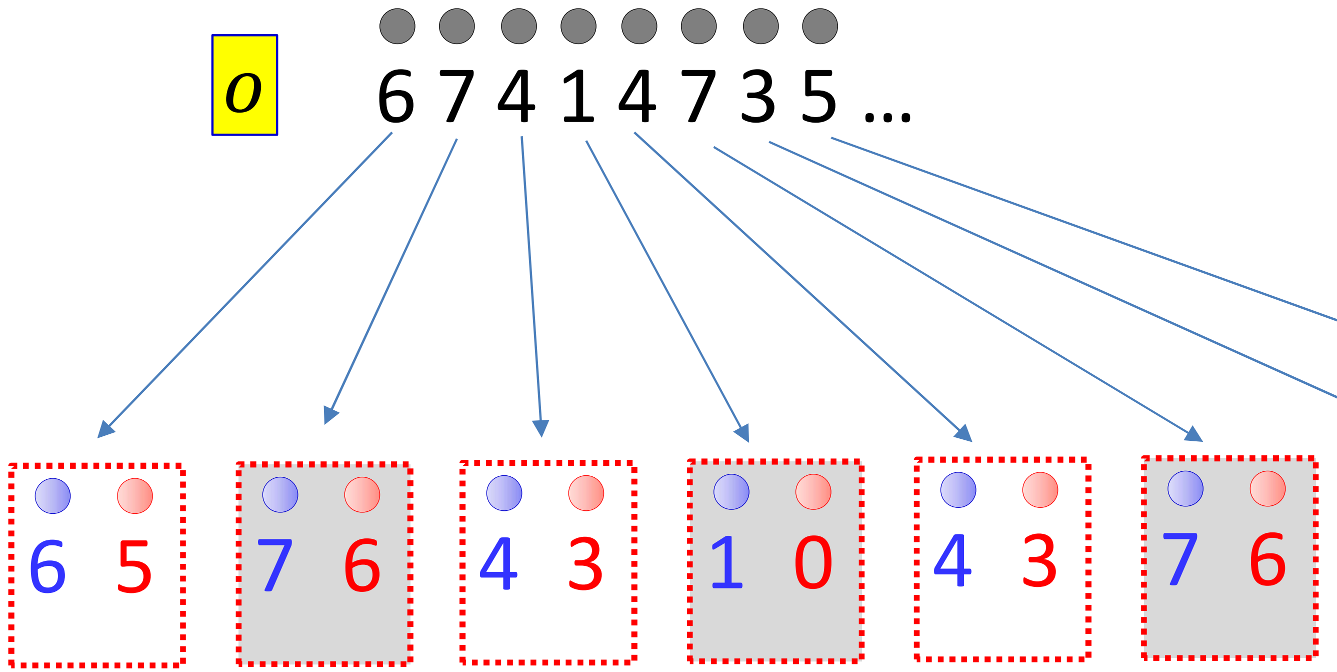
- Caller rolls a dice and flips a coin
- He calls out the number rolled if the coin shows head
- Otherwise he calls the number+1
- Can we estimate p(heads) and p(number) for the dice from a collection of outputs

The dice and the coin

$\boxed{o}$ 6 7 4 1 4 7 3 5 …

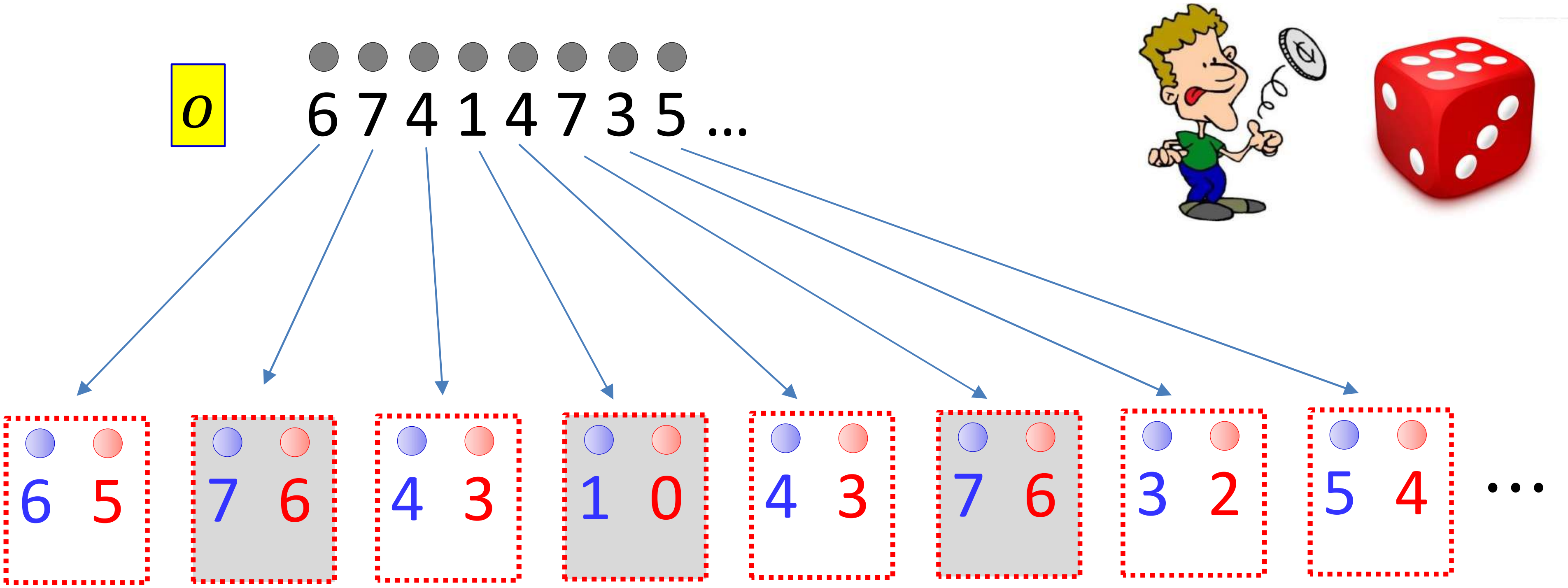
- The “face” of the coin is missing

The dice and the coin



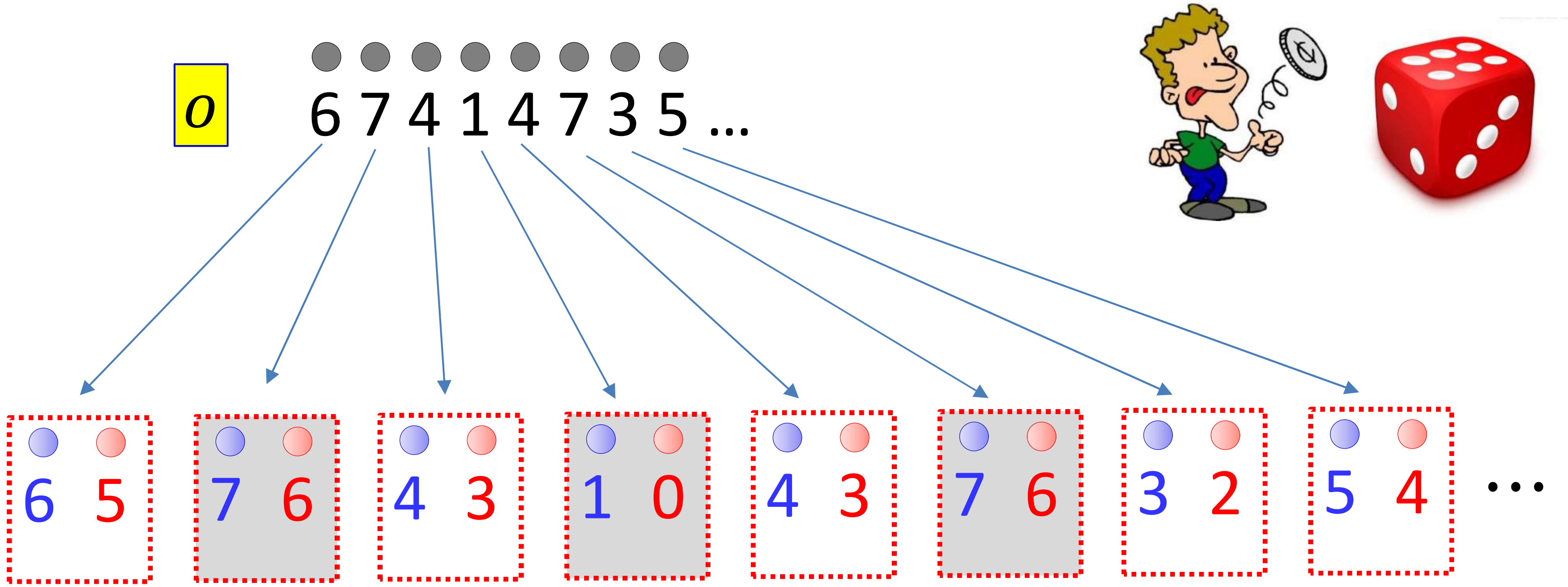
- The "face" of the coin is missing
- "Complete" each observation in every possible way:
  - assign each vector to every face
  - In proportion $P(f|o;\theta^l)$ (computed from current model estimate)
- Compute statistics from "completed" data

The dice and the coin



$$P(heads|o) = \frac{P(o)P(heads)}{P(o)P(heads) + P(o-1)P(tails)}$$

The dice and the coin

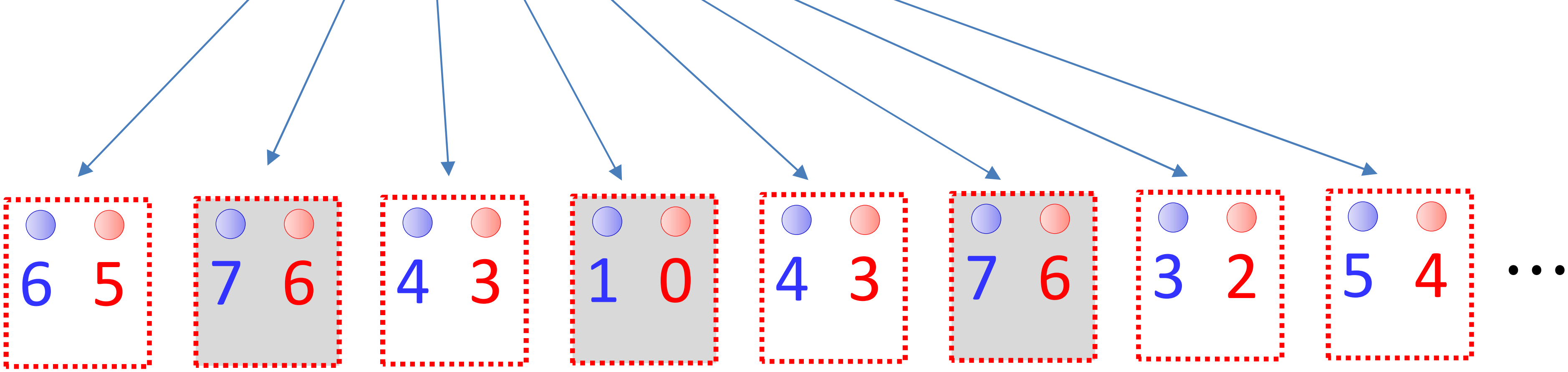


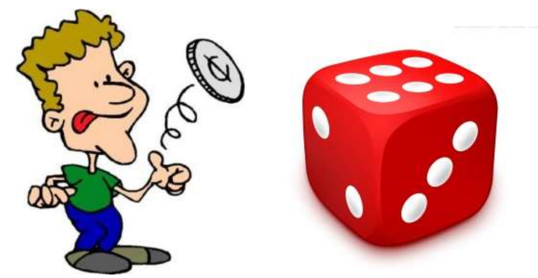
$$P(heads|o) = \frac{P(o)P(heads)}{P(o)P(heads) + P(o-1)P(tails)}$$

$$P(tails|o) = \frac{P(o-1)P(tails)}{P(o)P(heads) + P(o-1)P(tails)}$$

The dice and the coin

o 6 7 4 1 4 7 3 5 ...



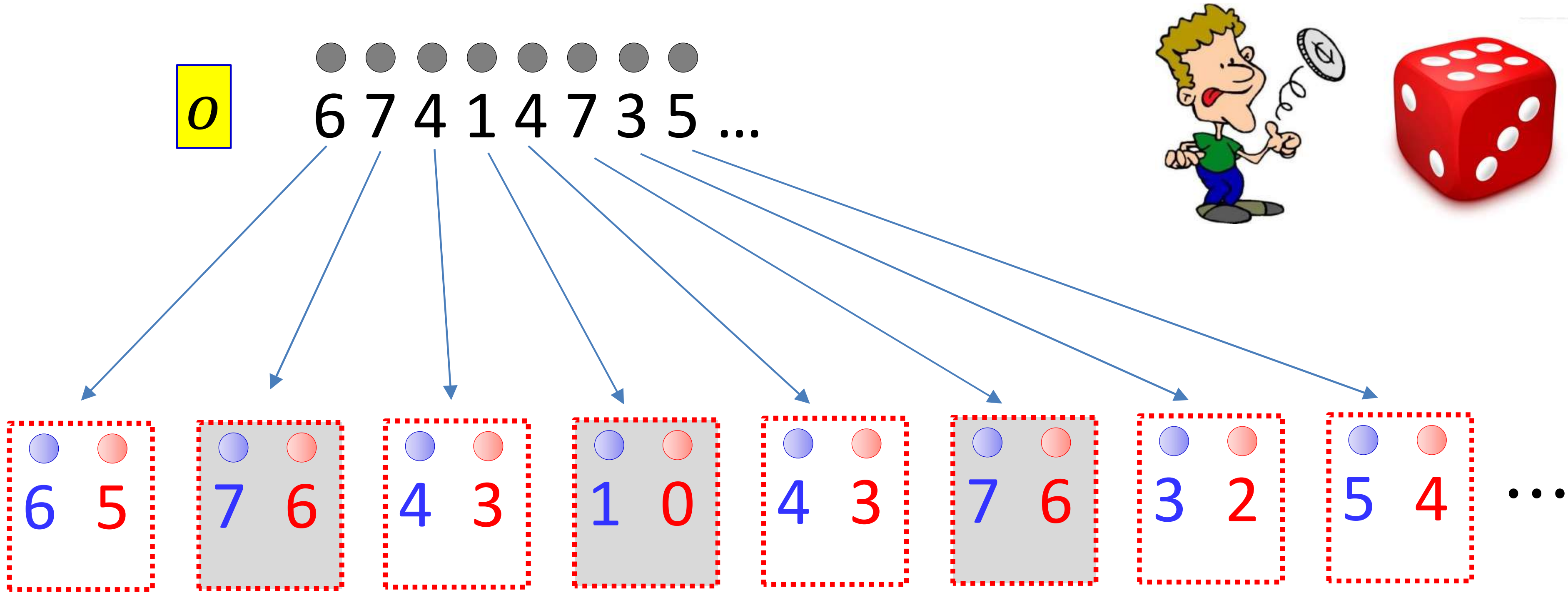


$$P(heads|o) = \frac{P(o)P(heads)}{P(o)P(heads) + P(o-1)P(tails)}$$

$$P(tails|o) = \frac{P(o-1)P(tails)}{P(o)P(heads) + P(o-1)P(tails)}$$

$$P(o) \propto N_o P(heads|o) + N_{o+1} P(tails|o+1)$$

The dice and the coin



$$P(heads|o) = \frac{P(o)P(heads)}{P(o)P(heads) + P(o-1)P(tails)}$$

$$P(tails|o) = \frac{P(o-1)P(tails)}{P(o)P(heads) + P(o-1)P(tails)}$$

$$P(o) \propto N_o P(heads|o) + N_{o+1} P(tails|o+1)$$

$$P(heads) \propto \sum_o N_o P(heads|o)$$

But now for something somewhat different

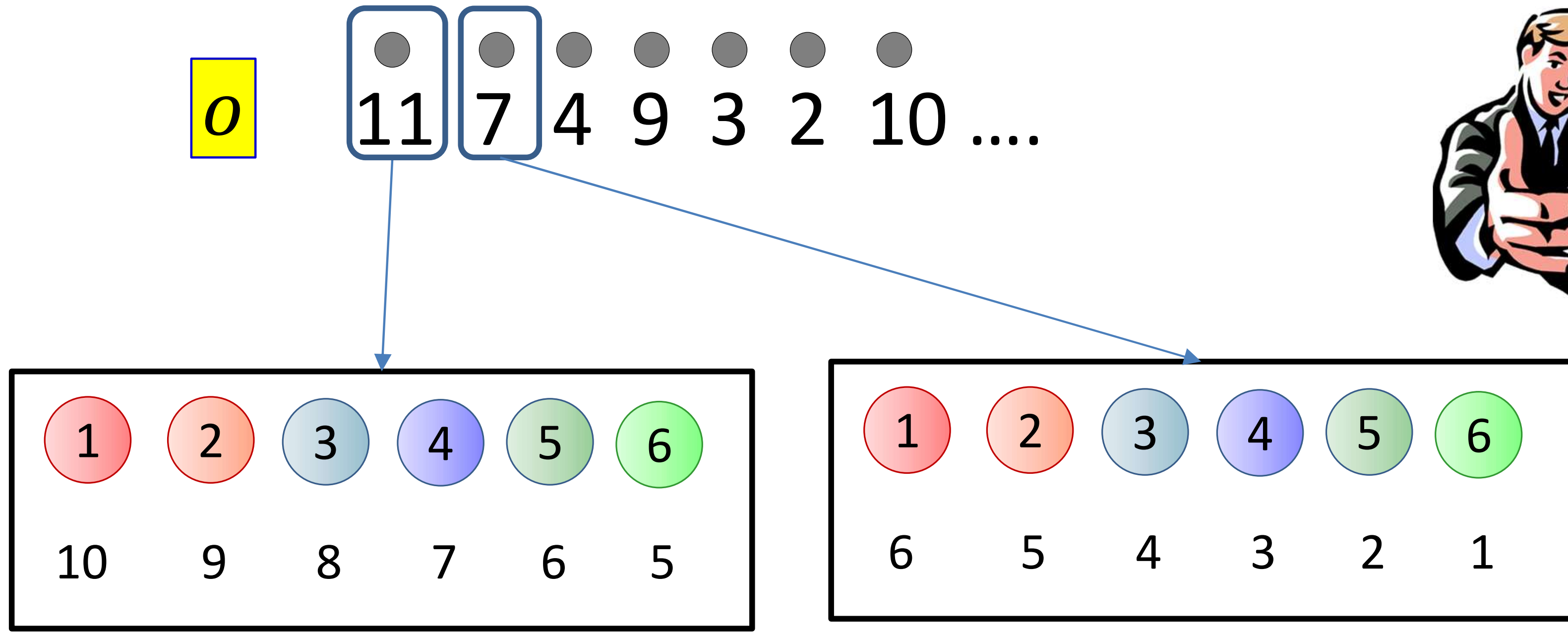
- Roller rolls two dice
- He calls out the sum
- Determine P(dice) from a collection of outputs

The sum of dice

$\boxed{o}$ 11 7 4 9 3 2 10 ....

- The “first” dice info is missing

The sum of dice



- The "first" dice info is missing
- Assign it to every value for the first dice
  - But note what happens to the second

The sum of dice

o 11 7 4 9 3 2 10 ....

1	2	3	4	5	6
10	9	8	7	6	5

1	2	3	4	5	6
6	5	4	3	2	1

$$P(n|o) = P(n, o-n|o) = \frac{P_1(n)P_2(o-n)}{\sum_{m=1}^{6} P_1(m)P_2(o-m)}$$

The sum of dice

o 11 7 4 9 3 2 10 ....

1	2	3	4	5	6
10	9	8	7	6	5

1	2	3	4	5	6
6	5	4	3	2	1

$$P(n, o-n|o) = \frac{P_1(n)P_2(o-n)}{\sum_{m=1}^{6} P_1(m)P_2(o-m)}$$

$$P_1(n) \propto \sum_{o=2}^{12} N_k P(n, o-n|o)$$

Poll 3: tinyurl.com/mlsp23-20231109-3

- The EM algorithm can be applied in any problem with missing data
  - True
  - False

- EM can also be applied when the observed data are drawn from the distribution obtained through the convolution of two component distributions which must be estimated
  - True
  - False

Poll 3

- The EM algorithm can be applied in any problem with missing data
  - **True**
  - False

- EM can also be applied when the observed data are drawn from the distribution obtained through the convolution of two component distributions which must be estimated
  - **True**
  - False

In closing

- Have seen a method for learning the parameters of generative models when some components of the data (or the underlying drawing process) are not observed
- The technique operates by “completing” incomplete data by filling in missing values in proportion to their posterior probabilities
- Coming up : apply this concept to various problems