Machine Learning II

Bjoern Andres
bjoern.andres@tu-dresden.de

Machine Learning for Computer Vision
Faculty of Computer Science
TU Dresden

Version 0.5β

Contents

Chapter 1

Introduction

1.1 Notation

We shall use the following notation:

- We write "iff" as shorthand for "if and only if".
- For any finite set A, we denote by $|A|$ the number of elements of A.
- For any set A, we denote by 2^A the power set of A.
- For any set A and any $m \in \mathbb{N}$, we denote by $\binom{A}{m}$ the set of all m-elementary subsets of A, i.e. $\binom{A}{m} = \{B \in 2^A \mid |B| = m\}$.
- For any sets A, B, we denote by B^A the set of all maps from A to B
- For any map $f \in B^A$, any $a \in A$ and any $b \in B$, we may write $b = f(a)$ or $b = f_a$ instead of $(a, b) \in f$
- Given any set J and, for any $j \in J$, a set S_j, we denote by $\prod_{j \in J} S_j$ the Cartesian product of the family $\{S_j\}_{j \in J}$, i.e.

$$\prod_{j \in J} S_j = \left\{ f \colon J \to \bigcup_{j \in J} S_j \;\middle|\; \forall j \in J \colon f(j) \in S_j \right\} \tag{1.1}$$

- We denote by $\langle \cdot, \cdot \rangle$ the standard inner product, and by $\| \cdot \|$ the Euclidean norm.
- For any $m \in \mathbb{N}$, we define $[m] = \{0, \ldots, m-1\}$.

Chapter 2

Supervised learning

2.1 Intuition

Informally, supervised learning is the problem of finding in a family $g : \Theta \to Y^X$ of functions, one $g_\theta : X \to Y$ that minimizes a weighted sum of two objectives:

1. g deviates little from a finite set $\{(x_s, y_s)\}_{s \in S}$ of input-output-pairs
2. g has low complexity, as quantified by a function $R : \Theta \to \mathbb{R}_0^+$

We note that the family g can have meaning beyond a mere parameterization of functions from X to Y. For instance, Θ can be a set of forms, g the functions defined by these forms, and R the length of forms. In that case, supervised learning is really an optimization problem over forms of functions, and R penalizes the complexity of these forms. Moreover, g can be chosen so as to constrain the set of functions from X to Y in the first place.

We concentrate exclusively on the special case where Y is finite. In fact, we concentrate on the case where $Y = \{0, 1\}$ in this chapter and reduce more general cases to this case in Chapter 4.

Moreover, we allow ourselves to take a detour by not optimizing over a family $g : \Theta \to \{0, 1\}^X$ directly but instead optimizing over a family $f : \Theta \to \mathbb{R}^X$ and defining g w.r.t. f via a function $L : \mathbb{R} \times \{0, 1\} \to \mathbb{R}_0^+$, called a *loss function*, such that

$$\forall \theta \in \Theta \; \forall x \in X : \quad g_\theta(x) = \operatorname*{argmin}_{\hat{y} \in \{0,1\}} \; L(f_\theta(x), \hat{y}) \; . \tag{2.1}$$

2.2 Definition

Definition 1 For any $S \neq \varnothing$ finite, called a set of *samples*, any $X \neq \varnothing$, called an *attribute space* and any $x : S \to X$, the tuple (S, X, x) is called *unlabeled data*.

For any $y : S \to \{0, 1\}$, given in addition and called a *labeling*, the tuple (S, X, x, y) is called *labeled data*.

Definition 2 For any labeled data $T = (S, X, x, y)$, any $\Theta \neq \varnothing$ and family of functions $f : \Theta \to \mathbb{R}^X$, any $R : \Theta \to \mathbb{R}_0^+$, called a *regularizer*, any $L : \mathbb{R} \times \{0, 1\} \to \mathbb{R}_0^+$, called a *loss function*, and any $\lambda \in \mathbb{R}_0^+$, called a *regularization parameter*, the instance of the *supervised learning problem* w.r.t. T, Θ, f, R, L and λ is defined as

$$\inf_{\theta \in \Theta} \quad \lambda R(\theta) + \frac{1}{|S|} \sum_{s \in S} L(f_\theta(x_s), y_s) \tag{2.2}$$

Definition 3 For any unlabeled data $T = (S, X, x)$, any $\hat{f} : X \to \mathbb{R}$ and any $L : \mathbb{R} \times \{0, 1\} \to \mathbb{R}_0^+$, the instance of the *inference problem* w.r.t. T, f and L is defined as

$$\min_{y' \in \{0,1\}^S} \sum_{s \in S} L(\hat{f}(x_s), y'_s) \tag{2.3}$$

Lemma 1 *The solutions to the inference problem are the $y : S \to \{0,1\}$ such that*

$$\forall s \in S\colon \quad y_s \in \operatorname*{argmin}_{\hat{y}\in\{0,1\}} L(\hat{f}(x_s), \hat{y}) \ . \tag{2.4}$$

Moreover, if

$$\hat{f}(X) \subseteq \{0,1\} \tag{2.5}$$

and

$$\forall r \in \mathbb{R}\ \forall \hat{y} \in \{0,1\}\colon \quad L(r,\hat{y}) = \begin{cases} 0 & \text{if } r = \hat{y} \\ 1 & \text{otherwise} \end{cases} \tag{2.6}$$

then

$$\forall s \in S\colon \quad y'_s = \hat{f}(x_s) \ . \tag{2.7}$$

Proof Generally, we have

$$\min_{y\in\{0,1\}^S} \sum_{s\in S} L(\hat{f}(x_s), y_s) = \sum_{s\in S} \min_{y_s\in\{0,1\}} L(\hat{f}(x_s), y_s) \tag{2.8}$$

By (2.5), $L(\hat{f}(x_s), \hat{f}(x_s))$ is well-defined for any $s \in S$. By (2.6) and non-negativity of L, we have

$$\forall y_s \in \{0,1\}\colon \quad L(\hat{f}(x_s), \hat{f}(x_s)) = 0 \leq L(\hat{f}(x_s), y_s) \ . \tag{2.9}$$

Thus, $y_s = \hat{f}(x_s)$ is optimal for any $s \in S$.

We note that the exact supervised learning problem formalizes a philosophical principle known as Ockham's razor.

Chapter 3

Deciding

3.1 Linear functions

3.1.1 Data

Throughout Section 3.1, we consider real attributes. More specifically, we consider some finite set $V \neq \varnothing$ and labeled data $T = (S, X, x, y)$ with $X = \mathbb{R}^V$. Hence, $x\colon S \to \mathbb{R}^V$ and $y\colon S \to \{0, 1\}$.

3.1.2 Familiy of functions

Throughout Section 3.1, we consider linear functions. More specifically, we consider $\Theta = \mathbb{R}^V$ and $f : \Theta \to \mathbb{R}^X$ such that

$$\forall \theta \in \Theta \; \forall \hat{x} \in X\colon \quad f_\theta(\hat{x}) = \langle \theta, \hat{x} \rangle \ . \tag{3.1}$$

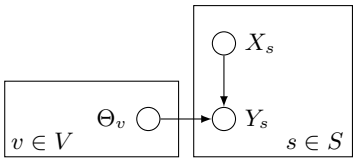
3.1.3 Probabilistic model

Random variables

- For any $s \in S$, let X_s be a random variable whose realization is a vector $x_s \in \mathbb{R}^V$, called the *attribute vector* of s
- For any $s \in S$, let Y_s be a random variable whose realization is a binary number $y_s \in \{0, 1\}$, called the *label* of s
- For any $v \in V$, let Θ_v be a random variable whose realization is a real number $\theta_v \in \mathbb{R}$, called a *parameter*

Conditional independence assumptions

We assume a probability distribution that factorizes according to the Bayesian net depicted below.



Factorization

- Firstly:

$$P(X, Y, \Theta) = \prod_{s \in S} P(Y_s \mid X_s, \Theta) P(X_s) \prod_{v \in V} P(\Theta_v) \tag{3.2}$$

- Secondly:

$$\begin{aligned} P(\Theta \mid X, Y) &= \frac{P(X, Y, \Theta)}{P(X, Y)} \\ &= \frac{P(Y \mid X, \Theta)\, P(X)\, P(\Theta)}{P(X, Y)} \\ &\propto P(Y \mid X, \Theta)\, P(\Theta) \\ &= \prod_{s \in S} P(Y_s \mid X_s, \Theta) \prod_{v \in V} P(\Theta_v) \end{aligned} \tag{3.3}$$

Forms

We consider:

- The *logistic distribution*

$$\forall s \in S: \qquad p_{Y_s|X_s,\Theta}(1) = \frac{1}{1 + 2^{-f_\theta(x_s)}} \tag{3.4}$$

- A $\sigma \in \mathbb{R}^+$ and the *normal distribution*:

$$\forall v \in V: \qquad p_{\Theta_v}(\theta_v) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\theta_v^2/2\sigma^2} \tag{3.5}$$

3.1.4 Learning problem

Lemma 2 (Logistic regression) *Estimating maximally probable parameters θ, given attributes x and labels y, i.e.,*

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

is identical to the supervised learning problem w.r.t. L, R and λ such that

$$\forall r \in \mathbb{R}\ \forall \hat{y} \in \{0, 1\}: \quad L(r, \hat{y}) = -\hat{y}r + \log\left(1 + 2^r\right) \tag{3.6}$$

$$\forall \theta \in \Theta: \quad R(\theta) = \|\theta\|_2^2 \tag{3.7}$$

$$\lambda = \frac{\log e}{2\sigma^2} \tag{3.8}$$

Proof Firstly,

$$\begin{aligned} & \operatorname*{argmax}_{\theta \in \mathbb{R}^m} \quad p_{\Theta|X,Y}(\theta, x, y) \\ \overset{(3.3)}{=} & \operatorname*{argmax}_{\theta \in \mathbb{R}^m} \quad \prod_{s \in S} p_{Y_s|X_s,\Theta}(y_s, x_s, \theta) \prod_{v \in V} p_{\Theta_v}(\theta_v) \\ = & \operatorname*{argmax}_{\theta \in \mathbb{R}^m} \quad \sum_{s \in S} \log p_{Y_s|X_s,\Theta}(y_s, x_s, \theta) + \sum_{v \in V} \log p_{\Theta_v}(\theta_v) \end{aligned} \tag{3.9}$$

Substituting in (3.9) the linearization

$$\begin{aligned} & \log p_{Y_s|X_s,\Theta}(y_s, x_s, \theta) \\ = \quad & y_s \log p_{Y_s|X_s,\Theta}(1, x_s, \theta) + (1 - y_s) \log p_{Y_s|X_s,\Theta}(0, x_s, \theta) \\ = \quad & y_s \log \frac{p_{Y_s|X_s,\Theta}(1, x_s, \theta)}{p_{Y_s|X_s,\Theta}(0, x_s, \theta)} + \log p_{Y_s|X_s,\Theta}(0, x_s, \theta) \end{aligned} \tag{3.10}$$

as well as (3.4) and (3.5) yields the form (3.11) below that is called the instance of the l_2-regularized *logistic regression problem* with respect to x, y and σ.

$$\operatorname*{argmin}_{\theta \in \mathbb{R}^m} \quad \sum_{s \in S} \left(-y_s \langle \theta, x_s \rangle + \log\left(1 + 2^{\langle \theta, x_s \rangle}\right)\right) + \frac{\log e}{2\sigma^2} \|\theta\|_2^2 \tag{3.11}$$

Exercise 1 *a) Derive* (3.11) *from* (3.9) *using* (3.10), (3.4) *and* (3.5)
b) Is the objective function of (3.11) *convex?*

3.1.5 Inference problem

Lemma 3 *Estimating maximally probable labels y, given attributes x' and parameters θ, i.e.,*

$$\operatorname*{argmax}_{y \in \{0,1\}^S} \quad p_{Y|X,\Theta}(y, x', \theta) \tag{3.12}$$

is identical to the inference problem w.r.t. f and L. It has the solution

$$\forall s \in S' \; : \quad y_s = \begin{cases} 1 & \text{if } f_\theta(x'_s) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.13}$$

Proof Firstly,

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

Secondly,

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

3.1.6 Inference algorithm

The inference problem is solved by computing independently for each $s \in S'$ the label

$$y_s = \begin{cases} 1 & \text{if } \langle \theta, x'_s \rangle > 0 \\ 0 & \text{otherwise} \end{cases} \; . \tag{3.14}$$

The time complexity is $O(|V||S'|)$.

Chapter 4

Semi-supervised and unsupervised learning

4.1 Intuition

So far, we have considered learning problems w.r.t. labeled data (S, X, x, y) where, for every $s \in S$, a label $y_s \in \{0, 1\}$ is given, and inference problems w.r.t. unlabeled data (S', X', x) where no label is given and every combination of labels $y' : S \to \{0, 1\}$ is a feasible solution.

Next, we consider learning problems where not every label is given and inference problems where not every combination of labels is feasible. Unlike before, the data we look at in both problems coincides, consisting of tuples $(S, X, x, \mathcal{Y})$ where $\mathcal{Y} \subseteq \{0, 1\}^S$ is a set of feasible labelings. In particular, $\mathcal{Y} = \{0, 1\}^S$ is the special case of unlabeled data, and $|\mathcal{Y}| = 1$ is the special case of labeled data. Non-trivial choices of $\mathcal{Y}$ allow us to express problems of learning and inferring finite structures such as maps (Chapter 5).

4.2 Definition

Definition 4 For any $S \neq \varnothing$ finite, called a set of *samples*, any $X \neq \varnothing$, called an *attribute space*, any $x : S \to X$ and any $\varnothing \neq \mathcal{Y} \subseteq \{0, 1\}^S$, called a set of *feasible labelings*, the tuple $T = (S, X, x, \mathcal{Y})$ is called *constrained data*.

Definition 5 For any constrained data $T = (S, X, x, \mathcal{Y})$, any $\Theta \neq \varnothing$ and family of functions $f : \Theta \to \mathbb{R}^X$, any $R : \Theta \to \mathbb{R}_0^+$, called a *regularizer*, any $L : \mathbb{R} \times \{0, 1\} \to \mathbb{R}_0^+$, called a *loss function* and any $\lambda \in \mathbb{R}_0^+$, called a *regularization parameter*, the instance of the *learning and inference problem* w.r.t. T, Θ, f, R, L and λ is defined as

$$\min_{y \in \mathcal{Y}} \inf_{\theta \in \Theta} \quad \lambda R(\theta) + \frac{1}{|S|} \sum_{s \in S} L(f_\theta(x_s), y_s) \tag{4.1}$$

The special case of one-elementary $\mathcal{Y} = \{y\}$ is called the *supervised learning problem*.

The special case of one-elementary $\Theta = \{\hat{\theta}\}$ written below is called the *inference problem*.

$$\min_{y \in \mathcal{Y}} \quad \sum_{s \in S} L(f_\theta(x_s), y_s) \tag{4.2}$$

Chapter 5

Classifying

5.1 Maps

For any finite set $A \neq \varnothing$ whose elements we seek to classify and any finite set $B \neq \varnothing$ of class labels, we are interested in *maps* $\varphi : A \to B$ that assign to every element $a \in A$ precisely one class label $\varphi(a) \in B$. Maps are precisely those subsets of $\varphi \subseteq A \times B$ that satisfy

$$\forall a \in A \; \exists b \in B : \quad (a,b) \in \varphi \tag{5.1}$$

$$\forall a \in A \; \forall b, b' \in B : \quad (a,b) \in \varphi \wedge (a,b') \in \varphi \Rightarrow b = b' \ . \tag{5.2}$$

They are characterized by those functions $y : A \times B \to \{0,1\}$ that satisfy

$$\forall a \in A \colon \quad \sum_{b \in B} y_{ab} = 1 \ . \tag{5.3}$$

We reduce the problem of learning and inferring maps to the problem of learning and inferring decisions, by choosing constrained data with

$$S = A \times B \tag{5.4}$$

$$\mathcal{Y} = \left\{ y \colon A \times B \to \{0,1\} \;\middle|\; \forall a \in A \colon \sum_{b \in B} y_{ab} = 1 \right\} \ . \tag{5.5}$$

5.2 Linear functions

5.2.1 Data

Throughout Section 5.2, we consider some finite set $V \neq \varnothing$ and constrained data $(S, X, x, \mathcal{Y})$ with $S = A \times B$ as in (5.4), $X = B \times \mathbb{R}^V$, and $\mathcal{Y}$ as in (5.5). More specifically, we assume that, for any $(a,b) \in A \times B$, the class label b is the first attribute of (a,b), i.e.,

$$\forall a \in A \; \forall b \in B \; \exists \hat{x} \in \mathbb{R}^V : \quad x_{ab} = (b, \hat{x}) \tag{5.6}$$

As a special case, we consider labeled data where we are given just one $\mathcal{Y} = \{y\}$ with y satisfying the constraints (5.3).

5.2.2 Familiy of functions

Throughout Section 5.2, we consider linear functions. More specifically, we consider $\Theta = \mathbb{R}^{B \times V}$ and $f : \Theta \to \mathbb{R}^X$ such that

$$\forall \theta \in \Theta \; \forall b \in B \; \forall \hat{x} \in \mathbb{R}^V : \quad f_\theta((b, \hat{x})) = \sum_{v \in V} \theta_{bv} \, \hat{x}_v = \langle \theta_{b\cdot}, \hat{x} \rangle \ . \tag{5.7}$$

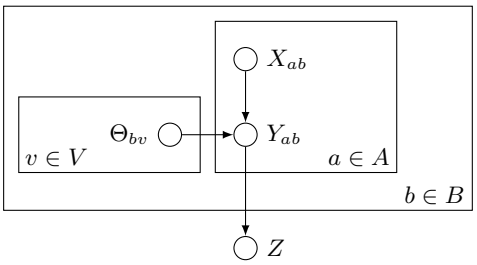
5.2.3 Probabilistic model

Random variables

- For any $(a, b) \in A \times B$, let X_{ab} be a random variable whose realization is a vector $x_{ab} \in B \times \mathbb{R}^V$, called the *attribute vector* of (a, b).
- For any $(a, b) \in A \times B$, let Y_{ab} be a random variable whose realization is a binary number $y_{ab} \in \{0, 1\}$, called the *decision* of classifying a as b
- For any $b \in B$ and any $v \in V$, let Θ_{bv} be a random variable whose realization is a real number $\theta_{bv} \in \mathbb{R}$, called a *parameter*
- Let Z be a random variable whose realization is a subset $z \subseteq \{0, 1\}^{A \times B}$. For multiple label classification, we are interested in $z = \mathcal{Y}$, the set of the characteristic functions of all maps from A to B.

Conditional independence assumptions

We assume a probability distribution that factorizes according to Bayesian net depicted below.



Factorization

These conditional independence assumptions imply the following factorizations:

- Firstly:

$$P(X, Y, Z, \Theta) = P(Z \mid Y) \prod_{(a,b) \in A \times B} P(Y_{ab} \mid X_{ab}, \Theta) \prod_{(b,v) \in B \times V} P(\Theta_{bv}) \prod_{(a,b) \in A \times B} P(X_{ab}) \tag{5.8}$$

- Secondly:

$$\begin{aligned} P(\Theta \mid X, Y, Z) &= \frac{P(X, Y, Z, \Theta)}{P(X, Y, Z)} \\ &= \frac{P(Z \mid Y)\, P(Y \mid X, \Theta)\, P(X)\, P(\Theta)}{P(Z \mid X, Y)\, P(X, Y)} \\ &= \frac{P(Z \mid Y)\, P(Y \mid X, \Theta)\, P(X)\, P(\Theta)}{P(Z \mid Y)\, P(X, Y)} \\ &= \frac{P(Y \mid X, \Theta)\, P(X)\, P(\Theta)}{P(X, Y)} \\ &\propto P(Y \mid X, \Theta)\, P(\Theta) \\ &= \prod_{(a,b) \in A \times B} P(Y_{ab} \mid X_{ab}, \Theta) \prod_{(b,v) \in B \times V} P(\Theta_{bv}) \end{aligned} \tag{5.9}$$

- Thirdly,

$$\begin{aligned} P(Y \mid X, Z, \theta) &= \frac{P(X, Y, Z, \Theta)}{P(X, Z, \Theta)} \\ &= \frac{P(Z \mid Y)\, P(Y \mid X, \Theta)\, P(X)\, P(\Theta)}{P(X, Z, \Theta)} \\ &\propto P(Z \mid Y)\, P(Y \mid X, \Theta) \\ &= P(Z \mid Y) \prod_{(a,b) \in A \times B} P(Y_{ab} \mid X_{ab}, \Theta) \end{aligned} \tag{5.10}$$

Forms

Here, we consider:

- The *logistic distribution*

$$\forall (a,b) \in A \times B: \qquad p_{Y_{ab} | X_{ab}, \Theta}(1) = \frac{1}{1 + 2^{-f_\theta(x_{ab})}} \tag{5.11}$$

- A $\sigma \in \mathbb{R}^+$ and the *normal distribution*:

$$\forall (b,v) \in B \times V: \qquad p_{\Theta_{bv}}(\theta_{bv}) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\theta_{bv}^2 / 2\sigma^2} \tag{5.12}$$

- A uniform distribution on a subset:

$$\forall z \subseteq \{0,1\}^{A \times B}: \quad p_{Z|Y}(z) \propto \begin{cases} 1 & \text{if } y \in z \\ 0 & \text{otherwise} \end{cases} \tag{5.13}$$

  Note that $p_{Z|Y}(\mathcal{Y})$ is non-zero iff the relation $y^{-1}(1) \subseteq A \times B$ is a map.

5.2.4 Learning problem

Lemma 4 *Estimating maximally probable parameters θ, given attributes x and decisions y, i.e.,*

$$\operatorname*{argmax}_{\theta \in \mathbb{R}^{B \times V}} \quad p_{\Theta | X, Y}(\theta, x, y)$$

is identical to the supervised learning problem w.r.t. L, R and λ such that

$$\forall r \in \mathbb{R}\ \forall \hat{y} \in \{0,1\}: \quad L(r, \hat{y}) = -\hat{y} r + \log\left(1 + 2^r\right) \tag{5.14}$$

$$\forall \theta \in \Theta: \qquad R(\theta) = \|\theta\|_2^2 \tag{5.15}$$

$$\lambda = \frac{\log e}{2\sigma^2} \tag{5.16}$$

Moreover, this problem separates into $|B|$ independent supervised learning problems, each w.r.t. parameters in $\mathbb{R}^V$, with L and λ as above, and with

$$\forall \theta' \in \mathbb{R}^V: \qquad R'(\theta') = \|\theta'\|_2^2 \tag{5.17}$$

PROOF Analogous to the case of binary classification from Section 3.1, we now obtain:

$$\begin{aligned} &\operatorname*{argmax}_{\theta \in \mathbb{R}^{B \times V}} \quad p_{\Theta | X, Y}(\theta, x, y) \\ &= \operatorname*{argmin}_{\theta \in \mathbb{R}^{B \times V}} \quad \sum_{(a,b) \in A \times B} \left(-y_{ab} f_\theta(x_{ab}) + \log\left(1 + 2^{f_\theta(x_{ab})}\right) \right) + \frac{\log e}{2\sigma^2} \|\theta\|_2^2 \ . \end{aligned} \tag{5.18}$$

Consider the unique $x' : A \times B \to \mathbb{R}^V$ such that, for any $(a, b) \in A \times B$, we have $x_{ab} = (b, x'_{ab})$.

Problem (5.18) separates into $|B|$ many l_2-regularized logistic regression problems, one for each $b \in B$, because

$$\begin{aligned}
&\min_{\theta \in \mathbb{R}^{B \times V}} \sum_{(a,b) \in A \times B} \left(-y_{ab} \langle \theta_{b\cdot}, x'_{ab} \rangle + \log \left(1 + 2^{\langle \theta_{b\cdot}, x'_{ab} \rangle} \right) \right) + \frac{\log e}{2\sigma^2} \|\theta\|_2^2 \\
= &\min_{\theta \in \mathbb{R}^{B \times V}} \sum_{b \in B} \left(\sum_{a \in A} \left(-y_{ab} \langle \theta_{b\cdot}, x'_{ab} \rangle + \log \left(1 + 2^{\langle \theta_{b\cdot}, x'_{ab} \rangle} \right) \right) + \frac{\log e}{2\sigma^2} \|\theta_{b\cdot}\|_2^2 \right) \\
= &\sum_{b \in B} \min_{\theta_{b\cdot} \in \mathbb{R}^V} \left(\sum_{a \in A} \left(-y_{ab} \langle \theta_{b\cdot}, x'_{ab} \rangle + \log \left(1 + 2^{\langle \theta_{b\cdot}, x'_{ab} \rangle} \right) \right) + \frac{\log e}{2\sigma^2} \|\theta_{b\cdot}\|_2^2 \right) \quad .
\end{aligned}$$

5.2.5 Inference problem

Lemma 5 *For any constrained data as defined above, any $\theta \in \mathbb{R}^{B \times V}$ and any $\hat{y} : A \times B \to \{0, 1\}$, $\hat{y}$ is a solution to the inference problem*

$$\min_{y \in \mathcal{Y}} \sum_{(a,b) \in A \times B} L(f_\theta(x_{ab}), y_{ab}) \tag{5.19}$$

iff there exists an $\varphi : A \to B$ such that

$$\forall a \in A\colon \quad \varphi(a) \in \max_{b \in B} \langle \theta_{b\cdot}, x'_{ab} \rangle \tag{5.20}$$

and

$$\forall (a, b) \in A \times B\colon \quad \hat{y}_{ab} = 1 \Leftrightarrow \varphi(a) = b \quad . \tag{5.21}$$

Proof

$$\begin{aligned}
&\sum_{(a,b) \in A \times B} L(f_\theta(x_{ab}), y_{ab}) \\
= &\sum_{(a,b) \in A \times B} \left(L(f_\theta(x_{ab}), 1)\, y_{ab} + L(f_\theta(x_{ab}), 0)\, (1 - y_{ab}) \right) \\
= &\sum_{(a,b) \in A \times B} \left(L(f_\theta(x_{ab}), 1) - L(f_\theta(x_{ab}), 0) \right) y_{ab} + \text{const.} \\
= &\sum_{(a,b) \in A \times B} \left(-f_\theta(x_{ab}) \right) y_{ab} && \text{by (5.14)} \\
= &\sum_{(a,b) \in A \times B} \left(-\langle \theta_{b\cdot}, x'_{ab} \rangle \right) y_{ab} && x_{ab} = (b, x'_{ab}) \\
= &\sum_{a \in A} \sum_{b \in B} \left(-\langle \theta_{b\cdot}, x'_{ab} \rangle \right) y_{ab}
\end{aligned}$$

5.2.6 Inference algorithm

The inference problem is solved by solving (5.20) independently for each $a \in A$. The time complexity is $O(|A||B||V|)$.

Chapter 6

Supervised structured learning

6.1 Intuition

Even in the most general learning and inference problem w.r.t. constrained data $(S, X, x, \mathcal{Y})$ we have considered so far, attributes $x_s \in X$ are defined for single elements $s \in S$ only, and solutions are such that decisions $y_s, y_{s'} \in \{0,1\}$ for distinct $s, s' \in S$ are independent unless they are tied by constraints of a feasible set $\mathcal{Y} \subset \{0,1\}^S$.

This mathematical abstraction of learning is too restrictive for certain applications. For example, consider a task where we are given a digital image and need to decide for every pixel $s \in S$, by the contents of the image around that pixel, whether the pixel is of interest ($y_s = 1$) or not of interest ($y_s = 0$). Typically, decisions at neighboring pixels $s, s' \in S$ are more likely to be equal ($y_s = y_{s'}$) than unequal ($y_s \neq y_{s'}$), and we may wish to learn how this increased probability depends on the contents of the image. None of the mathematical abstractions of learning we have considered so far is sufficient to express this dependency.

In order to lift this restriction, we will now define a supervised learning problem as well as an inference problem in which attributes are associated with subsets of S, and in which decisions can be tied by probabilistic dependencies. Therefor, we will introduce a family $H : \Theta \to \mathbb{R}^{X \times Y}$ of functions that quantify by $H_\theta(x, y)$ how incompatible attributes $x \in X$ are with a combination of decisions $y \in \{0,1\}^S$. We will define supervised structured learning as a problem of finding one function from this family. We will define structured inference as the problem of finding a combination of decisions $y \in \{0,1\}^S$ that minimizes $H_\theta(x, \cdot)$.

6.2 Definition

Definition 6 A triple (S, F, E) is called a *factor graph* with *variable nodes* S and *factor nodes* F iff $S \cap F = \varnothing$ and $(S \cup F, E)$ is a bipartite graph such that $\forall e \in E \, \exists s \in S \, \exists f \in F \colon e = \{s, f\}$.

For any factor node $f \in F$, we denote by $S_f = \{s \in S \,|\, \{s, f\} \in E\}$ the set of those variable nodes that are neighbors of f.

Definition 7 A tuple $T = (S, F, E, \{X_f\}_{f \in F}, x)$ is called *unlabeled structured data* iff (S, F, E) is a factor graph, every set X_f is non-empty, called the *attribute space* of f, and $x \in \prod_{f \in F} X_f$, where the Cartesian product $\prod_{f \in F} X_f$ is called the *attribute space* of T. A tuple $(S, F, E, \{X_f\}_{f \in F}, x, y)$ is called *labeled structured data* iff $(S, F, E, \{X_f\}_{f \in F}, x)$ is unlabeled structured data, and $y \in \{0,1\}^S$.

Definition 8 For any labeled structured data $T = (S, F, E, \{X_f\}_{f \in F}, x, y)$, the attribute space $X = \prod_{f \in F} X_f$, the set $Y = \{0,1\}^S$, any $\Theta \neq \varnothing$ and family of functions $H : \Theta \to \mathbb{R}^{X \times Y}$, any $R : \Theta \to \mathbb{R}_0^+$, called a *regularizer*, any $L : \mathbb{R}^Y \times Y \to \mathbb{R}_0^+$, called a *loss function*, and any $\lambda \in \mathbb{R}_0^+$, called a *regularization parameter*, the instance of the *supervised structured learning problem*

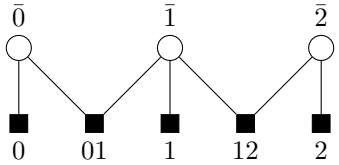


Figure 6.1: The factor graph with $S = \{\bar{0}, \bar{1}, \bar{2}\}$ and $F = \{0, 1, 2, 01, 12\}$ depicted above makes explicit that a function $H : \{0,1\}^S \to \mathbb{R}$ factorizes according to $H(y) = h_0(y_{\bar{0}}) + h_1(y_{\bar{1}}) + h_2(y_{\bar{2}}) + h_{01}(y_{\bar{0}}, y_{\bar{1}}) + h_{12}(y_{\bar{1}}, y_{\bar{2}})$.

w.r.t. T, Θ, H, R, L and λ is defined as

$$\inf_{\theta \in \Theta} \quad \lambda R(\theta) + L(H_\theta(x, \cdot), y) \tag{6.1}$$

Intuitively, H_θ is a function that quantifies by $H_\theta(x, y)$ how incompatible attributes $x \in X$ are with a combination of decisions $y \in \{0,1\}^S$. Consequently, $H_\theta(x, \cdot)$ is a functional that assigns an incompatibility to every combination of decisions.

Definition 9 For any unlabeled structured data $T = (S, F, E, \{X_f\}_{f \in F}, x)$ and any $\hat{H} \colon X \times \{0,1\}^S \to \mathbb{R}$, the instance of the *inference problem* w.r.t. T and $\hat{H}$ is defined as

$$\min_{y \in \{0,1\}^S} \quad \hat{H}(x, y) \tag{6.2}$$

6.3 Conditional graphical models

6.3.1 Data

Throughout Section 6.3, we consider labeled data $(S, F, E, \{X_f\}_{f \in F}, x, y)$ and an attribute space $X = \prod_{f \in F} X_f$ such that, for every $f \in F$, there is an $n_f \in \mathbb{N}$ and $X_f = \mathbb{R}^{n_f}$.

6.3.2 Family of functions

Definition 10 For any factor graph $G = (S, F, E)$, a function $H : \{0,1\}^S \to \mathbb{R}$ is said to *factorize* w.r.t. G iff, for every $f \in F$, there exists a function a function $h_f : \{0,1\}^{S_f} \to \mathbb{R}$, called a *factor* of H, such that

$$\forall y \in \{0,1\}^S \colon \quad H(y) = \sum_{f \in F} h_f(y_{S_f}) \ . \tag{6.3}$$

An example is shown in Fig. 6.1.

Definition 11 A tuple $(S, F, E, \{X_f\}_{f \in F}, \Theta, \{h_f\}_{f \in F})$ is called a *conditional graphical model* with *attribute space* $\prod_{f \in F} X_f = X$ and *parameter space* Θ iff (S, F, E) is a factor graph, $\Theta \neq \varnothing$ and, for every $f \in F$, $X_f \neq \varnothing$, called the *attribute space* of f, and $h_f : \Theta \to \mathbb{R}^{X_f \times \{0,1\}^{S_f}}$, called a *factor*.

The $H : \Theta \to \mathbb{R}^{X \times \{0,1\}^S}$ defined below is called the *energy function* of the conditional graphical model.

$$\forall \theta \in \Theta \ \forall x \in X \ \forall y \in \{0,1\}^S \colon \quad H_\theta(x, y) = \sum_{f \in F} h_{f\theta}(x_f, y_{S_f}) \tag{6.4}$$

Throughout Section 6.3, we consider such a conditional graphical model. We make two additional assumptions: Firstly, we assume that Θ is a finite-dimensional, real vector space, i.e., there exists a finite, non-empty set J and $\Theta = \mathbb{R}^J$. Secondly, we assume that every function h_f is linear in

Θ, i.e., for every $f \in F$, there exists a $\varphi_f : X_f \times \{0,1\}^{S_f} \to \mathbb{R}^J$ such that for any $x_f \in X_f$, any $y_{S_f} \in \{0,1\}^{S_f}$ and any $\theta \in \Theta$:

$$h_{f\theta}(x_f, y_{S_f}) = \langle \theta, \varphi_f(x_f, y_{S_f}) \rangle \tag{6.5}$$

For convenience, we define $\xi : X \times \{0,1\}^S \to \mathbb{R}^J$ such that for any $x \in X$ and any $y \in \{0,1\}^S$:

$$\xi(x,y) = \sum_{f \in F} \varphi_f(x_f, y_{S_f}) \tag{6.6}$$

Thus, we obtain for any $\theta \in \Theta$, any $x \in X$ and any $y \in Y$:

$$\begin{aligned} H_\theta(x,y) &= \sum_{f \in F} h_{f\theta}(x_f, y_{S_f}) \\ &= \sum_{f \in F} \langle \theta, \varphi_f(x_f, y_{S_f}) \rangle \\ &= \left\langle \theta, \sum_{f \in F} \varphi_f(x_f, y_{S_f}) \right\rangle \\ &= \langle \theta, \xi(x,y) \rangle \end{aligned} \tag{6.7}$$

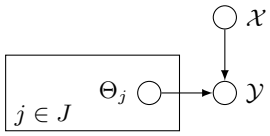
6.3.3 Probabilistic model

Random Variables

- Let $\mathcal{X}$ be a random variable whose realization is an element $x \in X$ of the attribute space.
- Let $\mathcal{Y}$ be a random variable whose realization is a combination of decisions $y \in \{0,1\}^S$
- For any $j \in J$, let Θ_j a random variable whose realization is a $\theta_j \in \mathbb{R}$

Conditional independence assumptions

We assume a probability distribution that factorizes according to the Bayesian net depicted below.



Factorization

- Firstly:

$$P(\mathcal{X}, \mathcal{Y}, \Theta) = P(\mathcal{Y} \mid \mathcal{X}, \Theta)\, P(\mathcal{X}) \prod_{j \in J} P(\Theta_j) \tag{6.8}$$

- Secondly:

$$\begin{aligned} P(\Theta \mid \mathcal{X}, \mathcal{Y}) &= \frac{P(\mathcal{X}, \mathcal{Y}, \Theta)}{P(\mathcal{X}, \mathcal{Y})} \\ &= \frac{P(\mathcal{Y} \mid \mathcal{X}, \Theta)\, P(\mathcal{X}) \prod_{j \in J} P(\Theta_j)}{P(\mathcal{X}, \mathcal{Y})} \\ &\propto P(\mathcal{Y} \mid \mathcal{X}, \Theta) \prod_{j \in J} P(\Theta_j) \end{aligned} \tag{6.9}$$

Forms

Definition 12 For any conditional graphical model, the *partition function* $Z\colon X \times \Theta \to \mathbb{R}$ and *Gibbs distribution* $p\colon X \times \{0,1\}^S \times \Theta \to [0,1]$ are defined by the forms

$$Z(x,\theta) = \sum_{y \in \{0,1\}^S} e^{-H_\theta(x,y)} \tag{6.10}$$

$$p(y,x,\theta) = \frac{1}{Z(x,\theta)} e^{-H_\theta(x,y)} \tag{6.11}$$

We consider in (6.9) the Gibbs distribution of our conditional graphical model, i.e.

$$p_{\mathcal{Y}|\mathcal{X},\Theta}(y,x,\theta) = \frac{1}{Z(x,\theta)} e^{-H_\theta(x,y)} \quad . \tag{6.12}$$

Moreover, we consider in (6.9) a $\sigma \in \mathbb{R}^+$ and, for every $j \in J$, the *normal distribution*

$$p_{\Theta_j}(\theta_j) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\theta_j^2/2\sigma^2} \quad . \tag{6.13}$$

6.3.4 Learning problem

Lemma 6 *Estimating maximally probable parameters θ, given attributes x and decisions y, i.e.,*

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

is identical to the supervised structured learning problem w.r.t. L, R and λ such that

$$L(H_\theta(x,\cdot),y) = H_\theta(x,y) + \ln Z(x,\theta) \tag{6.14}$$

$$= H_\theta(x,y) + \ln \sum_{y' \in \{0,1\}^S} e^{-H_\theta(x,y')} \tag{6.15}$$

$$= \langle \theta, \xi(x,y) \rangle + \ln \sum_{y' \in \{0,1\}^S} e^{-\langle \theta, \xi(x,y') \rangle} \tag{6.16}$$

$$R(\theta) = \|\theta\|_2^2 \tag{6.17}$$

$$\lambda = \frac{1}{2\sigma^2} \tag{6.18}$$

Exercise 2 *Prove Lemma 6.*

Lemma 7 *The first and second partial derivatives of the logarithm of the partition function have the forms*

$$\frac{\partial}{\partial \theta_j} \ln Z = \frac{1}{Z(x,\theta)} \sum_{y' \in \{0,1\}^S} (-\xi_j(x,y')) e^{-\langle \theta, \xi(x,y') \rangle} \tag{6.19}$$

$$= \mathbb{E}_{y' \sim p_{\mathcal{Y}|\mathcal{X},\Theta}}(-\xi_j(x,y')) \tag{6.20}$$

$$\frac{\partial^2}{\partial \theta_j \, \partial \theta_k} \ln Z = \mathbb{E}_{y' \sim p_{\mathcal{Y}|\mathcal{X},\Theta}}(\xi_j(x,y')\xi_k(x,y')) - \mathbb{E}_{y' \sim p_{\mathcal{Y}|\mathcal{X},\Theta}}(\xi_j(x,y')) \mathbb{E}_{y' \sim p_{\mathcal{Y}|\mathcal{X},\Theta}}(\xi_k(x,y'))$$

$$= \mathrm{COV}_{y' \sim p_{\mathcal{Y}|\mathcal{X},\Theta}}(\xi_j(x,y'), \xi_k(x,y')) \tag{6.21}$$

Exercise 3 *Prove Lemma 7.*

Lemma 8 *Supervised structured learning of a conditional graphical model is a convex optimization problem.*

Exercise 4 *Prove Lemma 8 using* (6.21).

6.3.5 Inference problem

Lemma 9 *Estimating maximally probable decisions y, given attributes x and parameters θ, i.e.*

$$\operatorname*{argmax}_{y \in \{0,1\}^S} \quad p_{\mathcal{Y}|\mathcal{X},\Theta}(x, y, \theta) \tag{6.22}$$

is identical to the structured inference problem with $\hat{H}(x, y) = H_\theta(x, y)$.

Exercise 5 *Prove Lemma 9.*

6.3.6 Learning algorithm

On the on hand, the supervised structured learning problem for conditional graphical models can be solved exactly by means of the steepest descent algorithm, due to its convexity (Lemma 8).

Algorithm 1 Steepest descent with 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, the time complexity of computing the gradient is $O(2^{|S|})$, due to the summations involved in computing the partition function $Z(x, \theta)$ and expectation values (6.19). More specifically, computing a derivative

$$\begin{aligned}
-\frac{\partial}{\partial \theta_j} \ln Z &= \mathbb{E}_{y' \sim p_{\mathcal{Y}|\mathcal{X},\Theta}}(\xi_j(x, y')) \\
&= \frac{1}{Z(x,\theta)} \sum_{y' \in \{0,1\}^S} \xi_j(x, y')\, e^{-\langle \theta, \xi(x,y') \rangle} \\
&= \frac{1}{Z(x,\theta)} \sum_{y' \in \{0,1\}^S} \sum_{f \in F} \varphi_{fj}(x_f, y'_{S_f})\, e^{-\langle \theta, \xi(x,y') \rangle} \\
&= \frac{1}{Z(x,\theta)} \sum_{f \in F} \sum_{y'_{S(f)} \in \{0,1\}^{S(f)}} \sum_{y'_{S \setminus S(f)} \in \{0,1\}^{S \setminus S(f)}} \varphi_{fj}(x_f, y'_{S(f)})\, e^{-\langle \theta, \xi(x,y') \rangle} \\
&= \sum_{f \in F} \sum_{y'_{S(f)} \in \{0,1\}^{S(f)}} \varphi_{fj}(x_f, y'_{S(f)}) \frac{1}{Z(x,\theta)} \sum_{y'_{S \setminus S(f)} \in \{0,1\}^{S \setminus S(f)}} e^{-\langle \theta, \xi(x,y') \rangle} \qquad (6.23) \\
&= \sum_{f \in F} \sum_{y'_{S(f)} \in \{0,1\}^{S(f)}} \varphi_{fj}(x_f, y'_{S(f)})\, p_{\mathcal{Y}_{S(f)}|\mathcal{X},\Theta}(y'_{S(f)} \mid x, \theta) \qquad (6.24) \\
&= \sum_{f \in F} \mathbb{E}_{y'_{S(f)} \sim p_{\mathcal{Y}_{S(f)}|\mathcal{X},\Theta}}(\varphi_{fj}(x_f, y'_{S(f)})) \qquad (6.25)
\end{aligned}$$

requires computing

- the partition function

$$Z(x, \theta) = \sum_{y' \in \{0,1\}^S} e^{-\langle \theta, \xi(x,y') \rangle} \tag{6.26}$$

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

$$p_{\mathcal{Y}_{S(f)}|\mathcal{X},\Theta}(y'_{S(f)} \mid x, \theta) = \frac{1}{Z(x,\theta)} \sum_{y'_{S\setminus S(f)} \in \{0,1\}^{S \setminus S(f)}} e^{-\langle \theta, \xi(x,y') \rangle} \tag{6.27}$$

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

$$\sum_{y'_{S(f)} \in \{0,1\}^{S(f)}} \varphi_{fj}(x_f, y'_{S(f)})\, p_{\mathcal{Y}_{S(f)}|\mathcal{X},\Theta}(y'_{S(f)} \mid x, \theta) \ . \tag{6.28}$$

In the special case where the degree $\max_{f \in F} S(f)$ of the conditional graphical model is bounded by a constant, computing (6.28) from the factor marginal takes constant time. The challenge in (6.27) or all (6.26) is to sum the function

$$\psi_\theta(x, y') := e^{-\langle \theta, \xi(x,y') \rangle} \tag{6.29}$$

over assignments to some (6.27) or all (6.26) variables y'. Defining

$$\psi_{f\theta}(x_f, y'_{S(f)}) = e^{-\langle \theta, \varphi_f(x_f, y'_{S(f)}) \rangle} \tag{6.30}$$

and exploiting factorization (6.6), we obtain

$$\begin{aligned} & e^{-\langle \theta, \xi(x,y') \rangle} \\ &= e^{-\sum_{f \in F} \langle \theta, \varphi_f(x_f, y_{S(f)}) \rangle} && (6.31) \\ &= \prod_{f \in F} e^{-\langle \theta, \varphi_f(x_f, y_{S(f)}) \rangle} && (6.32) \\ &= \prod_{f \in F} \psi_{f\theta}(x_f, y_{S(f)}) \ . && (6.33) \end{aligned}$$

Thus, the challenge in (6.27) and (6.26) is to compute a sum of a product of functions. Specifically:

$$Z(x,\theta) = \sum_{y' \in \{0,1\}^S} \prod_{f \in F} \psi_{f\theta}(x_f, y_{S(f)}) \tag{6.34}$$

$$p_{\mathcal{Y}_{S(f)}|\mathcal{X},\Theta}(y'_{S(f)} \mid x, \theta) = \frac{1}{Z(x,\theta)} \sum_{y'_{S\setminus S(f)} \in \{0,1\}^{S \setminus S(f)}} \prod_{f \in F} \psi_{f\theta}(x_f, y_{S(f)}) \tag{6.35}$$

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

$$m_{s\to f}, m_{f\to s} \colon \{0,1\} \to \mathbb{R} \ , \tag{6.36}$$

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

$$m_{s\to f}(y_s) = \prod_{f' \in F(s)\setminus\{f\}} m_{f'\to s}(y_s) \tag{6.37}$$

$$m_{f\to s}(y_s) = \sum_{y_{S(f)\setminus\{s\}}} \psi_{f\theta}(x_f, y_{S(f)}) \prod_{s' \in S(f)\setminus\{s\}} m_{s'\to f}(y_{s'}) \tag{6.38}$$

Lemma 10 *If the factor graph is acyclic, messages are defined recursively by* (6.37) *and* (6.38)*, beginning with the messages from leaves. Moreover, for any $s \in S$ and any $f \in F$:*

$$Z(x, \theta) = \sum_{y_s \in \{0,1\}} \prod_{f' \in F(s)} m_{f' \to s}(y_s) \tag{6.39}$$

$$p_{\mathcal{Y}_{S(f)} | \mathcal{X}, \Theta}(y'_{S(f)} \mid x, \theta) = \frac{1}{Z(x, \theta)} \psi_{f\theta}(x_f, y_{S(f)}) \prod_{s' \in S(f)} m_{s' \to f}(y_{s'}) \tag{6.40}$$

Exercise 6 *Prove Lemma 10.*

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

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, by Lemma 8 and Lemma 10.

For conditional graphical models whose factor graph is cyclic, the definition of messages by (6.37) and (6.38) 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 update messages according to some schedule, e.g., synchronously. This heuristic is known as *loopy belief propagation* and has proven suitable for some applications.

6.3.7 Inference algoritms

Iterated conditional modes (ICM)

For the inference problem

$$\operatorname*{argmin}_{y \in \{0,1\}^S} \; H_\theta(x, y) \;\; , \tag{6.41}$$

a heuristic that is guaranteed to converge and terminate in a (possibly sub-optimal) feasible solution is local search w.r.t. transformations that change one variable at a time:

Definition 14 For any $s \in S$, let $\mathrm{flip}_s \colon \{0,1\}^S \to \{0,1\}^S$ such that for any $y \in \{0,1\}^S$ and any $t \in S$:

$$\mathrm{flip}_s[y](t) = \begin{cases} 1 - y_t & \text{if } t = s \\ y_t & \text{otherwise} \end{cases} \;\; . \tag{6.42}$$

Algorithm 2 Greedy local search w.r.t. transformations that change one variable at a time is defined by the recursion below. In the context of graphical models and probabilistic inference, this algorithm is also known as *iterated conditional modes*, or ICM (Besag, 1986).

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

---

choose $s \in \operatorname*{argmin}_{s' \in S} \; H_\theta(x, \mathrm{flip}_{s'}[y]) - H_\theta(x, y)$
if $H_\theta(x, \mathrm{flip}_s[y]) < H_\theta(x, y)$
    $y' := \mathrm{icm}(\mathrm{flip}_s[y])$
else
    $y' := y$

Message passing

The inference problem

$$\operatorname*{argmin}_{y \in \{0,1\}^S} \; \sum_{f \in F} h_{f\theta}(x_f, y_{S(f)}) \tag{6.43}$$

consists in computing the minimum of a sum of of functions. This problem is analogous to that of computing the sum of a product of functions (Section 6.3.6) 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 15 (Kschischang et al. (2001)) For any variable node $s \in S$ and any factor node $f \in F$, the functions

$$\mu_{s \to f}, \mu_{f \to s} \colon \{0,1\} \to \mathbb{R} \;\; , \tag{6.44}$$

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

$$\mu_{s \to f}(y_s) = \sum_{f' \in F(s) \setminus \{f\}} \mu_{f' \to s}(y_s) \tag{6.45}$$

$$\mu_{f \to s}(y_s) = \min_{y_{S(f) \setminus \{s\}}} h_{f\theta}(x_f, y_{S(f)}) + \sum_{s' \in S(f) \setminus \{s\}} \mu_{s' \to f}(y_{s'}) \tag{6.46}$$

Lemma 11 *If the factor graph is acyclic, messages are defined recursively by* (6.45) *and* (6.46)*, beginning with the messages from leaves. Moreover, for any $s \in S$:*

$$\min_{y \in \{0,1\}^S} \; \sum_{f \in F} h_{f\theta}(x_f, y_{S(f)}) = \min_{y_s \in \{0,1\}} \; \sum_{f' \in F(s)} \mu_{f' \to s}(y_s)$$

PROOF Analogous to Lemma 10.

For conditional graphical models whose factor graph is acyclic, the inference problem can be solved efficiently by means of message passing, by Lemma 11.

For conditional graphical models whose factor graph is cyclic, the definition of messages by (6.45) and (6.46) is cyclic as well. The inference problem cannot be solved by message passing in general. A heuristic without guarantee of correctness or even convergence is to initialize all messages as constant zero and update messages according to some schedule, e.g., synchronously. This heuristic is also known as *loopy belief propagation* and has proven suitable for some applications.

Chapter 7

Clustering

7.1 Decompositions and multicuts

This section is concerned with learning and inferring decompositions (clusterings) of a graph. We introduce some terminology of Horňáková et al. (2017):

Definition 16 Let $G = (A, E)$ be any graph. A subgraph $G' = (A', E')$ of G is called a *component* of G iff G' is non-empty, node-induced[1] and connected[2]. A partition Π of the node set A is called a *decomposition* of G iff, for every $U \in \Pi$, the subgraph $(U, E \cap \binom{U}{2})$ of G induced by U is connected (and thus a component of G).

For any graph G, we denote by D_G the set of all decompositions of G. Useful in the study of decompositions are the multicuts of a graph:

Definition 17 For any graph $G = (A, E)$, a subset $M \subseteq E$ of edges is called a *multicut* of G iff, for every cycle $C \subseteq E$ of G, we have $|C \cap M| \neq 1$.

For any graph G, we denote by M_G the set of all multicuts of G. For any decomposition of a graph G, the set of those edges that straddle distinct components is a multicut of G. This multicut is said to be induced by the decomposition. In fact, the map from decompositions to induced multicuts is a bijection from D_G to M_G (Horňáková et al., 2017, Lemma 2). This bijection allows us to state the problem of learning and inferring decompositions as one of learning and inferring multicuts.

The characteristic function $y\colon E \to \{0, 1\}$ of a multicut $y^{-1}(1)$ decides, for every edge $\{a, a'\} = e \in E$, whether the incident nodes belong to the same component ($y_e = 0$) or distinct components ($y_e = 1$). By the definition of a multicut, these decisions are not necessarily independent. More specifically:

Lemma 12 *For any graph $G = (V, E)$ and any $y\colon E \to \{0, 1\}$, the set $y^{-1}(1)$ of those edges that are mapped to 1 is a multicut of G iff the following inequalities are satisfied:*

$$\forall C \in \text{cycles}(G)\ \forall e \in C\colon \quad y_e \leq \sum_{e' \in C \setminus \{e\}} y_{e'} \tag{7.1}$$

Exercise 7 *a) Prove Lemma 12.*

b) Show that it is sufficient in (7.1) *to consider only chordless cycles.*

[1] I.e. $E' = E \cap \binom{A'}{2}$

[2] A component is not necessarily maximal w.r.t. the subgraph relation.

Now that we have a finite set E, decisions $y\colon E \to \{0,1\}$ and constraints (7.1), we can state the problem of learning and inferring multicuts as a learning and inference problem (4.1) with

$$S = E \tag{7.2}$$

$$\mathcal{Y} = \left\{ y\colon S \to \{0,1\} \;\middle|\; \forall C \in \text{cycles}(G)\ \forall e \in C\colon y_e \leq \sum_{e' \in C \setminus \{e\}} y_{e'} \right\} \tag{7.3}$$

7.2 Linear functions

7.2.1 Data

Throughout Section 7.2, we consider some graph $G = (A, E)$ and constrained data $(S, X, x, \mathcal{Y})$ with $S = E$, as in (7.2), $\mathcal{Y}$ defined as in (7.3), and $X = \mathbb{R}^V$ with some finite, non-empty set V. As a special case, we consider labeled data, i.e., $\mathcal{Y} = \{y\}$ with y satisfying the constraints (7.1).

7.2.2 Familiy of functions

Throughout Section 7.2, we consider linear functions. More specifically, we consider $\Theta = \mathbb{R}^V$ and $f : \Theta \to \mathbb{R}^X$ such that

$$\forall \theta \in \Theta\ \forall \hat{x} \in \mathbb{R}^V\colon \quad f_\theta(\hat{x}) = \langle \theta, \hat{x} \rangle \ . \tag{7.4}$$

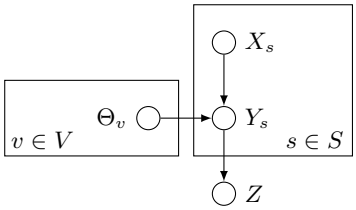
7.2.3 Probabilistic model

Random variables

- For any $\{a, a'\} \in S$, let $X_{\{a,a'\}}$ be a random variable whose realization is a vector $x_{\{a,a'\}} \in \mathbb{R}^V$, called the *attribute vector* of the pair $\{a, a'\}$.
- For any $\{a, a'\} \in S$, let $Y_{\{a,a'\}}$ be a random variable whose realization is a binary number $y_{\{a,a'\}} \in \{0,1\}$, called the *decision* of assigning a and a' to distinct components
- For any $v \in V$, let Θ_v be a random variable whose realization is a real number $\theta_v \in \mathbb{R}$, called a *parameter*
- Let Z be a random variable whose realization is a subset $z \subseteq \{0,1\}^S$. We are interested in $z = \mathcal{Y}$, a characterization of all multicuts (and hence, decompositions) of G

Conditional independence assumptions

We assume a probability distribution that factorizes according to the Bayesian net depicted below.



Factorization

These conditional independence assumptions imply the following factorizations:

- Firstly:

$$P(X, Y, Z, \Theta) = P(Z \mid Y) \prod_{s \in S} P(Y_s \mid X_s, \Theta) \prod_{s \in S} P(X_s) \prod_{v \in V} P(\Theta_v) \tag{7.5}$$

- Secondly:

$$\begin{aligned} P(\Theta \mid X, Y, Z) &= \frac{P(X, Y, Z, \Theta)}{P(X, Y, Z)} \\ &= \frac{P(Z \mid Y)\, P(Y \mid X, \Theta)\, P(X)\, P(\Theta)}{P(Z \mid X, Y)\, P(X, Y)} \\ &= \frac{P(Z \mid Y)\, P(Y \mid X, \Theta)\, P(X)\, P(\Theta)}{P(Z \mid Y)\, P(X, Y)} \\ &= \frac{P(Y \mid X, \Theta)\, P(X)\, P(\Theta)}{P(X, Y)} \\ &\propto P(Y \mid X, \Theta)\, P(\Theta) \\ &= \prod_{s \in S} P(Y_s \mid X_s, \Theta) \prod_{v \in V} P(\Theta_v) \end{aligned} \tag{7.6}$$

- Thirdly,

$$\begin{aligned} P(Y \mid X, Z, \theta) &= \frac{P(X, Y, Z, \Theta)}{P(X, Z, \Theta)} \\ &= \frac{P(Z \mid Y)\, P(Y \mid X, \Theta)\, P(X)\, P(\Theta)}{P(X, Z, \Theta)} \\ &\propto P(Z \mid Y)\, P(Y \mid X, \Theta) \\ &= P(Z \mid Y) \prod_{s \in S} P(Y_s \mid X_s, \Theta) \end{aligned} \tag{7.7}$$

Forms

Here, we consider:

- The *logistic distribution*

$$\forall s \in S: \qquad p_{Y_s \mid X_s, \Theta}(1) = \frac{1}{1 + 2^{-f_\theta(x_s)}} \tag{7.8}$$

- A $\sigma \in \mathbb{R}^+$ and the *normal distribution*:

$$\forall v \in V: \qquad p_{\Theta_v}(\theta_v) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\theta_v^2 / 2\sigma^2} \tag{7.9}$$

- A uniform distribution on a subset:

$$\forall z \subseteq \{0,1\}^S: \quad p_{Z \mid Y}(z) \propto \begin{cases} 1 & \text{if } y \in z \\ 0 & \text{otherwise} \end{cases} \tag{7.10}$$

  Note that $p_{Z \mid Y}(\mathcal{Y})$ is non-zero iff $y^{-1}(1)$ is a multicut and hence defines a decomposition of G.

7.2.4 Learning problem

Corollary 1 *Estimating maximally probable parameters θ, given attributes x and labels y, i.e.,*

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

is identical to the supervised learning problem w.r.t. L, R and λ such that

$$\forall r \in \mathbb{R}\ \forall \hat{y} \in \{0,1\}: \quad L(r, \hat{y}) = -\hat{y}r + \log\left(1 + 2^r\right) \tag{7.11}$$

$$\forall \theta \in \Theta: \qquad R(\theta) = \|\theta\|_2^2 \tag{7.12}$$

$$\lambda = \frac{\log e}{2\sigma^2} \tag{7.13}$$

7.2.5 Inference problem

Corollary 2 *For any constrained data as defined above and any $\theta \in \mathbb{R}^V$, the inference problem has the form of* CORRELATION-CLUSTERING*, i.e.*

$$\min_{y\colon S \to \{0,1\}} \quad \sum_{\{a,a'\} \in S} \left(-\langle \theta, x_{\{a,a'\}} \rangle\right) y_{\{a,a'\}} \tag{7.14}$$

$$\text{subject to} \quad \forall C \in \text{cycles}(G)\ \forall e \in C\colon \quad y_e \leq \sum_{e' \in C \setminus \{e\}} y_{e'} \ . \tag{7.15}$$

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

Lemma 13 (Bansal et al. (2004)) CORRELATION-CLUSTERING *is* NP*-hard.*

Bansal et al. (2004) establish NP-hardness of CORRELATION-CLUSTERING by a reduction of k-TERMINAL-CUT whose NP-hardness is an important result of Dahlhaus et al. (1994).

7.2.6 Inference algorithm

Below, we discuss three local search algorithms for CORRELATION-CLUSTERING. For simplicity, we define $c : S \to \mathbb{R}$ such that

$$\forall \{a, a'\} \in S\colon \quad c_{\{aa'\}} = -\langle \theta, x_{\{a,a'\}} \rangle \tag{7.16}$$

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

$$\forall y \in \{0,1\}^S\colon \quad \varphi(y) = \sum_{\{a,a'\} \in S} c_{\{a,a'\}}\, y_{\{a,a'\}} \tag{7.17}$$

Greedy joining

The greedy joining algorithm starts from any initial decomposition and searches for decompositions with lower objective value by joining pairs of components recursively. By this procedure, components can only grow, and the number of components decreases by precisely one in every step. Thus, one typically starts from the finest decomposition Π_0 of $G = (A, E)$ into one-elementary components.

Definition 18 For any graph $G = (A, E)$ and any disjoint sets $B, C \subseteq A$, the pair $\{B, C\}$ is called *neighboring* in G iff there exist nodes $b \in B$ and $c \in C$ such that $\{b, c\} \in E$.

For any decomposition Π of a graph $G = (A, E)$, we define

$$\mathcal{E}_\Pi = \left\{ \{B, C\} \in \tbinom{\Pi}{2} \,\middle|\, \exists b \in B\, \exists c \in C\colon \{b, c\} \in E \right\} \ . \tag{7.18}$$

Definition 19 For any decomposition Π of $G = (A, E)$ and any $\{B, C\} \in \mathcal{E}_\Pi$, let $\text{join}_{BC}[\Pi]$ be the decomposition of G obtained by joining the sets B and C in Π, i.e.

$$\text{join}_{BC}[\Pi] = (\Pi \setminus \{B, C\}) \cup \{B \cup C\} \ . \tag{7.19}$$

Algorithm 3 The greedy joining algorithm is defined by the recursion below.

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

---

choose $\{B, C\} \in \underset{\{B', C'\} \in \mathcal{E}_\Pi}{\text{argmin}} \ \varphi(y^{\text{join}_{B'C'}[\Pi]}) - \varphi(y^\Pi)$
if $\varphi(y^{\text{join}_{BC}[\Pi]}) - \varphi(y^\Pi) < 0$
    $\Pi' := \text{greedy-joining}(\text{join}_{BC}[\Pi])$
else
    $\Pi' := \Pi$

Exercise 8 *a) Write the difference $\varphi(y^{\text{join}_{B'C'}[\Pi]}) - \varphi(y^\Pi)$ in terms of the c defined in* (7.16)*.*
b) Implement greedy joining efficiently.
c) Establish a bound on the time complexity of your implementation.

Greedy moving

The greedy moving algorithm starts from any initial decomposition, e.g., the fixed point of greedy joining. It seeks to lower the objective value by recursively moving individual nodes from one component to a neighboring component, or to a new component. When a node is moved to a new component, the number of components can increase. When the last node is moved from a component, the number of components decreases.

Definition 20 For any graph G, a component G' of G is called *maximal* iff there is no subgraph of G that is both a strict supergraph of G' and a component of G.

Definition 21 For any graph $G = (A, E)$ and any decomposition Π of G, the decomposition Π is called *coarsest* iff, for every $U \in \Pi$, the component $(U, E \cap \binom{U}{2})$ induced by U is maximal.

Lemma 14 *For any graph G, the coarsest decomposition is unique.*

For any graph G, we denote the coarsest decomposition by Π^*_G.

Definition 22 For any graph $G = (A, E)$, any decomposition Π of A and any $a \in A$, choose U_a to be the unique $U_a \in \Pi$ such that $a \in U_a$, and let

$$\mathcal{N}_a = \{\varnothing\} \cup \{W \in \Pi \,|\, a \notin W \wedge \exists w \in W \colon \{a, w\} \in E\} \tag{7.20}$$

$$G_a = \left(U_a \setminus \{a\},\, E \cap \tbinom{U_a \setminus \{a\}}{2}\right) \tag{7.21}$$

For any $U \in \mathcal{N}_a$, let $\mathrm{move}_{aU}[\Pi]$ the decomposition of A obtained by moving the node a to the set U, i.e.

$$\mathrm{move}_{aU}[\Pi] = \Pi \setminus \{U_a, U\} \cup \{U \cup \{a\}\} \cup \Pi^*_{G_a} \ . \tag{7.22}$$

Algorithm 4 The greedy moving algorithm is defined by the recursion below.

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

choose $(a, U) \in \underset{(a', U') \in A \times (\Pi \cup \{\varnothing\})}{\operatorname{argmin}} \varphi(y^{\mathrm{move}_{a'U'}[\Pi]}) - \varphi(y^{\Pi})$
if $\varphi(y^{\mathrm{move}_{aU}[\Pi]}) - \varphi(y^{\Pi}) < 0$
    $\Pi' := \text{greedy-moving}(\mathrm{move}_{aU}[\Pi])$
else
    $\Pi' := \Pi$

Exercise 9 *a) Write the difference $\varphi(y^{\mathrm{move}_{aU}[\Pi_t]}) - \varphi(y^{\Pi_t})$ in terms of the c defined in* (7.16).
b) Implement greedy moving.

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

Both algorithms discussed above terminate as soon as no transformation (join and move, resp.) leads to a partition with strictly lower objective value. This can be sub-optimal in case transformations that increase the objective value at one point in the recursion can lead to transformations that decrease the objective value at later points in the recursion and the decrease overcompensates the increase. A generalization of greedy local search introduced by Kernighan and Lin (1970) can escape such sub-optimal fixed points. It is applied to greedy moving below.

Algorithm 5 The greedy moving algorithm generalized by the technique of Kernighan and Lin (1970) is defined by the recursion below.

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

$\Pi_0 := \Pi$
$\delta_0 := 0$
$A_0 := A$
$t := 0$
repeat (build sequence of moves)
  choose $(a_t, U_t) \in \underset{(a,U) \in A_t \times (\Pi \cup \{\varnothing\})}{\operatorname{argmin}} \varphi(y^{\operatorname{move}_{aU}[\Pi_t]}) - \varphi(y^{\Pi_t})$
  $\Pi_{t+1} := \operatorname{move}_{a_t U_t}[\Pi_t]$
  $\delta_{t+1} := \varphi(y^{\Pi_{t+1}}) - \varphi(y^{\Pi_t}) < 0$
  $A_{t+1} := A_t \setminus \{a_t\}$ (move a_t only once)
  $t := t + 1$
until $A_t = \varnothing$
$\hat{t} := \min \underset{t' \in \{0,\ldots,|A|\}}{\operatorname{argmin}} \sum_{\tau=0}^{t'} \delta_\tau$ (choose sub-sequence)
if $\sum_{\tau=0}^{\hat{t}} \delta_\tau < 0$
  $\Pi' := \text{greedy-moving-kl}(\Pi_{\hat{t}})$ (recurse)
else
  $\Pi' := \Pi$ (terminate)

Exercise 10 *a) Implement greedy moving using the technique of Kernighan and Lin (1970).*
b) Generalize the greedy joining algorithm using the technique of Kernighan and Lin (1970).
c) Implement greedy joining using the technique of Kernighan and Lin (1970).

Bibliography

Bansal, N., A. Blum, and S. Chawla
2004. Correlation clustering. *Machine Learning*, 56(1–3):89–113.

Besag, J.
1986. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B (Methodological)*, 48(3):259–302.

Chopra, S. and M. Rao
1993. The partition problem. *Mathematical Programming*, 59:87–115.

Dahlhaus, E., D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis
1994. The complexity of multiterminal cuts. *SIAM Journal on Computing*, 23(4):864–894.

Demaine, E. D., D. Emanuel, A. Fiat, and N. Immorlica
2006. Correlation clustering in general weighted graphs. *Theoretical Computer Science*, 361(2):172–187.

Horňáková, A., J.-H. Lange, and B. Andres
2017. Analysis and optimization of graph decompositions by lifted multicuts. In *ICML*.

Kernighan, B. W. and S. Lin
1970. An efficient heuristic procedure for partitioning graphs. *Bell Systems Technical Journal*, 49(2):291–307.

Kschischang, F. R., B. J. Frey, and H.-A. Loeliger
2001. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519.