

# Statistical Machine Learning (BE4M33SSU)

## Lecture 11: Hidden Markov Models

Czech Technical University in Prague

- ◆ Hidden Markov Models
- ◆ Inference algorithms for HMMs
- ◆ Parameter learning for HMMs

## Hidden Markov Models

- ◆ Let  $\mathbf{s} = (s_1, s_2, \dots, s_n)$  denote a sequence of hidden states from a finite set  $K$ .
- ◆ Let  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  denote a sequence of features from some feature space  $\mathcal{X}$ .

**Definition 1.** A joint p.d. on  $\mathcal{X}^n \times K^n$  is a Hidden Markov model if

- (a) the prior p.d.  $p(\mathbf{s})$  for the sequences of hidden states is a Markov model, and
- (b) the conditional distribution  $p(\mathbf{x} | \mathbf{s})$  for the feature sequence is independent, i.e.

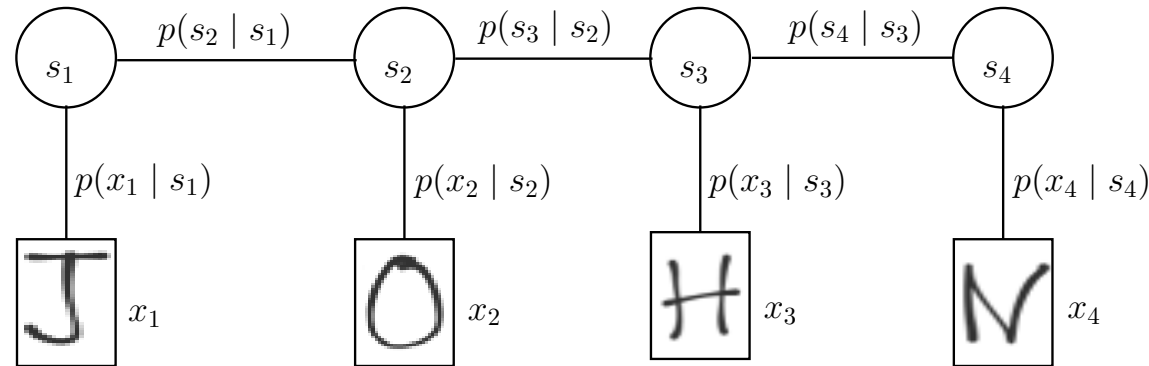
$$p(\mathbf{x} | \mathbf{s}) = \prod_{i=1}^n p(x_i | s_i).$$

The joint model is given by

$$p(\mathbf{x}, \mathbf{s}) = p(s_1) \prod_{i=2}^n p(s_i | s_{i-1}) \prod_{i=1}^n p(x_i | s_i).$$

# Hidden Markov Models

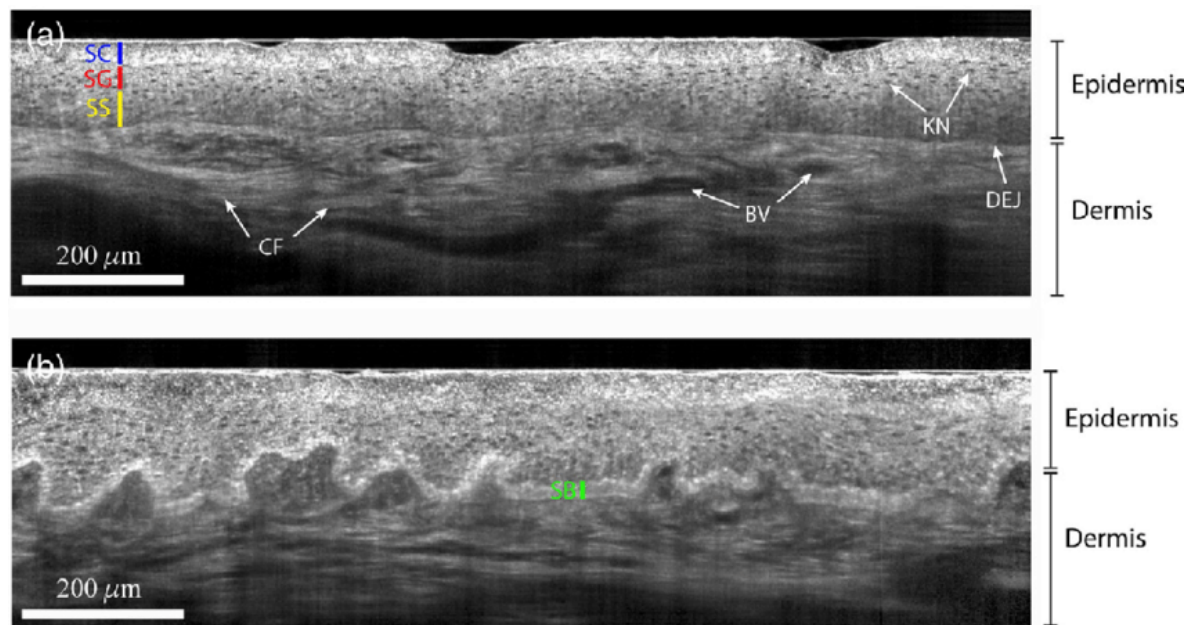
**Example 1** (Text recognition, OCR).



- ◆  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  – sequence of images with characters,
- ◆  $\mathbf{s} = (s_1, s_2, \dots, s_n)$  – sequence of alphabetic characters,
- ◆  $p(s_i | s_{i-1})$  – language model,
- ◆  $p(x_i | s_i)$  – appearance model for characters.

## Hidden Markov Models

**Example 2** (Skin layer segmentation, OCT).



- ◆  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  – sequence of image columns (features),
- ◆  $\mathbf{s} = (s_1, s_2, \dots, s_n)$  – sequence of boundary height values,
- ◆  $p(s_i | s_{i-1})$  – boundary model,
- ◆  $p(x_i | s_i)$  – appearance model for image columns.

## Inference algorithms for HMMs

How to find the most probable sequence of hidden states given the sequence of features  $x$

$$\mathbf{s}^* \in \arg \max_{\mathbf{s} \in K^n} p(s_1) \prod_{i=2}^n p(s_i | s_{i-1}) \prod_{i=1}^n p(x_i | s_i)$$

Take the logarithm, with

$$g_1(s_1, x_1) = \log[p(s_1)p(x_1 | s_1)]$$
$$g_i(s_{i-1}, s_i, x_i) = \log[p(s_i | s_{i-1})p(x_i | s_i)], \quad i = 2, \dots, n$$

Solve the task

$$\mathbf{s}^* \in \arg \max_{\mathbf{s} \in K^n} \left[ g_1(s_1, x_1) + \sum_{i=2}^n g_i(s_{i-1}, s_i, x_i) \right]$$

by dynamic programming as before for Markov models.

## Inference algorithms for HMMs

How to compute marginal probabilities for hidden states given the sequence of features?

We want to compute  $p(s_i = k, \mathbf{x})$ ,  $\forall k \in K$  in some sequence position  $j$ .

$$p(s_j, \mathbf{x}) = \sum_{s_1 \in K} \cdots \cancel{\sum_{s_j \in K}} \cdots \sum_{s_n \in K} p(s_1) p(x_1 | s_1) \prod_{i=2}^n [p(s_i | s_{i-1}) p(x_i | s_i)]$$

This is now more complicated, because we need to sum over the leading and trailing hidden state variables. Do this by dynamic matrix-vector multiplication from the left and from the right.

Initialise  $\phi_1(s_1) = p(s_1)p(x_1 | s_1)$  and  $\psi_n(s_n) \equiv 1$  and recursively compute

$$\phi_i(s_i) = \sum_{s_{i-1} \in K} p(x_i | s_i) p(s_i | s_{i-1}) \phi_{i-1}(s_{i-1}) \quad \forall s_i \in K$$

$$\psi_i(s_i) = \sum_{s_{i+1} \in K} p(x_{i+1} | s_{i+1}) p(s_{i+1} | s_i) \psi_{i+1}(s_{i+1}) \quad \forall s_i \in K$$

Denoting the transition probability matrices by  $P(i)$ , we can write this equivalently as matrix-vector multiplications:  $\phi_i = P(i)\phi_{i-1}$  and  $\psi_i = P^T(i+1)\psi_{i+1}$ .

## Inference algorithms for HMMs

The marginal probabilities are then obtained from

$$p(s_i = k, \mathbf{x}) = \phi_i(s_i = k)\psi_i(s_i = k)$$

The computational complexity for computing *all* marginal probabilities in *all* positions  $i = 1, \dots, n$  is thus  $\mathcal{O}(nK^2)$ .

### Remark 1.

- ◆ We can also compute *pairwise* marginal probabilities from the  $\phi$ -s and  $\psi$ -s

$$p(s_{i-1}, s_i, \mathbf{x}) = \phi_{i-1}(s_{i-1}) [p(s_i | s_{i-1})p(x_i | s_i)] \psi_i(s_i)$$

- ◆ Computing conditional marginal probabilities is then easy:

$$p(s_i = k | \mathbf{x}) = \frac{p(s_i = k, \mathbf{x})}{\sum_{k' \in K} p(s_i = k', \mathbf{x})}$$

- ◆ The same holds for computing the probability that the model will generate the sequence of features  $\mathbf{x}$ :  $p(\mathbf{x}) = \sum_{k \in K} p(s_i = k, \mathbf{x})$ .

## Learning algorithms for HMMs

### Supervised learning:

Given i.i.d. training data  $\mathcal{T}^m = \{(\mathbf{x}^j, \mathbf{s}^j) \in \mathcal{X}^n \times K^n \mid j = 1, \dots, m\}$ , estimate the parameters of the HMM by the maximum likelihood estimator.

This is done by simple “counting” as before for Markov models.

- ◆ Denote by  $a_i(s_{i-1} = \ell, s_i = k)$  the number of examples in  $\mathcal{T}^m$  for which  $s_{i-1} = \ell$  and  $s_i = k$ .
- ◆ Denote by  $b_i(s_i = k, x_i = x)$  the number of examples in  $\mathcal{T}^m$  for which  $s_i = k$  and  $x_i = x$ .

The estimates for the model parameters are then given by

$$p(s_i = k \mid s_{i-1} = \ell) = \frac{a_i(s_{i-1} = \ell, s_i = k)}{\sum_{k'} a_i(s_{i-1} = \ell, s_i = k')}, \quad p(x_i = x \mid s_i = k) = \frac{b_i(s_i = k, x_i = x)}{\sum_{x'} b_i(s_i = k, x_i = x')}$$

**Remark 2.** This is easy to generalise for the case that  $\mathcal{X} = \mathbb{R}^m$  and  $p(x_i \mid s_i)$  is from some parametric distribution family.

**Remark 3.** Learning HMMs by empirical risk minimisation has been discussed in Lecture 5. (Structured output SVMs).



## Learning algorithms for HMMs

### Unsupervised learning:

Given i.i.d. training data  $\mathcal{T}^m = \{\mathbf{x}^j \in \mathcal{X}^n \mid j = 1, \dots, m\}$ , estimate the parameters of the HMM by the maximum likelihood estimator.

We apply the EM-algorithm (aka Baum-Welch algorithm): Initialise the model, then iterate

**E-step** Use the current model estimate and compute the pairwise marginal probabilities

$$\alpha_{\mathbf{x}}(s_i, s_{i-1}) = p(s_i, s_{i-1} \mid \mathbf{x}) \quad s_{i-1}, s_i \in K$$

for each training example  $\mathbf{x} \in \mathcal{T}^m$  and all positions  $i = 2, \dots, n$ . See Remark 1.

**M-step** Use the  $\alpha$ -s as soft labels for computing the counts as in supervised learning

$$a_i(s_{i-1}, s_i) = \sum_{\mathbf{x} \in \mathcal{T}^m} \alpha_{\mathbf{x}}(s_i, s_{i-1}), \quad b_i(s_i, x) = \sum_{\mathbf{x} \in \mathcal{T}(x_i=x)} \alpha_{\mathbf{x}}(s_i)$$

where  $\mathcal{T}(x_i = x)$  is the set of training examples with  $x_i = x$  and  $\alpha_{\mathbf{x}}(s_i)$  is given by  $\alpha_{\mathbf{x}}(s_i) = \sum_{s_{i-1}} \alpha_{\mathbf{x}}(s_i, s_{i-1})$ .

## Summary

- ◆ Hidden Markov models are statistical models for pairs of processes (sequences)  $(s, x)$ , where  $s$  is a sequence of hidden states and not directly observable.
- ◆ Markov models and HMMs are exponential families.
- ◆ Their parameters can be estimated by supervised learning using either Maximum likelihood estimates or Empirical risk minimisation.
- ◆ Their parameters can be estimated by unsupervised learning using the EM-algorithm.
- ◆ All important inference and learning algorithms for HMMs have time complexity linear in the length of the sequence and quadratic in the size of the hidden state space.
- ◆ What if the space of hidden states is very large, e.g.  $s_i \in \mathbb{Z}^k$ ? We can use recurrent neural networks for modelling  $p(s_i | s_{i-1})$ , e.g. Gated recurrent Units (GRU) or Long short-term memory models (LSTM).