Structured Prediction in Computer Vision: Take-off ahead

Sebastian Nowozin

Machine Learning and Perception Group
Microsoft Research Cambridge

25th January 2009

Structured Prediction

- *Prediction Function*: input domain $\mathcal{X}$, output domain $\mathcal{Y}$

$$f : \mathcal{X} \rightarrow \mathcal{Y}$$

- *Structured Prediction*: $\mathcal{Y}$ defined over multiple variables, which are subject to
  - dependencies, constraints, and relations.
- *Structured Output Learning*: given $\{(x_i, y_i)\}_{i=1,\ldots,N}$, learn f

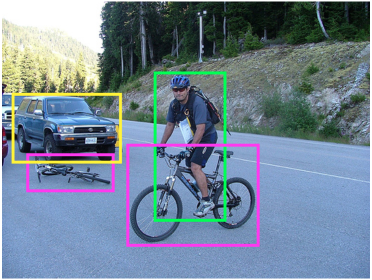
Structured Prediction

- *Prediction Function*: input domain $\mathcal{X}$, output domain $\mathcal{Y}$

$$f : \mathcal{X} \rightarrow \mathcal{Y}$$

- *Structured Prediction*: $\mathcal{Y}$ defined over multiple variables, which are subject to
  - dependencies, constraints, and relations.
- *Structured Output Learning*: given $\{(x_i, y_i)\}_{i=1,\dots,N}$, learn f

Examples: Structured Prediction



Object recognition

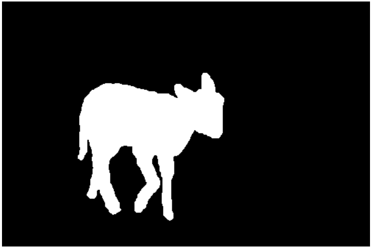
- $\mathcal{X}$: image
- $\mathcal{Y}$: bounding box object annotations

Examples: Structured Prediction

Denoising

- $\mathcal{X}$: image
- $\mathcal{Y}$: image

Examples: Structured Prediction



Segmentation

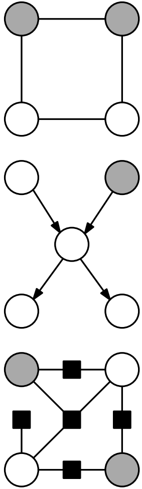
- $\mathcal{X}$: image
- $\mathcal{Y}$: binary segmentation mask

Advances in Structured Prediction

Advances in...

1. Graphical models: standard language, tools and best practises, discriminative models
2. Approximate inference: message passing algorithms, energy minimization
3. MAP-based parameter learning: max-margin approaches

Advances in Graphical Models



Graphical Models

- Statistical models for multiple random variables
- In a sense: universal
- Multiple forms,
  - Undirected graphical models (Markov networks, Markov random fields, Conditional random fields)
  - Directed graphical models (Bayesian networks)
  - Factor graphs (2000-)

2010: cross-domain defacto standard for structured models

- Books: (Koller and Friedman, 2009), (Wainwright and Jordan, 2008), (Bishop, 2007)
- Conferences: UAI, AISTATS, NIPS, ICML
- Journals: JMLR, MLJ

Advances in Approximate Inference

Interesting questions about graphical models are hard:

- computing marginal distributions
- computing modes
- computing normalizing constants

Progress

- Message passing algorithms (1997-): loopy BP, TRW, higher-order decompositions
- Graph-based energy minimization (1998-): graphcut methods, α-expansion, QPBO

Advances in Parameter Learning

Parameter learning, traditionally

- fixed or few parameters, cross-validation
- maximum likelihood estimation

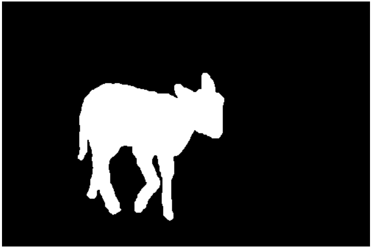
MAP-based training

- Often: computing mode is easier than computing marginals
- Max-margin methods (2001-): structured SVM, structured Perceptron
- Extends to other structured models (graph matching, sliding window classifiers, etc.)

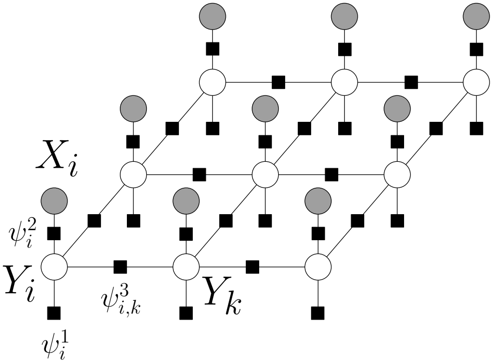
Talk

1. Higher-order Interactions in MRFs
2. Parameter Learning in MRFs

Challenge: Higher-order Potentials



Challenge: Higher-order Potentials



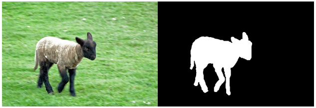
- X_i: observation variables (image statistics)
- Y_i: dependent variables (foreground/background)
- ψ_i^2: observation interactions
- $\psi_{i,k}^3$: pairwise interactions

Challenge: Higher-order Potentials (cont)

Sometimes one *knows* that a labeling must satisfy global properties.

Consider object segmentation

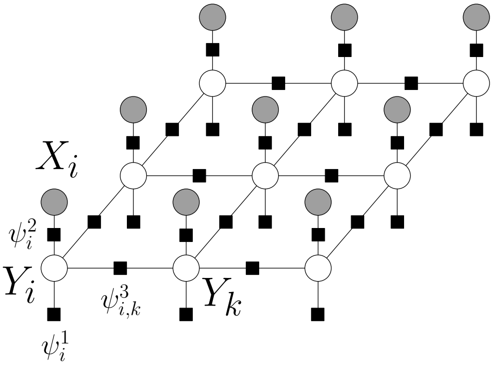
- "Connectedness": the resulting object segmentations should be connected
- "Hole-free": the object segmentations should have no holes
- etc.



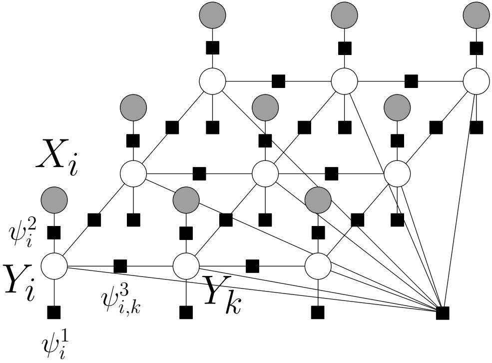
These properties are

- global properties,
- cannot be modelled by pairwise potentials,
- have not been successfully addressed.

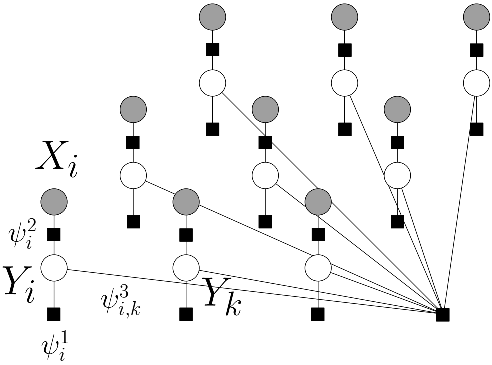
Challenge: Higher-order Potentials (cont)



Challenge: Higher-order Potentials (cont)



Challenge: Higher-order Potentials (cont)



Connectivity: Connected Subgraph Polytope

(Nowozin and Lampert, CVPR 2009),
(Nowozin and Lampert, SIAM IMS 2010, accepted)

Roadmap

- Global potential ψ_V: connectivity
- We want to restrict output labeling to labelings which are *globally connected* in the graph structure
- Derive a polyhedral set which captures connected subgraphs
- This set is the *connected subgraph polytope*
- Use MAP-MRF linear programming relaxation, but *intersect* with this set

Connected Subgraph Polytope (cont)

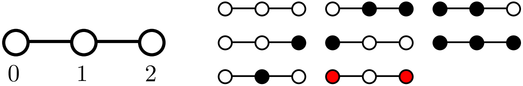
Definition (Connected Subgraph Polytope)

Given a simple, connected, undirected graph $G = (V, E)$, consider indicator variables $y_i \in \{0, 1\}$, $i \in V$. Let $C = \{\mathbf{y} : G' = (V', E')$ connected, with $V' = \{i : y_i = 1\}, E' = (V' \times V') \cap E\}$ denote the finite set of connected subgraphs of G. Then we call the convex hull $Z = \mathrm{conv}(C)$ the *connected subgraph polytope*.

Connected Subgraph Polytope (cont)

Definition (Connected Subgraph Polytope)

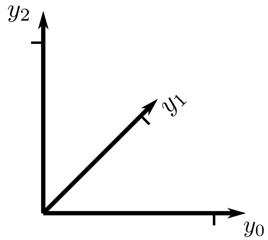
Given a simple, connected, undirected graph $G = (V, E)$, consider indicator variables $y_i \in \{0, 1\}$, $i \in V$. Let $C = \{\mathbf{y} : G' = (V', E') \text{ connected, with } V' = \{i : y_i = 1\}, E' = (V' \times V') \cap E\}$ denote the finite set of connected subgraphs of G. Then we call the convex hull $Z = \mathrm{conv}(C)$ the *connected subgraph polytope*.

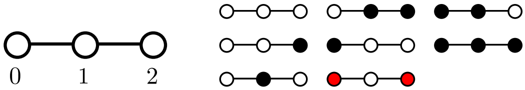


Connected Subgraph Polytope (cont)

Definition (Connected Subgraph Polytope)

Given a simple, connected, undirected graph $G = (V, E)$, consider indicator variables $y_i \in \{0, 1\}$, $i \in V$. Let $C = \{\mathbf{y} : G' = (V', E') \text{ connected, with } V' = \{i : y_i = 1\}, E' = (V' \times V') \cap E\}$ denote the finite set of connected subgraphs of G. Then we call the convex hull $Z = \mathrm{conv}(C)$ the *connected subgraph polytope*.

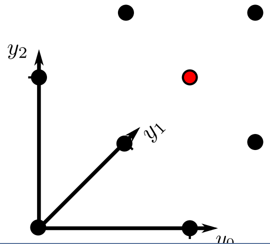


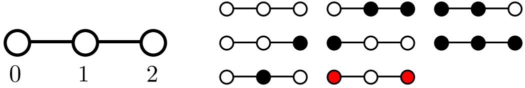


Connected Subgraph Polytope (cont)

Definition (Connected Subgraph Polytope)

Given a simple, connected, undirected graph $G = (V, E)$, consider indicator variables $y_i \in \{0, 1\}$, $i \in V$. Let $C = \{\mathbf{y} : G' = (V', E')$ connected, with $V' = \{i : y_i = 1\}, E' = (V' \times V') \cap E\}$ denote the finite set of connected subgraphs of G. Then we call the convex hull $Z = \mathrm{conv}(C)$ the *connected subgraph polytope*.

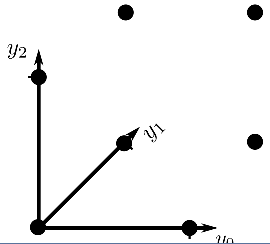


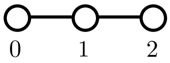


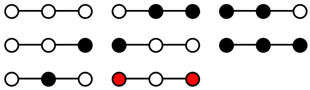
Connected Subgraph Polytope (cont)

Definition (Connected Subgraph Polytope)

Given a simple, connected, undirected graph $G = (V, E)$, consider indicator variables $y_i \in \{0, 1\}$, $i \in V$. Let $C = \{\mathbf{y} : G' = (V', E')$ connected, with $V' = \{i : y_i = 1\}, E' = (V' \times V') \cap E\}$ denote the finite set of connected subgraphs of G. Then we call the convex hull $Z = \mathrm{conv}(C)$ the *connected subgraph polytope*.



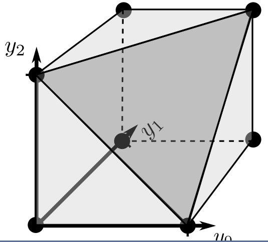


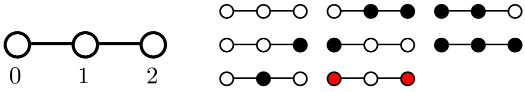


Connected Subgraph Polytope (cont)

Definition (Connected Subgraph Polytope)

Given a simple, connected, undirected graph $G = (V, E)$, consider indicator variables $y_i \in \{0, 1\}$, $i \in V$. Let $C = \{\mathbf{y} : G' = (V', E')$ connected, with $V' = \{i : y_i = 1\}, E' = (V' \times V') \cap E\}$ denote the finite set of connected subgraphs of G. Then we call the convex hull $Z = \mathrm{conv}(C)$ the *connected subgraph polytope*.





Hardness Results

Theorem (Karp, 2002)

It is NP-hard to optimize a linear function over $Z = conv(C)$.

The problem is known as Maximum-Weight Connected Subgraph Problem and has been shown to be NP-hard (Karp, 2002). Therefore,

- we plan to intersect $\mathrm{conv}(C)$ with the MAP-MRF LP relaxation
- hence, we will optimize a linear function over this polytope,
- from the theorem it follows that optimizing a linear function over $\mathrm{conv}(C)$ is NP-hard.
- (moreover: no additional results about Z known)

What to do?

- From insight into the polytope we will derive a tight relaxation to $\mathrm{conv}(C)$ which is polynomially solvable.

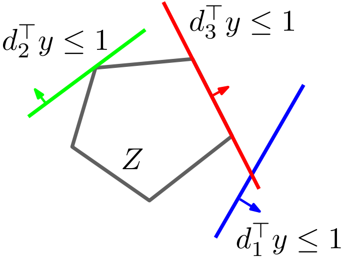
Facets and Valid Inequalities

Convex polytopes have two equivalent representations

- As a convex combination of extreme points
- As a set of facet-defining linear inequalities

A linear inequality with respect to a polytope can be

- *valid*, does not cut off the polytope,
- *representing a face*, valid and touching,
- *facet-defining*, representing a face of dimension one less than the polytope.



Warmup

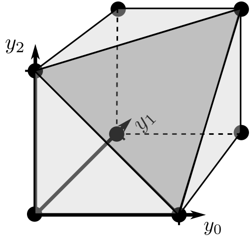
Some basic properties about the connected subgraph polytope Z. Note that Z depends on the graph structure.

Lemma

$dim(Z) = |V|$, that is, Z has full dimension.

Lemma

For all $i \in V$, the inequalities $y_i \geq 0$ and $y_i \leq 1$ are facet-defining for Z.

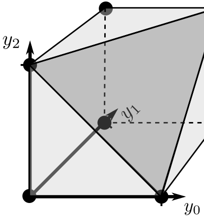


An Exponential-sized Class of Facet-defining Inequalities

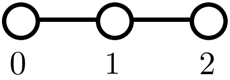
Theorem

The following linear inequalities are facet-defining for $Z = conv(C)$.

$$y_i + y_j - \sum_{k \in S} y_k \leq 1, \quad \forall (i,j) \notin E : \forall S \in \bar{\mathcal{S}}(i,j). \tag{1}$$



$y_0 + y_2 - y_1 \leq 1.$



Intuition

$$y_i + y_j - \sum_{k \in S} y_k \leq 1, \quad \forall (i,j) \notin E : \forall S \in \bar{\mathcal{S}}(i,j)$$

If two vertices i and j are selected ($y_i = y_j = 1$, shown in black), then any set of vertices which separate them (set S) must contain at least one selected vertex.

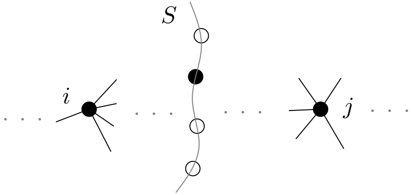


Figure: Vertex i and j and one vertex separator set $S \in \bar{\mathcal{S}}(i,j)$.

Formulation

Theorem

C, the set of all connected subgraphs, can be described exactly by the following constraint set.

$$y_i + y_j - \sum_{k \in S} y_k \leq 1, \forall (i,j) \notin E : \forall S \in \mathcal{S}(i,j), \qquad (2)$$

$$y_i \in \{0,1\}, \qquad i \in V. \qquad (3)$$

This means

- inequalities together with integrality are a *formulation* of the set of connected subgraphs,
- we can attempt to relax (3) to

$$y_i \in [0;1], \qquad i \in V.$$

- Problem: number of inequalities (2) is exponential in $|V|$.

Separation Problem

Optimization over the relaxed formulation

- *still tractable*,
- finding violated inequalities – the *separation problem* – can be solved efficiently.

Theorem (Polynomial-time Separation)

For a given point $\mathbf{y} \in [0;1]^{|V|}$ *to find a violated inequality (1) or prove that no such violated inequality exists requires only time polynomial in* $|V|$*.*

Summary: Connected Subgraph Polytope

- Convex hull of all connected subgraphs
- Convex and described by finite set of linear inequalities
- NP-hard to optimize over, exponentially sized description
- Identified a general class of facet-defining, polynomial-time separable inequalities $\rightarrow$ relaxation
- Devised an efficient separation procedure (by solving linear max-flow problems on a auxiliary graph)

$\rightarrow$ Let's put this into practise for random fields!

From Polytopes to Potentials

Remember the MAP-MRF LP relaxation

- $\mu^j(\mathbf{y}) = [\mu_1(y_j), \dots, \mu_{|V|}(y_j)]^\top \in [0;1]^{|V|}$,
  the set of variables indicating assignment to class j over all vertices

Enforce connectivity for the vertices assigned to the j'th class:

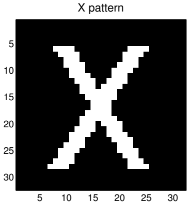
$$E^V_{\mathrm{hard(j)}}(\mathbf{y}) = \begin{cases} 0 & \mu^j(\mathbf{y}) \in Z \\ \infty & \text{otherwise} \end{cases}$$

- Realized by intersecting the feasible set of $\mu^j(\mathbf{y})$ with the Connected Subgraph Polytope.
- Alternatively: *soft potential* $E^V_{\mathrm{soft(j)}}$

Toy Experiment: Denoising

Simple denoising task

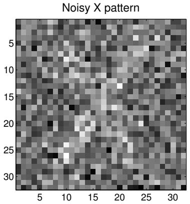
- 30×30 grid graph, 4-nn connectivity
- Two classes: foreground, background
- Denoise X-pattern from noisy measurements



Setup

- Noisy observations (Gaussian noise, σ)
- Associative/attractive pairwise Potts potentials (noise level k)

1. MRF
2. MRFcomp: MRF + select largest connected foreground component
3. CMRF (MRF with hard connectedness potential)



Results

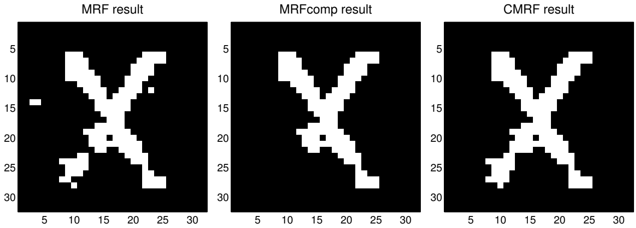


Figure: MRF/MRFcomp/CMRF: $E = -985.61$, $E = -974.16$, $E = -984.21$

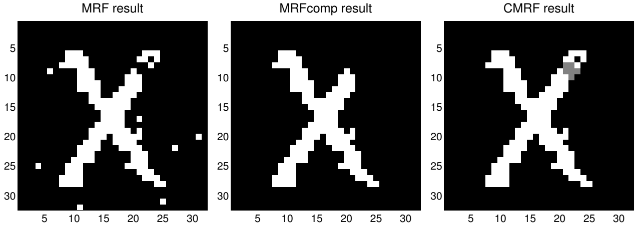


Figure: MRF/MRFcomp/CMRF: $E = -980.13$, $E = -974.03$, $E = -976.83$

Results (cont)

Discretized (σ, k)-parameter plane, mean error over 30 runs

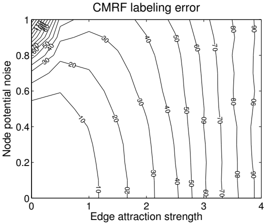


Figure: Connected MRF labeling error.

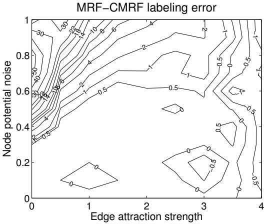


Figure: Error difference MRF-CMRF.

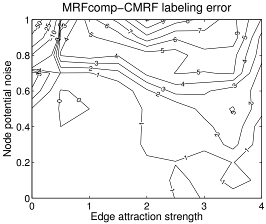


Figure: Error diff. MRFcomp-CMRF.

Results (cont)

For this toy experiment

- Connectivity assumption is known to be true
- Connectivity prior produces excellent results
- Truly global potential is tractable

Experiment: Recognition and Segmentation

(Nowozin and Lampert, CVPR 2009), PASCAL VOC 2008

Figure: Image/CRF/CRF+conn. Case where connectedness helps: the local evidence is scattered, enforcing connectedness (right) helps.

Figure: Image/CRF/CRF+conn. Connectedness can remove clutter: local evidence (edges on the runway) is overridden.

Conclusions

Summary

- Experimentally: connectedness prior reduces error on synthetic and real tasks
- Overcome the limitation of only considering local interactions in discrete random field models
- Principled way to derive global potential functions
- Polyhedral combinatorics opens a way to better model high-level vision tasks

Challenge: Parameter Learning

Parameter learning *required* for

- structured models in general,
- high level vision tasks,
- combining multiple features.

→ plenty of methods exist

(→ even just for CRFs, even just for image segmentation)

Challenge: Parameter Learning

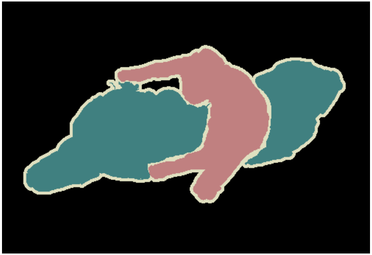
Parameter learning *required* for

- structured models in general,
- high level vision tasks,
- combining multiple features.

$\longrightarrow$ plenty of methods exist

($\longrightarrow$ even just for CRFs, even just for image segmentation)

Task: Object Class Image Segmentation



- PASCAL VOC 2009 segmentation challenge

Reference	Structure	Learning
Szummer et al., 2008	pixel grid	struct. SVM
Nowozin & Lampert, 2008	superpixels	struct. SVM
Reynolds & Murphy, 2007	superpixel tree	piecewise, CV
Plath et al., 2009	superpixel tree	piecewise, CV
Winn & Shotton, 2006	pixel grid, pixel blocks	CV
Shotton et al., 2007	pixel grid	piecewise, holdout validation
Kohli et al., 2008	pixel grid, superpixels	piecewise, CV
Ladický et al., 2009	pixel grid, superpixels	piecewise, heuristic
Gould et al., 2008	superpixels	piecewise
Batra et al., 2008	superpixels	CMLE (BP)
Schnitzspan et al., 2008	multiscale grid	mixed SVM and CMLE (BP)
Kumar & Hebert, 2003	pixel blocks	pseudolikelihood
Munoz et al., 2009	pixel grid, superpixels	piecewise, struct. SVM
He et al., 2004	pixel grid, blocks	piecewise, CMLE (contrastive div.)

Parameter Learning

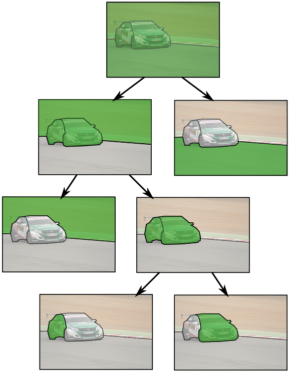
Status quo

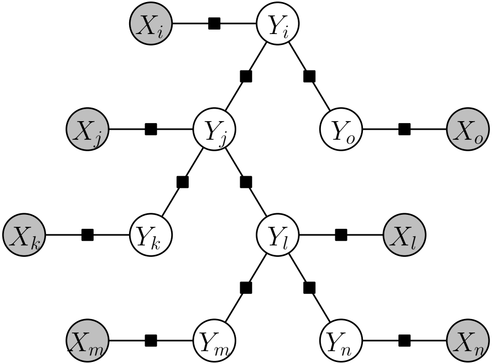
- method of choice: piecewise training and cross validation
- advanced methods are used, but advantage is unclear: structured SVM, approximations (pseudolikelihood, loopy BP, contrastive divergence)

This work

- Simple and tractable model
- Examine some effects and choices in parameter learning

Model





- Log-linear CRF on hierarchical segmentation ($\approx$ 100 superpixels)
- $\geq 10^5$ parameters, jointly learned, multiple features
- Loss due to representation, but still $\geq$ 90% VOC 2009 segmentation measure possible

Result 1: Learning Tradeoff

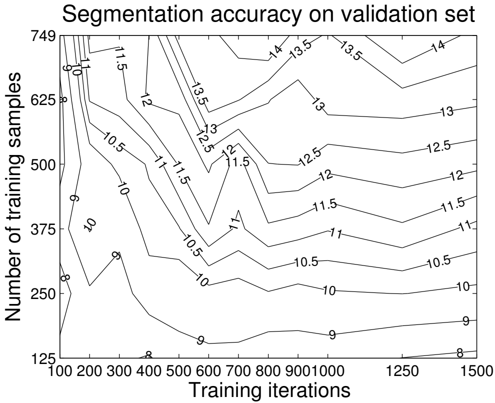


Figure: VOC 2009 segmentation accuracy on the validation set as a function of the training set size and number of LBFGS iterations.

- Training set size is the *limiting dimension*

Result 2: Feature Combination

Unary features	seg-val	Train time	D
SIFT	6.13%	22h01m	11,193
QHOG	8.40%	19h30m	11,193
QPHOG	7.35%	36h03m	11,193
STF	6.76%	39h36m	42,945
QHOG,QPHOG	10.92%	24h35m	21,945
SIFT,QHOG,QPHOG	14.54%	26h17m	32,697
SIFT,QHOG,QPHOG,STF	15.04%	41h39m	75,201

Table: The result of feature combination at the unary factors.

- No surprise: the more features, the better
- Despite many parameters: no overfitting observed

Result 3: Piecewise Training

Model	seg-val	Training time
Unary only,	9.98%	2h15m
Piecewise, Potts	14.50%	(2h15)+10h28m
Joint	14.54%	26h17m

- Piecewise training competitive

Result 4: Structured SVM

Pairwise factor	Accuracy (val) CMLE	Accuracy (val) SVM	Training time CMLE	Training time SVM
$E^{2,P}$	13.65%	13.21%	24h11m	165h10m
...	...	...		

- Performance competitive with maximum likelihood
- For many parameters and large values of C: intractable using simple cutting-plane model

Conclusion

Best practises for CRF parameter learning in *tractable* models

- Our observation: many parameters do not hurt, infact they help
- Limiting dimension: training data
- Piecewise training works well

Open questions and future directions

- More robust structured SVM methods (recent works: "1-slack" formulation, bundle methods)
- *Intractable* models: what conclusions hold?
- *Intractable* models: good approximate inference → good parameter learning? cf. (Kulesza and Pereira, 2007), (Finley and Joachims, 2008), (Martins et al., 2009)

References

M. Szummer, P. Kohli, and D. Hoiem.
Learning CRFs using graph cuts.
In *European Conference on Computer Vision*. Springer, 2008.

MAP-MRF LP Relaxation

(Integer) linear programming formulation for MAP-MRF

$$\min_{\boldsymbol{\mu}} \quad \sum_{i \in V} \sum_{y_i \in \mathcal{Y}_i} \mu_i(y_i) \left(E^{\{i\}}(y_i; \mathbf{x}, \mathbf{w}) \right)$$

$$+ \sum_{\substack{(i,j) \\ \in E}} \sum_{\substack{(y_i, y_j) \\ \in \mathcal{Y}_i \times \mathcal{Y}_j}} \mu_{i,j}(y_i, y_j) \left(E^{\{i,j\}}(y_i, y_j; \mathbf{x}, \mathbf{w}) \right)$$

$$\text{sb.t.} \quad \sum_{y_i \in \mathcal{Y}_i} \mu_i(y_i) = 1, \qquad i \in V,$$

$$\sum_{y_j \in \mathcal{Y}_j} \mu_{i,j}(y_i, y_j) = \mu_i(y_i), \; (i,j) \in E, y_i \in \mathcal{Y}_i,$$

$$\mu_i(y_i) \in \{0, 1\}, \quad i \in V, y_i \in \mathcal{Y}_i,$$

$$\mu_{i,j}(y_i, y_j) \in \{0, 1\}, \; (i,j) \in E, (y_i, y_j) \in \mathcal{Y}_i \times \mathcal{Y}_j.$$

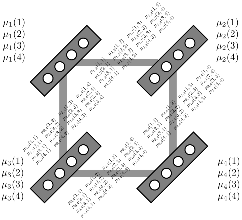


Figure: Variables in the LP for our example: $4 \cdot 4$ node variables, $4 \cdot 4 \cdot 4 = 64$ edge variables.

MAP-MRF LP Relaxation

(Integer) linear programming formulation for MAP-MRF

$$
\begin{aligned}
\min_{\boldsymbol{\mu}} \quad & \sum_{i \in V} \sum_{y_i \in \mathcal{Y}_i} \mu_i(y_i) \left(E^{\{i\}}(y_i; \mathbf{x}, \mathbf{w}) \right) \\
& + \sum_{\substack{(i,j) \\ \in E}} \sum_{\substack{(y_i, y_j) \\ \in \mathcal{Y}_i \times \mathcal{Y}_j}} \mu_{i,j}(y_i, y_j) \left(E^{\{i,j\}}(y_i, y_j; \mathbf{x}, \mathbf{w}) \right) \\
\text{sb.t.} \quad & \sum_{y_i \in \mathcal{Y}_i} \mu_i(y_i) = 1, \qquad i \in V, \\
& \sum_{y_j \in \mathcal{Y}_j} \mu_{i,j}(y_i, y_j) = \mu_i(y_i), \; (i,j) \in E, y_i \in \mathcal{Y}_i, \\
& \color{red}{\mu_i(y_i) \in [0,1]}, \quad i \in V, y_i \in \mathcal{Y}_i, \\
& \color{red}{\mu_{i,j}(y_i, y_j) \in [0,1]}, \; (i,j) \in E, (y_i, y_j) \in \mathcal{Y}_i \times \mathcal{Y}_j.
\end{aligned}
$$

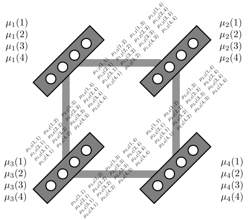


Figure: Variables in the LP for our example: $4 \cdot 4$ node variables, $4 \cdot 4 \cdot 4 = 64$ edge variables.