Machine Learning I

Bjoern Andres, Shengxian Zhao

Machine Learning for Computer Vision
TU Dresden

Winter Term 2022/2023

Contents.

- This part of the course is about the problem of learning to order a finite set.
- This problem is introduced as an **unsupervised learning** problem w.r.t. **constrained data**.

We consider any finite, non-empty set A that we seek to order.

Definition. A strict order on A is a binary relation $< \subseteq A \times A$ that satisfies the following conditions:

$$\forall a \in A\colon \quad \neg a < a \tag{1}$$

$$\forall \{a, b\} \in \binom{A}{2}\colon \quad a < b \ \text{ xor } \ b < a \tag{2}$$

$$\forall \{a, b, c\} \in \binom{A}{3}\colon \quad a < b \ \wedge \ b < c \ \Rightarrow \ a < c \tag{3}$$

Lemma. The strict orders on A are characterized by the bijections $\alpha : \{0, \dots, |A| - 1\} \to A$. For any such bijection, consider the order $<_\alpha$ such that

$$\forall a, b \in A\colon \quad a < b \;\Leftrightarrow\; \alpha^{-1}(a) < \alpha^{-1}(b) \ . \tag{4}$$

Lemma. The strict orders on A are characterized by those

$$y : \{(a, b) \in A \times A \mid a \neq b\} \to \{0, 1\} \tag{5}$$

that satisfy the following conditions:

$$\forall a \in A \; \forall b \in A \setminus \{a\}\colon \quad y_{ab} + y_{ba} = 1 \tag{6}$$

$$\forall a \in A \; \forall b \in A \setminus \{a\} \; \forall c \in A \setminus \{a, b\}\colon \quad y_{ab} + y_{bc} - 1 \leq y_{ac} \tag{7}$$

Constrained Data

We reduce the problem of learning and inferring orders to the problem of learning and inferring decisions, by defining **constrained data** (S, X, x, Y) with

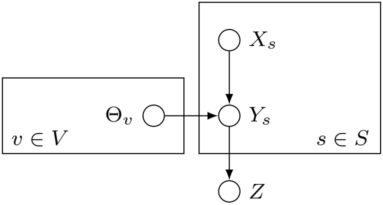
$$S = \{(a,b) \in A \times A \mid a \neq b\} \tag{8}$$

$$\mathcal{Y} = \Big\{ y \in \{0,1\}^S \;\Big|\; \forall a \in A \; \forall b \in A \setminus \{a\} \colon \quad y_{ab} + y_{ba} = 1 \\ \forall a \in A \; \forall b \in A \setminus \{a\} \; \forall c \in A \setminus \{a,b\} \colon \\ y_{ab} + y_{bc} - 1 \leq y_{ac} \Big\} \tag{9}$$

Familiy of functions

- We consider a finite, non-empty set V, called a set of **attributes**, and the **attribute space** $X = \mathbb{R}^V$
- We consider **linear functions**. Specifically, we consider $\Theta = \mathbb{R}^V$ and $f : \Theta \to \mathbb{R}^X$ such that

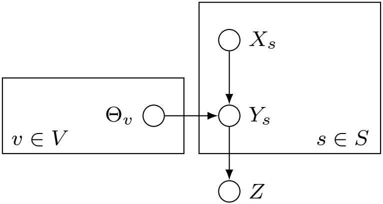
$$\forall \theta \in \Theta \; \forall \hat{x} \in \mathbb{R}^V : \quad f_\theta(\hat{x}) = \sum_{v \in V} \theta_v \, \hat{x}_v = \langle \theta, \hat{x} \rangle \; . \tag{10}$$



Random Variables

- For any $(a, b) = s \in S = E$, let X_s be a random variable whose value is a vector $x_s \in \mathbb{R}^V$, the **attribute vector** of s.
- For any $(a, b) = s \in S$, let Y_s be a random variable whose value is a binary number $y_s \in \{0, 1\}$, called the **decision** placing a before b.
- For any $v \in V$, let Θ_v be a random variable whose value is a real number $\theta_v \in \mathbb{R}$, a **parameter** of the function we seek to learn.
- Let Z be a random variable whose value is a subset $\mathcal{Z} \subseteq \{0, 1\}^S$ called the set of **feasible decisions**. For ordering, we are interested in $\mathcal{Z} = \mathcal{Y}$, the set of characteristic functions of strict orders on A.

Ordering



Factorization

$$P(X, Y, Z, \Theta) = P(Z \mid Y) \prod_{s \in S} P(Y_s \mid X_s, \Theta) \prod_{v \in V} P(\Theta_v) \prod_{s \in S} P(X_s)$$

Factorization

- Supervised learning:

$$\begin{aligned} P(\Theta \mid X, Y, Z) &= \frac{P(X, Y, Z, \Theta)}{P(X, Y, Z)} \\ &= \frac{P(Z \mid Y)\, P(Y \mid X, \Theta)\, P(X)\, P(\Theta)}{P(Z \mid X, Y)\, P(X, Y)} \\ &= \frac{P(Z \mid Y)\, P(Y \mid X, \Theta)\, P(X)\, P(\Theta)}{P(Z \mid Y)\, P(X, Y)} \\ &= \frac{P(Y \mid X, \Theta)\, P(X)\, P(\Theta)}{P(X, Y)} \\ &\propto P(Y \mid X, \Theta)\, P(\Theta) \\ &= \prod_{s \in S} P(Y_s \mid X_s, \Theta) \prod_{v \in V} P(\Theta_v) \end{aligned}$$

Factorization

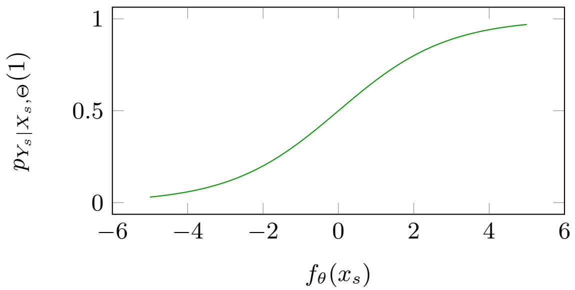
- Inference:

$$
\begin{aligned}
P(Y \mid X, Z, \theta) &= \frac{P(X, Y, Z, \Theta)}{P(X, Z, \Theta)} \\
&= \frac{P(Z \mid Y)\, P(Y \mid X, \Theta)\, P(X)\, P(\Theta)}{P(X, Z, \Theta)} \\
&\propto P(Z \mid Y)\, P(Y \mid X, \Theta) \\
&= P(Z \mid Y) \prod_{s \in S} P(Y_s \mid X_s, \Theta)
\end{aligned}
$$

Distributions

- **Logistic distribution**

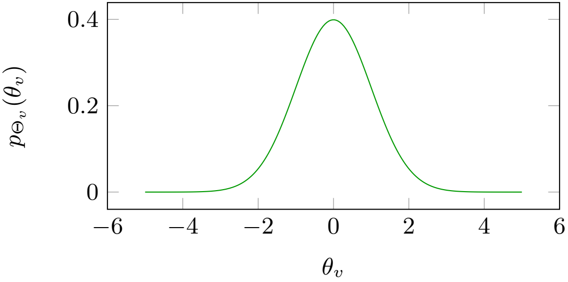
$$\forall s \in S: \qquad p_{Y_s|X_s,\Theta}(1) = \frac{1}{1 + 2^{-f_\theta(x_s)}} \qquad (11)$$



Distributions

- **Normal distribution** with $\sigma \in \mathbb{R}^+$:

$$\forall v \in V : \qquad p_{\Theta_v}(\theta_v) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\theta_v^2/2\sigma^2} \tag{12}$$



Distributions

- **Uniform distribution on a subset**

$$\forall \mathcal{Z} \subseteq \{0,1\}^S \; \forall y \in \{0,1\}^S \quad p_{Z|Y}(\mathcal{Z}, y) \propto \begin{cases} 1 & \text{if } y \in \mathcal{Z} \\ 0 & \text{otherwise} \end{cases}$$

Note that $p_{Z|Y}(\mathcal{Y}, y)$ is non-zero iff the labeling $y\colon S \to \{0,1\}$ defines an order on A.

Lemma. Estimating maximally probable parameters θ, given attributes x and decisions y, i.e.,

$$\underset{\theta \in \mathbb{R}^V}{\operatorname{argmax}} \quad p_{\Theta|X,Y,Z}(\theta, x, y, \mathcal{Y})$$

is an l_2-regularized logistic regression problem.

Proof. Analogous to the case of deciding, we obtain:

$$\begin{aligned} & \underset{\theta \in \mathbb{R}^V}{\operatorname{argmax}} \quad p_{\Theta|X,Y,Z}(\theta, x, y, \mathcal{Y}) \\ = & \underset{\theta \in \mathbb{R}^V}{\operatorname{argmin}} \quad \sum_{s \in S} \left(-y_s\, f_\theta(x_s) + \log\left(1 + 2^{f_\theta(x_s)}\right)\right) + \frac{\log e}{2\sigma^2} \|\theta\|_2^2 \ . \end{aligned}$$

Lemma. Estimating maximally probable decisions y, given attributes x, given the set of feasible decisions $\mathcal{Y}$, and given parameters θ, i.e.,

$$\underset{y \in \{0,1\}^S}{\operatorname{argmax}} \quad p_{Y|X,Z,\Theta}(y, x, \mathcal{Y}, \theta) \tag{13}$$

assumes the form of the **linear ordering problem**:

$$\underset{y\colon S \to \{0,1\}}{\operatorname{argmin}} \quad \sum_{s \in S} (-\langle \theta, x_s \rangle)\, y_s \tag{14}$$

$$\text{subject to} \quad \forall a \in A \;\forall b \in A \setminus \{a\}\colon \quad y_{ab} + y_{ba} = 1 \tag{15}$$

$$\forall a \in A \;\forall b \in A \setminus \{a\} \;\forall c \in A \setminus \{a, b\}\colon$$

$$y_{ab} + y_{bc} - 1 \leq y_{ac} \tag{16}$$

Theorem. The linear ordering problem is NP-hard.

The linear ordering problem has been studied intensively. A comprehensive survey is by Martí and Reinelt (2011). Pioneering research is by Grötschel (1984).

We define two **local search algorithms** for the linear ordering problem.

For simplicity, we define $c : S \to \mathbb{R}$ such that

$$\forall s \in S\colon \quad c_s = -\langle \theta, x_s \rangle \tag{17}$$

and write the (linear) cost function $\varphi : \{0,1\}^S \to \mathbb{R}$ such that

$$\forall y \in \{0,1\}^S\colon \quad \varphi(y) = \sum_{s \in S} c_s \, y_s \tag{18}$$

Greedy transposition algorithm:

- The greedy transposition algorithm starts from any initial strict order.
- It searches for strict orders with lower objective value by swapping pairs of elements

Definition. For any bijection $\alpha : \{0, \ldots, |A| - 1\} \to A$ and any $j, k \in \{0, \ldots, |A| - 1\}$, let $\mathrm{transpose}_{jk}[\alpha]$ the bijection obtained from α by swapping α_j and α_k, i.e.

$$\forall l \in \{0, \ldots, |A| - 1\} : \quad \mathrm{transpose}_{jk}[\alpha](l) = \begin{cases} \alpha_k & \text{if } l = j \\ \alpha_j & \text{if } l = k \\ \alpha_l & \text{otherwise} \end{cases} . \tag{19}$$

$\alpha' = \text{greedy-transposition}(\alpha)$

choose $(j, k) \in \underset{0 \leq j' < k' < |A|}{\operatorname{argmin}} \varphi(y^{\text{transpose}_{j'k'}[\alpha]}) - \varphi(y^{\alpha})$

if $\varphi(y^{\text{transpose}_{jk}[\alpha]}) - \varphi(y^{\alpha}) < 0$

$\quad \alpha' := \text{greedy-transposition}(\text{transpose}_{jk}[\alpha])$

else

$\quad \alpha' := \alpha$

Greedy transposition using the technique of Kernighan and Lin (1970)

$\alpha' =$ greedy-transposition-kl(α)

$\alpha^0 := \alpha$
$\delta_0 := 0$
$J_0 := \{0, \ldots, |A| - 1\}$
$t := 0$
repeat (build sequence of swaps)
  choose $(j,k) \in \underset{\{(j',k') \in J_t^2 \mid j' < k'\}}{\operatorname{argmin}} \varphi(y^{\text{transpose}_{j'k'}[\alpha^t]}) - \varphi(y^{\alpha^t})$
  $\alpha^{t+1} :=$ transpose$_{jk}[\alpha_t]$
  $\delta_{t+1} := \varphi(y^{\alpha^{t+1}}) - \varphi(y^{\alpha^t}) < 0$
  $J_{t+1} := J_t \setminus \{j, k\}$ (move α_j and α_k only once)
  $t := t + 1$
until $|J_t| < 2$

$\hat{t} := \min \underset{t' \in \{0,\ldots,|A|\}}{\operatorname{argmin}} \sum_{\tau=0}^{t'} \delta_\tau$ (choose sub-sequence)

if $\sum_{\tau=0}^{\hat{t}} \delta_\tau < 0$
  $\alpha' :=$ greedy-transposition-kl$(\alpha^{\hat{t}})$ (recurse)
else
  $\alpha' := \alpha$ (terminate)

Summary.

- Learning and inferring orders on a finite set A is an unsupervised learning problem w.r.t. constrained data whose feasible labelings characterize the strict orders on A.
- The supervised learning problem can assume the form of l_2-regularized logistic regression where samples are pairs $(a, b) \in A^2$ such that $a \neq b$ and decisions indicate whether $a < b$.
- The inference problem assumes the form of the NP-hard linear ordering problem
- Local search algorithms for tackling this problem are greedy transposition and greedy transposition using the technique of Kernighan and Lin.