MDI341 Structured Data

2 - Energy-based approaches for structured prediction

Florence d'Alché

Contact: `florence.dalche@telecom-paris.fr`,
Télécom Paris, Institut Polytechnique de France

Structured output prediction with energy-based methods

Statistical framework of structured prediction

Let $P(x, y)$ be the unknown true distribution of the data
Let $\mathcal{S} = \{(x_i, y_i), i = 1, \ldots, n\}$ be iid sample from P.
Let $\Delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ a loss function

- Find g that minimizes the expected loss $\mathbb{E}_{(x,y)}[\Delta(y, f(x))]$
- With $f(x) = \arg\max_{y \in \mathcal{Y}} g(x, y, \mathbf{w})$

Energy-based learning

- Choose the space of functions g
- Define an appropriate loss $\Delta(y, y')$,
- Define a surrogate loss ℓ and solve an optimization problem to learn g,

Energy-based learning: score functions

- $g(x,y) = w^T \phi(x,y)$:struct perceptron, struct-SVM
- $g(x,y) = \sum_r g_r(x, y_r)$ with $g_r(x, y_r) = NN_r(x,y)$ and r index subsets of components: deep structured prediction (Chen et al., 2015), structured prediction neural networks (Bellanger and Mc Callum, 2016)
- $g(x,y) = \hat{P}(Y = y|x)$: logistic regression, CRF, graphical probabilistic models

Losses for multiple classes

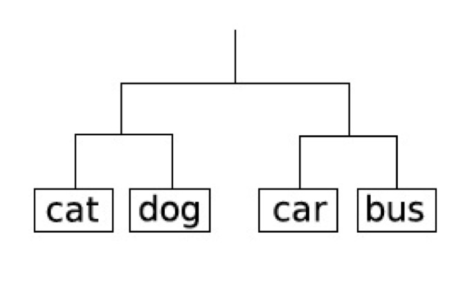
Without a hierarchy among classes

$$\Delta(y, y') = 1, \text{if } y \neq y', 0 \text{ otherwise}$$

Loss for a hierarchy of classes

$$\Delta(y, y') = \frac{1}{2} D_{tree}(y, y')$$

For the shown hierarchy: $\Delta(cat, cat) = 0$, $\Delta(cat, dog) = 1$, $\Delta(cat, bus) = 2; \ldots$



Loss for localization of an object in an image

Object detection: find a bounding box and a class

Credit Blaschko & Lampert, ECCV2008.

Localization of an object in an image: choice of loss function

Blatschko and Lampert (2008) propose to solve it as a a structured prediction problem.
$\mathcal{X} = \{images\}$ and $\mathcal{Y} = \{bounding\ boxes\ in\ images\}$
More precisely, $\mathcal{Y} = \{(\omega, t, l, b, r) | \omega \in \{+1, -1\}, (t, l, b, r) \in \mathbb{R}^4\}$
For $\omega = -1$, the vector (t,l,b,r) is ignored.

Loss function

$\Delta(y, y_i) = \Delta_i(y) = 1 - \frac{Area(y_i \cap y)}{area(y_i \cup y)}$ if $y_{i\omega} = y_\omega = 1$,
$1 - (\frac{1}{2}(y_{i\omega} y_\omega + 1))$ otherwise.

Learning with Max margin Approaches

Structured SVM

$$\ell(x, y_i, \mathbf{w}) = [\max_y(\Delta(y_i, y) + \mathbf{w}^T\phi_i(y) - \mathbf{w}^T\phi_i(y_i))]_+$$

ℓ is an upper bound of Δ, continuous and convex. Regularized Empirical Risk Minimization :

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{n}\sum_i \max(0, \max_y[\Delta(y_i, y) + \mathbf{w}^T\phi_i(y) - \mathbf{w}^T\phi_i(y_i)])$$

Max Margin Approaches: pb in the primal

Margin maximization with slack variables

$\min_{\mathbf{w}} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_i \xi_i$

s.t. :

$\forall i, \forall y \neq y_i, \mathbf{w}^T \phi_i(y_i) + \xi_i \geq \mathbf{w}^T \phi_i(y) + \Delta_i(y)$

$\forall i, \xi_i \geq 0$

$\Delta_i(y) = \Delta(y_i, y)$ reflects the loss between y and y_i

Pb in the dual

We note: $\delta\phi(x_i, y_i, y) = \phi(x_i, y_i) - \phi(x_i, y) = \phi_i(y_i) - \phi_i(y)$

Dual Struct-SVM pb

$\max_\alpha \sum_{iy} \alpha_{iy} \Delta_i(y) - \frac{1}{2} \sum_{i,j,y,y'} \alpha_{iy}\alpha_{jy'} < \delta\phi(x_i, y_i, y), \delta\phi(x_j, y_j, y') >$

s.t. :

$\forall i = 1, \dots n, \sum_{y \in \mathcal{Y}} \alpha_{i,y} \leq \frac{C}{N}$

- Prediction function:
  $f(x) = \arg\max_y (\sum_{iy'} \alpha_{iy'}(\phi_i(y') - \phi_i(y_i)))^T \phi(x, y)$

Le Cun et al. Energy-based models (2006)

Solving the optimization pb

- Solving the problem in the dual
- However there are many constraints (size $|\mathcal{Y}|$ n)
- Working set training (active constraints)
- Eventually : easy to kernelize (we'll see that later)

Algorithm

Working set S-SVM - Learning Algorithm

1. input: $(x_1, y_1), (x_n, y_n)$, C, ϵ
2. $S \leftarrow 0$
3. repeat
   - for i=1,..., n
     - solution to the quadratic pb with constraints from the working set S (subspace ascent)
     - Define: H(y) $= \Delta(y, y_i) - \mathbf{w}^T \phi_i(y_i) + \mathbf{w}^T \phi_i(y)$
     - with $\mathbf{w} := \sum_j \sum_{y' \in S_j} \alpha_{jy'} (\phi(x_j, y_j) - \phi(x_j, y'))$
     - Compute $\hat{y} = \arg\max_{y \in \mathcal{Y}} H(y)$
     - Compute $\xi_i = \max(0, \max_{y \in S_i} H(y))$
     - If $H(\hat{y}) > \xi_i + \epsilon$ then
       - $S \leftarrow S \cup \{(i, \hat{y})\}$ add the most violated constraint
   - End for
4. until S has not significantly changed.

More efficient: subgradient algorithm

For the penalized unconstrained formulation :

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{n}\sum_i \max(0, \max_y[\Delta(y_i, y) + \mathbf{w}^T\phi_i(y) - \mathbf{w}^T\phi_i(y_i)])$$

Convex, non-differentiable loss:

We can use sub-gradient method instead of gradient decent algorithm.

Subgradient Descent algorithm for Struct-SVM

Algo

- $\mathbf{w} = 0$
- for t=1,..., T
    - for i=1,..., n
    - $\hat{y} := \arg\max_{y \in \mathcal{Y}} \Delta(y, y_i) + < \mathbf{w}, \phi(x_i, y) >$ (loss augmented step)
    - $v_i := \phi(x_i, \hat{y}) - \phi(x_i, y_i)$
    - endfor
- $\mathbf{w} := \mathbf{w} - \eta_t(\mathbf{w} - \frac{C}{n} \sum_i v_i)$
- endfor

N.B.: Stochastic updates are usually chosen

Subgradient

Want to minimize: $\min_{\mathbf{w}} \frac{1}{2} \left\| \mathbf{w} \right\|^2 + \frac{C}{n} \ell_i(\mathbf{w})$
with

$$\begin{aligned} \ell_i^y(\mathbf{w}) &= \Delta(y_i, y) + \mathbf{w}^T \phi_i(y) - \mathbf{w}^T \phi_i(y_i) \\ \ell_i(\mathbf{w}) &= \sum_i \max(0, \max_y \ell_i^y(\mathbf{w})) \end{aligned}$$

The subgradient of $\ell_i(\mathbf{w})$ for $\mathbf{w} = \mathbf{w}_0$ is the vector v defined as follows:

$$\begin{aligned} \hat{y} &= \arg\max \ell_i(\mathbf{w}_0) \\ v &= \nabla \ell_i^{\hat{y}}(\mathbf{w}_0) \end{aligned}$$

More efficient Struct SVM learning

- Distributedly training structured support vector machines based on a distributed block-coordinate descent method: Lee, Chang, Upaydin and Roth, NIPS 2016
- Alternating DirectionMethod of Multipliers (ADMM) for structural SVM : Balamurugan et al; SDM 2011, and SIAM 2016.

To go further: Kernelization

Ref: Blatchko and Lampert 2008.

- A joint kernel function k:$(\mathcal{X} \times \mathcal{Y}) \times (\mathcal{X} \times \mathcal{Y}) \rightarrow \mathbb{R}$
    - $k((x,y),(x',y')) =< \phi(x,y), \phi(x',y') >$
    - k measures how the two pairs (x,y) and (x',y') are similar
- We can show that $< \delta\phi(x_i, y_i, y), \delta\phi(x_j, y_j, y') >= k((x_i, y),(x_j, y_j)) - k((x_i, y_i),(x_j, y')) - k((x_i, y),(x_j, y_j)) + k((x_i, y),(x_j, y'))$
- Let us denote $< \delta\phi(x_i, y_i, y), \delta\phi(x_j, y_j, y') >= k_{ijyy'}$

Pb in the dual

Dual Struct-SVM pb

$\max_{\alpha} \sum_{i,y} \alpha_{iy} \Delta_i(y) - \frac{1}{2} \sum_{i,j,y,y'} \alpha_{iy} \alpha_{jy'} k_{ijyy'}$

s.t. :

$\forall i = 1, \ldots n, \sum_{y \in \mathcal{Y}} \alpha_{i,y} \leq \frac{C}{N}$

- Prediction function:
  $f(x) = \arg\max_{y \in \mathcal{Y}} \sum_{iy'} \alpha_{iy'} [k((x_i, y'), (x, y)) - k((x_i, y_i), (x, y))]$

Examples of joint kernels

Both x and y decompose into components

$$k((x,y),(x',y')) = \sum_p k_p((x_p,y_p),(x'_p,y'_p))$$

Kernel for localization of an object in an image

Now the kernel used here is defined by

$$k_{restr}((x,y),(x',y')) = k(x_{|y}, x_{|y'})$$

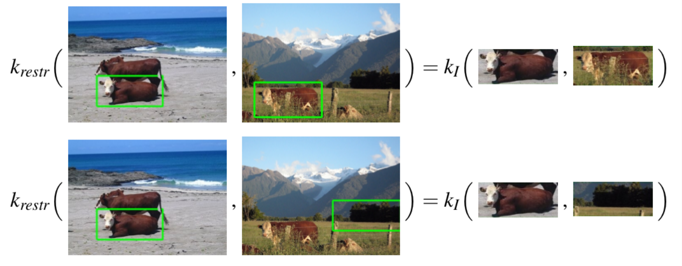


Figure from Blaschko and Lampert
(`http://www.robots.ox.ac.uk/~vgg/research/joint_kernel_detection/`)

Restriction kernel

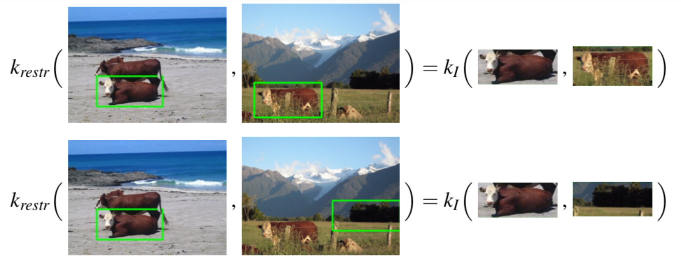


Figure from Blaschko and Lampert (`http://www.robots.ox.ac.uk/~vgg/research/joint_kernel_detection/`)

Prediction phase

- During training computation of ξ_i
- Prediction function:
  $f(x) = \arg\max_y (\sum_{iy'} \alpha_{iy'} (\phi_i(y') - \phi_i(y_i)))^T \phi(x, y)$

Very expensive, need to be done for each possible window (or a sample of them) in the image:

Alternative
Branch and Bound algorithm to search $\mathcal{Y}$.

Branch and bound algorithm

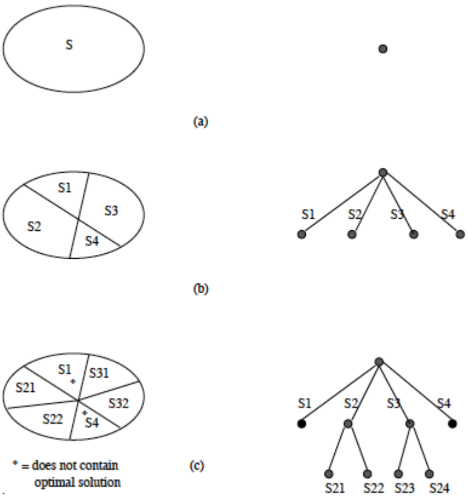


Figure 1: Illustration of the search space of B&B.

Results on VOC 2006 (Lampert, Blaschko 2006)

Visual Object Recognition Challenge 2006.

Area under the Precision Recall curve (Pr = TP/P, Rec = TP)

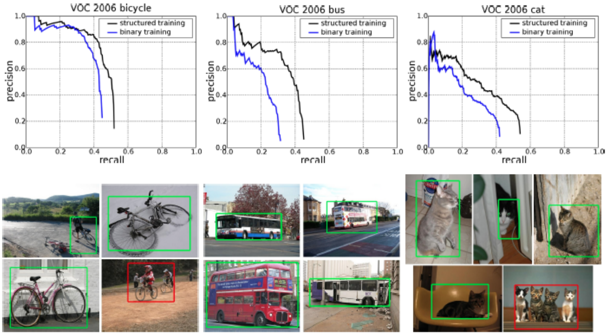


Fig. 5. Precision–recall curves and example detections for the PASCAL VOC **bicycle**, **bus** and **cat** category (from left to right). Structured training improves both, precision and recall. Red boxes are counted as mistakes by the VOC evaluation routine, because they are too large or contain more than one object.

Software libraries

SVM^{struct}

Author: Thorsten Joachims

Cornelll University

Language : in C but Python interface exists

`https://www.cs.cornell.edu/people/tj/svm_light/svm_struct.html`

▸ SVM-struct

PyStruct

Authors: Andreas C. Mueller, Sven Behnke

University of Bonn

Structured Output Prediction

`https://pystruct.github.io/` ▸ PyStruct

PyTorch (2016)

Authors: Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan

`https://pytorch.org/` ▸ PyTorch Based on Torch: Ronan Collobert, Koray Kavukcuoglu, Clement Farabet.

Many Deep Structured prediction tools are written in PyTorch

GitHub, Inc. (US) https://github.com/sgrvinod/Deep-Tutorials-for-PyTorch

In each tutorial, we will focus on a specific application or area of interest by implementing a model from a research paper.

Application	Paper	Tutorial	Status
Image Captioning	*Show, Attend, and Tell*	a PyTorch Tutorial to Image Captioning	Complete
Sequence Labeling	*Empower Sequence Labeling with Task-Aware Neural Language Model*	a PyTorch Tutorial to Sequence Labeling	Complete
Text Classification	*Hierarchical Attention Networks for Document Classification*	a PyTorch Tutorial to Text Classification	Code complete, tutorial in-progress
Object Detection	*SSD: Single Shot MultiBox Detector*	a PyTorch Tutorial to Object Detection	Code complete, tutorial in-progress
Text Recognition	*An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition*	a PyTorch Tutorial to Text Recognition	Planned
Text Summarization	*Get To The Point: Summarization with Pointer-Generator Networks*	a PyTorch Tutorial to Text Summarization	Planned
Machine Translation	*Attention Is All You Need*	a PyTorch Tutorial to Machine Translation	Planned
Semantic Segmentation	*Pyramid Scene Parsing Network*	a PyTorch Tutorial to Semantic Segmentation	Planned

Conclusion and References

Conclusion

Structured Output Prediction

- Enlarges considerably the scope of supervised learning
- Refers to a large set of problems
- Prediction in the original output space is an issue (usually expensive)
- Learning can be implemented either by learning a score or a surrogate problem
- No restriction to a specific model (MLP, CNN, SVM, trees)
- End-to-end learning refers to approaches in which the decoding problem is rather simple
- Theoretical framework for structured output learning: PAC-Bayes for SOP, Calibration theory

New challenges

- Bridge the gap between approaches that work for millions of data (deep learning) but still very computationally demanding, with nearly no theory and surrogate loss approaches
- Combination of deep architectures with margin losses, or with a kernel-based last layer
- Learning surrogate loss it self is one of the next challenge

References

Main references for this lecture

- Chapter Structured prediction, Book of HalII Daumé, online pdf, 2017.
- Nowozin and Lampert, Structured prediction, Foundations and trends in Machine Learning, 2011.

To go further: References for this lecture

- Belanger and Mc Callum, Structured Energy Networks, ICML 2016
- Crammer, Koby and Singer, Yoram, On the Algorithmic Implementation of Multiclass Kernel-based Vector Machines, J. Mach. Learn. Res., 3/1/2002.
- Tsochantaridis, I. and Joachims, T. and Hofmann, T. and Altun, Y., Large margin methods for structured and interdependent output variables, JMLR, 6,2005
- Collins, Michael, Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms, Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10,2002.
- Ben Taskar, Learning structured prediction models, a large margin approach, PhD thesis (`http://www.seas.upenn.edu/~taskar/pubs/thesis.pdf`), U. Pennsylvany, USA, 2004.
- Blaschko, Lampert, Learning to localize objects with structured output regression, ECCV 2008.
- Deep structured learning, tutorial Raquel Urtasun, U. Toronto
- Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Improving neural networks by preventing co-adaptation of feature detectors.
- Hossain, Sohel, Laga: A comprehensive survey of automatic captioning, arixv
- Chen, Schwing, Yuille, Urtasun, Learning Deep structured models, ICML 2015