

# Statistical Machine Learning (BE4M33SSU)

## Lecture 5: Structured Output Support Vector Machines

Czech Technical University in Prague  
V. Franc

## Linear classifier

- ◆  $\mathcal{X}$  ... set of observations
- ◆  $\mathcal{Y}$  ... finite set of hidden state
- ◆  $\phi: \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}^n$  ... joint feature map
- ◆  $\mathbf{w} \in \mathbb{R}^n$  ... vector of parameters
- ◆ Generic linear classifier  $h: \mathcal{X} \rightarrow \mathcal{Y}$

$$h(x; \mathbf{w}) \in \underset{y \in \mathcal{Y}}{\text{Argmax}} \langle \mathbf{w}, \phi(x, y) \rangle$$

## Example: two-class linear classifier

- ◆  $\mathcal{X} \in \mathbb{R}^n$  ... set of observations
- ◆  $\mathcal{Y} = \{+1, -1\}$  ... binary labels
- ◆ Two-class linear classifier  $h: \mathbb{R}^n \rightarrow \{-1, +1\}$

$$h(\mathbf{x}; \mathbf{u}, v) = \text{sign}(\langle \mathbf{u}, \mathbf{x} \rangle + v) = \begin{cases} +1 & \text{if } \langle \mathbf{u}, \mathbf{x} \rangle + v \geq 0 \\ -1 & \text{if } \langle \mathbf{u}, \mathbf{x} \rangle + v < 0 \end{cases}$$

- ◆ We can write the two-class classifier as

$$h(\mathbf{x}; \mathbf{w}) \in \underset{y \in \{-1, +1\}}{\text{Argmax}} y(\langle \mathbf{u}, \mathbf{x} \rangle + v) = \underset{y \in \{-1, +1\}}{\text{Argmax}} \langle \mathbf{w}, \phi(\mathbf{x}, y) \rangle$$

where  $\phi: \mathbb{R}^n \times \{-1, +1\} \rightarrow \mathbb{R}^{n+1}$

$$\phi(\mathbf{x}, y) = y(\mathbf{x}, 1) \quad \text{and} \quad \mathbf{w} = (\mathbf{u}, v)$$

## Example: multi-class linear classifier

- ◆  $\mathcal{X} = \mathbb{R}^n$  ... set of observations;  $\mathcal{Y} = \{1, \dots, Y\}$  ... set of class labels
- ◆ Multi-class linear classifier  $h: \mathcal{X} \rightarrow \mathcal{Y}$

$$h(x; \mathbf{w}) \in \underset{y \in \mathcal{Y}}{\text{Argmax}} \langle \mathbf{w}_y, \mathbf{x} \rangle$$

where  $\mathbf{w} = (\mathbf{w}_1, \dots, \mathbf{w}_Y) \in \mathbb{R}^{n \cdot Y}$  are parameters.

- ◆ We can write the multi-class classifier as

$$h(\mathbf{x}; \mathbf{w}) \in \underset{y \in \mathcal{Y}}{\text{Argmax}} \langle \mathbf{w}_y, \mathbf{x} \rangle = \underset{y \in \mathcal{Y}}{\text{Argmax}} \langle \mathbf{w}, \phi(\mathbf{x}, y) \rangle$$

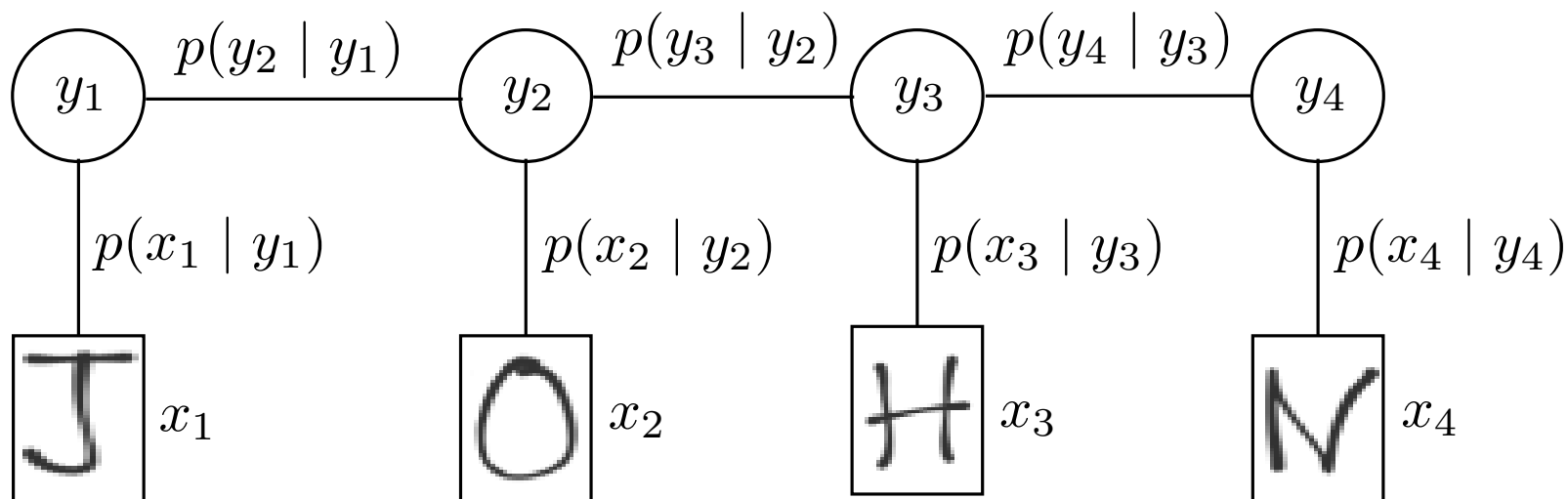
where  $\phi: \mathbb{R}^n \times \mathcal{Y} \rightarrow \mathbb{R}^{n \cdot Y}$  is

$$\phi(\mathbf{x}, y) = (0, \dots, \underbrace{\mathbf{x}}_{y\text{-th slot}}, \dots, 0)$$

## Example: sequence classifier for OCR

- ◆  $\mathbf{x} = (x_1, \dots, x_L) \in \mathcal{I}^L$  ... sequence of images with characters
- ◆  $\mathbf{y} = (y_1, \dots, y_L) \in \mathcal{A}^L$  ... seq. of chars. from  $\mathcal{A} = \{A, \dots, Z\}$
- ◆  $p(x_i | y_i)$  ... appearance model for characters
- ◆  $p(y_i | y_{i-1})$  ... language model
- ◆ Hidden Markov Chain model of the sequences:

$$p(x_1, \dots, x_L, y_1, \dots, y_L) = p(y_1) \prod_{i=2}^L p(y_i | y_{i-1}) \prod_{i=1}^L p(x_i | y_i)$$



## Example: sequence classifier for OCR

- ◆ The MAP estimate from HMC:

$$\hat{\mathbf{y}} \in \underset{\mathbf{y} \in \mathcal{A}^L}{\text{Argmax}} \left( \log p(y_1) + \sum_{i=2}^L \log p(y_i | y_{i-1}) + \sum_{i=1}^L \log p(x_i | y_i) \right)$$

- ◆ Let us assume the following parametrization:

$$\begin{aligned} \log p(y_1) &= \langle \mathbf{w}, \phi(y_1) \rangle \\ \log p(y_i | y_{i-1}) &= \langle \mathbf{w}, \phi(y_{i-1}, y_i) \rangle \\ \log p(x_i | y_i) &= \langle \mathbf{w}, \phi(x_i, y_i) \rangle \end{aligned}$$

- ◆ The MAP estimate becomes a linear classifier:

$$\hat{\mathbf{y}} = \underset{(y_1, \dots, y_k) \in \mathcal{A}^L}{\text{Argmax}} \left\langle \mathbf{w}, \underbrace{\phi(y_1) + \sum_{i=2}^L \phi(y_{i-1}, y_i) + \sum_{i=1}^L \phi(x_i, y_i)}_{\phi(\mathbf{x}, \mathbf{y})} \right\rangle$$

## Learning by Empirical Risk Minimization

- ◆  $\ell: \mathcal{Y} \times \mathcal{Y} \rightarrow [0, \infty)$  ... loss such that  $\ell(y, y') = 0$  iff  $y = y'$
- ◆ Find  $\mathbf{w}$  of  $h(x; \mathbf{w}) \in \text{Argmax}_{y \in \mathcal{Y}} \langle \mathbf{w}, \phi(x, y) \rangle$  which minimizes the risk

$$R(\mathbf{w}) = \mathbb{E}_{(x,y) \sim p} \left( \ell(y, h(x; \mathbf{w})) \right)$$

- ◆ ERM based learning leads to solving

$$\mathbf{w}^* \in \underset{\mathbf{w} \in \mathbb{R}^n}{\text{Argmin}} R_{\mathcal{T}^m}(\mathbf{w})$$

where the empirical risk is

$$R_{\mathcal{T}^m}(\mathbf{w}) = \frac{1}{m} \sum_{i=1}^m \ell(y^i, h(x^i; \mathbf{w}))$$

and  $\mathcal{T}^m = \{(x^i, y^i) \in (\mathcal{X} \times \mathcal{Y}) \mid i = 1, \dots, m\}$  are training examples drawn from i.i.d. with distribution  $p(x, y)$ .

## Learning linear classifier from separable examples

- ◆ An example  $(x^i, y^i)$  is correctly classified, that is,

$$y^i = h(x^i; \mathbf{w}) = \underset{y \in \mathcal{Y}}{\text{Argmax}} \langle \mathbf{w}, \phi(x^i, y) \rangle$$

is equivalent to

$$\langle \phi(x^i, y^i), \mathbf{w} \rangle > \langle \phi(x^i, y), \mathbf{w} \rangle, \quad \forall y \in \mathcal{Y} \setminus \{y^i\}$$

**Definition:** The examples  $\mathcal{T}^m = \{(x^i, y^i) \in (\mathcal{X} \times \mathcal{Y}) \mid i = 1, \dots, m\}$  are linearly separable w.r.t. joint feature map  $\phi: \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}^n$  if there exists  $\mathbf{w} \in \mathbb{R}^n$  such that

$$\langle \phi(x^i, y^i), \mathbf{w} \rangle > \langle \phi(x^i, y), \mathbf{w} \rangle, \quad \forall i \in \{1, \dots, m\}, y \in \mathcal{Y} \setminus \{y^i\}$$



## (Generic) Perceptron algorithm

- ◆ **Task:** given a set of points  $\{\mathbf{a}^i \in \mathbb{R}^n \mid i = 1, 2, \dots, K\}$  we want to find  $\mathbf{w} \in \mathbb{R}^n$  such that

$$\langle \mathbf{w}, \mathbf{a}^i \rangle > 0, \quad \forall i \in \{1, 2, \dots, K\} \quad (1)$$

- ◆ **Algorithm:**

1.  $\mathbf{w} \leftarrow \mathbf{0}$
2. Find a violating  $\langle \mathbf{w}, \mathbf{a}^i \rangle \leq 0, i \in \{1, 2, \dots, K\}$
3. If there is no violating inequality return  $\mathbf{w}$  otherwise update

$$\mathbf{w} \leftarrow \mathbf{w} + \mathbf{a}^i$$

and go to step 2.

- ◆ If the set of inequalities (1) is solvable then the Perceptron algorithm exits in a finite number of steps which does not depend on  $m$ .

# Structured Output Perceptron

- ◆ Learning  $h(x; \mathbf{w}) \in \text{Argmax}_{y \in \mathcal{Y}} \langle \mathbf{w}, \phi(x, y) \rangle$  from examples  $\mathcal{T}^m = \{(x^i, y^i) \in (\mathcal{X} \times \mathcal{Y}) \mid i = 1, \dots, m\}$  leads to solving

$$\langle \phi(x^i, y^i), \mathbf{w} \rangle - \langle \phi(x^i, y), \mathbf{w} \rangle > 0, \quad \forall i \in \{1, \dots, m\}, y \in \mathcal{Y} \setminus \{y^i\}$$

- ◆ **Algorithm:**

1.  $\mathbf{w} \leftarrow \mathbf{0}$
2. Find a misclassified example  $(x^i, y^i) \in \mathcal{T}^m$  such that

$$y^i \neq \hat{y}^i \in \text{Argmax}_{y \in \mathcal{Y}} \langle \mathbf{w}, \phi(x^i, y) \rangle \quad \text{prediction problem}$$

3. If there is no misclassified example return  $\mathbf{w}$  otherwise update

$$\mathbf{w} \leftarrow \mathbf{w} + \phi(x^i, y^i) - \phi(x^i, \hat{y}^i) \quad \text{parameter update}$$

and go to step 2.

# Structured Output Support Vector Machines

- ◆ Learning  $h(x; \mathbf{w}) \in \text{Argmax}_{y \in \mathcal{Y}} \langle \mathbf{w}, \phi(x, y) \rangle$  from examples  $\mathcal{T}^m = \{(x^i, y^i) \in (\mathcal{X} \times \mathcal{Y}) \mid i = 1, \dots, m\}$  by ERM leads to

$$\mathbf{w}^* \in \underset{\mathbf{w} \in \mathbb{R}^n}{\text{Argmin}} R_{\mathcal{T}^m}(\mathbf{w}) \quad \text{where} \quad R_{\mathcal{T}^m}(\mathbf{w}) = \frac{1}{m} \sum_{i=1}^m \ell(y^i, h(x^i; \mathbf{w}))$$

- ◆ The SO-SVM approximates the ERM by a convex problem

$$\mathbf{w}^* \in \underset{\mathbf{w} \in \mathcal{W}_r}{\text{Argmin}} R^\psi(\mathbf{w}) \quad \text{where} \quad R^\psi(\mathbf{w}) = \frac{1}{m} \sum_{i=1}^m \psi(x^i, y^i, \mathbf{w})$$

where

- $\mathcal{W}_r = \{\mathbf{w} \in \mathbb{R}^n \mid \|\mathbf{w}\| \leq r\}$  ... a ball of radius  $r$
- $\psi: \mathcal{X} \times \mathcal{Y} \times \mathbb{R}^n \rightarrow \mathbb{R}$  ... proxy approximating the true loss  $\ell$

## Margin rescaling loss

- ◆ The score for the correct label of an example  $(x^i, y^i)$  should be above scores for incorrect labels increased by margin proportional to loss  $\ell(y^i, y)$ :

$$\langle \mathbf{w}, \phi(x^i, y^i) \rangle \geq \langle \mathbf{w}, \phi(x^i, y) \rangle + \ell(y^i, y), \quad \forall y \in \mathcal{Y} \setminus \{y^i\}$$

- ◆ Example: Sequential OCR, Hamming distance  $\ell(\mathbf{y}, \mathbf{y}') = \sum_{i=1}^L [y_i \neq y'_i]$

$$\psi(x^i, y^i, \mathbf{w}) = \max \left\{ 0, \max \left\{ \begin{aligned} &4 + \langle \phi(\text{JOHN}, \text{AAAA}), \mathbf{w} \rangle - \langle \phi(\text{JOHN}, \text{JOHN}), \mathbf{w} \rangle, \\ &3 + \langle \phi(\text{JOHN}, \text{JAAA}), \mathbf{w} \rangle - \langle \phi(\text{JOHN}, \text{JOHN}), \mathbf{w} \rangle, \\ &2 + \langle \phi(\text{JOHN}, \text{JOAA}), \mathbf{w} \rangle - \langle \phi(\text{JOHN}, \text{JOHN}), \mathbf{w} \rangle, \\ &1 + \langle \phi(\text{JOHN}, \text{JOHA}), \mathbf{w} \rangle - \langle \phi(\text{JOHN}, \text{JOHN}), \mathbf{w} \rangle, \\ &\vdots \end{aligned} \right. \right.$$

## Margin rescaling loss

- ◆ The score for the correct label of an example  $(x^i, y^i)$  should be above scores for incorrect labels increased by margin proportional to loss  $\ell(y^i, y)$ :

$$\langle \mathbf{w}, \phi(x^i, y^i) \rangle \geq \langle \mathbf{w}, \phi(x^i, y) \rangle + \ell(y^i, y), \quad \forall y \in \mathcal{Y} \setminus \{y^i\}$$

- ◆ The margin rescaling loss

$$\psi(x^i, y^i, \mathbf{w}) = \max \left\{ 0, \max_{y \in \mathcal{Y} \setminus \{y^i\}} \left( \ell(y^i, y) + \langle \mathbf{w}, \phi(x^i, y) \rangle - \langle \mathbf{w}, \phi(x^i, y^i) \rangle \right) \right\}$$

- ◆ Upper bound of the true loss:

$$y^i \neq \hat{y} = h(x^i; \mathbf{w}) = \underset{y \in \mathcal{Y}}{\text{Argmax}} \langle \mathbf{w}, \phi(x^i, y) \rangle$$

implies  $\langle \mathbf{w}, \phi(x^i, \hat{y}) \rangle - \langle \mathbf{w}, \phi(x^i, y^i) \rangle \geq 0$  and hence

$$\psi(x^i, y^i, \mathbf{w}) \geq \ell(y^i, h(x^i, \mathbf{w})), \quad \forall \mathbf{w} \in \mathbb{R}^n$$

# SO-SVM as a convex unconstrained problem

- ◆ SO-SVM as **constrained convex problem**:

$$\mathbf{w}^* \in \underset{\mathbf{w} \in \mathcal{W}_r}{\text{Argmin}} R_r^\psi(\mathbf{w}) \quad \text{where} \quad R_r^\psi(\mathbf{w}) = \frac{1}{m} \sum_{i=1}^m \psi(x^i, y^i, \mathbf{w})$$

$$\text{and } \mathcal{W}_r = \{\mathbf{w} \in \mathbb{R}^n \mid \|\mathbf{w}\| \leq r\}$$

- ◆ SO-SVM as an **unconstrained convex problem**:

$$\begin{aligned} \mathbf{w}^* \in \underset{\mathbf{w} \in \mathbb{R}^n}{\text{Argmin}} R_\lambda^\psi(\mathbf{w}) \quad \text{where} \quad R_\lambda^\psi(\mathbf{w}) &= \frac{1}{m} \sum_{i=1}^m \left( \frac{\lambda}{2} \|\mathbf{w}\|^2 + \psi(x^i, y^i, \mathbf{w}) \right) \\ &= \frac{1}{m} \sum_{i=1}^m \psi_\lambda(x^i, y^i, \mathbf{w}) \end{aligned}$$

where  $\lambda > 0$  is a hyper-parameter which controls over-fitting.

# SO-SVM problem solved by Stochastic Gradient Descent

- ◆ The SO-SVM as an unconstrained convex problem:

$$\mathbf{w}^* \in \underset{\mathbf{w} \in \mathbb{R}^n}{\text{Argmin}} R_\lambda^\psi(\mathbf{w}) \quad \text{where} \quad R_\lambda^\psi(\mathbf{w}) = \frac{1}{m} \sum_{i=1}^m \psi_\lambda(x^i, y^i, \mathbf{w})$$

- ◆ **Algorithm:**

- 1: Choose an initial iterate  $\mathbf{w}_1 \in \mathbb{R}^n$
- 2: **for**  $k = 1, 2, \dots$
- 3:     Select an example  $(x^k, y^k) \in \mathcal{T}^m$  uniformly at random
- 4:     Compute subgradient  $\mathbf{g}_k$  of  $\psi_\lambda(x^k, y^k, \mathbf{w})$  at  $\mathbf{w}_k$

$$\mathbf{g}_k = \lambda \mathbf{w}_k + \phi(x^k, y^k) - \phi(x^k, \hat{y}^k)$$

$$\hat{y}^k \in \underset{y \in \mathcal{Y}}{\text{Argmax}} (\ell(y^k, y) + \langle \mathbf{w}_k, \phi(x^k, y) \rangle)$$

- 4:     Choose a stepsize  $\alpha_k > 0$      (e.g.  $\alpha_k = \frac{\text{constant}}{k}$ )
- 5:     Set the new iterate  $\mathbf{w}_{k+1} \leftarrow \mathbf{w}_k - \alpha_k \mathbf{g}_k$