# Computer Vision I

Jannik Irmai, Jannik Presberger, Bjoern Andres

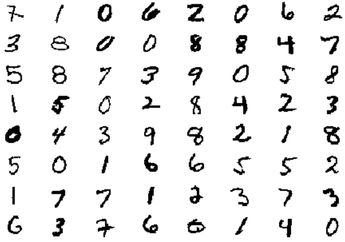
Machine Learning for Computer Vision
TU Dresden

https://mlcv.cs.tu-dresden.de/courses/24-winter/cv1/

Winter Term 2024/2025

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

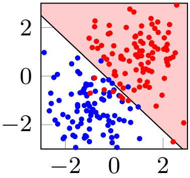
**Example.** Learning to identify precisely the images of the hand-written digit 7.



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

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

**Example.**



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 $\Theta_v$ 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:

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

We attempt to learn parameters by maximizing the conditional probability

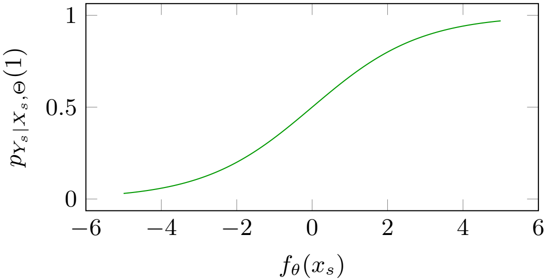
$$\begin{aligned} P(\Theta \mid X, Y) &= \frac{P(X, Y, \Theta)}{P(X, Y)} \\ &= \frac{P(Y \mid X, \Theta)\, P(X)\, P(\Theta)}{P(X, Y)} \\ &\propto P(Y \mid X, \Theta)\, P(\Theta) \\ &= \prod_{s \in S} P(Y_s \mid X_s, \Theta) \prod_{v \in V} P(\Theta_v) \ . \end{aligned}$$

We attempt to infer labels by maximizing the conditional probability

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

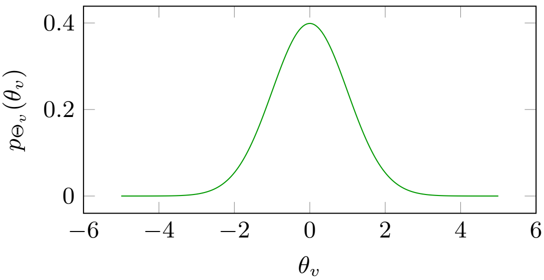
- **Sigmoid distribution**

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



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

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



**Lemma.** Estimating maximally probable parameters $\theta$, given attributes $x$ and labels $y$, i.e.,

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

is equivalent to the optimization problem

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

with $L$, $R$ and $\lambda$ such that

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

$$\forall \theta \in \Theta\colon \quad R(\theta) = \|\theta\|_2^2 \tag{6}$$

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

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

*Proof.* Firstly,

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

Secondly,

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

Thus, with (3) and (4):

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

**Lemma.** The objective function

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

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 $\theta$, i.e.,

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

is equivalent to the inference problem

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

It has the solution

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

*Proof.* Firstly,

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

Secondly,

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

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

- For any $v \in V$, let

$$P_v = \{u \in V \mid (u, v) \in E\} \qquad \text{the set of } \textbf{parents} \text{ of } v \tag{15}$$
$$C_v = \{w \in V \mid (v, w) \in E\} \qquad \text{the set of } \textbf{children} \text{ of } v \;. \tag{16}$$

- 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

$$A_v = \{u \in V \mid \mathcal{P}(u, v) \neq \emptyset\} \setminus \{v\} \qquad \text{the set of } \textbf{ancestors} \text{ of } v \tag{17}$$
$$D_v = \{w \in V \mid \mathcal{P}(v, w) \neq \emptyset\} \setminus \{v\} \qquad \text{the set of } \textbf{descendants} \text{ of } v \;. \tag{18}$$

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

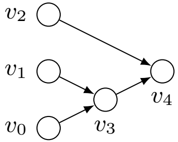
- $G = (V \cup D \cup D', E)$ is an acyclic digraph
- $\forall v \in V : P_v = \emptyset$
- $\forall v \in D' : C_v = \emptyset$
- $\forall v \in D : P_v \neq \emptyset$ and $C_v \neq \emptyset$

**Definition.** For any compute graph $(V, D, D', E, \Theta, \{g_{v\theta} \colon \mathbb{R}^{P_v} \to \mathbb{R}\}_{v \in (D \cup D') \setminus V, \theta \in \Theta})$, any $v \in V \cup D \cup D'$ and any $\theta \in \Theta$, let $\alpha_{v\theta} \colon \mathbb{R}^V \to \mathbb{R}$ such that for all $\hat{x} \in \mathbb{R}^V$:

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

We call $\alpha_{v\theta}(\hat{x})$ the **activation** of $v$ for **input** $\hat{x}$ and **parameters** $\theta$. For any $\theta \in \Theta$ let $f_\theta \colon \mathbb{R}^V \to \mathbb{R}^{D'}$ 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 $\theta$.

**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

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

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

**Remark.**

- (20) is a generalization of linear logistic regression.
- (20) can be non-convex for $f_\theta$ non-linear in $\theta$.
- 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**.

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

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

with

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

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

Proof (idea).

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

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

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

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 $\Delta_{uw}$ for one node $w$ and all nodes $u$. It is defined wrt. an arbitrary partial order $<_C$ of the nodes such that

$$\forall u \in V \cup D \quad \forall v \in C_u : \quad v <_C u \ . \tag{25}$$

---

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

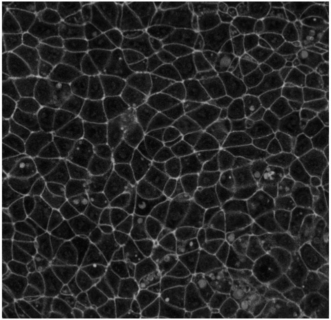
---

**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)

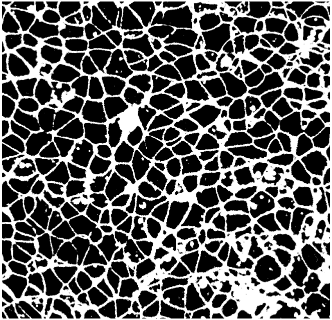
---

# Pixel classification

Digital image[1] $f\colon V \to C$



Classification $y\colon V \to \{0,1\}$



[1]By courtesy of Stephan Grill and his lab at the MPI of Molecular Cell Biology and Genetics.

**Definition.** Let $G = (V, E)$ a pixel grid graph and $g\colon V \to C$ a digital image. Let $m \in \mathbb{N}$ and $X = \mathbb{R}^m$ (a **feature space**). For any pixel $v \in V$, let $x_v^{(g)} \in X$ (a **feature vector** associated with the pixel $v$ of the digital image $g$). Let $f\colon X \to \mathbb{R}$ (e.g. a linear function learned by logistic regression).

The instance of the **trivial pixel classification problem** has the form

$$\min_{y \in \{0,1\}^V} \quad \sum_{v \in V} (-f(x_v))\, y_v \tag{26}$$

With the pixel grid graph $(V, E)$ and $c'\colon E \to \mathbb{R}_0^+$, the instance of the **smooth pixel classification problem** has the form

$$\min_{y \in \{0,1\}^V} \quad \underbrace{\sum_{v \in V} (-f(x_v))\, y_v + \sum_{\{v,w\} \in E} c'_{\{v,w\}}\, |y_v - y_w|}_{\varphi(y)} \tag{27}$$

**Remark.** Motivation: Prior knowledge that decisions at neighboring pixels $v, w$ are more likely to be equal $(y_v = v_w)$ than unequal $(y_v \neq y_w)$.

A naïve algorithm for the smooth pixel classification problem is **local search** with a transformation $T_v \colon \{0,1\}^V \to \{0,1\}^V$ that changes the decision for a single pixel, i.e., for any $y \colon V \to \{0,1\}$ and any $v, w \in V$:

$$T_v(y)(w) = \begin{cases} 1 - y_w & \text{if } w = v \\ y_w & \text{otherwise} \end{cases} .$$

**Algorithm.**
Initially, $y \colon V \to \{0,1\}$ and $W = V$
while $W \neq \emptyset$
    $W' := \emptyset$
    for each $v \in W$
        if $\varphi(T_v(y)) - \varphi(y) < 0$
            $y := T_v(y)$
            $W' := W' \cup \{w \in V \mid \{v, w\} \in E\}$
    $W := W'$

**Remark.**

- On the one hand, this algorithm is easy to implement and has straight-forward generalizations, e.g., to the case of more than two classes.
- On the other hand, it does not necessarily solve smooth pixel classification with two classes to optimality.
- Next, we will reduce the smooth pixel classification problem with two classes to the well-known **minimum** $st$-**cut problem** that can be solved exactly and efficiently.

**Definition.** A 5-tuple $N = (V, E, s, t, \gamma)$ is called a **network** iff $(V, E)$ is a directed graph and $s \in V$ and $t \in V$ and $s \neq t$ and $\gamma : E \to \mathbb{R}_0^+$. The nodes $s$ and $t$ are called the **source** and the **sink** of $N$, respectively. For any edge $e \in E$, $\gamma_e$ is called the **capacity** of $e$ in $N$.

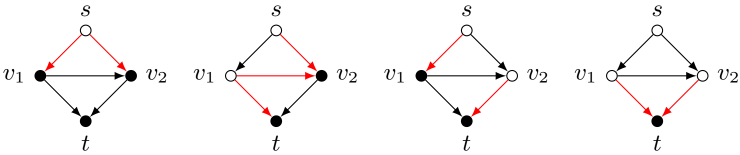
**Definition.** The instance of the **minimum** $st$**-cut problem** wrt. a network $N = (V, E, s, t, \gamma)$ has the form

$$\min_{x \in \{0,1\}^V} \quad \sum_{vw \in E} \gamma_{vw} \, (1 - x_v) \, x_w \tag{28}$$

$$\text{subject to} \quad x_s = 0 \tag{29}$$

$$x_t = 1 \tag{30}$$

**Example.**



**Lemma.** The smooth pixel classification problem is reducible to the minimum $st$-cut problem.

*Proof (sketch).* For any instance of the smooth pixel classification problem,

$$\min_{y \in \{0,1\}^V} \quad \underbrace{\sum_{v \in V} c_v \, y_v + \sum_{\{v,w\} \in E} c'_{\{v,w\}} \, \left( y_v (1 - y_w) + (1 - y_v) y_w \right)}_{\varphi(y)} \;, \tag{31}$$

define the instance of the induced minimum $st$-cut problem in terms of the network $(V', E', s, t, \gamma)$ such that

$$V' = V \cup \{s, t\} \tag{32}$$

$$E' = \{(s,v) \in V'^2 \mid c_v > 0\} \cup \{(v,t) \in V'^2 \mid c_v < 0\} \cup \{(v,w) \in V'^2 \mid \{v,w\} \in E\} \tag{33}$$

and $\gamma \colon E' \to \mathbb{R}_0^+$ such that

$$\forall (s,v) \in E' \colon \quad \gamma_{(s,v)} = c_v \tag{34}$$

$$\forall (v,t) \in E' \colon \quad \gamma_{(v,t)} = -c_v \tag{35}$$

$$\forall \{v,w\} \in E \colon \quad \gamma_{(v,w)} = \gamma_{(w,v)} = c'_{\{v,w\}} \;. \tag{36}$$