Machine learning: lecture 17

Tommi S. Jaakkola
MIT AI Lab
tommi@csail.mit.edu

Topics

- Clustering
  - semi-supervised clustering
  - clustering by dynamics: Markov models
- Structured probability models
  - Hidden Markov models

Semi-supervised clustering

- Let's assume we have some additional *relevance* information about the examples to be clustered

$\mathbf{x}_i$	Training example (e.g., a text document)
y	Relevance variable (e.g., a word)
$P(y\|\mathbf{x}_i)$	Relevance information (e.g., word distribution)

where $i = 1, \ldots, n$.

- We wish to cluster documents into larger groups without loosing information about words contained in the documents

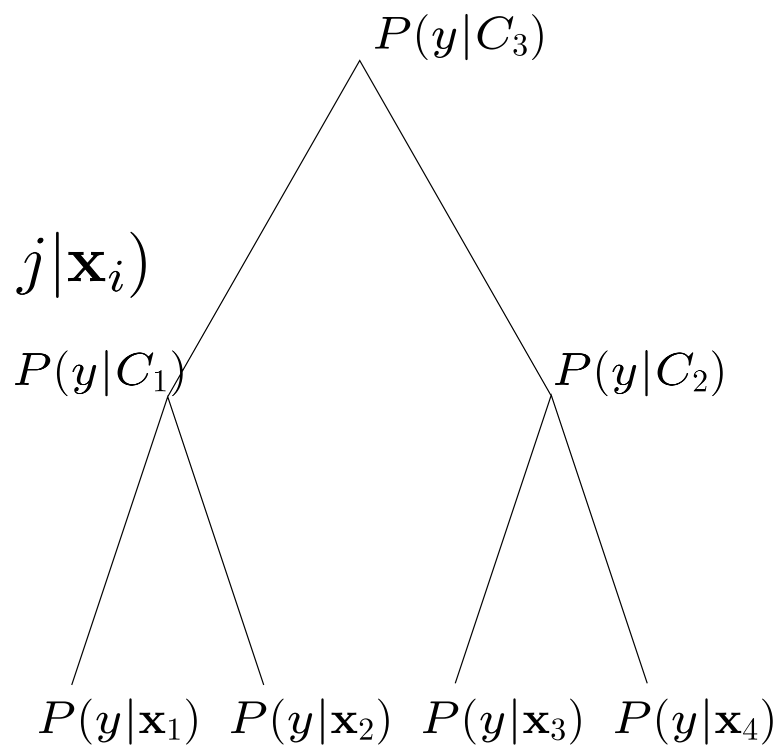
  (documents with similar word frequencies should be merged into a single cluster)

Semi-supervised clustering cont'd

- We cluster the examples $\{\mathbf{x}_i\}$ on the basis of $\{P(y|\mathbf{x}_i)\}$, the predictive distributions
- For any cluster C we define the predictive word distribution based on randomly picking a document in the cluster

$$\hat{P}(y = j|C) = \frac{1}{|C|} \sum_{i \in C} P(y = j|\mathbf{x}_i)$$

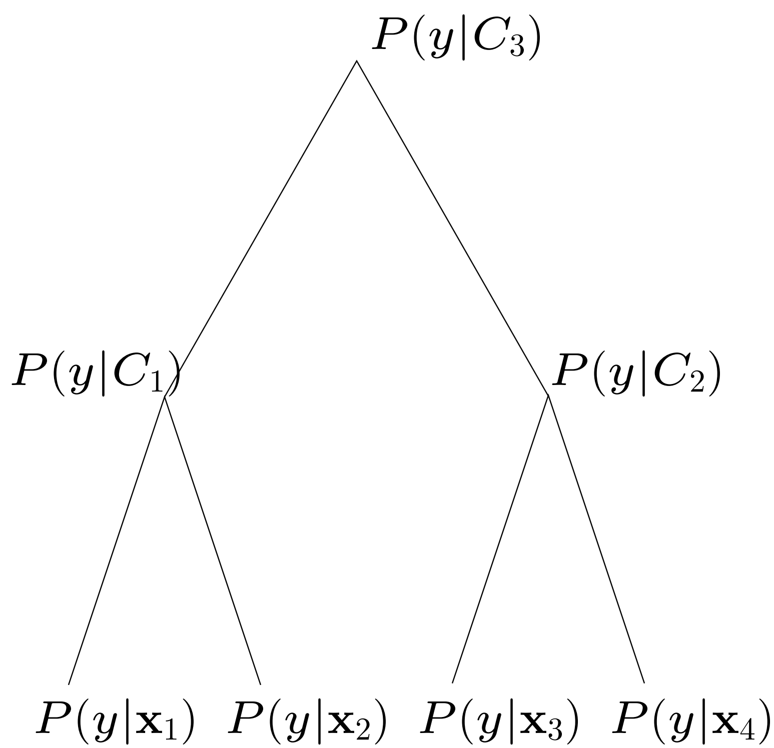
$$\hat{P}(C) = \frac{|C|}{n}$$



Semi-supervised clustering cont'd

- The distance between any two clusters measures how much information we loose about the words if the clusters are merged

$$d(C_l, C_k) \;=\; (\hat{P}(C_l) + \hat{P}(C_k)) \cdot I(\, y \,;\, \text{cluster identity}\,)$$



Semi-supervised clustering cont'd

- The distance between the clusters measures how much information we loose about the words if the clusters are merged

$$d(C_l, C_k) \;=\; (\hat{P}(C_l) + \hat{P}(C_k)) \cdot I(\,y\,;\,\text{cluster identity}\,)$$

where

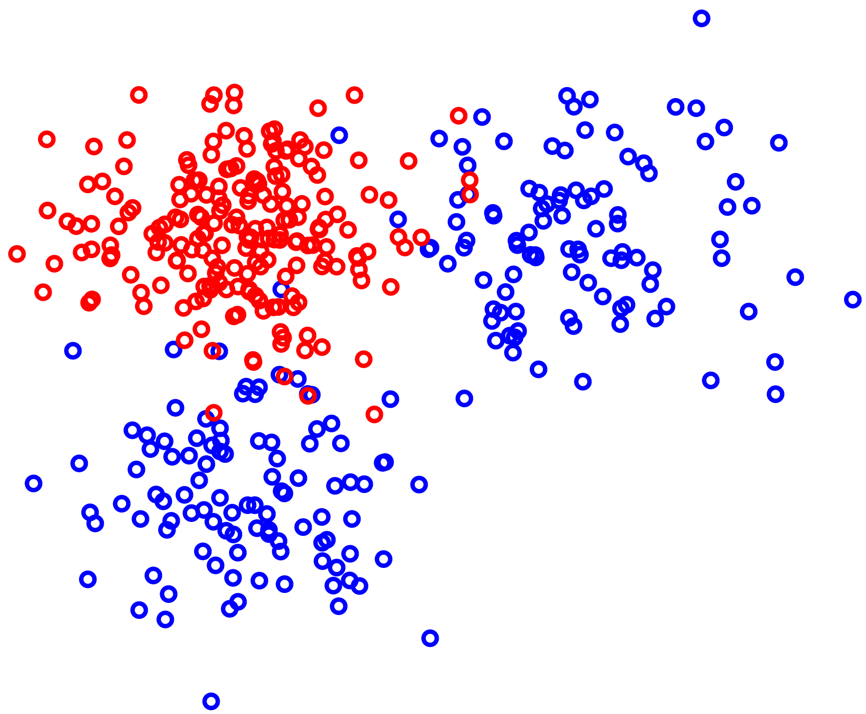
$$I(\,y\,;\,\text{cluster identity}\,) =$$

$$\frac{1}{\hat{P}(C_l) + \hat{P}(C_k)}\Bigg[\hat{P}(C_l)\sum_{j=1}^{m}\hat{P}(y=j|C_l)\log\frac{\hat{P}(y=j|C_l)}{\hat{P}(y=j|C_l\cup C_k)}$$

$$+\hat{P}(C_k)\sum_{j=1}^{m}\hat{P}(y=j|C_k)\log\frac{\hat{P}(y=j|C_k)}{\hat{P}(y=j|C_l\cup C_k)}\Bigg]$$

Semi-supervised clustering: example

- Suppose we have a set of labeled examples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$



- We can take the label as the relevance variable.

$$P(y|\mathbf{x}_i) = 1, \text{ if } y = y_i \text{ and zero otherwise}$$

Topics

- Clustering
  - semi-supervised clustering
  - clustering by dynamics: Markov models
- Structured probability models
  - Hidden Markov models

Clustering by dynamics

- We may wish to cluster time course signals not by direct comparison but in terms of dynamics that governs the signals
  - system behavior monitoring (anomaly detection)
  - biosequencies, processes

    etc.

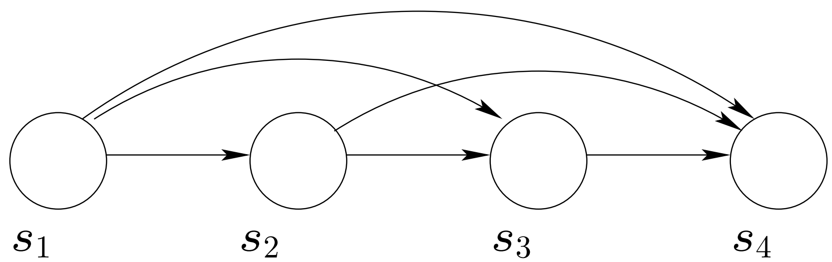
```
1.   0010011001000101000001000011101101010100...
2.   0101111110100110101000001000000101011001...
3.   1101011000000110110010001101111101011101...
4.   1101010111101011110111101101101101000101...
```

- We will use *Markov models* to capture the dynamics and a model selection criterion to induce an appropriate similarity measure

Modeling time course signals

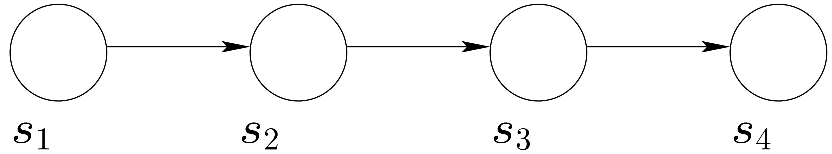
- Full probability model

$$P(s_1, \ldots, s_t) = P(s_1)P(s_2|s_1)P(s_3|s_2, s_1)P(s_4|s_3, s_2, s_1)\cdots$$



- First order Markov model

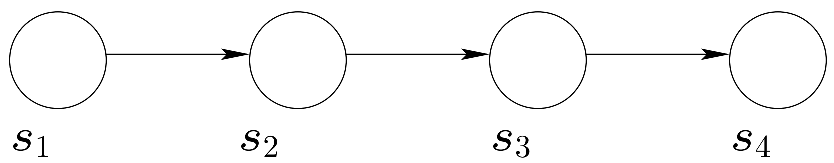
$$P(s_1, \ldots, s_t) = P(s_1)P(s_2|s_1)P(s_3|s_2)P(s_4|s_3)\cdots$$



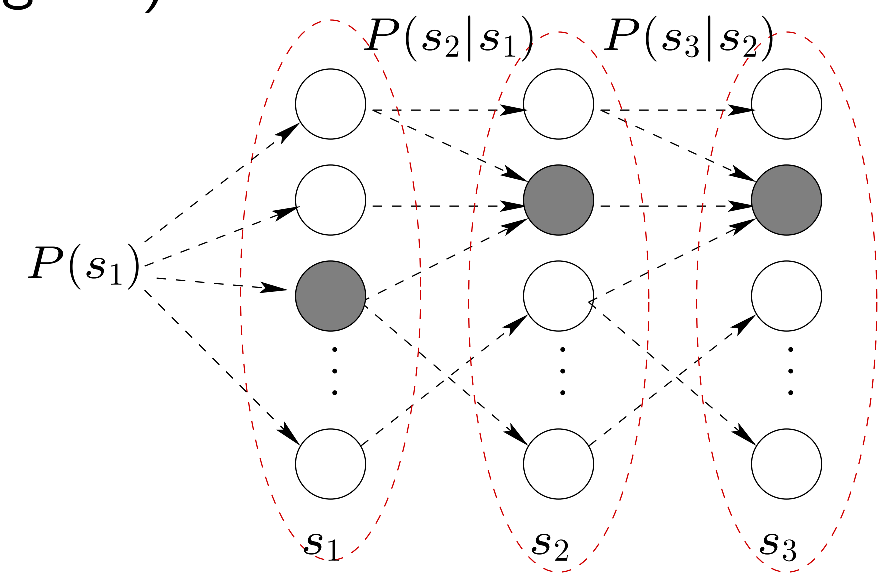
Discrete Markov models

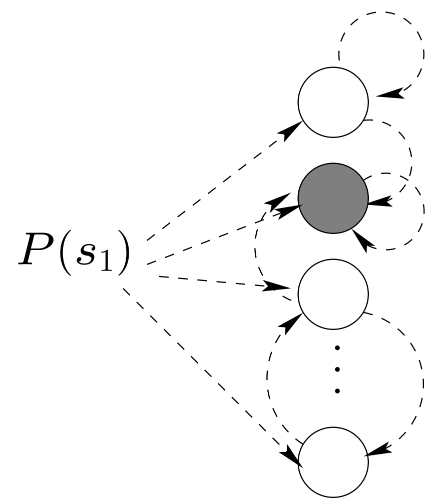
- Representation in terms of variables and dependencies (a graphical model):

$$P(s_1, \ldots, s_t) = P(s_1)P(s_2|s_1)P(s_3|s_2)P(s_4|s_3)\cdots$$



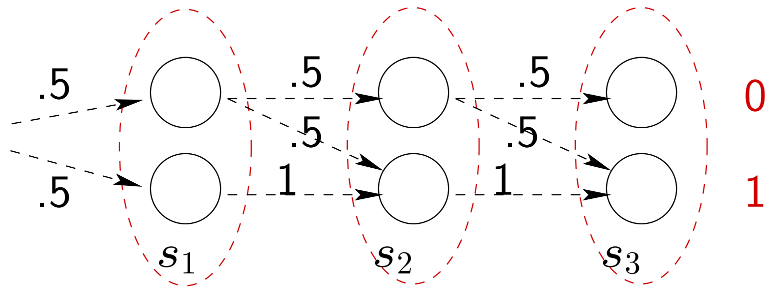
- Representation in terms of state transitions (transition diagram)





Discrete Markov models: properties

- The values of each s_t are known as *states*



$P(\cdot)$: 0.5
0.5

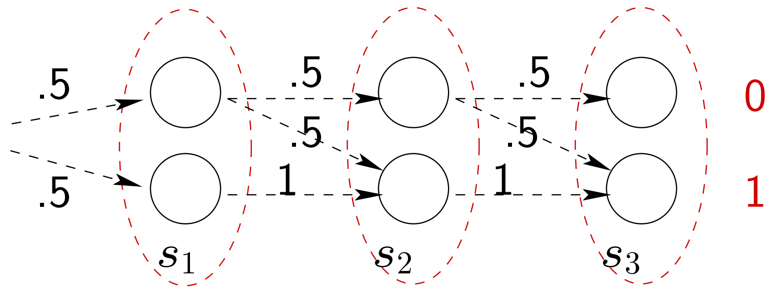
$P_1(\cdot|\cdot)$:

0.5	0.5
0.0	1.0

- When successive state transitions are governed by the same (one-step) transition probability matrix $P_1(s_t|s_{t-1})$, the Markov model is *homogeneous*

Discrete Markov models: properties

- The values of each s_t are known as *states*



$P(\cdot)$: $\begin{matrix} 0.5 \\ 0.5 \end{matrix}$

$P_1(\cdot|\cdot)$: $\begin{matrix} 0.5 & 0.5 \\ 0.0 & 1.0 \end{matrix}$

- When successive state transitions are governed by the same (one-step) transition probability matrix $P_1(s_t|s_{t-1})$, the Markov model is *homogeneous*

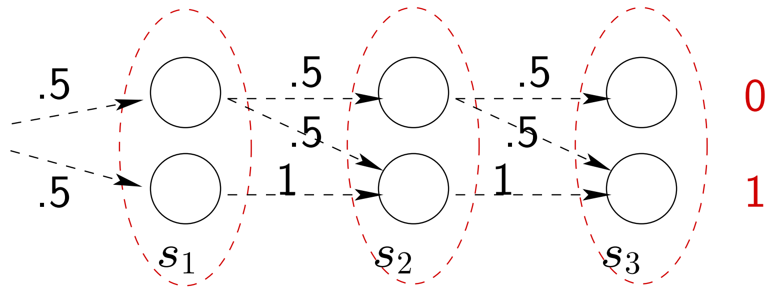
- Example: a language model

$$\text{This} \rightarrow \text{is} \rightarrow \text{a} \rightarrow \text{boring} \rightarrow \ldots$$

Is a homogeneous Markov model appropriate in this case?

Discrete Markov models: properties

- The values of each s_t are known as *states*



$P(\cdot)$:

0.5	
0.5	

$P_1(\cdot|\cdot)$:

0.5	0.5
0.0	1.0

- When successive state transitions are governed by the same (one-step) transition probability matrix $P_1(s_t|s_{t-1})$, the Markov model is *homogeneous*

- If after k transitions we can get from any state i to any other state j, the markov chain is *ergodic*.

  In other words, the k-step transition probabilities must satisfy $P_k(s_{k+1} = j|s_1 = i) > 0$ for all i, j.

Discrete Markov models: ML estimation

```
1.   00100110010001010000001000011101101010100...
2.   010111111010011010100000010000000101011001...
```

- ML estimates of the parameters (initial state and transition probabilities) are based on simple counts:

$$
\begin{aligned}
\hat{n}(i) &= \#\text{ of times } s_1 = i \\
\hat{n}(i,j) &= \#\text{ number of times } i \rightarrow j \\
\hat{P}(i) &= \frac{\hat{n}(i)}{\sum_{i'} \hat{n}(i')} \\
\hat{P}_1(j|i) &= \frac{\hat{n}(i,j)}{\sum_{j'} \hat{n}(i,j')}
\end{aligned}
$$

Simple clustering example cont'd

- Four binary sequences of length 50:

```
1.   001001100100010100000100000111011010100...
2.   010111111010011010100000100000010101 1001...
3.   110101100000011011001000110111110101 1101...
4.   110101011110101111011110110110101000101...
```

- We still need to derive the clustering metric based on the transitition probabilities (dynamics)
- We can turn the clustering problem into a model selection problem: which sequencies should be modeled with the same transition probabilities?

Cluster criterion

- To determine whether two arbitrary sequences

$$S^{(1)} = \{s_1^{(1)}, \ldots, s_{n_1}^{(1)}\} \text{ and } S^{(2)} = \{s_1^{(2)}, \ldots, s_{n_2}^{(2)}\}$$

should be in the same cluster, we compare (approximate) description lengths of either encoding the sequencies separately or jointly

$$\mathrm{DL}^{(1)} + \mathrm{DL}^{(2)} \gtrless \mathrm{DL}^{(1+2)}$$

where $\mathrm{DL}^{(1+2)}$ uses the same Markov model for both sequencies whereas $\mathrm{DL}^{(1)}$ and $\mathrm{DL}^{(2)}$ use models specific to each sequence.

Cluster criterion cont'd

- Approximate description lengths:

$$\begin{aligned}
\mathsf{DL}^{(1)} + \mathsf{DL}^{(2)} &= -\log P(S^{(1)}|\hat{\theta}_1) + \frac{d}{2}\log(n_1) \\
&\quad -\log P(S^{(2)}|\hat{\theta}_2) + \frac{d}{2}\log(n_2) \\
\mathsf{DL}^{(1+2)} &= -\log P(S^{(1)}|\hat{\theta}) - \log P(S^{(2)}|\hat{\theta}) \\
&\quad + \frac{d}{2}\log(n_1 + n_2)
\end{aligned}$$

  where the maximum likelihood parameter estimates $\hat{\theta}_1$, $\hat{\theta}_2$, and $\hat{\theta}$ include the initial state distribution and the transition probabilities; $d = 3$ for binary sequences.

- We are essentially testing here whether the two sequencies have the same first order Markov dynamics

Simple example cont'd

- Four binary sequences of length 50:

```
1.   0010011001000101000001000011101101010100...
2.   0101111110100110101000001000000101011001...
3.   1101011000000110110010001101111101011101...
4.   1101010101111011110111101101101101000101...
```

Evaluations:

$$\begin{aligned} \mathrm{DL}^{(1)} + \mathrm{DL}^{(2)} - \mathrm{DL}^{(1+2)} &= 6.6\,\text{bits} \\ \mathrm{DL}^{(1+2)} + \mathrm{DL}^{(3+4)} - \mathrm{DL}^{(1+2+3+4)} &= -0.9\,\text{bits} \end{aligned}$$

Agglomerative hierarchical clustering with Euclidean distance would give $(((2,3),4),1)$

Topics

- Clustering
  - semi-supervised clustering
  - clustering by dynamics: Markov models
- Structured probability models
  - Hidden Markov models

Beyond Markov models

- How can we model

  1. `0101010101010101010101010101010101010101...`

Beyond Markov models

- How can we model

  1. `0101010101010101010101010101010101010101...`
  2. `0010010010010010010010010010010010010010...`

Beyond Markov models

- How can we model

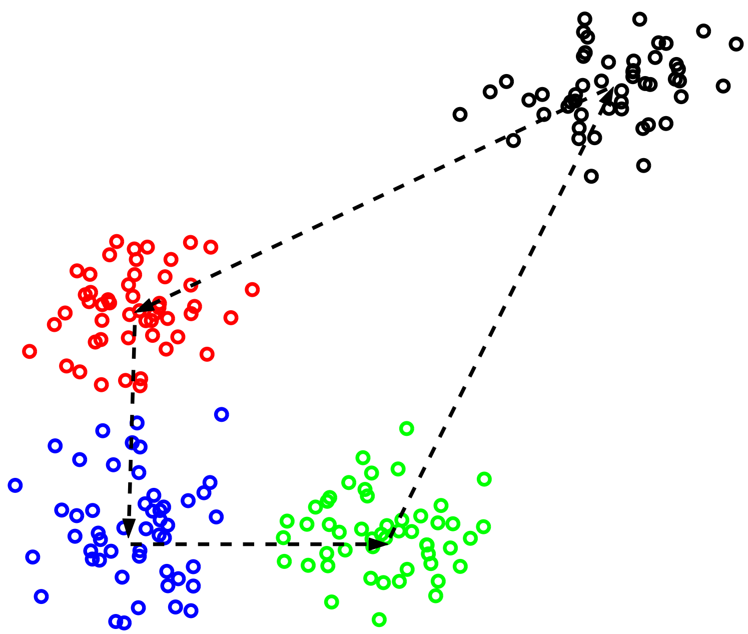
  1. `0101010101010101010101010101010101010101...`
  2. `0010010010010010010010010010010010010010...`
  3. `0100100010000101001000100001010010001000...`

Beyond Markov models

- How can we model

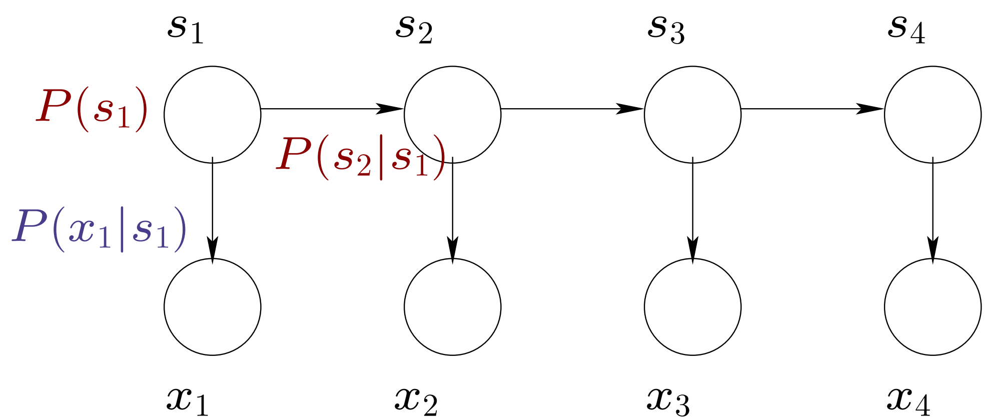
  1. 01010101010101010101010101010101010101010101...
  2. 00100100100100100100100100100100100100100010...
  3. 01001000100001010010001000010100100010000...

- What about



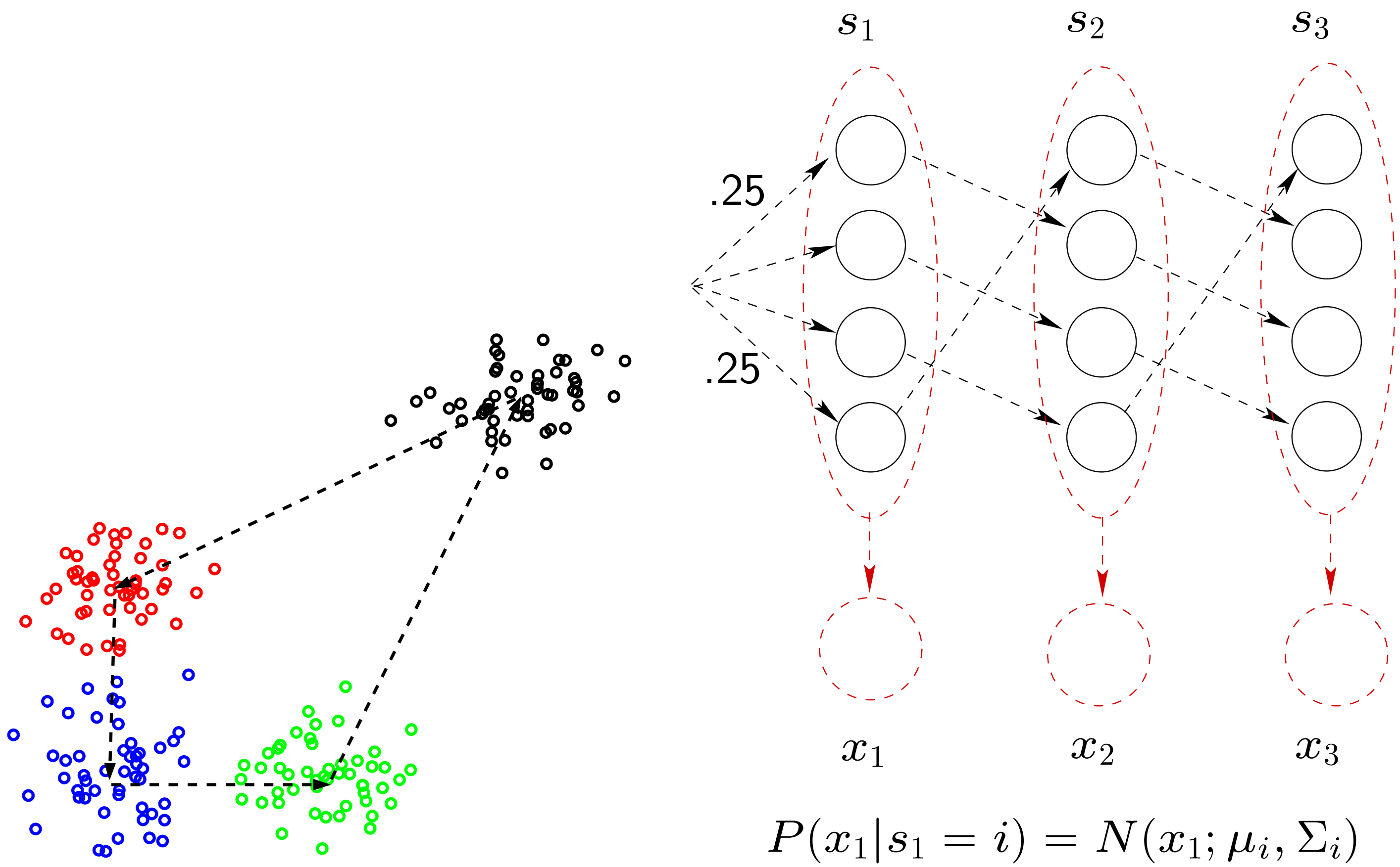
Hidden Markov models (HMMs)

- HMMs are Markov models with observations: the state variables s_t are not observed directly, only via associated observations x_t



$$P(s_1, x_1, s_2, x_2, \ldots) = P(s_1)P(x_1|s_1)P(s_2|s_1)P(x_2|s_2)\cdots$$

Hidden Markov models (HMMs)



$$P(x_1|s_1 = i) = N(x_1; \mu_i, \Sigma_i)$$