Statistical Machine Learning (BE4M33SSU)
Lecture 5: Structured Output Support Vector Machines

Czech Technical University in Prague
V. Franc

Generic linear classifier

- $\mathcal{X}$ ... set of observations
- $\mathcal{Y}$ ... finite set of hidden states
- $\boldsymbol{\phi}\colon \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^n$ ... joint feature map (chosen prior to learning)
- $\boldsymbol{w} \in \mathbb{R}^n$ ... vector of parameters (learned from examples)
- Generic linear classifier $h\colon \mathcal{X} \to \mathcal{Y}$

$$h(x; \boldsymbol{w}) \in \operatorname*{Argmax}_{y \in \mathcal{Y}} \langle \boldsymbol{w}, \boldsymbol{\phi}(x, y) \rangle$$

Example: two-class linear classifier

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

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

- We can write the two-class classifier as

$$h(\boldsymbol{x}; \boldsymbol{w}) \in \operatorname*{Argmax}_{y \in \{-1,+1\}} y(\langle \boldsymbol{u}, \boldsymbol{x} \rangle + v) = \operatorname*{Argmax}_{y \in \{-1,+1\}} \langle \boldsymbol{w}, \boldsymbol{\phi}(\boldsymbol{x}, y) \rangle$$

where $\boldsymbol{\phi}\colon \mathbb{R}^n \times \{-1, +1\} \to \mathbb{R}^{n+1}$

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

Example: multi-class linear classifier

- $\mathcal{X} = \mathbb{R}^n$ ... set of observations; $\mathcal{Y} = \{1, \ldots, Y\}$ ... set of class labels
- Multi-class linear classifier $h\colon \mathcal{X} \to \mathcal{Y}$

$$h(x; \boldsymbol{w}) \in \operatorname*{Argmax}_{y \in \mathcal{Y}} \langle \boldsymbol{w}_y, \boldsymbol{x} \rangle$$

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

- We can write the multi-class classifier as

$$h(\boldsymbol{x}; \boldsymbol{w}) \in \operatorname*{Argmax}_{y \in \mathcal{Y}} \langle \boldsymbol{w}_y, \boldsymbol{x} \rangle = \operatorname*{Argmax}_{y \in \mathcal{Y}} \langle \boldsymbol{w}, \boldsymbol{\phi}(\boldsymbol{x}, y) \rangle$$

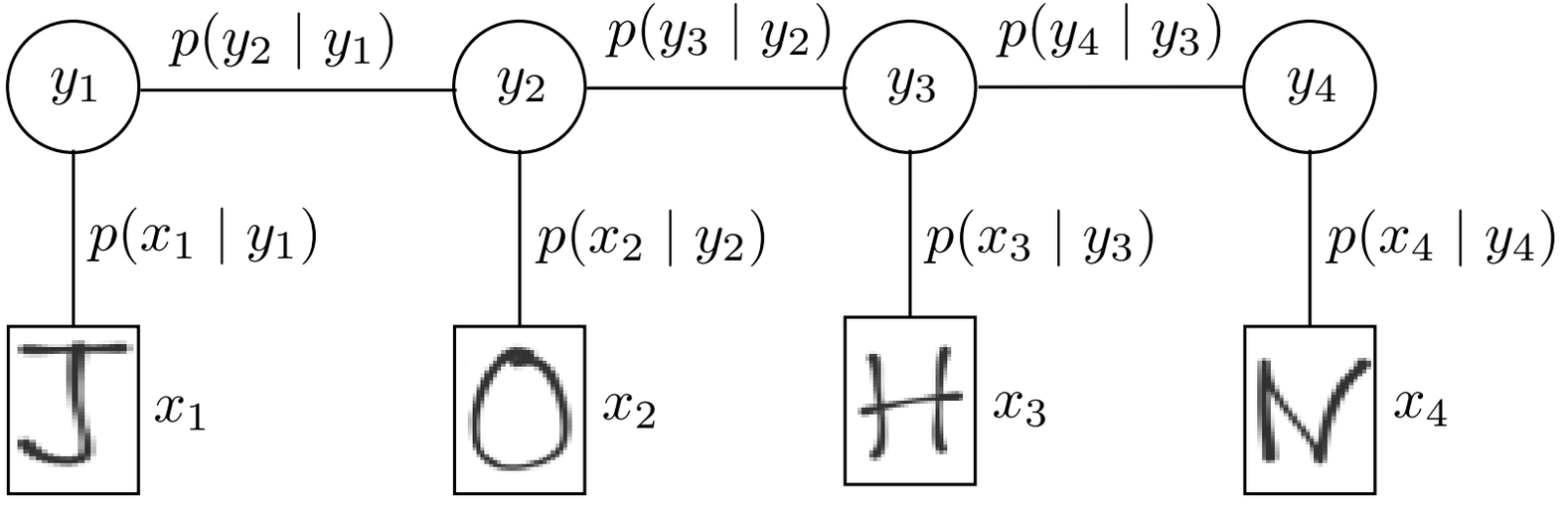
where $\boldsymbol{\phi}\colon \mathbb{R}^n \times \mathcal{Y} \to \mathbb{R}^{n \cdot Y}$ is

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

Example: sequence classifier for OCR

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

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



Example: sequence classifier for OCR

- The MAP estimate $\hat{\boldsymbol{y}} \in \operatorname{Argmax}_{\boldsymbol{y} \in \mathcal{A}^L} p(\boldsymbol{y} \mid \boldsymbol{x})$ from the HMC:

$$\hat{\boldsymbol{y}} \in \operatorname*{Argmax}_{\boldsymbol{y} \in \mathcal{A}^L} \left(\log p(y_1) + \sum_{i=2}^{L} \log p(y_i \mid y_{i-1}) + \sum_{i=1}^{L} \log p(x_i \mid y_i) \right)$$

- Let us assume the following linear parametrization:

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

- The MAP estimate becomes a linear classifier:

$$\hat{\boldsymbol{y}} = \operatorname*{Argmax}_{(y_1,\ldots,y_k) \in \mathcal{A}^L} \Big\langle \boldsymbol{w}, \underbrace{\boldsymbol{\phi}(y_1) + \sum_{i=2}^{L} \boldsymbol{\phi}(y_{i-1}, y_i) + \sum_{i=1}^{L} \boldsymbol{\phi}(x_i, y_i)}_{\boldsymbol{\phi}(\boldsymbol{x}, \boldsymbol{y})} \Big\rangle$$

Learning generic linear classifier by Emprical Risk Minimization

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

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

- ERM based learning leads to solving

$$\boldsymbol{w}^* \in \operatorname*{Argmin}_{\boldsymbol{w} \in \mathbb{R}^n} R_{\mathcal{T}^m}(\boldsymbol{w})$$

where the empirical risk is

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

and $\mathcal{T}^m = \{(x^i, y^i) \in (\mathcal{X} \times \mathcal{Y}) \mid i = 1, \ldots, m\}$ 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,

$$y^i = h(x^i; \boldsymbol{w}) = \operatorname*{Argmax}_{y \in \mathcal{Y}} \langle \boldsymbol{w}, \boldsymbol{\phi}(x^i, y) \rangle$$

  is equivalent to

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

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

$$\langle \boldsymbol{\phi}(x^i, y^i), \boldsymbol{w} \rangle > \langle \boldsymbol{\phi}(x^i, y), \boldsymbol{w} \rangle \,, \qquad \forall i \in \{1, \ldots, m\}, y \in \mathcal{Y} \setminus \{y^i\}$$

(Generic) Perceptron algorithm

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

$$\langle \boldsymbol{w}, \boldsymbol{a}^i \rangle > 0\,, \qquad \forall i \in \{1, 2, \ldots, K\} \tag{1}$$

- **Algorithm:**

1. $\boldsymbol{w} \leftarrow \boldsymbol{0}$
2. Find $k \in \{1, 2, \ldots, K\}$ such that $\langle \boldsymbol{w}, \boldsymbol{a}^k \rangle \leq 0$
3. If there is no such k return $\boldsymbol{w}$ otherwise update

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \boldsymbol{a}^k$$

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 $h(x; \boldsymbol{w}) \in \mathrm{Argmax}_{y \in \mathcal{Y}} \langle \boldsymbol{w}, \boldsymbol{\phi}(x, y) \rangle$ from examples $\mathcal{T}^m = \{(x^i, y^i) \in (\mathcal{X} \times \mathcal{Y}) \mid i = 1, \ldots, m\}$ leads to solving

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

- **Algorithm:**

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

$$y^k \neq \hat{y}^k = \operatorname*{Argmax}_{y \in \mathcal{Y}} \langle \boldsymbol{w}, \boldsymbol{\phi}(x^k, y) \rangle \qquad \text{prediction problem}$$

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

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \boldsymbol{\phi}(x^k, y^k) - \boldsymbol{\phi}(x^k, \hat{y}^k) \qquad \text{parameter update}$$

and go to step 2.

Structured Output Support Vector Machines

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

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

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

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

where

  - $\mathcal{W}_r = \{\boldsymbol{w} \in \mathbb{R}^n \mid \|\boldsymbol{w}\| \leq r\}$ … a ball of radius r
  - $\psi \colon \mathcal{X} \times \mathcal{Y} \times \mathbb{R}^n \to \mathbb{R}$ … proxy approximating the true loss ℓ

Margin rescaling loss

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

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

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

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

Margin rescaling loss

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

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

- The margin rescaling loss

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

- Upper bound of the true loss:

$$y^i \neq \hat{y} = h(x^i; \boldsymbol{w}) = \operatorname*{Argmax}_{y \in \mathcal{Y}} \langle \boldsymbol{w}, \boldsymbol{\phi}(x^i, y) \rangle$$

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

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

SO-SVM as a convex unconstrained problem

- SO-SVM as constrained convex problem:

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

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

- SO-SVM as an unconstrained convex problem:

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

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:

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

- **Algorithm**:

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

$$\hat{y}^k \in \operatorname*{Argmax}_{y\in\mathcal{Y}} \left(\ell(y^k, y) + \langle \boldsymbol{w}_k, \boldsymbol{\phi}(x^k, y)\rangle\right)$$

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

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

Summary

- General linear classifier.
- Perceptron learning algorithm.
- Structured Output Support Vector Machines.
- Training SO-SVM by Stochastic Gradient Descent.