Introduction to Statistical and Structural Pattern Recognition

Bogdan Savchynskyy

June 28, 2012

Contents

Chapter 1

Bayesian Decision Making

1.1 Basics

- $\mathcal{X} \ni x$ – observation set;
- $\mathcal{K} \ni k$ – the set of object states;
- $\mathcal{D} \ni d$ – set of decisions;
- $W : \mathcal{K} \times \mathcal{D} \to \mathbb{R}$ - penalty (loss) function;
- $p(k, x)$ - joint probability distribution.

Bayesian strategy $q : \mathcal{X} \to \mathcal{D}$:

$$\text{Risk } R(q) = \sum_{x\in\mathcal{X}} \sum_{k\in\mathcal{K}} p(k, x)W(k, q(x)) \to \min_{q}$$

i.e. for a given $x \in \mathcal{X}$ holds $p(k, x) = p(x)p(k|x)$ and thus

$$q(x) = d^* = \arg\min_{d\in\mathcal{D}} \sum_{k\in\mathcal{K}} p(x, k)W(k, d) = \arg\min_{d\in\mathcal{D}} p(x) \sum_{k\in\mathcal{K}} p(x|k)W(k, d) = \arg\min_{d\in\mathcal{D}} \sum_{k\in\mathcal{K}} p(x|k)W(k, d) \quad (1.1)$$

Take-home formula:

$$d^* = \arg\min_{d\in\mathcal{D}} \sum_{k\in\mathcal{K}} p(k|x)W(k, d)$$

Example 1.1.0.1 (Maximum aposteriory decision)*.* $D = K$,

$$W(k, d) = \begin{cases} 1, & d \neq k \\ 0, & d = k \end{cases} \quad (1.2)$$

Then

$$k^* = \arg\min_{k\in\mathcal{K}} \sum_{k'\neq k} p(k'|x) = \arg\min_{k\in\mathcal{K}} (1 - p(k|x)) = \arg\max_{k\in\mathcal{K}} p(k|x)\,. \quad (1.3)$$

Example 1.1.0.2 (Bayesian strategy with possible rejection). $\mathcal{K} = \{1, 2\}$, $\mathcal{D} = \mathcal{K} \cup \{\sharp\}$, $W(k,d) = \begin{cases} 0, & d = k \\ 1, & d \neq k, d \in \mathcal{K} \\ \varepsilon, & d = \sharp \end{cases}$

$$d^* = \arg\min_{d \in \mathcal{D}} \sum_{k \neq d} p(k|x) W(k,d) = \arg\min \begin{cases} 1 - p(d|x), & d \neq \sharp \\ \varepsilon, & d = \sharp \end{cases}$$

Discussion:

- $\varepsilon = 0$ – always refuse;
- $\varepsilon >= 1$ - never refuse;

thus $\varepsilon \in (0, 1)$.

Example 1.1.0.3 (Face Recognition). $\mathcal{K} = \mathcal{K}^+ \cup \mathcal{K}^-$, $\mathcal{D} = \{+, -\}$, $W(k,d) = \begin{cases} a, & d = -, & k \in \mathcal{K}^+ \\ b, & d = +, & k \in \mathcal{K}^- \\ 0, & & otherwise \end{cases}$

$$d = \arg\min_{d \in \mathcal{D}} \sum_{k \in \mathcal{K}} p(k|x) W(k,d) = \arg\min \begin{cases} a \cdot \sum_{k \in \mathcal{K}^+} p(k|x), & d = - \\ b \cdot \sum_{k \in \mathcal{K}^-} p(k|x), & d = + \end{cases}$$

Compare to the result of Example 1.1.0.1.

1.2 Deterministic vs. Random Strategies

Bayesian strategy is deterministic, i.e. $q(x)$ is selected deterministically even though the same x can correspond to different k. Let us show, that probabilistic strategy would be worse than the deterministic one. Let $q_r \colon X \times \mathcal{D} \to \mathbb{R}$ – randomized strategy (probability distribution). Then

$$R_{rand}(q_r) = \sum_{x \in \mathcal{X}} \sum_{k \in \mathcal{K}} p(k,x) \sum_{d \in \mathcal{D}} q_r(d|x) W(k,d)$$

Proposition 1.2.0.1. *For any randomized q_r exists deterministic q such that $R_{rand}(q_r) \geq R(q)$.*

$\rhd$

$$\begin{aligned} R_{rand}(q_r) = \sum_{x \in \mathcal{X}} \sum_{d \in \mathcal{D}} q_r(d|x) \sum_{k \in \mathcal{K}} p(k,x) W(k,d) \\ \geq \sum_{x \in \mathcal{X}} \min_{d \in \mathcal{D}} \sum_{k \in \mathcal{K}} p(k,x) W(k,d) = R(q)\,, \quad (1.4) \end{aligned}$$

where $q(x) = \arg\min_{d \in \mathcal{D}} \sum_{k \in \mathcal{K}} p(x,k) W(k,d)$

$\lhd$

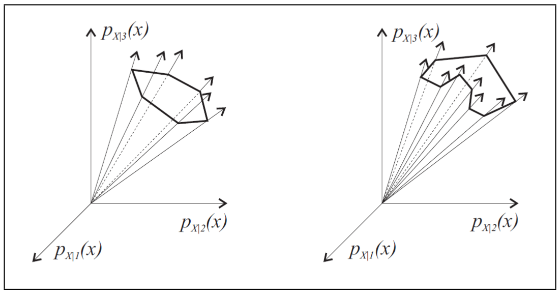


Figure 1.1: Convex and non-convex cones in $\mathbb{R}^3$. The image is taken from [1]

1.3 Convexity Property of Bayesian Strategies

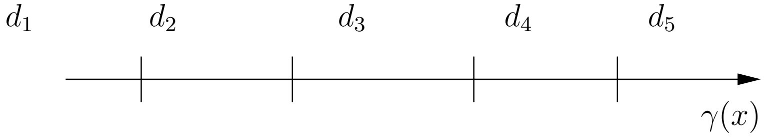
Let $\mathcal{K} = \{1, 2\}$. Then

$$\begin{aligned} q(x) &= \arg\min_{d\in\mathcal{D}} \left(p_{\mathcal{X},\mathcal{K}}(x,1)W(1,d) + p_{\mathcal{X},\mathcal{K}}(x,2)W(2,d)\right) \\ &= \arg\min_{d\in\mathcal{D}} \big(p_{\mathcal{X}/\mathcal{K}}(x/1)p_{\mathcal{K}}(1)W(1,d) + p_{\mathcal{X}/\mathcal{K}}(x/2)p_{\mathcal{K}}(2)W(2,d)\big) \\ &= \arg\min_{d\in\mathcal{D}} \left(\frac{p_{\mathcal{X}/\mathcal{K}}(x/1)}{p_{\mathcal{X}/\mathcal{K}}(x/2)} p_{\mathcal{K}}(1)W(1,d) + p_{\mathcal{K}}(2)W(2,d)\right) \\ &= \arg\min_{d\in\mathcal{D}} \left(\gamma(x)c_1(d) + c_2(d)\right) . \end{aligned} \tag{1.5}$$

Thus

$$\gamma(x)c_1(d^*) + c_2(d^*) \leq \gamma(x)c_1(d) + c_2(d), \forall d \in \mathcal{D}\backslash d^* \tag{1.6}$$

A solution of this system of liner inequalities is a convex set (possibly empty). The only non-trivial convex set in $\mathbb{R} \ni \gamma(x)$ is an interval.



Definition 1.3.0.1. The function $\gamma(x) = \frac{p_{\mathcal{X}/\mathcal{K}}(x/1)}{p_{\mathcal{X}/\mathcal{K}}(x/2)}$ is very important in decision making and has a special name *likelihood ratio.*

When additionally $D = K$ all strategies need only to compare $\gamma(x)$ to a given threshold. Let us consider the general case ($\mathcal{K} \neq \{1, 2\}$):

Proposition 1.3.0.2. *Let $q(x)$ be a Bayesian strategy and $\pi(x) = (p(x|k),\ k \in \mathcal{K})$ be points in a positive orthant of $\Pi = \mathbb{R}^{|\mathcal{K}|}$. Then among optimal strategists exists a strategy, that sets of points $\{\pi(x),\ q(x) = d\}$ are convex cones (Each convex cone corresponds to a certain decision d).*

$\triangleright$Let $n(d)$ - an index of the decision d. Than optimal strategy satisfies

$$\sum_{k\in\mathcal{K}} p(x|k)p(k)W(k,d^*) \leq \sum_{k\in\mathcal{K}} p(x|k)p(k)W(k,d),\ d \in \mathcal{D}\backslash d^* \tag{1.7}$$

$$\sum_{k\in\mathcal{K}} \pi_k(x)p(k)W(k,d^*) < \sum_{k\in\mathcal{K}} \pi_k(x)p(k)W(k,d),\ d \in \mathcal{D}\backslash d^* \tag{1.8}$$

Since constraints hold also for $\pi(x) = \alpha\pi(x)$, thus they define a cone. Since constraints are linear this cone is convex $\lhd$

Corollary 1.3.0.1.

- *Cones can be split by a hyperplane containing origin - $(\mathcal{K}-1)$-dimensional analogue of $\gamma(x)$;*
- *In conditions of Example 1.1.0.2 (possible rejection) - refuse to make any decision in case $p(x|k) < \theta\ \forall k \in \mathcal{K}$ - this not a bayesian strategy! Since $\{\pi_k(x) < \theta\}$ is not a cone.*
- *Example* 1.3.0.4. $\mathcal{K} = \{1,2,3,4\}$, $\mathcal{D} = \{1-2, 3-4\}$. Typical but **incorrect** solution is to compute $\arg\max_k p(k|x)$ and then decide whether it belongs to $1-2$ ar to $3-4$. It is incorrect, because union of two convex cones is not a convex cone anymore. The correct solution is...? :) See also Example 1.1.0.3

1.4 Discussion

An **advantage** of the Bayesian theory is its generality. Properties of $\mathcal{X}$, $\mathcal{K}$, $\mathcal{D}$, W are quite general (in fact, the theory is formulated even for general sets, not obligatory finite). They can represent numbers or non-numbers (symbols in abstract alphabet, graphs, sequences, functions, processes - almost anything!). The only numbers are $W(k,d)$ and $p(x,k)$.

Disadvantages:

1. $W(k,d)$ is a *numerical* function. But not all losses are representable as numbers! *Example:* medical diagnostics. Incorrect solution leads not only to additional costs for an operation, but also can be dangerous (lead to death).
2. $p(k,x) = p(k)p(x|k)$. The distribution $p(x|k)$ is always reasonably formulated and constitutes *a model of an object*. But $p(k)$ can be unknown or even have no statistical meaning. Example: radiolocation – is it an enemy airplane or not?
3. $p(x|k;z)$ - distribution depends on unknown (even non-random) parameter z. *Example:* OCR under condition of *unknown* language or/and font.

There is a Non-Bayesian decision theory. The most famous example: *Neyman-Peason* problem.

Bibliography

We mainly follow
[1] Schlesinger M.I., Hlavač V. *Ten Lectures on Statistical and Structural Pattern Recognition.* 2002 (c) Kluwer Academic Publishers.

Another book recommended for reading is:
[2] Richard O. Duda, Peter E. Hart, David G. Stork *Pattern Classification* (2nd Edition) Available on-line:

`http://www.ai.mit.edu/courses/6.891-f99/`

Chapter 2

Two Statistical Models of the Recognized Objects

2.1 Conditionally Independent Features

Let $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_n \ni (x_1, \ldots, x_n)$. Let also

$$p(x/k) = \prod_{i=1}^{n} p(x_i|k).$$

– conditionally independent features.

NB! In general, however

$$p(x) \neq \prod_{i=1}^{n} p(x_i)\,.$$

Let $\mathcal{K} = \{1, 2\}$. Then a decision $d \in \mathcal{D}$ (for a Bayesian problem) should be selected if

$$\theta^d_{min} < \log \frac{p(x|k=1)}{p(x|k=2)} \leq \theta^d_{max}\,. \tag{2.1}$$

Let $\mathcal{X}_i = \{0, 1\}$ $\forall i = 1, \ldots, n$. Then

$$\begin{aligned} \log \frac{p(x|k=1)}{p(x|k=2)} &= \sum_{i=1}^{n} \log \frac{p(x_i|k=1)}{p(x_i|k=2)} \\ &= \sum_{i=1}^{n} x_i \log \frac{p(1|k=1)p(0|k=2)}{p(1|k=2)p(0|k=1)} + \sum_{i=1}^{n} \log \frac{p(0|k=1)}{p(0|k=2)}\,. \end{aligned} \tag{2.2}$$

Thus (2.1) has a form

$$\theta^d_{min} < \sum_{i=1}^{n} \alpha_i x_i \leq \theta^d_{max}\,.$$

In a special case $\mathcal{D} = \mathcal{K}$ the set $\mathcal{X}$ should be splitted to $\mathcal{X}_1 \cup \mathcal{X}_2$ such that

$$x \in \begin{cases} X_1, & \text{if} \sum_{i=1}^{n} \alpha_i x_i \leq \theta \\ X_2, & \text{if} \sum_{i=1}^{n} \alpha_i x_i > \theta \end{cases}$$

Exercise 2.1.0.1. Show that decision strategy has the same form also for the case when $\mathcal{X}_i$ are general finite sets.

2.2 Gaussian Probability Distribution

Let $\mathcal{X} = \mathbb{R}^n$ - we consider probability densities $p(x|k)$ instead of probabilities. Let

$$p(x|k) = C(A(k)) \exp\left(-\frac{1}{2}\langle A(k)(x-\mu),(x-\mu)\rangle\right)$$

– Gaussian distribution. Let $\mathcal{K} = \mathcal{D} = \{1,2\}$. Then

$$\log\frac{p(x|k=1)}{p(x|k=2)} = \log(\frac{C(A(1))}{C(A(2))}) + \frac{1}{2}\left(\langle A(2)(x-\mu),(x-\mu)\rangle - \langle A(1)(x-\mu),(x-\mu)\rangle\right)$$

– quadratic function of x.

Thus recognition strategy again has a form: $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2$ and

$$x \in \left\{ \begin{array}{ll} X_1, & \text{if} \sum_i \sum_j \alpha_{ij} x_i x_j + \sum_i \beta_i x_i \leq \theta \\ X_2, & \text{if} \sum_i \sum_j \alpha_{ij} x_i x_j + \sum_i \beta_i x_i > \theta \end{array} \right. \tag{2.3}$$

Let us introduce new variables $y_l := (x_i x_j)$ for $\tau_l := \alpha_{ij}$ and $y_l := 1$ for $\tau_l := \beta_i$. Then (2.3) can be rewritten as

$$y \in \left\{ \begin{array}{ll} Y_1, & \text{if} \sum_{l=1} \tau_l y_l \leq \theta \\ Y_2, & \text{if} \sum_{l=1} \tau_l y_l > \theta \end{array} \right.$$

Such a technique of variables change is called *straightening of the feature space*.

Bibliography

We mainly follow

[1] Schlesinger M.I., Hlavač V. *Ten Lectures on Statistical and Structural Pattern Recognition.* 2002 (c) Kluwer Academic Publishers.

Chapter 3

Learning in Pattern Recognition

The distribution $p(x, k)$ is often unknown or known up to a parameter a, i.e. $p(x, k; a)$.

3.1 Non-regularized learning

3.1.1 Maximal Likelyhood Estimation (MLE) of Parameters

Other names: *Generative learning.*

Given a *multi-set* $\mathcal{L} = ((x_i, k_i), i = 1, l)$.

$$\begin{aligned}
a^* = (a_k^* \colon k \in \mathcal{K}) &= \arg \max_{a_k \colon k \in \mathcal{K}} \prod_{i=1}^{l} p(k_i) p(x_i | k_i; a_i) \\
&= \arg \max_{a_k \colon k \in \mathcal{K}} \prod_{x \in \mathcal{X}} \prod_{k \in \mathcal{K}} (p(k) p(x|k; a))^{\alpha(x,k)} \\
&= \arg \max_{a_k \colon k \in \mathcal{K}} \sum_{x \in \mathcal{X}} \sum_{k \in \mathcal{K}} \alpha(x, k) \log(p(k) p(x|k; a)) \\
&= \arg \max_{a_k \colon k \in \mathcal{K}} \sum_{k \in \mathcal{K}} \sum_{x \in \mathcal{X}} \alpha(x, k) \log p(x|k; a)
\end{aligned} \tag{3.1}$$

Thus

$$a_k^* = \arg \max_{a_k} \sum_{x \in \mathcal{X}} \alpha(x, k) \log p(x|k; a_k) \,. \tag{3.2}$$

In this case we do not need a priori probabilities $p(k)$ during estimating a_k^*.

Conditions: elements of the training multi-set are i.i.d. from the same distribution as during use of the recognition system. Convergence to true parameter values for big l. However the discribution is often unknown!

Example 3.1.1.1. OCR, a_k - template of the character $k \in \mathcal{K}$.

Example 3.1.1.2. Gaussian distribution - center in the mean value.

Example 3.1.1.3. Geologist.

3.1.2 Learning According to a Non-random Set

Instead of random learning *multi-set* - well-selected, highly probable *set*, which represents well the whole data.

$$a_k^* = \arg\max_{a_k} \min_{x \in \mathcal{X}(k)} p(x|k; a_k)\,,$$

where $\mathcal{X}(k)$– representatives of k-th class.

Example 3.1.2.1. Gaussian distribution - center of the smallest circle containing all training points.

3.1.3 Learning by Minimizing Empirical Risk

Problem is posed as minimization of the *empirical risk* – average loss value over the training set:

$$\hat{R}(a) = \frac{1}{l} \sum_{i=1}^{l} W(k_i, q(a)(x_i)) \to \min_a\ .$$

This formulation is connected to the algorithm (=recognition strategy) q and tries to tune it to achieve minimal loss on the training data.

Example 3.1.3.1. Gaussian distributions, $|\mathcal{K}| = |\mathcal{D}| = 2$ - find a separating hyperplane.

3.2 Discussion: General Properties of Learning Problems

Generalization property. Capacity of strategies. Learning by minimizing an empirical risk and just remembering the whole dataset.

3.3 Regularized Learning

3.3.1 Regularized MLE

Let $p(k) = p(k; a)$. Typically $p(k; a) = C_k \cdot \exp(-\lambda_k \|a\|)$. Then (3.1) takes the form

$$\begin{aligned} a^* = (a_k^*\colon k \in \mathcal{K}) &= \arg \max_{a_k\colon k \in \mathcal{K}} \sum_{k \in \mathcal{K}} \sum_{x \in \mathcal{X}} \alpha(x, k) \log(p(k; a) p(x|k; a_i)) \\ &= \arg \max_{a_k\colon k \in \mathcal{K}} -\lambda_k \|a\| + \sum_{x \in \mathcal{X}} \sum_{k \in \mathcal{K}} \alpha(x, k) \log p(x|k; a) \end{aligned} \tag{3.3}$$

If $p(k; a) = C_k \cdot \exp(-\lambda_k \|a_k\|)$ then

$$a_k^* = \arg\max_{a_k} \sum_{x \in \mathcal{X}} \alpha(x, k)(-\lambda_k \|a_k\| + \log p(x_i|k_i; a_i))\,.$$

– penalization of too complicated parameter values.

3.3.2 Regularized discriminative learning

Let us consider $p(x|k;a) = \frac{1}{C(x)}\exp(-W(k,q(a)(x)))$ (the more penalty the less probable is x). Then from (3.3) follows

$$a^* = (a_k^*, k \in \mathcal{K}) = \arg\max_{a_k, k\in\mathcal{K}} -\tilde{\lambda}\|a\| + \sum_{i=1}^{l} \log p(x_i|k_i;a_i) = \arg\min_{a_k, k\in\mathcal{K}} \lambda\|a\| + \hat{R}(a) \quad (3.4)$$

Bibliography

[1] Schlesinger M.I., Hlavač V. *Ten Lectures on Statistical and Structural Pattern Recognition.* 2002 (c) Kluwer Academic Publishers.
[2] Herbrich R. *Learning Kernel Classifiers.* Theory and Algorithms (MIT,2002)
[3] Duda, Hart *Pattern Classification and Scene Analysis*, 1973
[4] Vapnik V. *Statistical Learning Theory*, 1998
[5] Scholkopf B., Smola A.J. *Learning with Kernels. Support Vector Machines, Regularization, Optimization, and Beyond* 2001

Chapter 4

Linear Discriminant Analysis

The core problem of this topic is

Given two sets $\mathcal{X}_1$ and $\mathcal{X}_2$ in $\mathbb{R}^N$. Find such $\alpha \in \mathbb{R}^N$, that

$$\langle \alpha, x \rangle > 0,\ x \in \mathcal{X}_1 \tag{4.1}$$

$$\langle \alpha, x \rangle < 0,\ x \in \mathcal{X}_2\,. \tag{4.2}$$

4.1 Motivation for LDA

- Bayesian decision strategies in probability space can be formulated in terms of discriminating hyperplanes.
- More complicated strategies (having polynomial form) can be represented as hyperplanes in higher dimensional spaces using straightening of the feature space.
- Two considered models (and there are more!) lead to linear discriminative strategies.

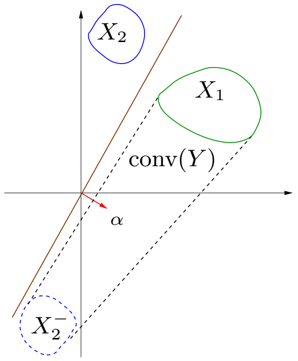
4.2 Equivalent Formulations in LDA

- affine vs. linear

  $\langle \alpha, x \rangle > b,\ x \in X_1$
  $\langle \alpha, x \rangle < b,\ x \in X_2$
  $\Rightarrow \mathbb{R}^{n+1} \ni y = (x, -1), \beta = (\alpha, b)$
  $\langle \beta, y \rangle > 0,\ y \in Y_1 = (X_1, -1)$
  $\langle \beta, y \rangle < 0,\ y \in Y_2 = (X_2, -1)$

- one set vs. two sets

  $\langle \alpha, x \rangle > 0,\ x \in X_1$
  $\langle \alpha, x \rangle < 0,\ x \in X_2$
  $\Rightarrow y = \begin{cases} x, & x \in X_1 \\ -x, & x \in X_2 \end{cases}$
  $\langle \alpha, y \rangle > 0,\ y \in Y = X_1 \bigcup X_2^-$
  Another direction is straightforward.



- necessary and sufficient condition for existence of the solution α

  convex hull does contain an origin: $\bar{0} \notin \operatorname{conv}(Y)$

- Fischer classifiers (template matching)

  $\mathcal{X} \subset \mathbb{R}^n, \quad \mathcal{K} = \{1, \dots, K\}$
  $\langle \alpha_k, x \rangle > \langle \alpha_j, x \rangle\,, \; x \in \mathcal{X}_k,\; j \neq k$
  Introduce $\beta = (\alpha_1, \dots, \alpha_K) \in \mathbb{R}^{nK}$
  and

  $$\mathcal{Y} = \{(0, \dots, \underbrace{x}_{k}, 0, \dots, 0, \underbrace{-x}_{j}, 0, \dots, 0), x \in \mathcal{X}_k, j \in \mathcal{K}\backslash\{k\},\; k \in \mathcal{K}\}$$

  Thus $\langle \beta, y \rangle > 0,\; y \in \mathcal{Y}$

4.3 Perceptron

Maybe the simplest algorithm for the LDA problem.

Perceptron algorithm

```
$0\colon w := \bar{0}$
$t\colon$ while $\exists x \in \mathcal{X}\;\; \langle w, x \rangle \leq 0 \quad w+ = x$
$t \leq D^2/\varepsilon^2$
$D = \sup_{x \in \mathcal{X}} \|x\|_2,\; \varepsilon = \min_{x \in \operatorname{conv}(\mathcal{X})} \|x\|.$
```

$\triangleright$

$$\|w_{t+1}\|^2 = \|w_t + x_t\|^2 = \|w_t\|^2 + 2\underbrace{\langle w_t, x_t \rangle}_{\leq 0} + \|x_t\|^2 \leq \|w_t\|^2 + \|x_t\|^2 \leq \|w_t\|^2 + D^2$$

Hence

$$\|w_{t+1}\|^2 \leq t \cdot D^2 \tag{4.3}$$

Let $w^* = \arg\min_{x \in \operatorname{conv}(\mathcal{X})} \|x\|$, thus $\|w^*\| = \varepsilon$ and $\left\langle \frac{w^*}{\|w^*\|}, x \right\rangle \geq \|w^*\| = \varepsilon,\; x \in \mathcal{X}$. Thus

$$\left\langle \frac{w^*}{\|w^*\|}, w_{t+1} \right\rangle = \left\langle \frac{w^*}{\|w^*\|}, w_t + x_t \right\rangle = \left\langle \frac{w^*}{\|w^*\|}, w_t \right\rangle + \left\langle \frac{w^*}{\|w^*\|}, x_t \right\rangle \geq \left\langle \frac{w^*}{\|w^*\|}, w_t \right\rangle + \varepsilon$$

From that follows:

$$\left\langle \frac{w^*}{\|w^*\|}, w_{t+1} \right\rangle \geq t \cdot \varepsilon$$

and

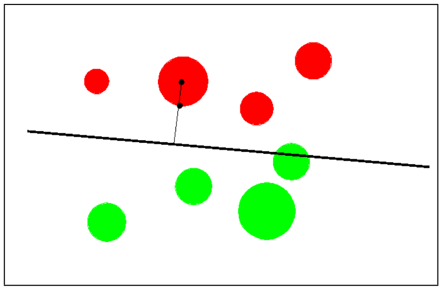
$$t \cdot \varepsilon \leq \left\langle \frac{w^*}{\|w^*\|}, w_{t+1} \right\rangle \leq \|w_{t+1}\| \cdot \left\| \frac{w^*}{\|w^*\|} \right\| \leq \|w_{t+1}\|$$

Hence

$$\|w_{t+1}\|^2 \geq t^2 \cdot \varepsilon^2 \tag{4.4}$$

Divide (4.3) to (4.4) and obtain $t \leq D^2/\varepsilon^2$. $\triangleleft$

- **Main property.** $|\mathcal{X}|$ can be arbitrary large, even infinite, continuum. Important is only existence of the oracle which finds $x\colon \langle w, x \rangle < 0$.
  Example: Separating 2 sets of balls



- **Size does not matter.** If w is a solution, then aw is also a solution for any $a > 0$.
- **Dual view.** $w = \sum_{i=1}^{N} x^i = \sum_{x\in\mathcal{X}} n(x)\cdot x$. Thus $C \cdot w \in \operatorname{conv}(\mathcal{X})$ for some $C > 0$. $\triangleright$Select $C = 1/\sum_{x\in\mathcal{X}} n(x))$ $C\cdot w = \sum_{x\in\mathcal{X}} \alpha(x)\cdot x$,
  $\alpha(x) = n(x)/\sum_{x\in\mathcal{X}} n(x)$, $\alpha \in \Delta_{\mathcal{X}}$ – simplex in $\mathbb{R}^{|\mathcal{X}|}$. $\triangleleft$
- **Dual perceptron algorithm:** $0\colon n(x) := 0,\ x\in\mathcal{X}$
  $i\colon$ while $\exists x \in \mathcal{X}\ \ \langle w, x\rangle < 0 \quad n(x){+}=1$.
- **Dual view to the decision rule:**

$$\langle w, x\rangle = \left\langle \sum_{x\in\mathcal{X}} n(x)\cdot x, z \right\rangle = \sum_{x\in\mathcal{X}} n(x)\left\langle x, z\right\rangle = \sum_{x\in\mathcal{X}} n(x)K(x,z)\,. \tag{4.5}$$

There is no need to compute a straightening mapping $x = (x_1, \cdots, x_n) \to \phi(x) = (\phi_1(x), \ldots, \phi_N(x))$ and then a scalar product in R^N ($N >> n$ typically) - it is just enough to know how to compute $\kappa(x,z)$ – scalar product in the original space=*kernel*.

4.4 Kernels

$\kappa\colon \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ is *a kernel* if a straightening mapping $x = (x_1, \cdots, x_n) \to \phi(x) = (\phi_1(x), \ldots, \phi_N(x))$ and scalar product $\langle\cdot,\cdot\rangle$ in $\mathbb{R}^N$ exist such that $\kappa(x,z) = \langle\phi(x), \phi(z)\rangle$.

Necessary conditions:

1. $\kappa(x,x) \geq 0$
2. $\kappa(x,z) = \kappa(z,x)$
3. $\kappa(x,z)^2 = \langle\phi(x),\phi(z)\rangle^2 \leq \|\phi(x)\|^2\|\phi(z)\|^2 = \langle\phi(x),\phi(x)\rangle\langle\phi(z),\phi(z)\rangle = \kappa(x,x)\kappa(z,z)$

These conditions are NOT SUFFICIENT!

Example 4.4.0.1. Consider κ - symmetric, but not positive semidefinite.

Definition 4.4.0.1. A square matrix κ is called positive semidefinite if all its eigenvalues are real and non-negative

Theorem 4.4.0.1. *A square $x \times n$ matrix κ is positive semidefinite iff $\forall x \in \mathbb{R}^n$ holds $x^T\kappa x \geq 0$.*

Theorem 4.4.0.2. *Let $\mathcal{X}$ be a non-empty set. A function $\kappa\colon \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ is kernel iff $\forall m \in \mathbb{N}$ and all $x_1, \ldots, x_m \in \mathcal{X}$ it gives rise to a symmetric positive semidefinite matrix $\kappa = (\kappa(x_i, x_j))$.*

Remark 4.4.0.1. Symmetric positive semidefinite means it can be represented (as any symmetric) as $\kappa = V\Lambda V^T$, where $\Lambda = (\lambda_j)_{j=1}^{|\mathcal{X}|}$ - diagonal matrix with eigenvalues of κ, V - ortogonal and $\Lambda_{jj} \geq 0$. Let us denote $\lambda_t = \Lambda_{tt}$ and $v_t = (v_{ti})_{i=1}^n$ be columns of V. Let $\mathcal{X}$ be finite. Let us construct

$$\phi(x_i) = \left(\sqrt{\lambda_t}v_{ti}\right)_{i=1}^n \in \mathbb{R}^n,\ i = 1, \ldots, n$$

Then

$$\langle\phi(x_i), \phi(x_j)\rangle = \sum_{t=1}^{n} \lambda_t v_{ti} v_{tj} = (V\Lambda V^T)_{ij} = \kappa_{ij} = \kappa(x_i, x_j)\,.$$

4.4.1 Making Kernels from Kernels

Proposition 4.4.1.1. *Let K_1 and K_2 be kernels over $\mathcal{X} \times \mathcal{X}$, $\mathcal{X} \in \mathbb{R}^n$, then the following functions are kernels:*

1. $\kappa(x, z) = \kappa_1(x, z) + \kappa_2(x, z)$
2. $\kappa(x, z) = a\kappa(x, z)$, $a \geq 0$
3. $\kappa(x, z) = \kappa_1(x, z)\kappa_2(x, z)$
4. $\kappa(x, z) = f(x)f(z)$, $f \colon \mathcal{X} \to \mathbb{R}$
5. $\kappa(x, z) = \kappa_3(\phi(x), \phi(z))$, $\phi \colon \mathcal{X} \to \mathbb{R}^m$ *and* K_3 *- a kernel over* $\mathbb{R}^m \times \mathbb{R}^m$
6. $K(x, z) = x^T Bz$, B *– symmetric positive semidefinite matrix.*

$\rhd$Fix a finite set of points $\{x_1, \ldots, x_l\}$ and let κ_i be korresponding matrices obtained by restricting corresp. kernels to these points. Let α be any vector in $\mathbb{R}^l$. Then

1. $\alpha^T(\kappa_1 + \kappa_2)\alpha = \alpha^T\kappa_1\alpha + \alpha^T\kappa_2\alpha \geq 0$
2. analog. to 1.
3. Based on Shur theorem ([2]), stating that elementwise product of matrices secures positive semidefinitness property.
4. $$\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j\kappa(x_i, x_j) = \sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j f(x_i)f(x_j) = \left(\sum_{i=1}^{l}\alpha_i f(x_i)\right)\left(\sum_{j=1}^{l}\alpha_j f(x_j)\right) = \left(\sum_{i=1}^{l}\alpha_i f(x_i)\right)^2 \geq 0 \quad (4.6)$$
5. Since K_3 is a kernel, the matrix obtained by restricting K_3 to the points $\phi(x_1), \ldots, \phi(x_l)$ is positive semi-definite as required.
6. $\kappa(x, z) = x^T Bz = x^T V^T \Lambda V z = x^T V^T \sqrt{\Lambda}\sqrt{\Lambda} V z = x^T A^T A z = \langle Ax, Az \rangle$ – it is an inner product using a feature mapping A.

$\lhd$

Corollary 4.4.1.1. *Let κ_1 be a kernel over $\mathcal{X} \times \mathcal{X}$ and $p(x)$ - a polynomial with positive coefficients. Then the following funcitons are also kernels:*

- $\kappa(x, z) = p(\kappa_1(x, z))$
- $\kappa(x, z) = \exp(\kappa_1(x, z))$
- $\kappa(x, z) = \exp(-\|x - z\|^2/\sigma^2)$

$\rhd$

- follows from 1-4 of Proposition 4.4.1.1. 4-for a constant of polynomial.
- exp is a limit of a sum of positive polynomials. The set of kernels are closed with respect to pointwise limit.
- $\exp(-\|x - z\|^2/\sigma^2) = \exp(-\|x\|^2/\sigma^2)\exp(-\|z\|^2/\sigma^2)\exp(2\langle x, z\rangle/\sigma^2)$. Hence the proof follows from 4,3,5.

$\lhd$

4.5 Support Vector Machines

Let us consider $K = \{-1, 1\}$, $\mathcal{X} = \mathcal{X}^{-1} \cup \mathcal{X}^{1}$, classifier $q(x) = \operatorname{sgn}(\langle w, x \rangle) = \begin{cases} 1, & x > 1 \\ -1, & x \leq 1 \end{cases}$

Loss-function

$$W(k, d) = \begin{cases} 0, & k = d \\ 1, & k \neq d \end{cases} \tag{4.7}$$

Training set: $\mathcal{L} = ((x_i, k_i), i = 1, m)$.

We should solve

$$\langle w, x \rangle > 0, x \in \mathcal{X}^1 \tag{4.8}$$

$$\langle w, x \rangle < 0, x \in \mathcal{X}^{-1} \tag{4.9}$$

We know that it can be represented as

$$\langle w, x \rangle > 0, x \in \mathcal{X}$$

Hence an empirical risk minimization problem:

$$R_{emp}(q_w) = \min_w \frac{1}{m} \sum_{i=1}^{m} W(k^i, q(x^i)) = \min_w \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}(\left\langle w, x^i \right\rangle \leq 0) \tag{4.10}$$

This is a *maximum feasible linear subsystem problem* – NP-hard.

Regularized version

$$\lambda \|w\| + \sum_{i=1}^{m} \mathbb{I}(\langle w, x \rangle \leq 1)$$

is equivalent to (4.10) for λ small enough - also hopeless problem. We should consider approximations.

Hinge Loss Consider

$$\min_{w,\xi} \lambda \|w\| + \sum_{i=1}^{m} \xi^i \tag{4.11}$$

$$\langle w, x_i \rangle \geq 1 - \xi^i, \ x_i$$

$$\xi^i \geq 0\,, \ i = 1, \ldots, m$$

Proposition 4.5.0.2. *Let (w, ξ) be any feasible point of (4.11). Then $\xi^i \geq W(k^i, q(x^i))$ for W and $q(x)$ defined above.*

$\triangleright$If $\langle w, x_i \rangle > 1$ then $\xi^i = 0 = W(k^i, q(x_i))$
Otherwise $\xi^i = 1 - \langle w, x_i \rangle$. If $\langle w, x_i \rangle > 0$ then $\xi^i > 0 = W(k^i, q(x_i))$.
If $\langle w, x_i \rangle \leq 0$ then $\xi^i \geq 1 = W(k^i, q(x_i))$ $\triangleleft$

4.6 Anderson Problem (Multiclass Discriminative Learning)

From $\left\langle w^{k^i}, x^i \right\rangle - \left\langle w^j, x^i \right\rangle > 0$ follows:

$$\begin{cases} \langle w, \phi(x^i, k^i, k') \rangle > 1 - \xi^i, \ k' \in \mathcal{K} \backslash \{k^i\}, \\ \xi^i, \ i = 1, \dots, m \\ \lambda \|w\| + \sum_{i=1}^m \xi^i \end{cases}$$

Disadvantage: very simple loss function (4.7). Let's consider an arbitrary $W(k, d) \geq 0$ such that $W(k, k) = 0$ $k \in \mathcal{K}$:

$$\begin{cases} \langle w, \phi(x^i, k^i, k') \rangle > W(k^i, k') - \xi^i, \ k' \in \mathcal{K}, \ i = 1, \dots, m \\ \lambda \|w\| + \sum_{i=1}^m \xi^i \end{cases} \tag{4.12}$$

NB! Constraint $\xi^i \geq 0$ is implicitly included (consider $k' = k$)

Proposition 4.6.0.3. *Let w, ξ be a feasible point of (4.12). Then $\xi^i \geq W(k^i, q(x^i))$.*

Proof is absolutely analogous to the proof of Proposition 4.5.0.2.

4.7 Regularizers

4.7.1 ℓ_2-Regularizer

Rewrite the problem (4.12) in a compact way and consider an ℓ_2-regularization.

$$\frac{1}{2} \lambda \|w\|^2 + \sum_{i=1}^m \xi^i \tag{4.13}$$

$$\langle w, x^{ki} \rangle > \Delta^{ki} - \xi^i, \ k \in \mathcal{K}, \ i = 1, \dots, m \tag{4.14}$$

Lagrangian:

$$F(w, \xi, \alpha) = \frac{1}{2} \lambda \|w\|^2 + \sum_{i=1}^m \xi^i + \sum_{k,i} \alpha^{ki} (\Delta^{ki} - \xi^i - \langle w, x^{ki} \rangle), \ \alpha \geq 0 \tag{4.15}$$

$$\frac{\partial F}{\partial w} = \lambda w - \sum_{k,i} \alpha^{ki} x^{ki} = 0 \Rightarrow w = \sum_{k,i} \alpha^{ki} x^{ki} / \lambda \tag{4.16}$$

$$\frac{\partial F}{\partial \xi^i} = 1 - \sum_k \alpha^{ki} = 0 \Rightarrow \sum_k \alpha^{ki} = 1 \text{– simplex constraint } \forall i = 1, \dots, m. \tag{4.17}$$

Plugging in w

$$\frac{1}{2} \lambda \|w\|^2 = \sum_{k,i} \sum_{k',i'} \alpha^{ki} \alpha^{k'i'} \underbrace{x^{ki} x^{k'i'}}_{K(x^{ki}, x^{k'i'})} / (2\lambda) = \frac{1}{2\lambda} \alpha^T K \alpha$$

and changing sign and min to max we receive a dual objective:

$$\min_\alpha \frac{1}{2\lambda} \alpha^T K \alpha - \sum_{k,i} \alpha^{ki} \Delta^{ki} \tag{4.18}$$

$$\sum_k \alpha^{ki} = 1, \ i = 1, \dots, m \tag{4.19}$$

$$\alpha \geq 0\,. \tag{4.20}$$

Both primal and dual are QP. Number of constraints of the primal is equal to the number of vars of the dual - one can switch between them to achieve the best optimization efficiency. The dual is representable in terms of kernel K, hence the kernel trick can be applied as in the case of a perceptron.

4.7.2 ℓ_1-Regularizer

We denote $|\cdot|$ an ℓ_1-norm.

$$\lambda|w| + \sum_{i=1}^{m} \xi^i \tag{4.21}$$

$$\left\langle w, x^{ki} \right\rangle > \Delta^{ki} - \xi^i,\ k \in \mathcal{K},\ i = 1, \dots, m \tag{4.22}$$

Let us show that this is an LP-problem. Let $w = a - b$ and $a > 0$ and $b > 0$:

$$\lambda(a+b) + \sum_{i=1}^{m} \xi^i \tag{4.23}$$

$$\left\langle a - b, x^{ki} \right\rangle > \Delta^{ki} - \xi^i,\ k \in \mathcal{K},\ i = 1, \dots, m \tag{4.24}$$

$$a \geq 0, b \geq 0 \tag{4.25}$$

Similar considerations lead to the dual:

$$\max \sum_{k,i} \alpha^{ki} \Delta^{ki} \tag{4.26}$$

$$\lambda \mathbb{I} - \sum_{k,i} \alpha^{ki} x^{ki} \geq 0 \tag{4.27}$$

$$\lambda \mathbb{I} + \sum_{k,i} \alpha^{ki} x^{ki} \geq 0 \tag{4.28}$$

$$\sum_{k} \alpha^{ik} = 0 \tag{4.29}$$

The dual is not representable through a kernel :(. But the parameters vector w is typically more sparse then in ℓ_2-case.

4.7.3 Forcing Dual Sparsity (ν-SVM)

For computational reasons (computing with kernels) one would like to get a sparse dual solution - only small amount of coordinates of α are non-zero. Recall (4.16): $w = \sum_{k,i} \alpha^{ki} x^{ki} / \lambda$. Plug it in to constraints of (4.13) and select an ℓ_1-regularizer for dual variables:

$$\min \lambda|\alpha| + \sum_{i=1}^{m} \xi^i \tag{4.30}$$

$$\sum_{i',k'} \alpha^{ki} \underbrace{\left\langle x^{k'i'}, x^{ki} \right\rangle}_{\kappa(x^{k'i'}, x^{ki})} \geq \Delta^{ki} - \xi^i,\ k \in \mathcal{K},\ i = 1, \dots, m \tag{4.31}$$

$$\alpha \geq 0 \tag{4.32}$$

- again an LP problem. However, it does not possess a primal variable w sparseness property anymore.

Bibliography

[1] http://www.kernel-machines.org/tutorials

[2] Shur theorem: Horn, Roger A.; Johnson, Charles R. (1985), *Matrix Analysis*, Cambridge University Press, ISBN 978-0-521-38632-6

[3] Nello Cristianini, John Shawe-Taylor *An introduction to support vector machines: and other kernel-based learning methods*

[4] Schölkopf B Person and Smola AJ : *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, 644, MIT Press, Cambridge, MA, USA, (December-2002).

[5] Statistical Pattern Recognition toolbox for Matlab by V.Franc.
http://cmp.felk.cvut.cz/cmp/software/stprtool/index.html

[6] Duality in convex programming: Stephen Boyd and Lieven Vandenberghe *Convex Optimization* - textbook. Available online: `http://www.stanford.edu/~boyd/cvxbook/`

Chapter 5

Hidden Markov Models (Acyclic)

Let $\bar{\mathcal{X}} = \mathcal{X}^1 \times \mathcal{X}^2 \dots \mathcal{X}^n \ni \bar{x}$ be the observation set, $\bar{x}$-observed sequence.
Let $\bar{\mathcal{K}} = \mathcal{K}^1 \times \mathcal{K}^2 \dots \mathcal{K}^n \ni \bar{k}$ the set of objects states (labelings, sequences of latent variables), $\bar{k}$-sequence of hidden (latent) variables or labeling.

$$p(\bar{x}, \bar{k}) = p_0(k_0) \prod_{i=1}^{n} p_i(x_i, k_i | k_{i-1}) \tag{5.1}$$

– joint probability distribution.

Remark 5.0.3.1. If $\bar{\mathcal{X}}$ or infinite, then $p(\bar{x}, \bar{k})$ is a density of a probability distribution. The set $\bar{\mathcal{K}}$ is considered to be finite here.

For the sake of notation (and without loss of generality) we will suppose that $\mathcal{X}^i = \mathcal{X}^j = \mathcal{X}$ and $\mathcal{K}^i = \mathcal{K}^j = \mathcal{K}$.

Example 5.0.3.1. Medical diagnistics: i - time, x_i - results of analysis, k_i - patient state.

Example 5.0.3.2. Recognition of a sequence of character images. $\mathcal{K}$ - set of characters, alphabet, $\mathcal{X}$ - set of charactes templates, $p(x|k)$ - distribution of the image x given character k, i.e. template images plus some kind of noise. $p_i(x_i, k_i|k_i) = p_i(x_i, k_i|k_i) = p(x_i|k_i)p(k_i|k_{i-1})$.

Example 5.0.3.3. Voice recognition: $\mathcal{K}$ - set of phonems, $\mathcal{X}$ - set of corresponding acoustic signals. In this case, however such factorization $p_i(x_i, k_i|k_i) = p(x_i|k_i)p(k_i|k_{i-1})$ is not valid (acoustic signal of a given phoneme depends on the previous phoneme) as well.

Example 5.0.3.4. License plates recognition. $\mathcal{X}$ and $\mathcal{K}$ similar to the one in Example 5.0.3.2, but $p_i(k_i|k_i)$ is dependent on the currect position i.

5.1 Inference For HMM(A)

5.1.1 Maximum A Posteriory Estimation of the Sequence of Hidden States (MAP-Inference)

Let $\mathcal{D} = \bar{\mathcal{K}}$ and the loss function be $W(\bar{k}, \bar{k}^*) = \begin{cases} 0, & \bar{k} = \bar{k}^* \\ 1, & \bar{k} \neq \bar{k}^* \end{cases}$. Then average risk minimization problem is MAP problem, i.e.:

$$\bar{k}^* = \arg\max_{\bar{k}\in\bar{\mathcal{K}}} p(\bar{k}|\bar{x}) = \arg\max_{\bar{k}\in\bar{\mathcal{K}}} \frac{p(\bar{x}, \bar{k})}{p(\bar{x})} = \arg\max_{\bar{k}\in\bar{\mathcal{K}}} p(\bar{x}, \bar{k}) \tag{5.2}$$

$$= \arg\max_{\bar{k}\in\bar{\mathcal{K}}} p_0(k_0) \prod_{i=1}^{n} p_i(x_i, k_i|k_{i-1}) = \arg\max_{\bar{k}\in\bar{\mathcal{K}}} q_0(k_0) + \sum_{i=1}^{n} q_i(k_i, k_{i-1})\,, \tag{5.3}$$

where $q_0(k_0) = \log p_0(k_0)$ and $q_i(k_i, k_{i-1}) = \log p_i(x_i, k_i|k_{i-1})$.

Lets use associative, distributive and commutative properties of operations max and +:

$$\bar{k}^* = \arg\max_{k_n,\dots,k_1} \left(\sum_{i=2}^{n} q_i(k_i, k_{i-1}) + \underbrace{\max_{k_0\in\mathcal{K}}(q_1(k_1, k_0) + q_0(k_0)))}_{Q_1(k_1)} \right) \tag{5.4}$$

$$= \arg\max_{k_n,\dots,k_2} \left(\sum_{i=3}^{n} q_i(k_i, k_{i-1}) + \underbrace{\max_{k_1\in\mathcal{K}}(q_2(k_2, k_1) + Q_1(k_1)))}_{Q_2(k_2)} \right) \tag{5.5}$$

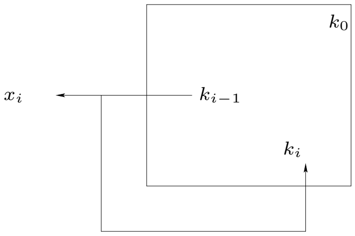
$$\dots \tag{5.6}$$

Summarizing: we've got an iterative algorithm:

1. $Q_0(k_0) = q_0(k_0)$
2. $Q_i(k_i) = \max_{k_{i-1}\in\mathcal{K}} q_i(k_i, k_{i-1}) + Q_{i-1}(k_{i-1})$.
3. $Q = \max_{k_n\in\mathcal{K}} Q_n(k_n)$.

Complexity of the algorithm $O(nK^2)$.

5.1.2 Recognition of Stochastic Finite Autonomous Automaton, $(+, \times)$ algorithm



Problem: Given m automata, i.e. $p^d(\bar{x}, \bar{k})$, $d \in \mathcal{D}\{1, \dots, m\}$ – the set of decisions. Find which of the automata generated an observed sequence $\bar{x}$.

Example 5.1.2.1. Language recognition (speech, text or image).

Reasonable loss function is $W(d,d') = \left\{ \begin{array}{ll} 0, & d = d' \\ 1, & d \neq d' \end{array} \right.$.

A max-probability stategy correspond to such a loss:

$$\bar{k}^* = \arg\max_{d\in\mathcal{D}} p(d|\bar{x}) = \arg\max_{d\in\mathcal{D}} \frac{p(\bar{x},d)}{p(\bar{x})} = \arg\max_{d\in\mathcal{D}} p(\bar{x},d).$$

$$p(\bar{x},d) = \sum_{\bar{k}\in\bar{\mathcal{K}}} p^d(\bar{x},\bar{k})\,.$$

We will omit the superscript d in $p^d(\bar{x},\bar{k})$.

$$= \sum_{\bar{k}\in\bar{\mathcal{K}}} p(\bar{x},\bar{k}) = \sum_{\bar{k}\in\bar{\mathcal{K}}} p_0(k_0) \prod_{i=1}^{n} p_i(x_i,k_i|k_{i-1}) \tag{5.7}$$

denoting $q_0(k_0) = p_0(k_0)$ and $q_i(k_i,k_{i-1}) = p_i(x_i,k_i|k_{i-1})$ receive

$$= \sum_{\bar{k}\in\bar{\mathcal{K}}} q_0(k_0) \cdot \prod_{i=1}^{n} q_i(k_i,k_{i-1}) \tag{5.8}$$

using associative, distributive and commutative properties of operations $\sum$ and ×:

$$= \arg \sum_{k_n,\ldots,k_1} \left(\prod_{i=2}^{n} q_i(k_i,k_{i-1}) \cdot \underbrace{\sum_{k_0\in\mathcal{K}} \left(q_1(k_1,k_0)\cdot q_0(k_0)\right))}_{Q_1(k_1)} \right) \tag{5.9}$$

$$= \arg \sum_{k_n,\ldots,k_2} \left(\prod_{i=3}^{n} q_i(k_i,k_{i-1}) \cdot \underbrace{\sum_{k_1\in\mathcal{K}} \left(q_2(k_2,k_1)\cdot Q_1(k_1)\right))}_{Q_2(k_2)} \right) \tag{5.10}$$

$$\ldots \tag{5.11}$$

Summarizing: we've got an iterative algorithm:

1. $Q_0(k_0) = q_0(k_0)$
2. $Q_i(k_i) = \sum_{k_{i-1}\in\mathcal{K}} q_i(k_i,k_{i-1}) \cdot Q_{i-1}(k_{i-1})$.
3. $Q = \sum_{k_n\in\mathcal{K}} Q_n(k_n)$.

5.1.3 Generalized Computational Scheme, $(\oplus,\otimes)$

The triple $(W,\oplus,\otimes)$ of the set W and two operations $\oplus$ and $\otimes$ is called *a commutative semiring with one* if

1. Operations $\oplus$ and $\otimes$ are associative, distributive and commutative.
2. There exist neutral elements (called *zero* and *one*) for both operations.

General iterative scheme:

1. $Q_0(k_0) = q_0(k_0)$
2. $Q_i(k_i) = \bigoplus_{k_{i-1}\in\mathcal{K}} q_i(k_i, k_{i-1}) \otimes Q_{i-1}(k_{i-1})$.
3. $Q = \bigoplus_{k_n\in\mathcal{K}} Q_n(k_n)$.

Important semirings:

1. $(R, \max, +)$
2. $(R, \min, +)$
3. $([0,1], \max, \times)$
4. $(0, 1, \vee, \wedge)$
5. $(R, \max, \min)$
6. $(R, \min, \max)$

Exercise 5.1.3.1. Write down corresponding algorithmic schemes and find out their meaning. Figure out zero and one elements for each of the semirings.

5.1.4 Locally Additive Penalty, Marginalization Problem

The MAP estimation of the hidden sequence $\bar{k}$ correspond to typically very non-natural loss, which penalizes equally ALL incorrect inference results. Let us consider a wide class of widely usesd loss functions of the form

$$W(\bar{k}, \bar{k}') = \sum_{i=0}^{n} w_i(k_i, k_i') \,. \tag{5.12}$$

The corresponding Bayesian problem reads

$$\begin{aligned} \bar{k}^* &= \arg\min_{\bar{k}'\in\bar{\mathcal{K}}} \sum_{\bar{k}\in\mathcal{K}} p(\bar{k}|\bar{x}) W(\bar{k}, \bar{k}') \qquad (5.13) \\ &= \arg\min_{\bar{k}'\in\bar{\mathcal{K}}} \sum_{\bar{k}\in\mathcal{K}} p(\bar{k}|\bar{x}) \sum_{i=1}^{n} w_i(k_i, k_i') \\ &= \arg\min_{\bar{k}'\in\bar{\mathcal{K}}} \sum_{i=0}^{n} \sum_{k_i\in\mathcal{K}} w_i(k_i, k_i') \underbrace{\sum_{\bar{k}''\in\mathcal{K}_i(k_i)} p(\bar{k}''|\bar{x})}_{p_i(k_i|\bar{x})} \,, \end{aligned}$$

where $\mathcal{K}_i(k_i)$ is the set of hidden sequences containing k_i at the i-th place. Hence the initial problem splits into n *independent small* subproblems given *marginal probalities* $p_i(k_i|\bar{x})$:

$$k_i^* = \arg\min_{k_i' \in \mathcal{K}} \sum_{k_i \in \mathcal{K}} w_i(k_i, k_i') p_i(k_i|\bar{x}) \,. \tag{5.14}$$

The only relatively difficult part is to compute $p_i(k_i|\bar{x})$. This problem is commonly nown as *marginalization problem.* For our model it can be done solved quite efficiently by the *forward-backward* Algorithm 1.

Algorithm 1 Forward-backward $(+, \times)$ algorithm

```
1. Compute $Q_i^F$ using forward variant of the $(+,\times)$ algorithm ($q_0(k_0) = p_0(k_0)$ and $q_i(k_i, k_{i-1}) = p_i(x_i, k_i|k_{i-1})$ ):
    (a) $Q_0^F(k_0) = q_0(k_0)$
    (b) $Q_i^F(k_i) = \sum_{k_{i-1} \in \mathcal{K}} q_i(k_i, k_{i-1}) \cdot Q_{i-1}^F(k_{i-1})$.
2. Compute $Q_i^B$ using backward variant of the $(+,\times)$ algorithm
    (a) $Q_n^B(k) = 1,\ k \in \mathcal{K}$
    (b) $Q_i^B(k_i) = \sum_{k_{i+1} \in \mathcal{K}} q_i(k_{i+1}, k_i) \cdot Q_{i+1}^B(k_{i+1})$.
    (c) $Q_0^B = \sum_{k_1 \in \mathcal{K}} q_0(k_0) Q_1^B(k_1)$.
3. Compute marginals $p_i(k_i|\bar{x}) = Q_i^F(k_i) \cdot Q_i^B(k_i)$.
```

Exercise 5.1.4.1. Consider generalization of this computational scheme for the case when loss depends also on pair of neighboring hidden states, i.e.
$W(\bar{k}, \bar{k}') = \sum_{i=0}^n w_i(k_i, k_i') + \sum_{i=1}^n w_{i-1,i}(k_{i-1}, k_i, k_{i-1}', k_i')$.

Bibliography

We mainly follow the excellent text-book:

[1] Schlesinger M.I., Hlavač V. *Ten Lectures on Statistical and Structural Pattern Recognition.* 2002 (c) Kluwer Academic Publishers.

5.2 Discriminative Learning of HMM. Structural SVM

Since we learned already one of the simplest (but indeed quite powerful) structural model (Hidden Markov Chains) let us consider approaches to learn its parameters.

First we will cooncentrate on a discriminative learning of the MAP classifier. We will denote $\bar{\mathcal{X}}$, $\bar{\mathcal{K}}$ sets of observable and hidden sequences as before. The Markovian probability distribution $p(\bar{x}, \bar{k}; w)$ is supposed to be known up to a parameter vector w. Denoting $q_i(k_i, k_{i-1}, x_i; w) = \log p_i(k_i, x_i|k_{i-1}; w)$ the MAP estimation problem reads

$$\bar{k}^* = \arg\max_{\bar{k} \in \mathcal{K}} p(\bar{k}, \bar{x}) = \arg\max_{\bar{k} \in \mathcal{K}} q_0(k_0; w) + \sum_{i=1}^n q_i(k_i, k_{i-1}, x_i; w)$$

for a given sequence $\bar{x}$. We will assume that $q_i(k_i, k_{i-1}, x_i; w)$ linearly depends on w, thus

$$q_0(k_0; w) + \sum_{i=1}^{n} q_i(k_i, k_{i-1}, x_i; w) = \left\langle w, \phi(\bar{k}, \bar{x}) \right\rangle .$$

Non-regularized discriminative learning problem Given the learning sample $\mathcal{L} = \{(\bar{k}^j, \bar{x}^j),\ j = 1 \ldots, m\}$ find parameter vector w such that

$$\left\langle w, \phi(\bar{k}^j, \bar{x}^j) \right\rangle > \left\langle w, \phi(\bar{k}, \bar{x}^j) \right\rangle \ , \ \bar{k} \in \mathcal{K} \backslash \{\bar{k}^j\},\ j = 1 \ldots, m \, .$$

5.2.1 Structural Perceptron

Our aim is to solve

$$\left\langle w, \phi(\bar{k}^j, \bar{x}^j) \right\rangle - \left\langle w, \phi(\bar{k}, \bar{x}^j) \right\rangle > 0 \, , \ \bar{k} \in \mathcal{K} \backslash \{\bar{k}^j\},\ j = 1 \ldots, m \, .$$

Crusial difficulty: extremely (exponentially) large number of inequalities. We know however, that it is not a problem for the perceptron algorithm as soon as *an oracle able to find a non-satisfied inequality* is available.

1. Select initial $w^0 = 0$.

2. Iterate t:

   (a) Find unsatisfied inequality:

   $$\bar{k}^{*j} = \arg\max_{\bar{k} \in \mathcal{K}} \left\langle w^t, \phi(\bar{k}, \bar{x}^j) \right\rangle ,\ j = 1 \ldots, m. \tag{5.15}$$

   (b) if $\exists l \in \{1 \ldots, m\} \colon \bar{k}^{*l} \neq \bar{k}^l$
   $w^{t+1} := w^t + \phi(\bar{k}^l, \bar{x}^l) - \phi(\bar{k}, \bar{x}^l)$
   else **Exit**.

Remark 5.2.1.1. A convex hull of the set of vector $\psi(\bar{k}^l, \bar{k}, \bar{x}^l) := \phi(\bar{k}^j, \bar{x}^l) - \phi(\bar{k}, \bar{x}^l),\ \bar{k} \in \mathcal{K} \backslash \{\bar{k}^j\},\ l = 1 \ldots, m$ should not contain an origin (it is called *a separable case* in literature).

Remark 5.2.1.2. It is not necessary to compute (5.15) for each $j = 1 \ldots, m$ (can be quite expensive for a large learning sample), it is enough to find a least one l for which $\bar{k}^{*l} \neq \bar{k}^l$ holds. Number of iterations however can depend on the choice of l a lot.

5.2.2 Structural SVM

The same reasoning as for the non-structural multi-class SVM (Fisher classifier) leads to the following formulation (ℓ_2-regularization):

$$\min_{w, \xi} \frac{1}{2} \|w\|^2 + \frac{C}{m} \sum_{j=1}^{m} \xi^j \tag{5.16}$$

$$\text{s.t. } \left\langle w, \psi(\bar{k}^j, \bar{k}, \bar{x}^j) \right\rangle \geq \Delta^{\bar{k}, j} - \xi^j,\ \bar{k} \in \mathcal{K},\ j = 1 \ldots, m \, , \tag{5.17}$$

where $\psi(\bar{k}^j, \bar{k}, \bar{x}^j) = \phi(\bar{k}^j, \bar{x}^j) - \phi(\bar{k}, \bar{x}^j)$ and $\Delta^{\bar{k}, j} = W(\bar{k}, \bar{k}^j)$ -loss such that $W(\bar{k}, \bar{k}') = 0$ if $\bar{k} = \bar{k}'$ and $W(\bar{k}, \bar{k}') > 0$ otherwise.

Again the same **difficulty** as for the structural perceptron: exponentially large number of inequalities. We will see however, that the overall problem is solvable as soon as *an oracle able to find a non-satisfied inequality* is available.

Let us consider the problem which should be solved by such an oracle:

$$\begin{aligned}\xi^{*j} &= \max_{\bar k \in \mathcal{K}} (\Delta^{\bar k, j} - \left\langle w, \psi(\bar k^j, \bar k, \bar x^j) \right\rangle) \\ &= -\left\langle w, \phi(\bar k^j, \bar x^j) \right\rangle + \max_{\bar k \in \mathcal{K}} (\left\langle w, \phi(\bar k, \bar x^j) \right\rangle + \Delta^{\bar k, j}) \\ &= -\left\langle w, \phi(\bar k^j, \bar x^j) \right\rangle + \max_{\bar k \in \mathcal{K}} (q(\bar k, \bar x^j; w) + \Delta^{\bar k, j}) \qquad (5.18)\end{aligned}$$

If $\Delta^{\bar k,j} = W(\bar k, \bar k^j) = \sum_{i=0}^n w_i(k_i, k_i') + \sum_{i=1}^n w_{i-1,i}(k_{i-1}, k_i, k'_{i-1}, k'_i))$ (locally additive loss) we can assign $q_i^j(k_i, k_{i-1}, \bar x^j; w) = q_i(k_i, k_{i-1}, \bar x^j; w) - w_i(k_i, k_i^j) - w_{i-1,i}(k_{i-1}, k_i, k_{i-1}^j, k_i^j)$ and solve a usual MAP estimation problem

$$\max_{\bar k \in \mathcal{K}} (q_0^j(k_0; w) + \sum_{i=1}^n q_i^j(k_i, k_{i-1}, x_i^j; w)) \,.$$

In this notation

$$\xi^{*j} = -\left\langle w, \phi(\bar k^j, \bar x^j) \right\rangle + \max_{\bar k \in \mathcal{K}} (q_0^j(k_0; w) + \sum_{i=1}^n q_i^j(k_i, k_{i-1}, x_i^j; w)) \,. \qquad (5.19)$$

Hence, the oracle is solvable at least for a locally additive loss.

5.2.3 Cutting Plane Algorithm

1. Select the initial $\tilde w$ and the initial constraint sets $\tilde I_j = \{(\bar k^j, j)\}$, $\tilde I = \cup_{j=1}^m \tilde I_j$.

2. Optimize (5.16) for a fixed $\tilde w$ with respect to ξ, i.e. compute $\tilde \xi = (\tilde \xi^j)$ according to (5.19), i.e.

$$\tilde\xi^j = -\left\langle \tilde w, \phi(\bar k^j, \bar x^j) \right\rangle + \max_{\bar k \in \mathcal{K}} (q_0^j(k_0; \tilde w) + \sum_{i=1}^n q_i^j(k_i, k_{i-1}, x_i^j; \tilde w))$$

   Let also

$$\tilde{\bar k}^j = \arg\max_{\bar k \in \mathcal{K}} (q_0^j(k_0; \tilde w) + \sum_{i=1}^n q_i^j(k_i, k_{i-1}, x_i^j; \tilde w))$$

   be the optimal (MAP) sequences corresponding to $\tilde \xi^j$.

3. Increase a constraint set $\tilde I_j := \tilde I_j \cup \{(\tilde{\bar k}^j, j)\}$.

4. Compute a dual to the restricted primal

$$\min_{w,\xi} Q_P(\tilde w, \tilde \xi) = \min_{w,\xi} \frac{1}{2}\|w\|^2 + \frac{C}{m}\sum_{j=1}^m \xi^j \qquad (5.20)$$

$$\text{s.t.} \ \left\langle w, \psi(\bar k^j, \bar k, \bar x^j) \right\rangle \geq \Delta^{\bar k, j} - \xi^j, \ \bar k \in \tilde I_j, \ j = 1 \ldots, m \,, \qquad (5.21)$$

Abusing the notation the dual reads:

$$\max_\alpha Q_D(\alpha) = \max_\alpha \sum_{l\in\tilde I} \alpha_l \Delta^l - \frac{1}{2}\sum_{l\in\tilde I}\sum_{s\in\tilde I} \alpha_l\alpha_s\kappa_{ls} \tag{5.22}$$

$$\sum_{l\in\tilde I_j} \alpha_l = \frac{C}{m}, j = 1\dots m \tag{5.23}$$

$$\alpha_l \geq 0,\ l \in \tilde I \quad \text{(selected constraints)} \tag{5.24}$$

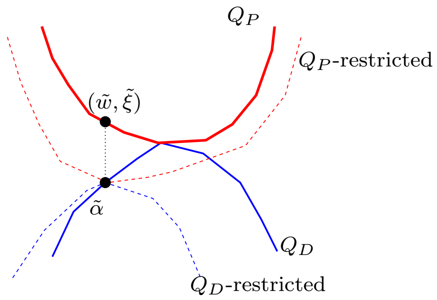
Here $\kappa_{ls} = \left\langle \psi(\bar k^j, \bar k, \bar x^j), \psi(\bar k^{j'}, \bar k', \bar x^{j'}) \right\rangle$ for $l = (\bar k, j)$ and $s = (\bar k', j')$. Let $\tilde\alpha$ be the solution of (5.22).

5. If

$$Q_P(\tilde w, \tilde\xi) - Q_D(\tilde\alpha) \leq \varepsilon \tag{5.25}$$

Exit
else
$\tilde w := \sum_{l\in\tilde I} \alpha_l \psi(\bar k^j, \bar k, j)$, where $l = (\bar k, j)$;
goto step 2.

Remark 5.2.3.1. Condition (5.25) is sufficient to get a precision ε with respect to the primal objective (5.16) value. This is due to the fact that $(\tilde w, \tilde\xi)$ and $\tilde\alpha$ are **feasible** points for the initial **non-restricted** primal (5.16) and its dual ((5.22) for $\tilde I = I$). Let Q^* be an optimal objective value of the initial non-restricted problem (5.16). Then $Q_P(\tilde w, \tilde\xi) \geq Q^*$, $Q_D(\tilde w, \tilde\xi) \leq Q^*$ and $Q_P(\tilde w, \tilde\xi) \geq Q_D(\tilde\alpha)$. Hence the condition (5.25) means $\varepsilon \geq Q_P(\tilde w, \tilde\xi) - Q_D(\tilde\alpha) \geq Q_P(\tilde w, \tilde\xi) - Q^*$.



Remark 5.2.3.2. With ℓ_2-regularizer as in (5.16) one can use a kernel trick, as it is clear from (5.22).

Remark 5.2.3.3. It is not obligatory to switch to the dual at the step 4 of the algorithm, since we need only optimal value $Q_D(\tilde\alpha)$ of the dual restricted problem (5.22) and corresponding optimal parameter vector $\tilde w$. Due to the strong duality both can be achieved by minimizing the primal restricted problem (5.20).

This remark is especially important for ℓ_1-regularization, since in that case it is a nontrivial procedure of getting $\tilde w$ from the dual solution (see lecture about SVM's and ℓ_1 regularization therein).

Bibliography

[1] Tutorial: Sebastian Nowozin and Christoph H. Lampert, *Structured Learning and Prediction in Computer Vision* http://www.nowozin.net/sebastian/papers/nowozin2011structured-tutorial.pdf

[2] Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann and Yasemin Altun (2005), Large Margin Methods for Structured and Interdependent Output Variables, *JMLR, Vol. 6, pages 1453-1484.*
[3] Vojtěch Franc, Bogdan Savchynskyy, Discriminative Learning of Max-Sum Classifiers *Journal of Machine Learning Research, 9(Jan):67–104, 2008, Microtome Publishing.*
http://www.jmlr.org/papers/volume9/franc08a/franc08a.pdf

5.3 Generative Learning of Hidden Markov Chains

We will denote $\bar{\mathcal{X}}$, $\bar{\mathcal{K}}$ sets of observable and hidden sequences as before. The Markovian probability distribution $p(\bar{x}, \bar{k}; w)$ is supposed to be known up to a parameter vector w. We additionally suppose that

$$p(\bar{x}, \bar{k}; w) = p_0(k_0; w) \prod_{i=1}^{m} p_i(x_i, k_i | k_{i-1}; w) = p_0(k_0; w) \prod_{i=1}^{m} \frac{p_i(x_i, k_i, k_{i-1}; w)}{\sum_{x \in \mathcal{X}, k \in \mathcal{K}} p_i(x, k, k_{i-1}; w)} ,$$

where we defined conditional probabilities $p_i(x_i, k_i | k_{i-1}; w)$ via joint probabilities $p_i(x, k, k_{i-1}; w)$.

Given the learning sample $\mathcal{L} = \{(\bar{k}^j, \bar{x}^j),\ j = 1 \ldots, m\}$ we would like find parameter vector w maximizing the regularized likelihood of the sample, i.e.:

$$w^* = \arg\max_w R(w) \cdot \prod_{j=1}^{m} p(\bar{k}^j, \bar{x}^j; w) = \arg\max_w R(w) \cdot \prod_{j=1}^{m} p_0(k_0^j; w) \prod_{i=1}^{n} \frac{p_i(x_i^j, k_i^j, k_{i-1}^j; w)}{\sum_{x \in \mathcal{X}, k \in \mathcal{K}} p_i(x, k, k_{i-1}^j; w)}$$

$$= \arg\max_w \lambda |w| + \sum_{j=1}^{m} \log p_0(k_0^j; w) + \sum_{i=1}^{n} \log \frac{p_i(x_i^j, k_i^j, k_{i-1}^j; w)}{\sum_{x \in \mathcal{X}, k \in \mathcal{K}} p_i(x, k, k_{i-1}^j; w)} \qquad (5.26)$$

Function (5.26) is convex for many important distributions $p_i(x_i^j, k_i^j, k_{i-1}^j; w)$ and thus can be optimized with convex optimization techniques. We consider several important special cases when $\lambda = 0$. (Case of $\lambda \neq 0$ will be considered later on.) Moreover, to simplify formulas we will always consider w to be consistent of two parts, i.e. $w = (w_0, w')$, such that $p_0(k_0^j; w) = p_0(k_0^j; w_0)$ and $p_i(x_i^j, k_i^j, k_{i-1}^j; w) = p_i(x_i^j, k_i^j, k_{i-1}^j; w')$. Thus the problem (5.26) splits into two independent subproblems

$$w_0^* = \arg\max_{w_0} \sum_{j=1}^{m} \log p_0(k_0^j; w) \qquad (5.27)$$

and

$$w'^* = \arg\max_{w'} \sum_{j=1}^{m} \sum_{i=1}^{n} \log \frac{p_i(x_i^j, k_i^j, k_{i-1}^j; w')}{\sum_{x \in \mathcal{X}, k \in \mathcal{K}} p_i(x, k, k_{i-1}^j; w')} . \qquad (5.28)$$

In what follows we will consider only the second part as typically more difficult to learn.

5.3.1 Time-dependent parameters

Let $w = (w_0, w_1, \ldots, w_n)$ and $p_i(x_i, k_i, k_{i-1}; w) = p_i(x_i, k_i, k_{i-1}; w_i)$. In this case (5.28) splits into independent subproblems for each $i = 0, \ldots, n$:

$$w_i^* = \arg\max_{w_i} \sum_{j=1}^{m} \log \frac{p_i(x_i^j, k_i^j, k_{i-1}^j; w)}{\sum_{x \in \mathcal{X}, k \in \mathcal{K}} p_i(x, k, k_{i-1}^j; w)}$$

$$= \arg\max_{w_i} \sum_{x \in \mathcal{X}, k \in \mathcal{K}, k' \in \mathcal{K}} \alpha_i(x, k, k') \log \frac{p_i(x, k, k'; w_i)}{\sum_{x'' \in \mathcal{X}, k'' \in \mathcal{K}} p_i(x'', k'', k'; w_i)} , \qquad (5.29)$$

where $\alpha_i(x, k, k')$ determines how many times the triple (x, k, k') happen in the training sample $\mathcal{L}$ in the i-th time step.

5.3.2 Non-parametric estimation

Let $w_i = p_i$, i.e. one wants to estimate numbers $p_i(x, k, k')$ for all $x \in \mathcal{X},\ k \in \mathcal{K},\ k' \in \mathcal{K}$. To compute (5.29) in this case we need the following famous lemma

Lemma 5.3.2.1 (Shannon)**.** *For all $\beta_i \geq 0$, $i = 1, \ldots, l$ such that*

$$\sum_{i=1}^{l} \beta_i = 1 \tag{5.30}$$

and all $\alpha_i \geq 0$, $i = 1, \ldots, l$ holds

$$\sum_{i=1}^{l} \alpha_i \log \beta_i \leq \sum_{i=1}^{l} \alpha_i \log \frac{\alpha_i}{\sum_{i=1}^{l} \alpha_i}\,.$$

$\triangleright$Function $f(\beta) = \sum_{i=1}^{l} \alpha_i \log \beta_i$ is concave (since log is concave and $\alpha_i \geq 0$), thus it achieves its global optimum over the convex set defined by constraint (5.30) and $\beta_i \geq 0$. Taking into account condition (5.30) the (partial - without taking into account positivity constraints $\beta_i \geq 0$) Lagrangian reads

$$F(\beta, \gamma) = \sum_{i=1}^{l} \alpha_i \log \beta_i + \gamma(1 - \sum_{i=1}^{l} \beta_i)$$

Its partial derivative reads

$$\frac{\partial F}{\partial \beta_i} = \frac{\alpha_i}{\beta_i} - \gamma\,.$$

Assigning it to zero leads to $\beta_i \propto \alpha_i$, and since $\alpha_i \geq 0$ positivity constraint is satisfied, which means this local minimum correspond to the constrained global one. Applying constraint (5.30) results in

$$\beta_i = \frac{\alpha_i}{\sum_{i=1}^{l} \alpha_i}\,.$$

$\triangleleft$

Taking into account that

$$\sum_{x \in \mathcal{X}, k \in \mathcal{K}} \frac{p_i(x, k, k')}{\sum_{x'' \in \mathcal{X}, k'' \in \mathcal{K}} p_i(x'', k'', k')} = 1,\ k' \in \mathcal{K} \quad \text{(these are β's from Lemma 5.3.2.1)}$$

and applying Lemma 5.3.2.1 to (5.29) we conclude that

$$p_i(x, k, k') \propto \alpha_i(x, k, k')$$

maximizes (5.29). It means, $p_i(x, k, k')$ are equal to frequencies of (x, k, k') in the i-th time step in the learning sample $\mathcal{L}$.

5.3.3 Conditionally independent state and observation

Let now $p_i(x,k|k') = p_i(k|k')p_i(x|k';w_i)$, i.e. k and x are conditionally independent given k' (think about sequence of observable character images corresponding to English words). In what follows we omit index i and denote $p_i(k|k')$ as $p_K(k|k')$ to distinguish between two probability distributions $p_i(k|k')$ and $p_i(x|k';w_i)$. The latter distribution will be used without any additional index.

Hence

$$p(x,k|k') = p_K(k|k')p(x|k';w) = \frac{p_K(k,k')}{\sum_{k''\in\mathcal{K}} p_K(k'',k')} \cdot p(x|k';w)\,.$$

After plugging this in into (5.29) it reads

$$\begin{aligned}(w^*,p_K^*) &= \arg\max_{w,p_K} \sum_{x\in\mathcal{X},k\in\mathcal{K},k'\in\mathcal{K}} \alpha_i(x,k,k') \log\left(\frac{p_K(k,k')}{\sum_{k''\in\mathcal{K}} p_K(k'',k')} \cdot p(x|k';w)\right) \\ &= \arg\max_{w,p_K} \sum_{x\in\mathcal{X},k\in\mathcal{K},k'\in\mathcal{K}} \alpha_i(x,k,k') \left(\log\frac{p_K(k,k')}{\sum_{k''\in\mathcal{K}} p_K(k'',k')} + \log p(x|k';w)\right)\,. \qquad (5.31)\end{aligned}$$

The problem (5.31) splits into two independent subproblems for parameters w^* and p_K^* respectively

$$w^* = \arg\max_w \sum_{x\in\mathcal{X},k\in\mathcal{K},k'\in\mathcal{K}} \alpha_i(x,k,k')\log p(x|k';w) = \arg\max_w \sum_{x\in\mathcal{X},k'\in\mathcal{K}} \alpha_i(x,k')\log p(x|k';w)\,, \qquad (5.32)$$

(where $\alpha_i(x,k') = \sum_{k\in\mathcal{K}} \alpha_i(x,k,k')$) and

$$\begin{aligned}p_K^* &= \arg\max_{p_K} \sum_{x\in\mathcal{X},k\in\mathcal{K},k'\in\mathcal{K}} \alpha_i(x,k,k') \log\frac{p_K(k,k')}{\sum_{k''\in\mathcal{K}} p_K(k'',k')} \\ &= \arg\max_{p_K} \sum_{k\in\mathcal{K},k'\in\mathcal{K}} \alpha_i(k,k') \log\frac{p_K(k,k')}{\sum_{k''\in\mathcal{K}} p_K(k'',k')}\,. \qquad (5.33)\end{aligned}$$

where $\alpha_i(k,k') = \sum_{x\in\mathcal{X}} \alpha_i(x,k,k')$. The first equation states a typical maximal likelihood estimation problem (compare to (3.2)), the second equation has the same form as (5.29) and thus the approach of the paragraph **Non-parametric estimation** can be applied here as well. As a result we receive that the estimate for $p_K(k,k')$ are frequencies of (k,k') corresponding to the given time step i.

5.3.4 Time homogeneous case

Let us consider another typical situation when probability distribution $p_i(x_i,k_i,k_{i-1};w)$ is the same for all time steps i, i.e.$p_i(x_i,k_i,k_{i-1};w) = p(x_i,k_i,k_{i-1};w)$. In this case

$$\begin{aligned}w^* &= \arg\max_w \sum_{j=1}^m \sum_{i=1}^n \log\frac{p(x_i^j,k_i^j,k_{i-1}^j;w)}{\sum_{x\in\mathcal{X},k\in\mathcal{K}} p(x,k,k_{i-1}^j;w)} \\ &= \arg\max_w \sum_{x\in\mathcal{X},k\in\mathcal{K},k'\in\mathcal{K}} \alpha(x,k,k')\log\frac{p(x,k,k';w)}{\sum_{x''\in\mathcal{X},k''\in\mathcal{K}} p(x'',k'',k';w)}\,, \qquad (5.34)\end{aligned}$$

where $\alpha(x,k,k')$ determines how many times the triple (x,k,k') appeared in the learning sample $\mathcal{L}$ WITHOUT taking into account the time step index i.

Comparing (5.34) to (5.28) one see that both special cases **Non-parametric estimation** and **Conditionally independent state and observation** can be treated exactly in the same way as before (for **Time-dependent parameters** case) with the only difference in values $\alpha(x, k, k')$.

5.3.5 Example: Markovian Sequence of Images of Characters. Time homogeneous case with conditionally independent state and observation

Let $\bar{\mathcal{K}}$ denotes the set of sequences of English characters corresponding to a natural language. Let $x \in \mathcal{X}$ be an image of some character k. The distribution $p_i(k, k, x; w) = p(k, k', x; w)$ does not depend on i and $p(x, k|k') = p(k|k')p(x|k'; w)$. The conditional probability of the picture x given the corresponding character k depends on the template image w_k. The task is to estimate all such template images $w^* = (w_k^*,\ k \in \mathcal{K})$.

Generative Learning

In this case (5.32) splits to $|\mathcal{K}|$ independent subproblems

$$w_k^* = \arg\max_{w_k} \sum_{x\in\mathcal{X},k\in\mathcal{K}} \alpha(x, k) \log p(x|k; w_k)\,, \tag{5.35}$$

Denoting as $x(l)$ the l-th pixel in the image x and under Gaussian noise assumption holds

$$p(x|k; w_k) = C \cdot \exp\left(-\sigma \sum_l (x(l) - w_k(l))^2\right)\,. \tag{5.36}$$

Let us plug it into (5.35) and obtain

$$w_k(l) = \sum_{x\in\mathcal{X},k\in\mathcal{K}} \frac{\alpha(x, k)x(l)}{\sum_{x'\in\mathcal{X},k'\in\mathcal{K}} \alpha(x', k')}\,,$$

which basically means that we have to average all images from the learning sample $\mathcal{L}$, which correspond to the character k.

Probabilities $p_K(k, k')$ of neighboring pairs of characters (k, k') should be taken equal to the frequencies of corresponding character pairs $\frac{\alpha(k,k')}{\sum_{k\in\mathcal{K},k'\in\mathcal{K}} \alpha(k,k')}$ summed up for all time steps i.

Discriminative Learning

Given the training set $\mathcal{L} = \{(\bar{x}^j, \bar{k}^j),\ j = 1\ldots, m\}$ find w to fulfill

$$\begin{aligned}
\bar{k}^j &= \mathrm{argmax}_{\bar{k}} p(\bar{k}|\bar{x}^j; w) = \mathrm{argmax}_{\bar{k}} \frac{p(\bar{x}^j, \bar{k}; w)}{p(\bar{x}^j)} \\
&= \mathrm{argmax}_{\bar{k}} p(\bar{x}^j, \bar{k}; w) = \mathrm{argmax}_{\bar{k}} \log p(\bar{x}^j, \bar{k}; w) \\
&= \mathrm{argmax}_{\bar{k}} \log p_0(k_0) + \sum_{i=1}^{n} \log p(k_i|k_{i-1}) + \log p(x_i|k_{i-1}; w)\,.
\end{aligned} \tag{5.37}$$

Here $\log p_0(k_0) = \alpha(k_0)$ and $\log p(k_i|k_{i-1}) = \beta(k_i, k_{i-1})$ are just numbers which have to be estimated and assuming (5.36)

$$\log p(x|k; w) = -\sigma \sum_l (x(l) - w_k(l))^2 = -\sigma \sum_l x^2(l) + w_k^2(l) - 2x(l)w_k(l)$$

Term $\sum_l x^2(l)$ does not depends on $\bar{k}$, hence it does not influence (5.37) and can be omitted.

Considering $-\sigma \sum_l w_k^2(l) = \gamma(k)$ as a separate variable, the objective (5.37) becomes linear with respect to parameters $w = (\alpha(k), \beta(k, k'), \gamma(k), w_k(l) \mid k', k \in \mathcal{K},\ l \in I)$ (I denotes a set of all pixes in a single image), thus the linear discriminative learning machinery can be applied.

The problem (5.37) in the new notation reads

$$\bar{k}^j = \text{argmax}_{\bar{k}} \left\langle w, \phi(\bar{k}) \right\rangle$$

for a suitably selected vector $\phi(\bar{k})$.

According to discriminative learning paradigm we have to solve the system of linear inequalities

$$\left\langle w, \phi(\bar{k}^j, \bar{x}^j) \right\rangle > \left\langle w, \phi(\bar{k}, \bar{x}^j) \right\rangle, \ \bar{k} \in \bar{\mathcal{K}} \backslash \bar{k}^j, \ j = 1 \dots m \tag{5.38}$$

or a corresponding SVM problem with respect to parameter vector w.

We will consider a perceptron algorithm as applied to (5.38). Let $\bar{k}' = \text{argmax}_{\bar{k}} \left\langle w, \phi(\bar{k}) \right\rangle$ and $\bar{k}' \neq \bar{k}^j$. Then the following steps have to be performed according to the perceptron algorithm:

$$\begin{array}{rclrcll}
\alpha(k_0^j) & += & 1 & \alpha(k_0') & -= & 1 & \\
\beta(k_i^j, k_{i-1}^j) & += & 1 & \beta(k_i', k_{i-1}') & -= & 1 & i = 1 \dots n \\
\gamma(k_i^j) & += & 1 & \gamma(k_i') & -= & 1 & i = 1 \dots n \\
w_{k_i^j}(l) & += & 2\sigma x(l) & w_{k_i'}(l) & += & 2\sigma x(l) & i = 1 \dots n,\ l \in I
\end{array} \tag{5.39}$$

Discriminative Learning with Gaussian RBF Kernels

It often happends that approximation (5.36) for distribution $p(x_i|k_{i-1}; w)$ does not work (works very bad) and true distribution is unknown. In this case the Radial Basis Functions are used as kernels for the kernel-SVM (kernel-perceptron). We already saw that using such kernels one can approximate very complex surfaces. In this case it is considered that

$$\log p(x|k'; w) = \sum_{j=1}^{m} \sum_{\substack{i:\ k_i^j = k' \\ 0 \le i \le n-1}} \gamma(j, i) \exp \sum_l -\sigma(x(l) - x_i^j(l))^2 = \sum_{j=1}^{m} \sum_{\substack{i:\ k_i^j = k' \\ 0 \le i \le n-1}} \gamma(j, i) \xi(x_i^j, x) \,.$$

Let us use notation $T(k') = \{(j, i) \colon j = 1, \dots, m,\ 0 \le i \le n-1,\ k_i^j = k'\}$. Then we would like to find out such parameters, that

$$\begin{aligned}
\bar{k}^j &= \text{argmax}_{\bar{k}} \log p_0(k_0) + \sum_{i=1}^{n} \log p(k_i|k_{i-1}) + \log p(x_i|k_{i-1}; w) \\
&= \text{argmax}_{\bar{k}} \alpha(k_0) + \beta(k_i, k_{i-1}) + \sum_{(j', i) \in T(k_{i-1})} \gamma(j', i) \xi(x_i^{j'}, x^j) \,,
\end{aligned} \tag{5.40}$$

where $\log p_0(k_0) = \alpha(k_0)$ and $\log p(k_i|k_{i-1}) = \beta(k_i, k_{i-1})$ as in the previous case.

In this case parameter vector has the form $w = (\alpha(k), \beta(k, k'), \gamma(j, i) \mid (j, i) \in T(k'),\ k', k \in \mathcal{K}\}$ and perceptron algprithm has a similar form as in (5.39).

Hybrid approach.

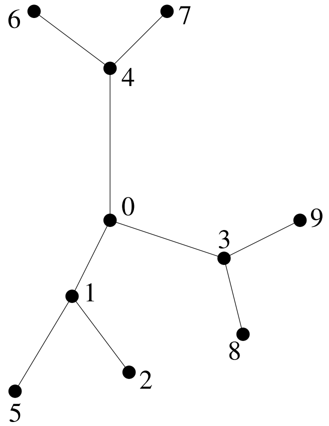
Quite popular (however typically delivering worse results than the previous method) is hybrid learning: parameters $\gamma(j,i)$ are learned independently via non-structural kernel SVM and probabilities $p_0(k_0)$ and $p(k_i|k_{i-1})$ are estimated from corresponding frequencies, as in **Generative Learning** case.

Bibliography

[1] Schlesinger M.I., Hlavač V. *Ten Lectures on Statistical and Structural Pattern Recognition.* 2002 (c) Kluwer Academic Publishers.
[2] Vojtěch Franc, Bogdan Savchynskyy, Discriminative Learning of Max-Sum Classifiers *Journal of Machine Learning Research, 9(Jan):67–104, 2008, Microtome Publishing.* http://www.jmlr.org/papers/volume9/franc08a/franc08a.pdf
[3] Tutorial: Sebastian Nowozin and Christoph H. Lampert, *Structured Learning and Prediction in Computer Vision* http://www.nowozin.net/sebastian/papers/nowozin2011structured-tutorial.pdf

5.4 Tree-structured HMM

Let $G = (\mathcal{V}, \mathcal{E})$ be a tree. Let $\bar{\mathcal{X}} = \mathcal{X}^1 \times \mathcal{X}^2 \ldots \mathcal{X}^n \ni \bar{x}$ be the observation set, $\bar{x}$-observed collection.
Let $\bar{\mathcal{K}} = \mathcal{K}^1 \times \mathcal{K}^2 \ldots \mathcal{K}^n \ni \bar{k}$ the set of objects states (labelings, collections of latent variables), $\bar{k}$- a collection of hidden (latent) variables or labeling.



Let the joint probability of observations $\bar{x}$ and hidden $\bar{k}$ states be equal to

$$p_0(\bar{x}, \bar{k}) = p(k_0) \prod_{i \in I \setminus \{0\}} p_i(x_{i,g(i)}, k_i | k_{g(i)}) \,,$$

where 0 is a tree-root, $i, g(i) \in I$, $I \sim \mathcal{V}$, $n : I \to \mathcal{V}$ - bijective mapping, $(n(i), n(g(i))) \in \mathcal{E}$ and a path from $n^{-1}(0)$ to $n^{-1}(i)$ contains $n^{-1}(g(i))$. The following algorithm enumerates tree vertices (constructs $n(i)$), such that $g(i) < i$.

An important property of the tree is that there is always at least one leave node ;))), i.e. node $n^{-1}(i^*) \in \mathcal{V}$ such, that $i^* \neq g(i)$, $\forall i \in I$.

Let us consider MAP problem and show how it can be solved by (basically) the same

Algorithm 2 Algorithm of tree vertices enumeration

```
• Initialize: select tree root $a_0 \in \mathcal{V}$, set $N = \{a_0\}$, $\bar{N} = \emptyset$, $i := 0$.
• Iterate (for $i \geq 0$)
    1. Select $a \in N \backslash \bar{N}$;
    2. $n(a) := i$
    3. $N := N \cup \{\text{neighbors}(a)\}$
    4. $\bar{N} = \bar{N} \cup \{a\}$
    5. i:=i+1, goto 1
```

trick as with sequences.

$$\begin{aligned}\bar{k}^* = \arg\max_{\bar{k}\in\bar{\mathcal{K}}} p(k_0) \prod_{i\in I\backslash\{0\}} p_i(x_{i,g(i)}, k_i|k_{g(i)}) \\ = \arg\max_{k_i,\ i\in I} \sum_{i\in I} \phi_i(k_i) + \sum_{i\in I\backslash\{0\}} q_i(k_i, k_{g(i)}) \qquad (5.41)\end{aligned}$$

Here we introduced notation

$$\phi_i(k_i) = \begin{cases} \log p_0(k_0), & i = 0 \\ 0, & \text{otherwise} \end{cases}$$

and $q_i(k_i, k_{g(i)}) = \log p_i(x_{i,g(i)}, k_i|k_{g(i)})$.

Let us now select $i^* \ :\ i^* \neq g(i),\ \forall i \in I$. We can rewrite (5.41) as

$$\begin{aligned}\bar{k}^* = \arg\max_{k_i,\ i\in I\backslash\{i^*\}} \sum_{i\in I\backslash\{i^*\}} \phi_i(k_i) + \sum_{i\in I\backslash\{0,i^*\}} q_i(k_i, k_{g(i)}) + \underbrace{\max_{k_{i^*}} \phi_{i^*}(k_{i^*}) + q_{i^*}(k_{i^*}, k_{g(i^*)})}_{\phi_{i'}(k_{i'}),\ i'=g(i^*)} \\ = \arg\max_{k_i,\ i\in I'} \sum_{i\in I'} \phi_i(k_i) + \sum_{i\in I'\backslash\{0\}} q_i(k_i, k_{g(i)})\,, \qquad (5.42)\end{aligned}$$

where $I' = I\backslash\{i^*\}$. Comparing the last equation to (5.41) we see that we've got a recursive rule.

Let us formulate this algorithm in the general $(\oplus, \otimes)$ semiring. We have to compute

$$\bigoplus_{(k_i, i\in I)} \bigotimes_{i\in I} \phi_i(k_i) \otimes \bigotimes_{i\in I\backslash\{0\}} q_i(k_i, k_{g(i)})$$

Algorithm 3 General $(\oplus, \otimes)$ inference algorithm on tree

```
repeat
    Find $i^* \ :\ i^* \neq g(i),\ \forall i \in I$
    $i' := g(i^*)$
    $\phi_{i'}(k_{i'}) := \bigoplus_{k_{i^*}} \phi_{i^*}(k_{i^*}) \otimes q_{i^*}(k_{i^*}, k_{i'})$
    $I := I \backslash \{i^*\}$
until remains to compute $\bigoplus_{k_0} \phi_0(k_0)$
```

Chapter 6

Markov Random Fields and Cyclic Hidden Markov Models

6.1 General Definitions and Properties

In the previous section we considered acyclic models - the underlying neighborhood structure was a sequence or a tree. From now on we will concentrate on more general case - without this restriction.

Definition 6.1.0.1. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph consisting of the finite set of *nodes* $\mathcal{V}$ and the set of node pairs $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. The set $\mathcal{E}$ will be also called *a neighborhood structure* of the set $\mathcal{V}$.

Definition 6.1.0.2. The function of the form $\bar{k} \colon \mathcal{V} \to \mathcal{K}$, where $\mathcal{K}$ is a finite set will be called *labeling* or (sometimes in connection with the word "Markov") *field.* We will denote as $k_\Omega := \bar{k}\big|_\Omega$ the restriction of $\bar{k}$ to the set Ω. In particular we will often use notations $k_v,\ v \in \mathcal{V}$ and $k_e = (k_u, k_v),\ (u,v) = e \in \mathcal{E}$.

Definition 6.1.0.3. If a probability distribution p of the form $p \colon \mathcal{K}^{\mathcal{V}} \to \mathbb{R}$ is given we will consider *a random labeling* to be specified. The probability of the labeling $\bar{k}$ is equal to $p(\bar{k})$.

Definition 6.1.0.4. The probability distribution

$$p_\Omega(k_\Omega) = \sum_{\bar{k}' \in \mathcal{K}_\Omega(k_\Omega)} p(\bar{k}')$$

will be called *marginal* with respect to the set $\Omega \subset V$. Here $\mathcal{K}_\Omega(k'_\Omega)$ denotes the set $\{\bar{k} \in \mathcal{K}^{\mathcal{V}} \colon \bar{k}_\Omega = \bar{k}'_\Omega\}$.

Definition 6.1.0.5. Two probability distributions p and p' are called *equivalent* ($p \sim p'$) with respect to the neighborhood structure $\mathcal{E}$ if for all $e \in \mathcal{E}$ and all labellings $\bar{k}$ corresponding marginal probabilities are equal, i.e. $p_e(k_e) = p'_e(k_e)$.

Definition 6.1.0.6. We will say, that a positive probability distribution p ($p(\bar{k}) > 0$) defines a *Markov random field (MRF) in the neighborhood structure* $\mathcal{E}$ if for any other distribution p', equivalent to p with respect to $\mathcal{E}$ holds

$$\sum_{\bar{k} \in \mathcal{K}^{\mathcal{V}}} p(\bar{k}) \log p(\bar{k}) < \sum_{\bar{k} \in \mathcal{K}^{\mathcal{V}}} p'(\bar{k}) \log p'(\bar{k})\,.$$

(function $\sum_i p_i \log p_i$ is strictly convex, hence the strict inequality holds). In other words Markovian random field has the largest entropy in its equivalence class.

The following two propositions are the most important in the theory of Markov random fields.

Proposition 6.1.0.1. *Let p be MRF in the neighborhood structure $\mathcal{E}$. Then it can be represented as*

$$p(\bar{k}) = \frac{1}{Z} \prod_{e \in \mathcal{E}} \exp(-\lambda_e(k_e)) = \frac{1}{Z} \exp(-\sum_{e \in \mathcal{E}} \lambda_e(k_e)) \tag{6.1}$$

for some functions $\lambda_e \colon \mathcal{K}_e \to \mathbb{R}$.

Distributions of the form (6.1) are called *Gibbs distributions.*
$\triangleright$Proposition claims that numbers $p(\bar{k}), \bar{k} \in \mathcal{K}^{\mathcal{V}}$ minimize

$$\sum_{\bar{k} \in \mathcal{K}^{\mathcal{V}}} p(\bar{k}) \log p(\bar{k})$$

given

$$p_e(k_e) = \sum_{\bar{k}' \in \mathcal{K}_e(k_e)} p(\bar{k}')\text{-fixed},$$

$$\sum_{\bar{k} \in \mathcal{K}^{\mathcal{V}}} p(\bar{k}) = 1 \tag{6.2}$$

and $p(\bar{k}) > 0, \bar{k} \in \mathcal{K}^{\mathcal{V}}$.

Corresponding partial (without taking into account $p(\bar{k}) > 0, \bar{k} \in \mathcal{K}^{\mathcal{V}}$) Lagrangian reads

$$\Phi(p, \lambda) = \sum_{\bar{k} \in \mathcal{K}^{\mathcal{V}}} p(\bar{k}) \log p(\bar{k}) + \lambda_0 \left(\sum_{\bar{k} \in \mathcal{K}^{\mathcal{V}}} p(\bar{k}) - 1 \right) + \sum_{e \in \mathcal{E}} \sum_{k_e \in \mathcal{K}_e} \lambda_e(k_e) \left(\sum_{\bar{k}' \in \mathcal{K}_e(k_e)} p(\bar{k}') - p_e(k_e) \right).$$

Taking derivative with respect to p and assigning them to 0 gives

$$\frac{\partial \Phi}{\partial p(\bar{k})} = 1 + \log p(\bar{k}) + \lambda_0 + \sum_{e \in \mathcal{E}} \lambda_e(k_e) = 0$$

Hence

$$p(\bar{k}) = \exp(-(1 + \lambda_0)) \prod_{e \in \mathcal{E}} \exp(-\lambda_e(k_e))$$

that finalizes our proof. $\triangleleft$

Remark 6.1.0.1. Constant Z can be computed from normalization condition (6.2) and thus it reads

$$Z = \sum_{\bar{k} \in \mathcal{K}^{\mathcal{V}}} \prod_{e \in \mathcal{E}} \exp(-\lambda_e(k_e))$$

and is called *partition function.* One typically talks about *log-partition function*, equal to $\log Z$.

Definition 6.1.0.7. Let $\Omega \subset \mathcal{V}$. We will call the set $N(\Omega) = \{u \in \mathcal{V} \colon (u, v) \in \mathcal{E},\ v \in \Omega\}$ *a neighborhood* of Ω.

The following proposition claims that all Gibbs distributions posses Markov property.

Proposition 6.1.0.2. *Let p be Markov in the structure $\mathcal{E}$ and Ω be any subset of $\mathcal{V}$. Then the conditional probability $p_\Omega(k_\Omega|k_{\mathcal{V}\setminus\Omega})$ depends only on $k_{N(\Omega)}$.*

$\triangleright$Let us express $p_\Omega(k_\Omega|k_{\mathcal{V}\setminus\Omega})$ via $p(\bar{k})$:

$$p_\Omega(k'_\Omega|k'_{\mathcal{V}\setminus\Omega}) = \frac{p(\bar{k}')}{\sum_{\bar{k}\in\mathcal{K}_{\mathcal{V}\setminus\Omega}(k'_{\mathcal{V}\setminus\Omega})} p(\bar{k})} = \frac{\frac{1}{Z}\prod_{e\in\mathcal{E}}\exp(-\lambda_e(k'_e))}{\frac{1}{Z}\sum_{\bar{k}\in\mathcal{K}_{\mathcal{V}\setminus\Omega}(k'_{\mathcal{V}\setminus\Omega})}\prod_{e\in\mathcal{E}}\exp(-\lambda_e(k_e))}\,.$$

Please note that all terms $\lambda_e(k_e)$ for $e\cap\Omega=\emptyset$ enter all summands in the denominator and thus cancels with the same terms in the numerator. Taking this into account we obtain

$$p_\Omega(k'_\Omega|k'_{\mathcal{V}\setminus\Omega}) = \frac{\prod_{\substack{e\in\mathcal{E}\\ e\cap\Omega\neq\emptyset}}\exp(-\lambda_e(k'_e))}{\sum_{\bar{k}\in\mathcal{K}_{\mathcal{V}\setminus\Omega}(k'_{\mathcal{V}\setminus\Omega})}\prod_{\substack{e\in\mathcal{E}\\ e\cap\Omega\neq\emptyset}}\exp(-\lambda_e(k_e))}\,,$$

which finalizes proof of the lemma, since the right-hand side depends only on $k'_{N(\Omega)}$. $\triangleleft$

Due to the proved proposition Proposition 6.1.0.1 receives an important applied value: if is enough to define arbitrary functions λ_e and define distribution in the form (6.1) to **guarantee** its Markov property.

In the previous lectures we considered acyclic models (i.e. hidden Markov chains). According to the main consideration the probability distribution $p(\bar{k})$ was equal to (5.1):

$$p(\bar{k}) = p_0(k_0)\prod_{i=1}^{n} p_i(k_i|k_{i-1})\,. \tag{6.3}$$

Such representation (via *probabilities* $p_i(k_i|k_j)$) has a very important advantage for learning algorithms, which we will discuss later. It turns out, however, that if the underlying graph $\mathcal{G}$ for a MRF is acyclic, the Gibbs distribution (6.1) can be represented as a product of probabilities (6.3). To simplify notations we will prove this for a case when $\mathcal{G}$ is a chain, however the proof only slightly differs for a general acyclic case.

Proposition 6.1.0.3. *Let*

$$p(\bar{k}) = \frac{1}{Z}\prod_{i=1}^{n}\exp(-\lambda_i(k_i,k_{i-1})) = \frac{1}{Z}\prod_{i=1}^{n} f_i(k_i,k_{i-1})\,, \tag{6.4}$$

where $f_i(k_i,k_{i-1})>0,\ i=1\dots n, k_i,k_{i-1}\in\mathcal{K}$. Then $p(\bar{k})$ can be represented as (6.3).

$\triangleright$We will use few facts in our proof:

$$p_i(k_i|k_{i-1}) = \frac{p_i(k_i,k_{i-1})}{p_i(k_{i-1})}\,, \tag{6.5}$$

representation (6.4) allows us to use the $+,\times$ forward-backward Algorithm 1 to compute marginal probabilities

$$p_i(k_i,k_{i-1}) = \sum_{\bar{k}'\in\mathcal{K}_i(k_i,k_{i-1})} p(\bar{k}') = \frac{1}{Z}Q^F_{i-1}(k_{i-1})f(k_i,k_{i-1})Q^B_i(k_i),\ i=1\dots,n\,, \tag{6.6}$$

$$p_i(k_i) = \sum_{\bar{k}'\in\mathcal{K}_i(k_i)} p(\bar{k}') = \frac{1}{Z}Q^F_i(k_i)Q^B_i(k_i)\,, \tag{6.7}$$

where values $Q^F_i(k)$ and $Q^B_i(k)$ are computed by Algorithm 1 from functions $q_i(k_i,k_{i-1}) = f_i(k_i,k_{i-1})$ and $q_0(k_0)=1, k_0\in\mathcal{K}$.

Let us now consider

$$
\begin{aligned}
p_0(k_0)\prod_{i=1}^{n} p_i(k_i|k_{i-1}) &\overset{(6.5)}{=} p_0(k_0)\prod_{i=1}^{n}\frac{p_i(k_i,k_{i-1})}{p_i(k_{i-1})} \\
&\overset{(6.7),(6.6)}{=} \frac{1}{Z}Q_0^B(k_0)\frac{\prod_{i=1}^{n} Q_{i-1}^F(k_{i-1})f(k_i,k_{i-1})Q_i^B(k_i)}{\prod_{i=1}^{n} Q_i^F(k_{i-1})Q_i^B(k_{i-1})} \qquad (6.8)
\end{aligned}
$$

Canceling equal terms in numerator and denominator and taking into account that $Q_i^B(k_n) = 1,\ k_n \in \mathcal{K}$, we obtain (6.4), which finalizes the proof. ◁

Remark 6.1.0.2. All constructions remain the same if we consider joint probability of Markov labeling $\bar{k} \in \bar{\mathcal{K}}$ and (not obligatory Markov) observation $\bar{x} \in \bar{\mathcal{X}}$, i.e.

$$p(\bar{k},\bar{x}) = \frac{1}{Z(\bar{x})}\prod_{e\in\mathcal{E}}\exp(-\lambda_e(k_e,\bar{x})) \qquad (6.9)$$

it is also typical that $\lambda_e(k_e,\bar{x}) = -\lambda_e(k_e,x_e)$.

Remark 6.1.0.3. Each function of the form (6.9) can be written also in the form

$$p(\bar{k},\bar{x}) = \frac{1}{Z(\bar{x})}\prod_{v\in\mathcal{V}}\exp(-\lambda_v(k_v,\bar{x}))\prod_{e\in\mathcal{E}}\exp(-\lambda_e(k_e,\bar{x})) = \frac{1}{Z(\bar{x})}\exp\left(-\sum_{v\in\mathcal{V}}\lambda_v(k_v,\bar{x}) - \sum_{e\in\mathcal{E}}\lambda_e(k_e,\bar{x})\right) \qquad (6.10)$$

and vice versa. Function

$$E(\bar{k}|\bar{x}) = \sum_{v\in\mathcal{V}}\lambda_v(k_v,\bar{x}) + \sum_{e\in\mathcal{E}}\lambda_e(k_e,\bar{x})$$

is called *(Gibbs) energy* and $p(\bar{k},\bar{x}) = \frac{1}{Z}\exp(-E(\bar{k}|\bar{x}))$.

Example 6.1.0.1 (Potts model). Let $\mathcal{G}$ be a grid and $p(\bar{k})$ looks as (6.10), where $\lambda_v(k_v)$ are arbitrary and $\lambda_e(k_e) = \lambda_{uv}(k_u,k_v) = \begin{cases} 0, & k_u = k_v \\ 1, & k_u \neq k_v\,. \end{cases}$. This can be considered as a discrete analog of (multilabeling) Total Variation.

6.2 Bayesian Problems for MRF

6.2.1 0/1 loss

As in general case the loss (1.2) leads to the maximum a posteriory decision rule (1.3):

$$\bar{k}^* = \arg\max_{\bar{k}\in\bar{\mathcal{K}}} p(\bar{k}|\bar{x}) = \arg\max_{\bar{k}\in\bar{\mathcal{K}}} p(\bar{k},\bar{x}) \overset{(6.10)}{=} \arg\min_{\bar{k}\in\bar{\mathcal{K}}}\sum_{v\in\mathcal{V}}\lambda_v(k_v) + \sum_{uv\in\mathcal{E}}\lambda_{uv}(k_u,k_v)\,. \qquad (6.11)$$

Hence the MAP inference problem ≡ Energy minimization problem.

6.2.2 Locally additive loss

Let the loss has a form analogous to (5.12)

$$W(\bar{k},\bar{k}') = \sum_{v\in\mathcal{V}} w_v(k_v,k'_v)\,. \qquad (6.12)$$

Repeating the same transformation as in (5.13) leads to the result analogous to (5.14):

$$k_v^* = \arg\min_{k'_v \in \mathcal{K}_v} \sum_{k_v \in \mathcal{K}_v} w_v(k_v, k'_v) p_v(k_v|\bar{x})$$
$$= \arg\min_{k'_v \in \mathcal{K}_v} \sum_{k_v \in \mathcal{K}_v} w_v(k_v, k'_v) \frac{p_v(k_v, \bar{x})}{p(\bar{x})} = \arg\min_{k'_v \in \mathcal{K}_v} \sum_{k_v \in \mathcal{K}_v} w_v(k_v, k'_v) p_v(k_v, \bar{x}) \,, \quad (6.13)$$

where $p_v(k_v|\bar{x})$ denotes a conditional marginal probability

$$p_v(k_v|\bar{x}) = \sum_{\bar{k}'' \in \mathcal{K}_v(k_v)} p(\bar{k}''|\bar{x})$$

and $p_v(k_v, \bar{x})$ – the corresponding joint marginal probability

$$p_v(k_v, \bar{x}) = \sum_{\bar{k}'' \in \mathcal{K}_v(k_v)} p(\bar{k}'', \bar{x}) = \frac{1}{Z(\bar{x})} \sum_{\bar{k}' \in \mathcal{K}_v(k_v)} \prod_{e \in \mathcal{E}} \exp(-\lambda_e(k_e, \bar{x})) \,. \quad (6.14)$$

Computing (6.14) constitutes the main difficulty for solving (6.13) and is called (as in the case of acyclic HMM) *a marginalization inference problem.*

6.3 Learning MRF

Let us consider two most important learning problems formulations: discriminative and generative. In both cases we will suppose that the learning sample $\mathcal{L} = \{(\bar{k}^j, \bar{x}^j),\ j = 1, \ldots, m\}$ is given and the probability distribution $p(\bar{k}, \bar{x}; w)$ depends on parameter(s) w.

6.3.1 Structured Discriminative Learning

Discriminative learning is based on the assumption that MAP inference (6.11) is used. Let also $\lambda_c(k_c, \bar{x}) = \langle w_c, \phi_c(\bar{k}, \bar{x}) \rangle \,,\ c \in \mathcal{V} \cup \mathcal{E}$. Thus as for acyclic HMM case one has essentially to solve the (exponentially large) system of linear inequalities

$$E(\bar{k}^j|\bar{x}; w) = \langle w, \phi(\bar{k}^j, \bar{x}) \rangle < \langle w, \phi(\bar{k}, \bar{x}) \rangle = E(\bar{k}|\bar{x}; w),\ j = 1 \ldots, m,\ \bar{k} \in \bar{\mathcal{K}} \backslash \{\bar{k}^j\}$$

as good as possible ($\phi(\bar{k}^j, \bar{x})$ essentially is a collection of $\phi_c(\bar{k}, \bar{x})$ and w is a collection of w_c). In case of perceptron algorithm the key subproblem is finding unfulfilled inequality, which leads (as in case of acyclic HMM (5.15)) to the MAP inference problem

$$\bar{k}' = \arg\min_{\bar{k} \in \bar{\mathcal{K}}} \langle w, \phi(\bar{k}, \bar{x}^j) \rangle = \arg\min_{\bar{k} \in \bar{\mathcal{K}}} E(\bar{k}|\bar{x}; w) \,. \quad (6.15)$$

The only difference for a Structured SVM is that instead of the energy minimization pure problem (6.15) one has to solve the loss-augmented energy minimization problem analogous to (5.18):

$$\bar{k}' = \arg\min_{\bar{k} \in \bar{\mathcal{K}}} \langle w, \phi(\bar{k}, \bar{x}^j) \rangle - \Delta^{\bar{k},j} = \arg\min_{\bar{k} \in \bar{\mathcal{K}}} E(\bar{k}|\bar{x}) - \Delta^{\bar{k},j} \,.$$

Changing of the sign from + to − comparing to (5.18) is due to changing of max to min.

Summarizing, the MAP inference (energy minimization) problem is the key component of the discriminative learning of MRFs.

6.3.2 Generative Learning

As usual parameters w^* are inferred from the maximum likelihood principle, i.e.

$$w^* = \arg\max_w \prod_{j=1}^m p(\bar{k}^j, \bar{x}^j; w) = \arg\max_w \prod_{j=1}^m \frac{1}{Z(w, \bar{x}^j)} \exp(-E(\bar{k}^j | \bar{x}^j; w))$$

$$= \arg\min_w \sum_{j=1}^m Z(w, \bar{x}^j) + E(\bar{k}^j | \bar{x}^j; w)\,. \quad (6.16)$$

Regularized learning problem differs in a presence of the regularization term:

$$w^* = \arg\min_w C\|w\| + \sum_{j=1}^m Z(w, \bar{x}^j) + E(\bar{k}^j | \bar{x}^j; w)\,. \quad (6.17)$$

As soon as $E(\bar{k}^j|\bar{x}^j; w)$ considered to be linear with respect to w ($E(\bar{k}|\bar{x}; w) = \langle w, \phi(\bar{k}, \bar{x}) \rangle$) both regularized and non-regularized learning problems are convex and can be solved by convex optimization techniques as soon as the (sub-)gradient of the objective function and the function itself can be computed. Gradient of $E(\bar{k}|\bar{x}; w)$ is equal to $\phi(\bar{k}, \bar{x})$ and does not make any problem for ts computation. The problematic are computation of $Z(w, \bar{x}^j)$ and its gradient.

Since for $v \in \mathcal{V}$

$$p(\bar{x}; w) = \sum_{k_v \in \mathcal{K}} p_v(k_v, \bar{x}; w) = \frac{1}{Z(w, \bar{x})} \sum_{k_v \in \mathcal{K}} \sum_{\bar{k}'_c \in \bar{\mathcal{K}}_c(k_c)} \exp(-E(\bar{k}, \bar{x}; w))$$

thus

$$Z(w, \bar{x}) = \frac{1}{p(\bar{x}; w)} \sum_{k_v \in \mathcal{K}} \sum_{\bar{k}'_c \in \bar{\mathcal{K}}_c(k_c)} \exp(-E(\bar{k}, \bar{x}; w))$$

and computing $Z(w, \bar{x})$ is essentially the marginalization problem.

Bibliography

[1] Tutorial: Sebastian Nowozin and Christoph H. Lampert, *Structured Learning and Prediction in Computer Vision* http://www.nowozin.net/sebastian/papers/nowozin2011structured-tutorial.pdf