Machine Learning I

Lucas Fabian Naumann, David Stein, Bjoern Andres

Machine Learning for Computer Vision
TU Dresden

https://mlcv.cs.tu-dresden.de/courses/25-winter/ml1/

Winter Term 2025/2026

Contents. This part of the course introduces algorithms for supervised structured learning of conditional graphical models.

On the one hand, supervised structured learning of conditional graphical models whose factors are linear functions is a **convex** optimization problem.

Thus, it can be solved exactly by means of the **steepest descent algorithm** with a tolerance parameter $\epsilon \in \mathbb{R}_0^+$:

$\theta := 0$
repeat
  $d := \nabla_\theta L(H_\theta(x, \cdot), y)$
  $\eta := \operatorname{argmin}_{\eta' \in \mathbb{R}} L(H_{\theta - \eta' d}(x, \cdot), y)$ (line search)
  $\theta := \theta - \eta d$
  if $\|d\| < \epsilon$
    return θ

On the other hand, computing the gradient naïvely takes time $O(2^{|S|})$:

On the other hand, computing the gradient naïvely takes time $O(2^{|S|})$:

$$-\frac{\partial}{\partial \theta_j} \ln Z = \mathbb{E}_{y' \sim p_{\mathcal{Y}|\mathcal{X},\Theta}}(\xi_j(x, y'))$$

On the other hand, computing the gradient naïvely takes time $O(2^{|S|})$:

$$
\begin{aligned}
-\frac{\partial}{\partial\theta_j}\ln Z &= \mathbb{E}_{y'\sim p_{\mathcal{Y}|\mathcal{X},\Theta}}(\xi_j(x,y')) \\
&= \frac{1}{Z(x,\theta)}\sum_{y'\in\{0,1\}^S}\xi_j(x,y')\,e^{-\langle\theta,\xi(x,y')\rangle}
\end{aligned}
$$

On the other hand, computing the gradient naïvely takes time $O(2^{|S|})$:

$$
\begin{aligned}
-\frac{\partial}{\partial \theta_j} \ln Z &= \mathbb{E}_{y' \sim p_{\mathcal{Y}|\mathcal{X},\Theta}}(\xi_j(x, y')) \\
&= \frac{1}{Z(x,\theta)} \sum_{y' \in \{0,1\}^S} \xi_j(x, y')\, e^{-\langle \theta, \xi(x,y') \rangle} \\
&= \frac{1}{Z(x,\theta)} \sum_{y' \in \{0,1\}^S} \sum_{f \in F} \varphi_{fj}(x_f, y'_{S_f})\, e^{-\langle \theta, \xi(x,y') \rangle}
\end{aligned}
$$

On the other hand, computing the gradient naïvely takes time $O(2^{|S|})$:

$$
\begin{aligned}
-\frac{\partial}{\partial \theta_j} \ln Z &= \mathbb{E}_{y' \sim p_{\mathcal{Y}|\mathcal{X},\Theta}}(\xi_j(x, y')) \\
&= \frac{1}{Z(x,\theta)} \sum_{y' \in \{0,1\}^S} \xi_j(x, y')\, e^{-\langle \theta, \xi(x,y') \rangle} \\
&= \frac{1}{Z(x,\theta)} \sum_{y' \in \{0,1\}^S} \sum_{f \in F} \varphi_{fj}(x_f, y'_{S_f})\, e^{-\langle \theta, \xi(x,y') \rangle} \\
&= \frac{1}{Z(x,\theta)} \sum_{f \in F} \sum_{y'_{S(f)} \in \{0,1\}^{S(f)}} \sum_{y'_{S \setminus S(f)} \in \{0,1\}^{S \setminus S(f)}} \varphi_{fj}(x_f, y'_{S(f)})\, e^{-\langle \theta, \xi(x,y') \rangle}
\end{aligned}
$$

On the other hand, computing the gradient naïvely takes time $O(2^{|S|})$:

$$
\begin{aligned}
-\frac{\partial}{\partial \theta_j} \ln Z &= \mathbb{E}_{y' \sim p_{\mathcal{Y}|\mathcal{X},\Theta}}(\xi_j(x, y')) \\
&= \frac{1}{Z(x,\theta)} \sum_{y' \in \{0,1\}^S} \xi_j(x, y')\, e^{-\langle \theta, \xi(x,y') \rangle} \\
&= \frac{1}{Z(x,\theta)} \sum_{y' \in \{0,1\}^S} \sum_{f \in F} \varphi_{fj}(x_f, y'_{S_f})\, e^{-\langle \theta, \xi(x,y') \rangle} \\
&= \frac{1}{Z(x,\theta)} \sum_{f \in F} \sum_{y'_{S(f)} \in \{0,1\}^{S(f)}} \sum_{y'_{S \setminus S(f)} \in \{0,1\}^{S \setminus S(f)}} \varphi_{fj}(x_f, y'_{S(f)})\, e^{-\langle \theta, \xi(x,y') \rangle} \\
&= \sum_{f \in F} \sum_{y'_{S(f)} \in \{0,1\}^{S(f)}} \varphi_{fj}(x_f, y'_{S(f)}) \frac{1}{Z(x,\theta)} \sum_{y'_{S \setminus S(f)} \in \{0,1\}^{S \setminus S(f)}} e^{-\langle \theta, \xi(x,y') \rangle}
\end{aligned}
$$

On the other hand, computing the gradient naïvely takes time $O(2^{|S|})$:

$$
\begin{aligned}
-\frac{\partial}{\partial\theta_j}\ln Z &= \mathbb{E}_{y'\sim p_{\mathcal{Y}|\mathcal{X},\Theta}}(\xi_j(x,y')) \\
&= \frac{1}{Z(x,\theta)}\sum_{y'\in\{0,1\}^S}\xi_j(x,y')\,e^{-\langle\theta,\xi(x,y')\rangle} \\
&= \frac{1}{Z(x,\theta)}\sum_{y'\in\{0,1\}^S}\sum_{f\in F}\varphi_{fj}(x_f,y'_{S_f})\,e^{-\langle\theta,\xi(x,y')\rangle} \\
&= \frac{1}{Z(x,\theta)}\sum_{f\in F}\sum_{y'_{S(f)}\in\{0,1\}^{S(f)}}\sum_{y'_{S\setminus S(f)}\in\{0,1\}^{S\setminus S(f)}}\varphi_{fj}(x_f,y'_{S(f)})\,e^{-\langle\theta,\xi(x,y')\rangle} \\
&= \sum_{f\in F}\sum_{y'_{S(f)}\in\{0,1\}^{S(f)}}\varphi_{fj}(x_f,y'_{S(f)})\frac{1}{Z(x,\theta)}\sum_{y'_{S\setminus S(f)}\in\{0,1\}^{S\setminus S(f)}}e^{-\langle\theta,\xi(x,y')\rangle} \\
&= \sum_{f\in F}\sum_{y'_{S(f)}\in\{0,1\}^{S(f)}}\varphi_{fj}(x_f,y'_{S(f)})\,p_{\mathcal{Y}_{S(f)}|\mathcal{X},\Theta}(y'_{S(f)}\mid x,\theta)
\end{aligned}
$$

On the other hand, computing the gradient naïvely takes time $O(2^{|S|})$:

$$
\begin{aligned}
-\frac{\partial}{\partial\theta_j}\ln Z &= \mathbb{E}_{y'\sim p_{\mathcal{Y}|\mathcal{X},\Theta}}(\xi_j(x,y')) \\
&= \frac{1}{Z(x,\theta)} \sum_{y'\in\{0,1\}^S} \xi_j(x,y')\, e^{-\langle\theta,\xi(x,y')\rangle} \\
&= \frac{1}{Z(x,\theta)} \sum_{y'\in\{0,1\}^S} \sum_{f\in F} \varphi_{fj}(x_f, y'_{S_f})\, e^{-\langle\theta,\xi(x,y')\rangle} \\
&= \frac{1}{Z(x,\theta)} \sum_{f\in F} \sum_{y'_{S(f)}\in\{0,1\}^{S(f)}} \sum_{y'_{S\setminus S(f)}\in\{0,1\}^{S\setminus S(f)}} \varphi_{fj}(x_f, y'_{S(f)})\, e^{-\langle\theta,\xi(x,y')\rangle} \\
&= \sum_{f\in F} \sum_{y'_{S(f)}\in\{0,1\}^{S(f)}} \varphi_{fj}(x_f, y'_{S(f)}) \frac{1}{Z(x,\theta)} \sum_{y'_{S\setminus S(f)}\in\{0,1\}^{S\setminus S(f)}} e^{-\langle\theta,\xi(x,y')\rangle} \\
&= \sum_{f\in F} \sum_{y'_{S(f)}\in\{0,1\}^{S(f)}} \varphi_{fj}(x_f, y'_{S(f)})\, p_{\mathcal{Y}_{S(f)}|\mathcal{X},\Theta}(y'_{S(f)} \mid x,\theta) \\
&= \sum_{f\in F} \mathbb{E}_{y'_{S(f)}\sim p_{\mathcal{Y}_{S(f)}|\mathcal{X},\Theta}}(\varphi_{fj}(x_f, y'_{S(f)}))
\end{aligned}
$$

Computing the gradient requires that we compute

- the partition function

$$Z(x, \theta) = \sum_{y' \in \{0,1\}^S} e^{-\langle \theta, \xi(x, y') \rangle} \tag{1}$$

Computing the gradient requires that we compute

- the partition function

$$Z(x,\theta) = \sum_{y' \in \{0,1\}^S} e^{-\langle \theta, \xi(x,y') \rangle} \tag{1}$$

- for every factor $f \in F$, the so-called **factor marginal**

$$p_{\mathcal{Y}_{S(f)}|\mathcal{X},\Theta}(y'_{S(f)} \mid x,\theta) = \frac{1}{Z(x,\theta)} \sum_{y'_{S \setminus S(f)} \in \{0,1\}^{S \setminus S(f)}} e^{-\langle \theta, \xi(x,y') \rangle} \tag{2}$$

Computing the gradient requires that we compute

- the partition function

$$Z(x,\theta) = \sum_{y' \in \{0,1\}^S} e^{-\langle \theta, \xi(x,y') \rangle} \tag{1}$$

- for every factor $f \in F$, the so-called **factor marginal**

$$p_{\mathcal{Y}_{S(f)} \mid \mathcal{X},\Theta}(y'_{S(f)} \mid x,\theta) = \frac{1}{Z(x,\theta)} \sum_{y'_{S \setminus S(f)} \in \{0,1\}^{S \setminus S(f)}} e^{-\langle \theta, \xi(x,y') \rangle} \tag{2}$$

- for every factor $f \in F$, the expectation value

$$\sum_{y'_{S(f)} \in \{0,1\}^{S(f)}} \varphi_{fj}(x_f, y'_{S(f)}) \, p_{\mathcal{Y}_{S(f)} \mid \mathcal{X},\Theta}(y'_{S(f)} \mid x,\theta) \ . \tag{3}$$

Computing the gradient requires that we compute

- the partition function

$$Z(x,\theta) = \sum_{y' \in \{0,1\}^S} e^{-\langle \theta, \xi(x,y') \rangle} \tag{1}$$

- for every factor $f \in F$, the so-called **factor marginal**

$$p_{\mathcal{Y}_{S(f)}|\mathcal{X},\Theta}(y'_{S(f)} \mid x, \theta) = \frac{1}{Z(x,\theta)} \sum_{y'_{S \setminus S(f)} \in \{0,1\}^{S \setminus S(f)}} e^{-\langle \theta, \xi(x,y') \rangle} \tag{2}$$

- for every factor $f \in F$, the expectation value

$$\sum_{y'_{S(f)} \in \{0,1\}^{S(f)}} \varphi_{fj}(x_f, y'_{S(f)})\, p_{\mathcal{Y}_{S(f)}|\mathcal{X},\Theta}(y'_{S(f)} \mid x, \theta) \ . \tag{3}$$

The challenge is to sum the function

$$\psi_\theta(x, y') := e^{-\langle \theta, \xi(x,y') \rangle} \tag{4}$$

over assignments of 0 or 1 to linearly many (2) or all (1) variables y'.

Defining

$$\psi_{f\theta}(x_f, y'_{S(f)}) = e^{-\langle \theta, \varphi_f(x_f, y'_{S(f)}) \rangle} \tag{5}$$

we obtain

$$\begin{aligned}
\psi_\theta(x, y') =& e^{-\langle \theta, \xi(x,y') \rangle} \\
=& e^{-\sum_{f \in F} \langle \theta, \varphi_f(x_f, y_{S(f)}) \rangle} && (6) \\
=& \prod_{f \in F} e^{-\langle \theta, \varphi_f(x_f, y_{S(f)}) \rangle} && (7) \\
=& \prod_{f \in F} \psi_{f\theta}(x_f, y_{S(f)}) \ . && (8)
\end{aligned}$$

Thus, the challenge in (2) and (1) is to compute a sum of a product of functions. Specifically:

$$Z(x,\theta) = \sum_{y' \in \{0,1\}^S} \prod_{f \in F} \psi_{f\theta}(x_f, y_{S(f)}) \tag{9}$$

$$p_{\mathcal{Y}_{S(f)}|\mathcal{X},\Theta}(y'_{S(f)} \mid x,\theta) = \frac{1}{Z(x,\theta)} \sum_{y'_{S \setminus S(f)} \in \{0,1\}^{S \setminus S(f)}} \prod_{f \in F} \psi_{f\theta}(x_f, y_{S(f)}) \tag{10}$$

- One approach to tackle this problem is to sum over variables recursively.
- In order to avoid redundant computation, Kschischang et al. (2001) define partial sums.

Definition (Kschischang et al. (2001)) For any variable node $s \in S$ and any factor node $f \in F$, the functions

$$m_{s \to f}, m_{f \to s} \colon \{0, 1\} \to \mathbb{R} \ , \tag{11}$$

called **messages**, are defined such that for all $y_s \in \{0, 1\}$:

$$m_{s \to f}(y_s) = \prod_{f' \in F(s) \setminus \{f\}} m_{f' \to s}(y_s) \tag{12}$$

$$m_{f \to s}(y_s) = \sum_{y_{S(f) \setminus \{s\}}} \psi_{f\theta}(x_f, y_{S(f)}) \prod_{s' \in S(f) \setminus \{s\}} m_{s' \to f}(y_{s'}) \tag{13}$$

Lemma. If the factor graph is acyclic, messages are defined recursively by (12) and (13), beginning with the messages from leaves. Moreover, for any $s \in S$ and any $f \in F$:

$$Z(x, \theta) = \sum_{y_s \in \{0,1\}} \prod_{f' \in F(s)} m_{f' \to s}(y_s) \tag{14}$$

$$p_{\mathcal{Y}_{S(f)} \mid \mathcal{X}, \Theta}(y'_{S(f)} \mid x, \theta) = \frac{1}{Z(x, \theta)} \psi_{f\theta}(x_f, y_{S(f)}) \prod_{s' \in S(f)} m_{s' \to f}(y_{s'}) \tag{15}$$

The recursive computation of messages is known as **message passing**.

Summary

- For conditional graphical models whose factor graph is **acylic**, the supervised structured learning problem can be solved efficiently by means of the steepest descent algorithm and message passing.
- For conditional graphical models whose factor graph is **cyclic**, the definition of messages is cyclic as well. The partition function and marginals cannot be computed by message passing in general.
- A heuristic without guarantee of correctness or even convergence is to initialize all messages as normalized constant functions and to update messages according to some schedule, e.g., synchronously. This heuristic is commonly known as **loopy belief propagation**.

Contents. This part of the course introduces algorithms for supervised structured inference with conditional graphical models.

The **inference problem** w.r.t. a **conditional graphical model** has the form of an unconstrained binary optimization problem:

$$\underset{y \in \{0,1\}^S}{\operatorname{argmin}} \; H_\theta(x, y) \tag{16}$$

It is NP-hard. (This can be shown, e.g., by reduction of binary integer programming, which is one of Karp's 21 problems).

We consider transformations that change one decision at a time:

Definition. For any $s \in S$, let $\operatorname{flip}_s \colon \{0,1\}^S \to \{0,1\}^S$ such that for any $y \in \{0,1\}^S$ and any $t \in S$:

$$\operatorname{flip}_s[y](t) = \begin{cases} 1 - y_t & \text{if } t = s \\ y_t & \text{otherwise} \end{cases} . \tag{17}$$

The greedy local search algorithm w.r.t these transformations is known as **Iterated Conditional Modes**, or ICM (Besag 1986).

$y' = \mathsf{icm}(y)$

---

choose $s \in \operatorname*{argmin}_{s' \in S} \; H_\theta(x, \operatorname{flip}_{s'}[y]) - H_\theta(x, y)$

if $H_\theta(x, \operatorname{flip}_s[y]) < H_\theta(x, y)$

    $y' := \mathsf{icm}(\operatorname{flip}_s[y])$

else

    $y' := y$

- The **inference problem** consists in computing the minimum of a sum of functions:

$$
\begin{aligned}
& \underset{y \in \{0,1\}^S}{\operatorname{argmin}} \; H_\theta(x, y) \\
= & \underset{y \in \{0,1\}^S}{\operatorname{argmin}} \; \sum_{f \in F} h_{f\theta}(x_f, y_{S(f)})
\end{aligned}
\tag{18}
$$

- This problem is analogous to that of computing the sum of a product of functions (from the previous lecture) in that both $(\mathbb{R}, \min, +)$ and $(\mathbb{R}, +, \cdot)$ are commutative semi-rings.
- This analogy is sufficient to transfer the idea of **message passing**, albeit with messages adapted to the $(\mathbb{R}, \min, +)$ semi-ring:

Definition. (Kschischang 2001) For any variable node $s \in S$ and any factor node $f \in F$, the functions

$$\mu_{s \to f}, \mu_{f \to s} \colon \{0,1\} \to \mathbb{R} \ , \tag{19}$$

called **messages**, are defined such that for all $y_s \in \{0,1\}$:

$$\mu_{s \to f}(y_s) = \sum_{f' \in F(s) \setminus \{f\}} \mu_{f' \to s}(y_s) \tag{20}$$

$$\mu_{f \to s}(y_s) = \min_{y_{S(f) \setminus \{s\}} \in \{0,1\}^{S(f) \setminus \{s\}}} h_{f\theta}(x_f, y_{S(f)}) + \sum_{s' \in S(f) \setminus \{s\}} \mu_{s' \to f}(y_{s'}) \tag{21}$$

Lemma. If the factor graph is acyclic, messages are defined recursively by (20) and (21), beginning with the messages from leaves. Moreover, for any $s \in S$:

$$
\begin{aligned}
&\operatorname*{argmin}_{y \in \{0,1\}^S} \; H_\theta(x, y) \\
&= \min_{y \in \{0,1\}^S} \; \sum_{f \in F} h_{f\theta}(x_f, y_{S(f)}) \\
&= \min_{y_s \in \{0,1\}} \sum_{f' \in F(s)} \mu_{f' \to s}(y_s)
\end{aligned}
\tag{22}
$$

Proof. Analogous to that of Lemma 18 in the lecture notes.

Summary

- For conditional graphical models whose factor graph is **acylic**, the inference problem can be solved efficiently by means of **min-sum message passing**.
- For conditional graphical models whose factor graph is **cyclic**, one local search algorithm for the inference problem is known as **Iterated Conditional Modes (ICM)**.