8. Supervised learning of HMMs: Empirical risk minimisation

Given: i.i.d. training data \( T = \{(x_i, s_i) \mid x_i \in F, s_i \in K, j = 1, \ldots, m\} \)
and the loss function \( L(s, s') = \mathbb{1}\{s \neq s'\} \)

Recall: optimal predictor \( h: F^n \rightarrow K^n \) for 0/1 loss is
\[
h^*_u(x) \in \arg\max_{s \in K^n} P_u(x, s)
\]

Empirical risk minimisation:
\[
\frac{1}{m} \sum_{j=1}^{m} \mathbb{1}\{s^j \neq h_u(x^j)\} \rightarrow \min_u
\]

This task is not tractable because the objective function is piece-wise constant.

Special case: Suppose, \( \exists u^* \) s.t. the empirical risk is zero.

How to find it? Conditions for \( u^* \):
\[
s^j \in \arg\max_{s \in K^n} P_u(x^j, s) \quad \forall j = 1, \ldots, m
\]

or, equivalently
\[
\langle \Phi(x^j, s^j), u^* \rangle > \langle \Phi(x^j, s), u^* \rangle \quad \forall s \neq s^j, \quad \forall j = 1, \ldots, m
\]

This is a system of linear inequalities \( \Rightarrow \) perceptron algorithm

Start with arbitrary \( u \) and iterate

- find \( s^j = \arg\max_{s \in K^n} \langle \Phi(x^j, s), u \rangle \) \( \quad \forall j = 1, \ldots, m \)

This can be done by the algorithm in Sec. 4

- if for some \( j \), \( s^* \neq s^j \), update \( u \) by
\[
u \rightarrow u + \Phi(x^j, s^j) - \Phi(x^j, s^*)
\]
General case

Idea: overcome intractability by replacing the loss (as a function of $u$) by a convex upper bound. E.g., "margin rescaling" surrogate

$$ I\{S \neq h_u(x)\} \leq \max_{s' \in K^n} \left\{ \Pi\{S \neq s'\} + \langle \Phi(x; s') - \Phi(x; s), u \rangle \right\} $$

The approximation task reads

$$ \frac{1}{m} \sum_{j=1}^{m} \max_{s \in K^n} \left\{ \Pi\{S \neq s\} + \langle \Phi(x; s) - \Phi(x; s^j), u \rangle \right\} \rightarrow \min_u $$

Solve by subgradient descent, cutting plane algorithm, ... The inner optimisation tasks $\max_{s \in K^n} \{ \ldots \}$ are solved by the algorithm in Sec. 4.

Remark 1. This approach is designated as "Structured Output SVM" and can be generalised for more complex losses as e.g. the Hamming distance.
9. Unsupervised learning: EM algorithm for HMMs

Given: i.i.d. training data $T = \{ x^i \in \mathbb{F}^n \mid i=1,..,m \}$

ML estimator: $u^* \in \text{argmax}_u \frac{1}{|T|} \sum_{x \in T} \log \sum_{s \in K^n} p_u(x,s)$

Recall EM algorithm

$$L(u) = \frac{1}{|T|} \sum_{x \in T} \log \sum_{s \in K^n} \frac{\alpha(s|x)}{\alpha(s|x)} p_u(x,s),$$

where $\alpha(s|x) > 0$, $\sum_{s \in K^n} \alpha(s|x) = 1 \; \forall \; x \in T$

Using concavity of log, we get a lower bound

$$L(u) \geq L_B(u,\alpha) = \frac{1}{|T|} \sum_{x \in T} \sum_{s \in K^n} \alpha(s|x) \log \frac{p_u(x,s)}{\alpha(s|x)}$$

Equivalently

$$L_B(u,\alpha) = \mathbb{E}_{\pi} \left[ \log p_u(x) - D_{KL}(\alpha(s|x) \| \rho(s|x)) \right]$$

EM algorithm: Maximise $L_B(u,\alpha)$ by block-coordinate ascent w.r.t. $\alpha$ and $u$. Start with some $u^{(0)}$

E-step set $\alpha^{(t)}(s|x) = p_u(x,s) \; \forall \; s \in K^n, \forall \; x \in T$

M-step set

$$u^{(t+1)} \in \text{argmax}_u \frac{1}{|T|} \sum_{x \in T} \sum_{s \in K^n} \alpha^{(t)}(s|x) \log p_u(x,s)$$

Let us analyse the M-step for HMMs. The objective is

$$\frac{1}{|T|} \sum_{x \in T} \sum_{s \in K^n} \alpha^{(t)}(s|x) \langle \phi(x,s), u \rangle - \log Z(u) \rightarrow \max_u$$
Denoting
\[ \Psi = \frac{1}{|T|} \sum_{x \in T} \sum_{s \in K^n} \alpha^{(k)}(s|x) \Phi(x,s) \]
we get
\[ \langle \Psi, u \rangle - \log Z(u) \to \max_u \]
This is equivalent to the supervised learning task in Sec. 7. We know how to solve it, provided we can compute \( \Psi \).

Computing \( \Psi \):
For each \( x \in T \) compute
\[ \Psi^i(x) = \sum_{s \in K^n} \alpha^{(k)}(s|x) \Phi(x,s) = \mathbb{E}_{\mu(w|x)} \Phi(x,s), \]
i.e. we have to compute posterior pairwise marginals
\( \Phi(s_{i-1}, s_i|x) \) \( i = 2, \ldots, n \) and \( s_{i-1}, s_i \in K \). This can be done by an algorithm similar to the one discussed in Sec. 5.

The components of \( \Psi \) are then obtained by averaging the components of \( \Psi^i(x) \) over all \( x \in T \), i.e. \( \Psi = \mathbb{E}_T \Psi^i(x) \).

**Theorem 1** (w/o proof)
The sequence \( L(\Psi^{(k)}) \) is monotonously increasing and the sequence \( \alpha^{(k)} \) is convergent.

**Remark 1** The EM algorithm for HMMs is referred to as Baum-Welch algorithm.