Machine Learning I

B. Andres, J. Irmai, J. Presberger, D. Stein, S. Zhao

Machine Learning for Computer Vision
TU Dresden

Winter Term 2023/2024

Contents.

- This part of the course is about the problem of **partitioning** a set into subsets, without knowing the number, size or any other property of the subsets.
- This problem is introduced as an **unsupervised learning** problem w.r.t. **constrained data**.

Let A be any finite set.

Definition. A **partition** Π of A is a collection $\Pi \subseteq 2^A$ of non-empty, pairwise disjoint subsets of A whose union is A.

Definition.

- An **equivalence relation** $\equiv$ on A is a binary relation $\equiv \subseteq A \times A$ that is reflexive, symmetric and transitive.
- For any partition Π of A, the equivalence relation $\equiv_\Pi$ induced by Π is such that

$$\forall a, a' \in A\colon \quad a \equiv_\Pi a' \;\Leftrightarrow\; \exists U \in \Pi\colon a \in U \wedge a' \in U \quad . \tag{1}$$

Lemma. This map from the set of all partitions of A to the set of all equivalence relations on A is a bijection.

Lemma. The equivalence relations on A are characterized by those decisions $y : \binom{A}{2} \to \{0,1\}$ that indicate, for any pair $\{a,b\} \in \binom{A}{2}$:

- by $y_{\{a,b\}} = 1$ that a and b are in the same subset
- by $y_{\{a,b\}} = 0$ that a and b in distinct subsets.

These are precisely those $y : \binom{A}{2} \to \{0,1\}$ that satisfy

$$\forall a \in A \; \forall b \in A \setminus \{a\} \; \forall c \in A \setminus \{a,b\} \colon \quad y_{\{a,b\}} + y_{\{b,c\}} - 1 \leq y_{\{a,c\}} \tag{2}$$

Constrained Data

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

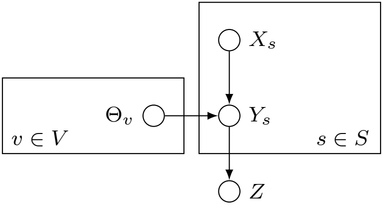
$$S = \binom{A}{2} \tag{3}$$

$$\mathcal{Y} = \left\{ y : \binom{A}{2} \to \{0,1\} \;\middle|\; \forall a \in A \; \forall b \in A \setminus \{a\} \; \forall c \in A \setminus \{a,b\} : \right.$$
$$\left. y_{\{a,b\}} + y_{\{b,c\}} - 1 \leq y_{\{a,c\}} \right\} \tag{4}$$

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

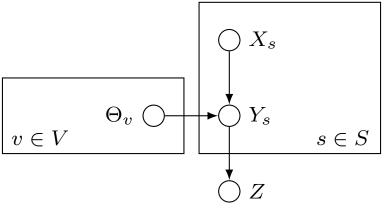
$$\forall \theta \in \Theta \; \forall \hat{x} \in \mathbb{R}^V : \quad f_\theta(\hat{x}) = \sum_{v \in V} \theta_v \, \hat{x}_v = \langle \theta, \hat{x} \rangle \; . \tag{5}$$



Random Variables

- For any $\{a,b\} = s \in S = \binom{A}{2}$, 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 $s \in S$, let Y_s be a random variable whose value is a binary number $y_s \in \{0,1\}$, called the **decision** of joining $\{a,b\} = s$.
- 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 partitioning, we are interested in $\mathcal{Z} = \mathcal{Y}$, the set characterizing equivalence relations on A.

Partitioning



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:

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

Partitioning

Factorization

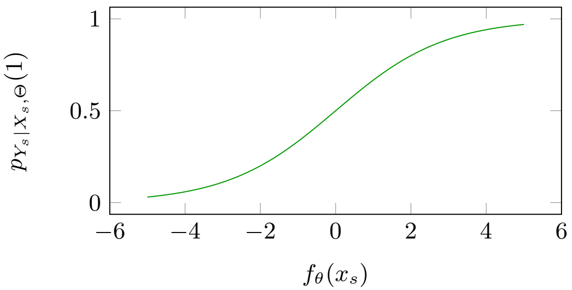
- Inference:

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

Distributions

- **Logistic distribution**

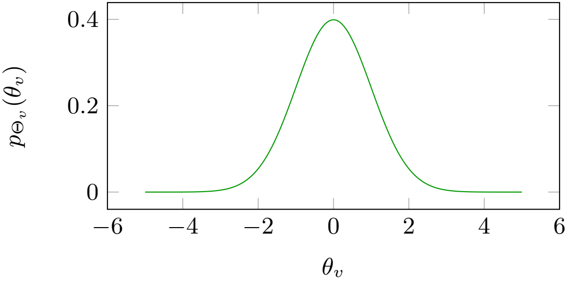
$$\forall s \in S: \qquad p_{Y_s \mid X_s, \Theta}(1) = \frac{1}{1 + 2^{-f_\theta(x_s)}} \tag{6}$$



Distributions

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

$$\forall v \in V : \qquad p_{\Theta_v}(\theta_v) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\theta_v^2/2\sigma^2} \tag{7}$$



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 equivalence relation 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:

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

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

$$\underset{y \in \{0,1\}^S}{\operatorname{argmax}} \quad p_{Y|X,Z,\Theta}(y, x, \mathcal{Y}, \theta) \tag{8}$$

assumes the form of the **set partition problem**

$$\begin{aligned} \underset{y:\, \binom{A}{2} \to \{0,1\}}{\operatorname{argmin}} \quad & \sum_{\{a,b\} \in S} \left(-\langle \theta, x_{\{a,b\}} \rangle\right) y_{\{a,b\}} & (9) \\ \text{subject to} \quad & \forall a \in A \; \forall b \in A \setminus \{a\} \; \forall c \in A \setminus \{a,b\}\colon \\ & \qquad y_{\{a,b\}} + y_{\{b,c\}} - 1 \leq y_{\{a,c\}} \quad . & (10) \end{aligned}$$

Theorem. The set partition problem is NP-hard.

It has been studied intensively, notably by Chopra and Rao (1993), Bansal et al. (2004) and Demaine et al. (2006).

We will discuss three **local search algorithms** for the set partition problem.

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

$$\forall \{a, a'\} \in S\colon \quad c_{\{aa'\}} = -\langle \theta, x_{\{a,a'\}} \rangle \tag{11}$$

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

$$\forall y \in \{0,1\}^S\colon \quad \varphi(y) = \sum_{\{a,a'\} \in S} c_{\{a,a'\}}\, y_{\{a,a'\}} \tag{12}$$

Greedy joining algorithm:

- The greedy joining algorithm is a local search algorithm that starts from any initial partition.
- It searches for partitions with lower cost by joining pairs of subsets recursively.
- As subsets can only grow and the number of subsets decreases by one in every step, one typically starts from the **finest partition** Π_0 of A into one-elementary subsets.

Definition. For any partition Π of A, and any $B, C \in \Pi$, let $\text{join}_{BC}[\Pi]$ be the partition of A obtained by **joining** the sets B and C in Π, i.e.:

$$\text{join}_{BC}[\Pi] = (\Pi \setminus \{B, C\}) \cup \{B \cup C\} \tag{13}$$

$\Pi' = \text{greedy-joining}(\Pi)$

choose $\{B, C\} \in \underset{\{B',C'\} \in \binom{\Pi}{2}}{\operatorname{argmin}} \varphi(y^{\text{join}_{B'C'}[\Pi]}) - \varphi(y^{\Pi})$

if $\varphi(y^{\text{join}_{BC}[\Pi]}) - \varphi(y^{\Pi}) < 0$

    $\Pi' := \text{greedy-joining}(\text{join}_{BC}[\Pi])$

else

    $\Pi' := \Pi$

Greedy moving algorithm:

- The greedy moving algorithm is a local search algorithm that starts from any initial partition, e.g., the fixed point of greedy joining.
- It searches for partitions with lower cost by recursively moving individual elements from one subset to another, or to a new subset.
- When an element is moved to a new subset, the number of subsets increases. When the last element is moved out of a subset, the number of subsets decreases.

Definition. For any partition Π of A, any $a \in A$ and any $U \in \Pi \cup \{\emptyset\}$, let $\mathsf{move}_{aU}[\Pi]$ the partition of A obtained by moving the element a to a subset $U \cup \{a\}$ in Π.

$$\begin{aligned}\mathsf{move}_{aU}[\Pi] = &\Pi \setminus \{U\} \setminus \{W \in \Pi \mid a \in W\} \\ &\cup \{U \cup \{a\}\} \cup \bigcup_{\{W \in \Pi \,\mid\, a \in W \wedge \{a\} \neq W\}} \{W \setminus \{a\}\} \ . \end{aligned} \tag{14}$$

$\Pi' = \text{greedy-moving}(\Pi)$

choose $(a, U) \in \underset{(a', U') \in A \times (\Pi \cup \{\emptyset\})}{\operatorname{argmin}} \varphi(y^{\mathsf{move}_{a'U'}[\Pi]}) - \varphi(y^{\Pi})$

if $\varphi(y^{\mathsf{move}_{aU}[\Pi]}) - \varphi(y^{\Pi}) < 0$

$\quad \Pi' := \text{greedy-moving}(\mathsf{move}_{aU}[\Pi])$

else

$\quad \Pi' := \Pi$

Greedy moving using the technique of Kernighan and Lin:

- Both algorithms discussed above terminate as soon as no transformation (join and move, resp.) leads to a partition with strictly lower cost.
- This can be sub-optimal in case transformations that increase the cost at one point in the recursion can lead to transformations that decrease the cost at later points in the recursion and the decrease overcompensates the increase.
- A generalization of local search introduced by Kernighan and Lin (1970) can escape such sub-optimal fixed points.
- Its application to greedy moving (next slide) builds a sequence of moves and then carries out the first t moves whose cumulative decrease in cost is optimal.

$\Pi' = \text{greedy-moving-kl}(\Pi)$

$\Pi_0 := \Pi$
$\delta_0 := 0$
$A_0 := A$
$t := 0$
repeat (build sequence of moves)
    choose $(a_t, U_t) \in \underset{(a,U) \in A_t \times (\Pi \cup \{\emptyset\})}{\operatorname{argmin}} \varphi(y^{\text{move}_{aU}[\Pi_t]}) - \varphi(y^{\Pi_t})$
    $\Pi_{t+1} := \text{move}_{a_t U_t}[\Pi_t]$
    $\delta_{t+1} := \varphi(y^{\Pi_{t+1}}) - \varphi(y^{\Pi_t}) < 0$
    $A_{t+1} := A_t \setminus \{a_t\}$ (move a_t only once)
    $t := t + 1$
until $A_t = \emptyset$

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

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

Summary.

- Learning and inferring partitions is an unsupervised learning problem w.r.t. constrained data whose feasible labelings characterize the equivalence relations on a set
- The supervised learning problem can assume the form of l_2-regularized logistic regression where samples are pairs of elements and decisions indicate whether these elements are in the same or distinct subsets
- The inference problem assumes the form of the NP-hard set partition problem
- Local search algorithms for tackling this problem are greedy joining, greedy moving, and greedy moving using the technique of Kernighan and Lin.