

Computer Vision I

Jannik Presberger, David Stein, Bjoern Andres

Machine Learning for Computer Vision
TU Dresden



<https://mlcv.cs.tu-dresden.de/courses/25-winter/cv1/>

Winter Term 2025/2026

Classification of digital images

Problem. Given a finite set V of pixels, the set $X = \mathbb{R}^V$ of images, a finite image collection $x : S \rightarrow X$ and binary decisions $y : S \rightarrow \{0, 1\}$, **find** a function $g : X \rightarrow \{0, 1\}$ to make such a decision for **any** image $x \in X$.

Example. Learning to identify precisely the images of the hand-written digit 7.

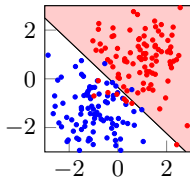
7	1	0	6	2	0	6	2
3	8	0	0	8	8	4	7
5	8	7	3	9	0	5	8
1	5	0	2	8	4	2	3
0	4	3	9	8	2	1	8
5	0	1	6	6	5	5	2
1	7	7	1	2	3	7	3
6	3	7	6	0	1	4	0

Logistic regression

To begin with, we consider **linear functions**. More specifically, we consider $\Theta = \mathbb{R}^V$ and $f : \Theta \rightarrow \mathbb{R}^X$ such that

$$\forall \theta \in \Theta \quad \forall \hat{x} \in X: \quad f_{\theta}(\hat{x}) = \langle \theta, \hat{x} \rangle = \sum_{v \in V} \theta_v \hat{x}_v \quad (1)$$

Example.



Logistic regression

We introduce a probabilistic model:

- ▶ For any sample $s \in S$, let X_s be a random variable whose value is a vector $x_s \in \mathbb{R}^V$, the **feature vector** of s
- ▶ For any sample $s \in S$, let Y_s be a random variable whose value is a binary number $y_s \in \{0, 1\}$, the **label** of 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 linear function we seek to learn

We assume that the joint probability factorizes according to:

$$P(X, Y, \Theta) = \prod_{s \in S} (P(Y_s \mid X_s, \Theta) P(X_s)) \prod_{v \in V} P(\Theta_v) \quad (2)$$

Logistic regression

We attempt to learn parameters by maximizing the conditional probability

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

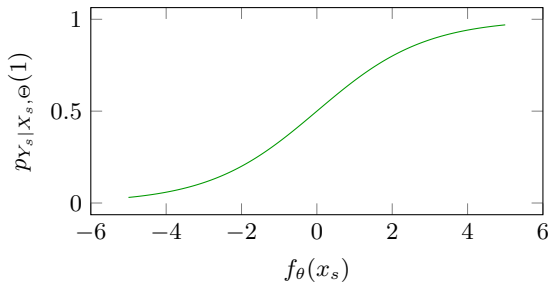
We attempt to infer labels by maximizing the conditional probability

$$P(Y \mid X, \Theta) = \prod_{s \in S} P(Y_s \mid X_s, \Theta) .$$

Logistic regression

► Sigmoid distribution

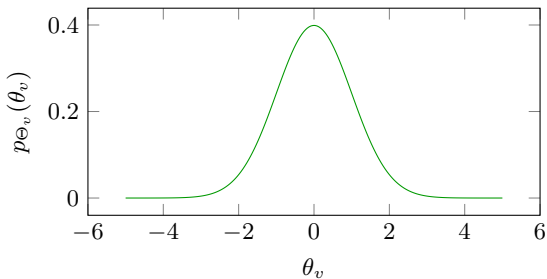
$$\forall s \in S : \quad p_{Y_s|X_s, \Theta}(1) = \frac{1}{1 + 2^{-f_{\theta}(x_s)}} \quad (3)$$



Logistic regression

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

$$\forall v \in V : \quad p_{\Theta_v}(\theta_v) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\theta_v^2/2\sigma^2} \quad (3)$$



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

$$\operatorname{argmax}_{\theta \in \mathbb{R}^m} p_{\Theta|X,Y}(\theta, x, y)$$

is equivalent to the optimization problem

$$\min_{\theta \in \Theta} \lambda R(\theta) + \sum_{s \in S} L(f_{\theta}(x_s), y_s) \quad (4)$$

with L , R and λ such that

$$\forall r \in \mathbb{R} \ \forall \hat{y} \in \{0, 1\}: \quad L(r, \hat{y}) = -\hat{y}r + \log(1 + 2^r) \quad (5)$$

$$\forall \theta \in \Theta: \quad R(\theta) = \|\theta\|_2^2 \quad (6)$$

$$\lambda = \frac{\log e}{2\sigma^2} . \quad (7)$$

It is called the l_2 -regularized **logistic regression problem** with respect to x , y and σ .

Logistic regression

Proof. Firstly,

$$\begin{aligned} & \operatorname{argmax}_{\theta \in \mathbb{R}^m} p_{\Theta|X,Y}(\theta, x, y) \\ &= \operatorname{argmax}_{\theta \in \mathbb{R}^m} \prod_{s \in S} p_{Y_s|X_s, \Theta}(y_s, x_s, \theta) \prod_{v \in V} p_{\Theta_v}(\theta_v) \\ &= \operatorname{argmax}_{\theta \in \mathbb{R}^m} \sum_{s \in S} \log p_{Y_s|X_s, \Theta}(y_s, x_s, \theta) + \sum_{v \in V} \log p_{\Theta_v}(\theta_v) \end{aligned} \quad (8)$$

Secondly,

$$\begin{aligned} & \log p_{Y_s|X_s, \Theta}(y_s, x_s, \theta) \\ &= y_s \log p_{Y_s|X_s, \Theta}(1, x_s, \theta) + (1 - y_s) \log p_{Y_s|X_s, \Theta}(0, x_s, \theta) \\ &= y_s \log \frac{p_{Y_s|X_s, \Theta}(1, x_s, \theta)}{p_{Y_s|X_s, \Theta}(0, x_s, \theta)} + \log p_{Y_s|X_s, \Theta}(0, x_s, \theta) \end{aligned} \quad (9)$$

Thus, with (3) and (4):

$$\operatorname{argmin}_{\theta \in \mathbb{R}^m} \sum_{s \in S} \left(-y_s \langle \theta, x_s \rangle + \log \left(1 + 2^{\langle \theta, x_s \rangle} \right) \right) + \frac{\log e}{2\sigma^2} \|\theta\|_2^2 \quad (10)$$

Lemma. The objective function

$$\varphi(\theta) = \lambda R(\theta) + \sum_{s \in S} L(f_{\theta}(x_s), y_s) \quad (11)$$

of the l_2 -regularized logistic regression problem is convex.

The l_2 -regularized logistic regression problem can be solved, e.g., by the steepest descent algorithm with a tolerance parameter $\epsilon \in \mathbb{R}_0^+$:

Algorithm. Steepest descent with line search

```
 $\theta := 0$ 
repeat
   $d := \nabla \varphi(\theta)$ 
   $\eta := \operatorname{argmin}_{\eta' \in \mathbb{R}} \varphi(\theta - \eta' d)$  (line search)
   $\theta := \theta - \eta d$ 
  if  $\|d\| < \epsilon$ 
    return  $\theta$ 
```

Lemma: Estimating maximally probable labels y , given attributes x' and parameters θ , i.e.,

$$\operatorname{argmax}_{y \in \{0,1\}^S} p_{Y|X,\Theta}(y, x', \theta) \quad (12)$$

is equivalent to the inference problem

$$\min_{y' \in \{0,1\}^S} \sum_{s \in S} L(f_\theta(x_s), y'_s) \quad (13)$$

It has the solution

$$\forall s \in S' : \quad y_s = \begin{cases} 1 & \text{if } f_\theta(x'_s) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

Logistic regression

Proof. Firstly,

$$\begin{aligned} & \operatorname{argmax}_{y \in \{0,1\}^{S'}} p_{Y|X,\Theta}(y, x', \theta) \\ &= \operatorname{argmax}_{y \in \{0,1\}^{S'}} \prod_{s \in S'} p_{Y_s|X_s,\Theta}(y_s, x'_s, \theta) \\ &= \operatorname{argmax}_{y \in \{0,1\}^{S'}} \sum_{s \in S'} \log p_{Y_s|X_s,\Theta}(y_s, x'_s, \theta) \\ &= \operatorname{argmax}_{y \in \{0,1\}^{S'}} \sum_{s \in S'} \left(y_s \log \frac{p_{Y_s|X_s,\Theta}(1, x'_s, \theta)}{p_{Y_s|X_s,\Theta}(0, x'_s, \theta)} + \log p_{Y_s|X_s,\Theta}(0, x'_s, \theta) \right) \\ &= \operatorname{argmin}_{y \in \{0,1\}^{S'}} \sum_{s \in S'} \left(-y_s f_\theta(x'_s) + \log \left(1 + 2^{f_\theta(x'_s)} \right) \right) \\ &= \operatorname{argmin}_{y \in \{0,1\}^{S'}} \sum_{s \in S'} L(f_\theta(x'_s), y_s) . \end{aligned}$$

Secondly,

$$\min_{y \in \{0,1\}^{S'}} \sum_{s \in S'} \left(-y_s f_\theta(x'_s) + \log \left(1 + 2^{f_\theta(x'_s)} \right) \right) = \sum_{s \in S'} \max_{y_s \in \{0,1\}} y_s f_\theta(x'_s) .$$

Notation. Let $G = (V, E)$ a digraph.

- For any $v \in V$, let

$$P_v = \{u \in V \mid (u, v) \in E\} \quad \text{the set of **parents** of } v \quad (15)$$

$$C_v = \{w \in V \mid (v, w) \in E\} \quad \text{the set of **children** of } v. \quad (16)$$

- For any $u, v \in V$, let $\mathcal{P}(u, v)$ denote the set of all uv -paths. (Any path is a subgraph. For any node u , the uu -path $(\{u\}, \emptyset)$ exists.)

Let G be **acyclic**.

- For any $v \in V$, let

$$A_v = \{u \in V \mid \mathcal{P}(u, v) \neq \emptyset\} \setminus \{v\} \quad \text{the set of **ancestors** of } v \quad (17)$$

$$D_v = \{w \in V \mid \mathcal{P}(v, w) \neq \emptyset\} \setminus \{v\} \quad \text{the set of **descendants** of } v. \quad (18)$$

Definition. A tuple $(V, D, D', E, \Theta, \{g_{v\theta} : \mathbb{R}^{P_v} \rightarrow \mathbb{R}\}_{v \in (D \cup D') \setminus V, \theta \in \Theta})$ is called a **compute graph**, iff the following conditions hold:

- ▶ $G = (V \cup D \cup D', E)$ is an acyclic digraph
- ▶ $\forall v \in V : P_v = \emptyset$
- ▶ $\forall v \in D' : C_v = \emptyset$
- ▶ $\forall v \in D : P_v \neq \emptyset$ and $C_v \neq \emptyset$

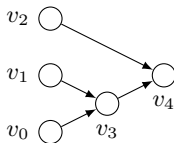
Definition. For any compute graph

$(V, D, D', E, \Theta, \{g_{v\theta} : \mathbb{R}^{P_v} \rightarrow \mathbb{R}\}_{v \in (D \cup D') \setminus V, \theta \in \Theta})$, any $v \in V \cup D \cup D'$ and any $\theta \in \Theta$, let $\alpha_{v\theta} : \mathbb{R}^V \rightarrow \mathbb{R}$ such that for all $\hat{x} \in \mathbb{R}^V$:

$$\alpha_{v\theta}(\hat{x}) = \begin{cases} \hat{x}_v & \text{if } v \in V \\ g_{v\theta}(\alpha_{P_v\theta}(\hat{x})) & \text{otherwise} \end{cases} . \quad (19)$$

We call $\alpha_{v\theta}(\hat{x})$ the **activation** of v for **input** $\hat{x}$ and **parameters** θ . For any $\theta \in \Theta$ let $f_\theta : \mathbb{R}^V \rightarrow \mathbb{R}^{D'}$ such that $f_\theta = \alpha_{D'\theta}$. We call $f_\theta(\hat{x})$ the **output** of the compute graph for input $\hat{x}$ and parameters θ .

Example. Consider the compute graph below with $V = \{v_0, v_1, v_2\}$, $D = \{v_3\}$ and $D' = \{v_4\}$.



Moreover, consider $\Theta = \{\theta_0, \theta_1\}$ and

- ▶ $g_{v_3\theta}: \mathbb{R}^{\{v_0, v_1\}} \rightarrow \mathbb{R}$ such that $g_{v_3\theta}(x) = x_{v_0} + \theta_0 x_{v_1}$
- ▶ $g_{v_4\theta}: \mathbb{R}^{\{v_2, v_3\}} \rightarrow \mathbb{R}$ such that $g_{v_4\theta}(x) = x_{v_2} + x_{v_3}^{\theta_1}$

This defines the function $f_\theta(x) = x_{v_2} + (x_{v_0} + \theta_0 x_{v_1})^{\theta_1}$.

In the following:

- ▶ We assume $\Theta = \mathbb{R}^J$ for some set J .
- ▶ We consider compute graphs with $|D'| = 1$, i.e. $f_{\theta}(\hat{x}) \in \mathbb{R}$ for every $\hat{x} \in \mathbb{R}^V$.

Problem: The l_2 -regularized non-linear logistic regression problem with respect to labeled data $T = (S, \mathbb{R}^V, x, y)$ and $\sigma \in \mathbb{R}^+$ is to solve

$$\operatorname{argmin}_{\theta \in \mathbb{R}^J} \sum_{s \in S} \left(-y_s f_{\theta}(x_s) + \log \left(1 + 2^{f_{\theta}(x)} \right) \right) + \frac{\log e}{2\sigma^2} \|\theta\|^2 . \quad (20)$$

Remark.

- ▶ (20) is a generalization of linear logistic regression.
- ▶ (20) can be non-convex for f_{θ} non-linear in θ .
- ▶ A local minimum $\hat{\theta} \in \mathbb{R}^J$ can be found by means of a steepest descent algorithm.
- ▶ In order to compute $\nabla_{\theta} f_{\theta}$, we describe the **backward propagation algorithm**.

Deep Learning

Lemma. Let $j \in J$. For any $v \in V$: $\frac{\partial \alpha_{v\theta}}{\partial \theta_j} = 0$. For any $v \in (D \cup D') \setminus V$:

$$\frac{\partial \alpha_{v\theta}}{\partial \theta_j} = \sum_{u \in (A_v \cup \{v\}) \setminus V} \frac{\partial g_{u\theta}}{\partial \theta_j} \Delta_{uv} \quad (21)$$

with

$$\Delta_{uv} := \sum_{(V', E') \in \mathcal{P}(u, v)} \prod_{(u', v') \in E'} \frac{\partial g_{v'\theta}}{\partial \alpha_{u'\theta}}. \quad (22)$$

Remark. For any node u : $\Delta_{uu} = 1$. For any u, v with $\mathcal{P}(u, v) = \emptyset$: $\Delta_{uv} = 0$.

Proof (idea).

$$\begin{aligned} \frac{\partial \alpha_{v\theta}}{\partial \theta_j} &= \frac{\partial g_{v\theta}}{\partial \theta_j} + \sum_{u \in P_v} \frac{\partial g_{v\theta}}{\partial \alpha_{u\theta}} \frac{\partial \alpha_{u\theta}}{\partial \theta_j} \\ &= \frac{\partial g_{v\theta}}{\partial \theta_j} + \sum_{u \in P_v} \frac{\partial g_{v\theta}}{\partial \alpha_{u\theta}} \frac{\partial g_{u\theta}}{\partial \theta_j} + \sum_{u \in P_v} \sum_{u' \in P_u} \frac{\partial g_{v\theta}}{\partial \alpha_{u\theta}} \frac{\partial g_{u\theta}}{\partial \alpha_{u'\theta}} \frac{\partial \alpha_{u'\theta}}{\partial \theta_j} \\ &= \text{repeated application (23)} \\ &= \sum_{u \in (A_v \cup \{v\}) \setminus V} \frac{\partial g_{u\theta}}{\partial \theta_j} \sum_{(V', E') \in \mathcal{P}(u, v)} \prod_{(u', v') \in E'} \frac{\partial g_{v'\theta}}{\partial \alpha_{u'\theta}} \end{aligned} \quad (23)$$

Lemma (backward propagation). For all nodes $u \neq w$ such that $\mathcal{P}(u, w) \neq \emptyset$:

$$\Delta_{uw} = \sum_{v \in C_u} \frac{\partial g_{v\theta}}{\partial \alpha_{u\theta}} \Delta_{vw} \quad (24)$$

Proof.

$$\begin{aligned} \Delta_{uw} &= \sum_{(V', E') \in \mathcal{P}(u, w)} \prod_{(u', v') \in E'} \frac{\partial g_{v'\theta}}{\partial \alpha_{u'\theta}} \\ &= \sum_{v \in C_u} \sum_{(V'', E'') \in \mathcal{P}(v, w)} \prod_{(u', v') \in E'' \cup \{(u, v)\}} \frac{\partial g_{v'\theta}}{\partial \alpha_{u'\theta}} \\ &= \sum_{v \in C_u} \frac{\partial g_{v\theta}}{\partial \alpha_{u\theta}} \sum_{(V'', E'') \in \mathcal{P}(v, w)} \prod_{(u', v') \in E''} \frac{\partial g_{v'\theta}}{\partial \alpha_{u'\theta}} \\ &= \sum_{v \in C_u} \frac{\partial g_{v\theta}}{\partial \alpha_{u\theta}} \Delta_{vw} \end{aligned}$$

□

The **backward propagation algorithm** computes Δ_{uw} for one node w and all nodes u . It is defined wrt. an arbitrary partial order $<_C$ of the nodes such that

$$\forall u \in V \cup D \quad \forall v \in C_u: \quad v <_C u . \quad (25)$$

Input:

Compute graph $(V, D, D', E, \Theta, \{g_{v\theta}: \mathbb{R}^{P_v} \rightarrow \mathbb{R}\}_{v \in (D \cup D') \setminus V, \theta \in \Theta})$

Node $w \in V \cup D \cup D'$

for u ordered by $<_C$ (25)

if $u = w$

$\Delta_{uw} := 1$

else if $\mathcal{P}(u, w) = \emptyset$

$\Delta_{uw} := 0$

else

$\Delta_{uw} := \sum_{v \in C_u} \frac{\partial g_{v\theta}}{\partial \alpha_{u\theta}} \Delta_{vw}$ (24)
