Machine Learning

Introduction to Structural Models

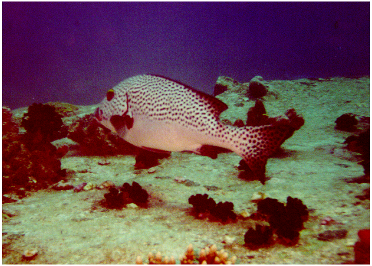
Dmitrij Schlesinger

WS2013/2014, 07.02.2014

COMPUTER VISION LAB DRESDEN

TECHNISCHE UNIVERSITÄT DRESDEN

The Dream ...

Color   Shape   Spatial relations   Relation “consists of”

Complete scene interpretation

Structural Models:
Data that consists of several parts, and not only the parts themselves contain information, but also the way in which the parts belong together.

A bit reality

- The set of parts is given (e.g. the set of pixels in low-level vision)
- An interpretation (label) should be assigned to each part
- There are only relations between parts, the description is not hierarchical – no relation “consists of” (at least not explicitly)

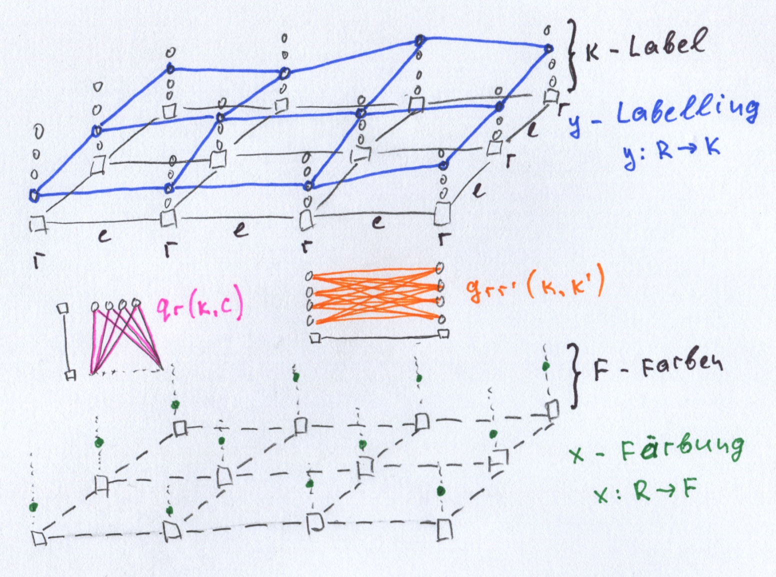
$\Rightarrow$ **Labeling Problems**

These problems can be at least formulated :-),
exact solutions are however very seldom :-(

A special case – the set of “pixels” is a chain
$\Rightarrow$ **Markov Chains**

Both formulations and algorithms are relatively simple

Markov Random Fields (simplified)



Markov Random Fields (simplified)

Graph $G = (R, E)$, K – label set,
F – the set of "elementary" observation (e.g. colors)
$y : R \to K$ – labeling, $y \in \mathcal{Y}$
$x : R \to F$ – observation (coloring), $x \in \mathcal{X}$

An elementary event is a pair (x, y), the (negative) energy:

$$E(x,y) = \sum_{rr' \in E} g_{rr'}(y_r, y_{r'}) + \sum_{r \in R} q_r(x_r, y_r)$$

The joint probability:

$$p(x,y) = \frac{1}{Z} \exp\Big[-E(x,y)\Big]$$

Partition Function:

$$Z = \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} \exp\Big[-E(x,y)\Big]$$

Inference – Bayesian Decision Theory

The principle is the same as for unstructured models – **minimize the Bayesian Risk**, i.e. the expected loss:

$$R\Big(e(x)\Big) = \sum_y p(x,y) \cdot C\Big(y, e(x)\Big) \to \min_e$$

remember that y are labellings $\to$ more complex algorithms

Special case $D = \mathcal{Y}$, $C(y,d) = \delta(y \neq d)$
$\to$ Maximum A-posteriori Decision

$$\begin{aligned} y^* &= \arg\max_y p(x,y) = \arg\min_y E(x,y) = \\ &= \arg\min_y \Big[\sum_{rr' \in E} g_{rr'}(y_r, y_{r'}) + \sum_{r \in R} q_r(x_r, y_r) \Big] \end{aligned}$$

Such tasks are known as **Energy Minimization** problems.

Additive loss is widely used as well (even more often in some particular domains) $\to$ leads to **Marginal based** decisions

Learning

Again, the basic principles are the same as for unstructured models:

- Statistical Learning – Maximum (Conditional) Likelihood
  - Supervised – gradient ascent (usually convex functions), stochastic gradient ...
  - Unsupervised – Expectation-Maximization Algorithm, gradient ascent ...
- Discriminative Learning – Empirical Risk minimization (sub-gradient algorithms), Large Margin learning (quadratic optimization) etc.

The difference to unstructured models – more complex algorithms, because for structured models practically nothing can be computed exactly :-(

MRF-s are members of the Exponential Family

The energy can be written in form of scalar product

$$E(x, y; \theta) = E(x, y; w) = \langle \phi(x, y), w \rangle$$

it is sometimes called "overcomplete" representation
(see the board for an example).

$\rightarrow$ For almost any kind of learning the sufficient statistics $\phi(x, y)$ are crucial.

Interesting: the Perceptron Algorithm (that is indeed very simple) is applicable for some tasks of discriminative learning.

Large Margin, SVM, Kernels etc. are possible as well

Some popular MRF-s

... of second order over the pixel grid, 4-neighborhood (because simple) – segmentation, denoising, deconvolution, stereo, motion fields etc.

... with **continuous** label spaces – denoising, stereo

... with **dense** neighborhood structure – shape modeling (e.g. curvature), segmentation

... of **higher order** – all the stuff above

Conditional Random Fields (CRF) – MRF-s that model posterior distributions of labellings instead of the joint ones

---

Energy Minimization $\rightarrow$ Hopfield Networks

(Hopfield) Networks with stochastic Neurons $\rightarrow$ MRF-s
also known as **Bolzmann Machines** (Feed-Forward as well)

Labeling Problems – a generalization

Constraint Satisfaction Problems (CSP) – OrAnd

$$\bigvee_{y} \left[\bigwedge_{r \in R} q_r(y_r) \wedge \bigwedge_{rr' \in E} g_{rr'}(y_r, y_{r'}) \right]$$

Energy Minimization – MinSum

$$\min_{y} \left[\sum_{r} q_r(y_r) + \sum_{rr'} g_{rr'}(y_r, y_{r'}) \right]$$

Partition Function – SumProd

$$\sum_{y} \left[\prod_{r} q_r(y_r) \cdot \prod_{rr'} g_{rr'}(y_r, y_{r'}) \right]$$

Generalized formulation

$$\bigoplus_{y} \left[\bigotimes_{r} q_r(y_r) \otimes \bigotimes_{rr'} g_{rr'}(y_r, y_{r'}) \right]$$

Labeling Problems – state-of-the-art

All labeling problems are NP-complete in general

All labeling problems can be solved exactly and efficiently by Dynamic Programming, if the graph is simple (chains, trees, cycles, partial w-trees of low tree-width)

For OrAnd problems over general graphs there is a dichotomy (polynomial $\leftrightarrow$ NP) with respect to the properties of g-functions

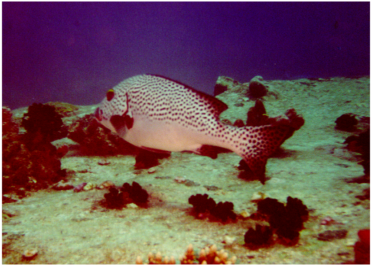
Submodular MinSum Problems are exactly solvable

There are many (good) approximate algorithms for MinSum over general graphs – relaxation based, search based, partial optimality etc.

There are also approximations for SumProd

There is also a dichotomy for MinSum and SumProd (?)

So what?



... see you in summer semester at “Computer Vision II” ...