Structured Prediction

(with Application in Information Retrieval)

Thomas Hofmann

Google, Switzerland

thofmann@google.com

1 Structured Classification

Supervised Machine Learning

Learn functional dependencies between inputs and outputs from **training data** (inductive inference)

Most basic form: **classification**

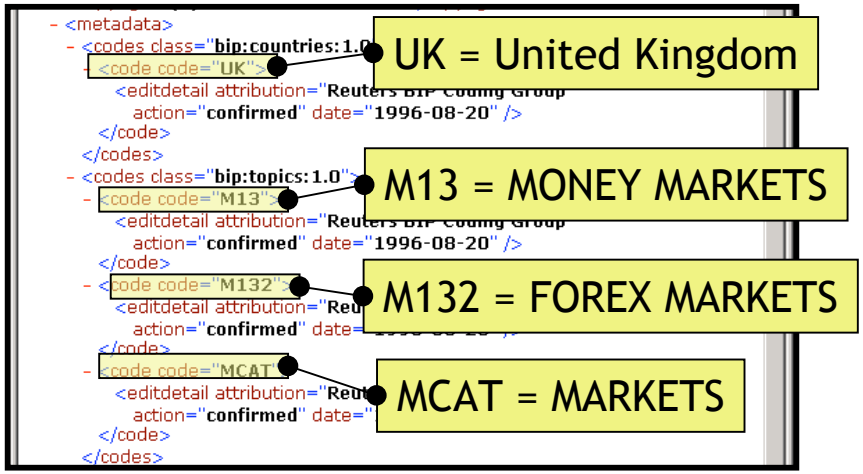
f

optical character recognition

u

f

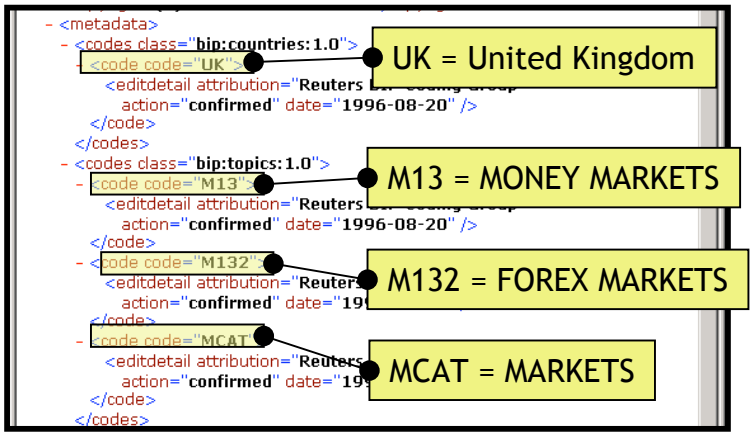
document categorization



Automatic Document Categorization

document categorization

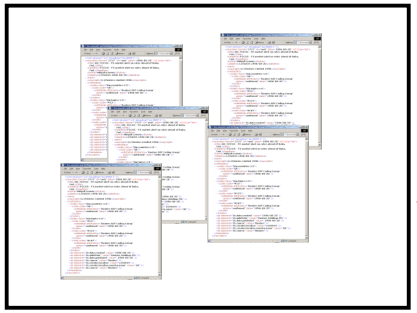
f



training examples

M132 = FOREX MARKETS

expert



learning machine

inductive inference

/* some 'complicated' algorithm */

M132 = FOREX MARKETS

advantages:

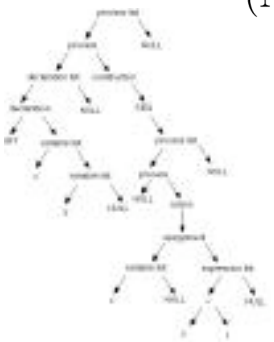
- implicit knowledge (in action)
- no knowledge engineering
- fully automatic & adaptive
- human-level accuracy

Structured Classification

Generalize classification methods to deal with structured outputs and/or with multiple, interdependent outputs

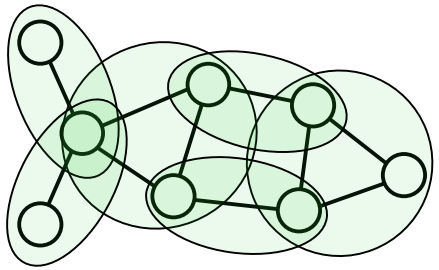
Outputs are either

(N,V,P,N,N,',',ADV,...)



- **Structured objects** such as sequences, strings, trees, etc.

- **Multiple response variables** that are interdependent *(e.g. dependencies modeled by probabilistic graphical models)* = **collective classification**



Part-Of-Speech Tagging

[Example taken from Michael Collins]

INPUT

Profits soared at Boeing Co., easily topping forecasts on Wall Street, as their CEO Alan Mulally announced first quarter results.

x

$\updownarrow$

OUTPUT

Profits/N soared/V at/P Boeing/N Co./N ,/, easily/ADV topping/V forecasts/N on/P Wall/N Street/N ,/, as/P their/POSS CEO/N Alan/N Mulally/N announced/V first/ADJ quarter/N results/N ./.

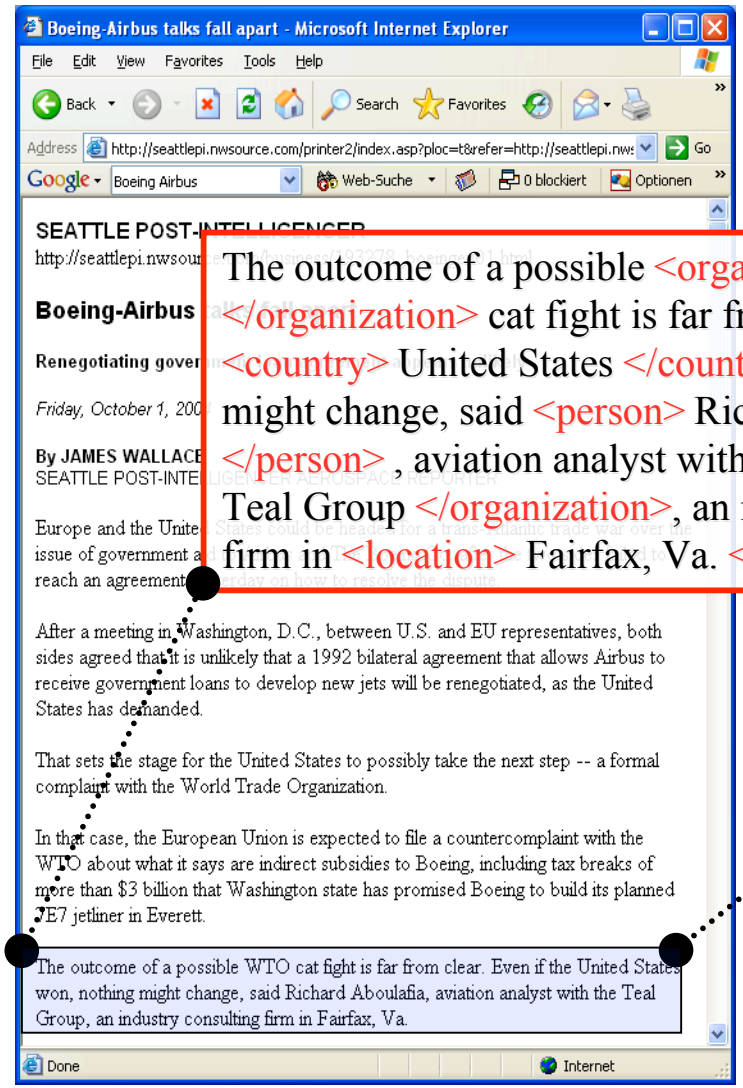
$y = (\mathrm{N},\mathrm{V},\mathrm{P},\mathrm{N},\mathrm{N},\ldots)$

N = noun
V = verb
P = preposition
ADV = adverb
...

Challenge:
Predict a tag sequence

- R. Garside, *The CLAWS Word-tagging System*. In: R. Garside, G. Leech and G. Sampson (eds), The Computational Analysis of English: A Corpus-based Approach. London: Longman, 1987.
- Y.Altun, I. Tsochantaridis, T. Hofmann, *Hidden Markov Support Vector Machines*, ICML, 2003

Information Extraction



The outcome of a possible <organization> WTO </organization> cat fight is far from clear. Even if the <country> United States </country> won, nothing might change, said <person> Richard Aboulafia </person> , aviation analyst with the <organization> Teal Group </organization>, an industry consulting firm in <location> Fairfax, Va. </location>

y

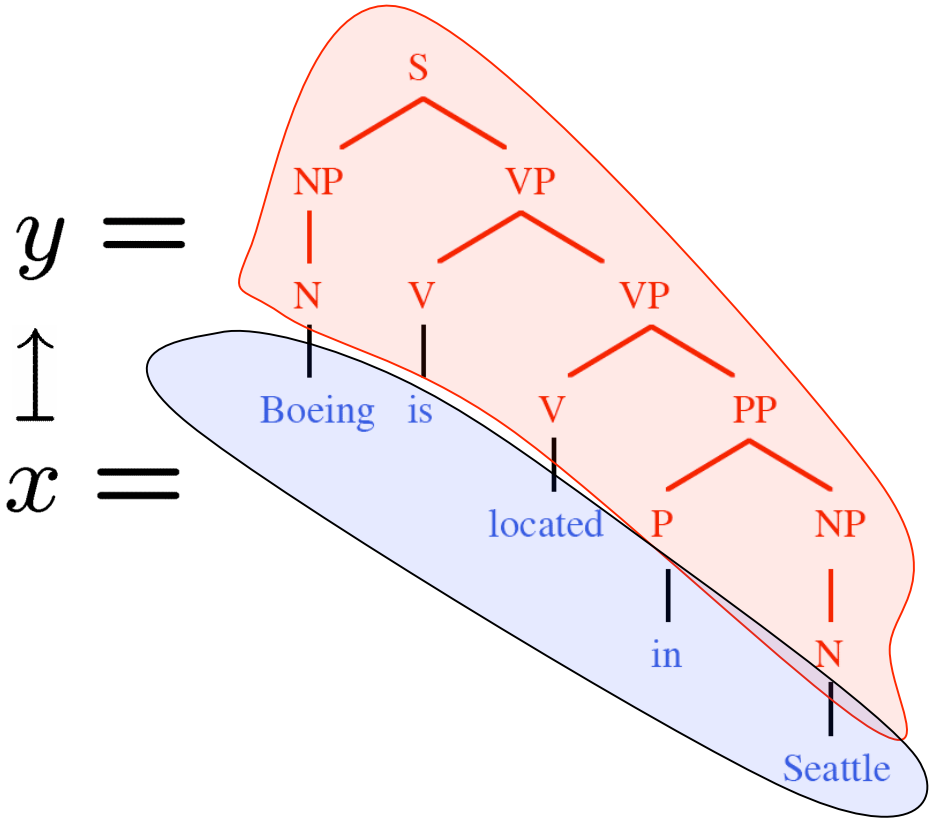
x

Challenge:
Predict a labeled segmentation

- J. Lafferty, F. Pereira, and A. McCallum. *Conditional random fields: Probabilistic models for segmenting and labeling sequence data*. ICML, 2001
- Y.Altun, I. Tsochantaridis, T. Hofmann, *Hidden Markov Support Vector Machines*, ICML, 2003

Natural Language Parsing

Syntactical analysis of natural language,
e.g. based on a context-free grammar

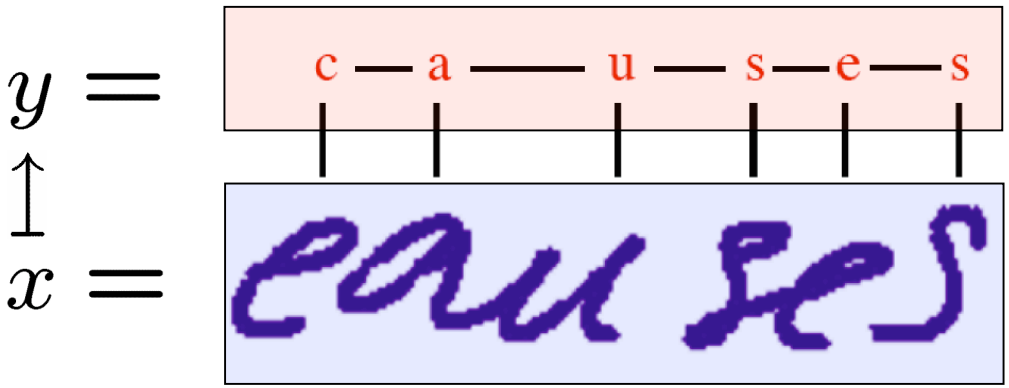


Challenge:
Predict a labeled parse tree

- M Collins, *Discriminative Reranking for Natural Language Parsing*, ICML 2000.
- B. Taskar, D. Klein, M. Collins, D. Koller, and C. Manning, Max-Margin Parsing, EMNLP, 2004.
- I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun, *Support Vector Machine Learning for Interdependent and Structured Output Spaces*, ICML 2004.

Optical Character Recognition

Recognize handwritten words



Training set

Challenge:

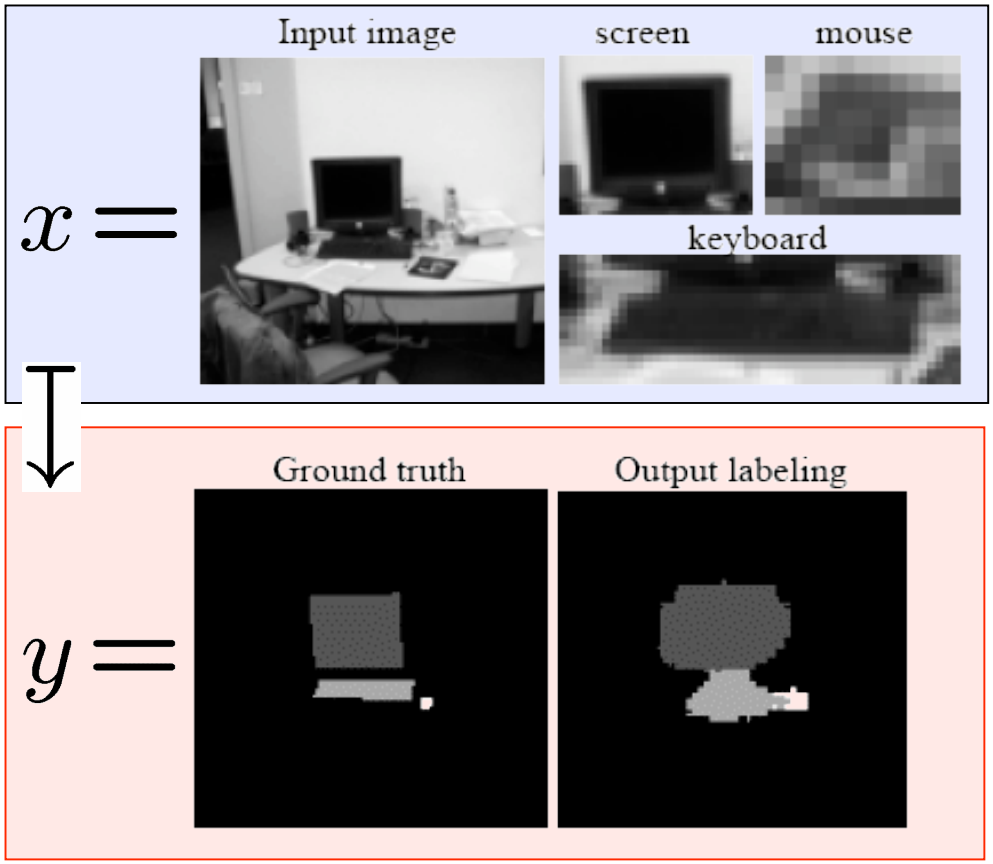
Predict a character sequence

- B. Taskar, C. Guestrin, and D. Koller. *Max margin Markov networks*. NIPS, 2003.

Visual Scene Analysis

Recognize objects in a camera image or video

[Example taken from Torralba et al.]



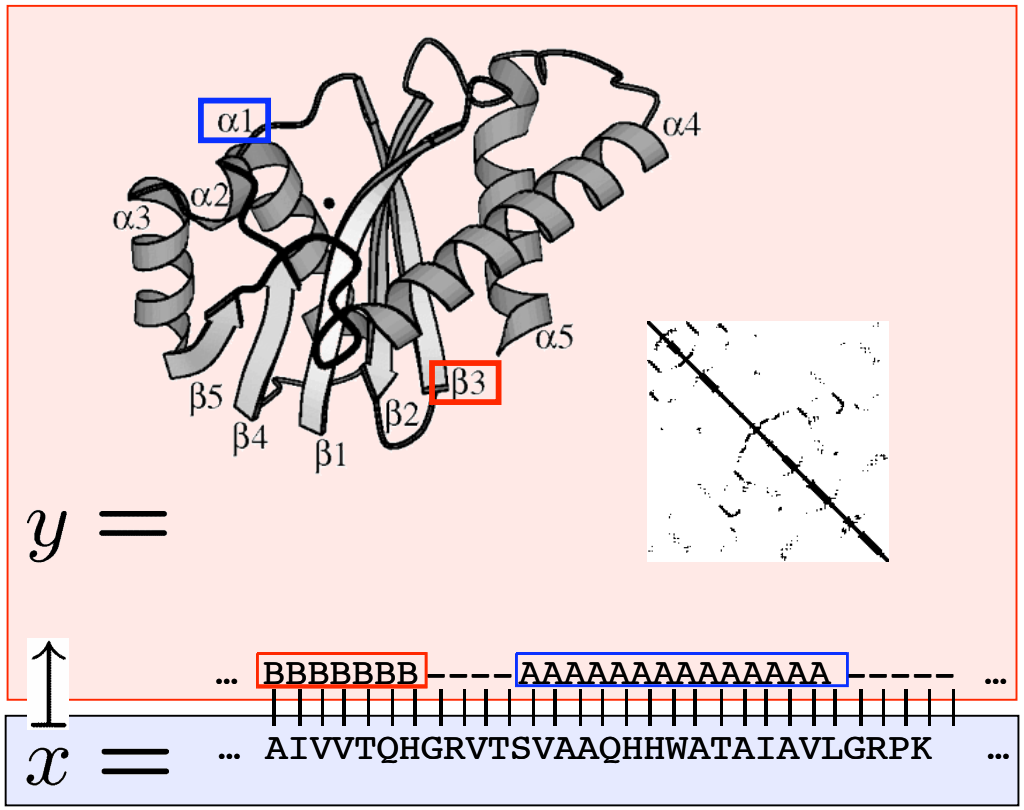
Challenge:
Recognize objects in context

Challenge:
Predict a scene graph

- A. Torralba, K. Murphy and W. Freeman, *Contextual Models for Object Detection using Boosted Random Fields*, NIPS 2004.

Protein Structure Prediction

Predict protein secondary and tertiary structure from its primary structure



Challenge:
Binary contact matrix

Challenge:
Predict labeled segmentation

- Y. liu, E. P. Xing, and J. Carbonell, *Predicting protein folds with structural repeats using a chain graph model*, ICML 2005

2 Multiclass Classification

Linear Multiclass Classification

- Let us see what we can do on the output side...
- First step towards structured outputs: multiclass problems
- Input space $\mathcal{X}$, output space $\mathcal{Y}$
- $\mathcal{Y} = \{y^{(1)}, \ldots, y^{(K)}\}$, K classes or categories
- Introduce weight vectors $\mathbf{w}_y$ for each class $y \in \mathcal{Y}$
- Define discriminant functions $\mathbf{f_y(\cdot) := \langle w_y, \Phi(\cdot)\rangle}$
- For convenience also define a function of input-output pairs $f(\mathbf{x}, y) := f_y(\mathbf{x})$
- Predict class y^* according to

$$y^*(\mathbf{x}) = \arg\max_{y \in \mathcal{Y}} f(\mathbf{x}, y) = \arg\max_{y \in \mathcal{Y}} \langle \mathbf{w}_y, \Phi(\mathbf{x}) \rangle$$

- Note: class with highest score is selected as the prediction

Linear Multiclass Classification - Geometry

- Geometrically: Partition of input space with hyperplanes
- Class y favored over y'

$$\iff \langle \mathbf{w}_y, \Phi(\mathbf{x}) \rangle > \langle \mathbf{w}_{y'}, \Phi(\mathbf{x}) \rangle$$
$$\iff \langle \mathbf{w}_y - \mathbf{w}_{y'}, \Phi(\mathbf{x}) \rangle > 0$$

- Can also introduce additional bias terms: b_y, so that $f(\mathbf{x}, y) = \langle \mathbf{w}_y, \mathbf{x} \rangle + b_y$ (not done for simplicity)

Multiclass Perceptron

- Simplest learning algorithm: perceptron algorithm
- In each step with training pattern $(\mathbf{x}_i, y_i)$
  - predict $y_i^* := y^*(\mathbf{x}_i)$ using current set of weights
  - check if $y_i^* = y_i$ (do nothing if prediction is correct)
  - if prediction is incorrect, then update:

$$\mathbf{w}_{y^*} = \mathbf{w}_{y^*} - \Phi(\mathbf{x}_i), \qquad \mathbf{w}_y = \mathbf{w}_y + \Phi(\mathbf{x}_i),$$

- Move weight vector of correct class *towards* pattern, move weight vector of incorrect class *away from* pattern.
- Like in standard perceptron, cycle through data until converged (assuming linear multi-class separability)

- Michael Collins, Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms, EMNLP 2002

Multiclass SVMs: Separation Margin

- There are a number of ways to generalize support vector machines to the multiclass setting
- One simple way is to define the notion of a **separation margin** as follows:

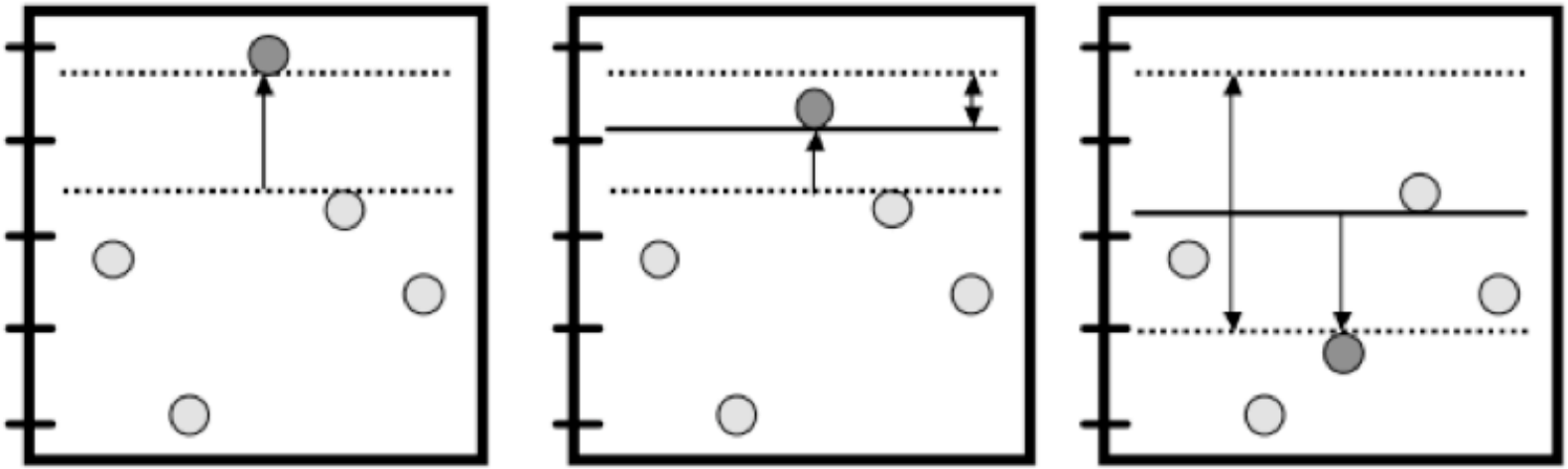
$$\gamma_i := \langle \mathbf{w}_{y_i}, \Phi(\mathbf{x}_i) \rangle - \langle \mathbf{w}_{\hat{y}}, \Phi(\mathbf{x}_i) \rangle$$

where

$$\hat{y} := \arg\max_{y \neq y_i} \langle \mathbf{w}_y, \Phi(\mathbf{x}_i) \rangle$$

- This means: one compares the score of the correct class with the score of the highest ranked incorrect class.
- $\gamma_i > 0$ guarantees that the i-th training example is classified correctly

Multiclass SVM: Separation Margin, Illustrated



- y-value = score, circles = labels, filled = correct
- Case 1: correct separation margin obtained
- Case 2: correct classification, but insufficient margin
- Case 3: incorrect classification

Multiclass SVMs: Formulation

- Now one can generalize binary SVM classification to the multiclass setting as follows [CS01]:
- Define concatenation of per class weight vectors

$$\mathbf{w} := (\mathbf{w}_1', \ldots, \mathbf{w}_K')'$$

- Multiclass (soft margin) SVM formulation

$$\min_{\mathbf{w},\xi} \left\{ \|\mathbf{w}\|^2 + \sum_{i=1}^{n} \xi_i \right\}, \quad \text{s.t.} \quad \gamma_i \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad (\forall i)$$

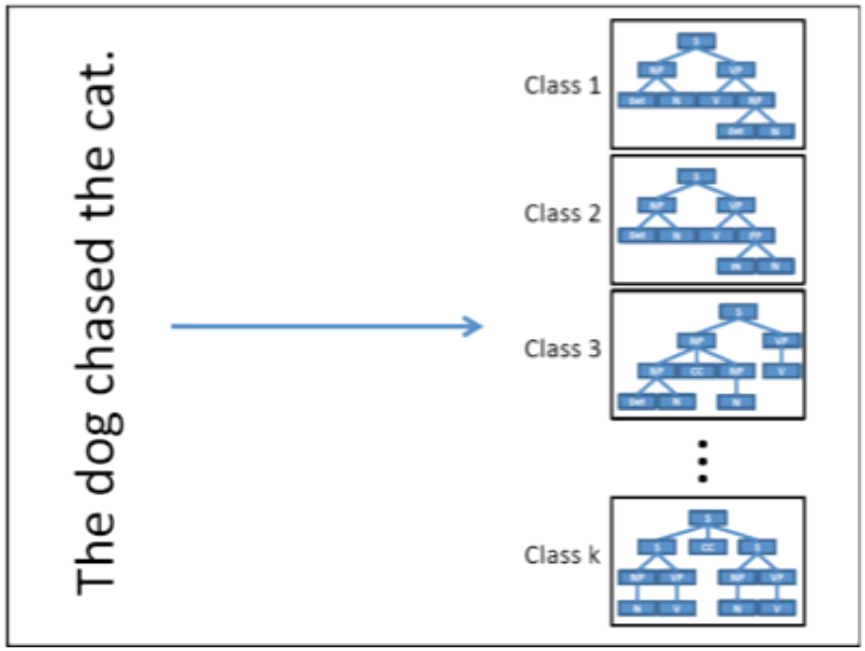
- Note that one can explicitly expand the margin constraints

$$\gamma_i \geq 1 - \xi_i \Leftrightarrow \langle \mathbf{w}_{y_i}, \Phi(\mathbf{x}_i) \rangle - \langle \mathbf{w}_y, \Phi(\mathbf{x}_i) \rangle \geq 1 - \xi_i \quad (\forall y \neq y_i)$$

- Comment: the number of linear inequality constraints equals $n \cdot |\mathcal{Y}|$

- Koby Crammer, Yoram Singer, On the Algorithmic Implementation of Multiclass Kernel-based Vector Machines, Journal of Machine Learning Research, 2001.

From Multiclass to Structured Prediction



- One can naively think of each structured output as a different class of its own
- Can we then apply multiclass methods?
- Challenges
  1. Compact representation
  2. Efficient search
  3. Refined error metric
  4. Sub-linear training time (in number of outputs)

- Thorsten Joachims, Thomas Hofmann, Yisong Yue, Chun-Nam Yu, Predicting Structured Objects with Support Vector Machines, Communications of the ACM, 2009 (to appear).

From Multiclass to Structured Prediction

1. Compact Representation: Define joint **input-output feature functions** and/or kernels that ensure generalization capabilities across inputs and outputs.

2. Efficient search: Develop problem or representation specific **search algorithms**, e.g. based on dynamic programming, min flow, etc., to find optimal output for given input

3. Refined error metric: Integrate application-specific **loss function** into problem formulation

4. Training algorithm: Use efficient output search in conjunction with (sparse) cutting plane **optimization method**

3 Function Classes for Structured Classification

Compatibility Functions

Linear compatibilty functions in **joint feature representation** of input/output pairs

$$\Phi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^m$$

$$f(x, y; w) = \langle w, \Phi(x, y) \rangle$$

encodes combined properties of inputs and outputs:
which aspect of the input is relevant for which part of the output?

Prediction

$$F(x) = \operatorname*{argmax}_y f(x, y; w)$$

for given input predict most compatible output (may involve search)

Binary classification

$$\Phi(x, y) = y\Phi(x),$$

$$F(x) = \operatorname*{argmax}_{y \in \{-1,1\}} y\langle w, \Phi(x) \rangle = \operatorname{sign}(\langle w, \Phi(x) \rangle)$$

Kernels for Discrete Inputs

Structured inputs are usually handled with appropriate feature extraction methods, i.e. for input space X, find a **feature map**:

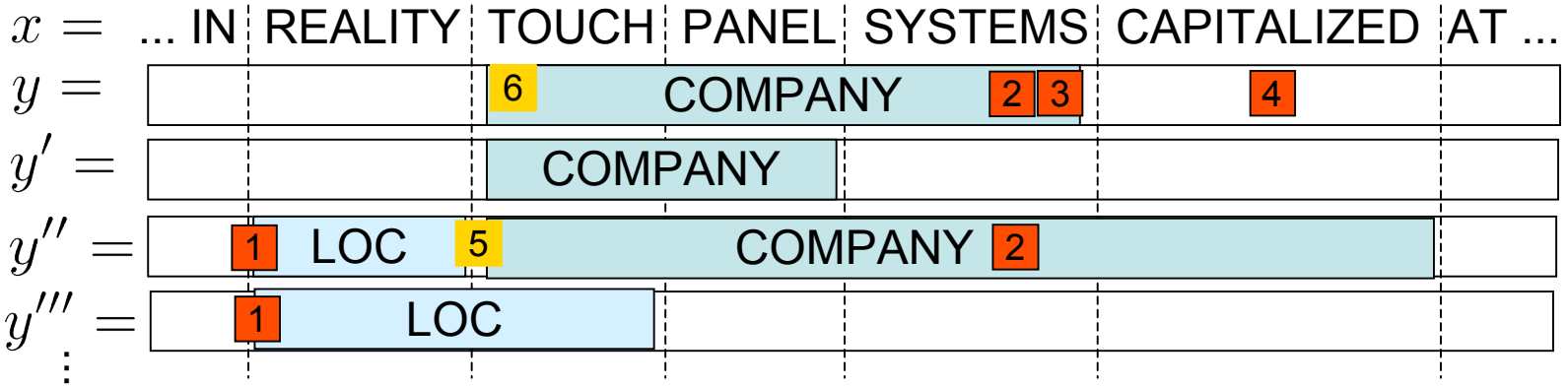
$$\Phi : \mathcal{X} \rightarrow \mathbb{R}^d$$

With the use of **kernels**, the feature map can be implicit and is not restricted to finite d.

Examples:

- String kernels [LSST+02], [LK04], [VS03]: number of common subsequences (with gaps)
- Tree kernels and convolutional kernels [CD01]
- Graph kernels [GLF02], [Gär03]
- Set kernels [KJ03]
- Research topics: modeling power, applications & experimental investigations, computational improvements

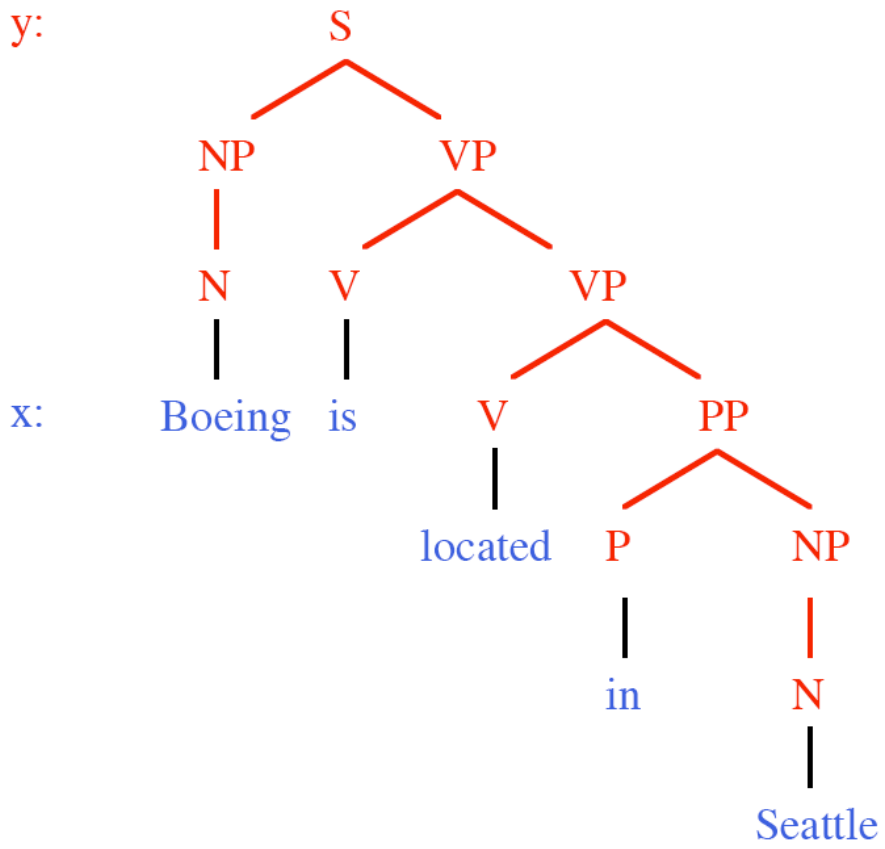
Joint Feature Map: Information Extraction



$$\Phi(x,y) = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ \dots \end{pmatrix} \quad \Phi(x,y') = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \dots \end{pmatrix} \quad \Phi(x,y'') = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ \dots \end{pmatrix} \quad \Phi(x,y''') = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \dots \end{pmatrix}$$

- Y.Altun, I. Tsochantaridis, T. Hofmann, *Hidden Markov Support Vector Machines*, ICML, 2003
- S. Sarawagi & W. W. Cohen, *Semi-Markov Conditional Random Fields for Information Extraction*, NIPS 2004

Joint Feature Maps: Parsing



$$\Phi(x,y) = \begin{pmatrix} 1 \\ 0 \\ 2 \\ 1 \\ 1 \\ 1 \\ 1 \\ \dots \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \begin{matrix} \text{S} \rightarrow \text{NP VP} \\ \text{S} \rightarrow \text{VP} \\ \text{NP} \rightarrow \text{N} \\ \text{VP} \rightarrow \text{V VP} \\ \text{VP} \rightarrow \text{V PP} \\ \text{PP} \rightarrow \text{P NP} \\ \dots \\ \text{N} \rightarrow \text{Boeing} \\ \text{N} \rightarrow \text{Airbus} \\ \text{V} \rightarrow \text{is} \\ \text{V} \rightarrow \text{located} \\ \text{P} \rightarrow \text{in} \\ \text{N} \rightarrow \text{Seattle} \end{matrix}$$

- I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun, *Support Vector Machine Learning for Interdependent and Structured Output Spaces*, ICML 2004.
- B. Taskar, D. Klein, M. Collins, D. Koller, and C. Manning, *Max-Margin Parsing*, EMNLP, 2004.

Tensor Product Feature Maps

- Simple way to define joint feature maps: via tensor products

$$\Phi(\mathbf{x}, y) := \Phi^{\mathcal{X}}(\mathbf{x}) \otimes \Phi^{\mathcal{Y}}(y)$$
$$\phi_{ij}(\mathbf{x}, y) = \phi_i^{\mathcal{X}}(\mathbf{x}) \phi_j^{\mathcal{Y}}(y)$$

  - design feature maps for inputs and outputs (independently)
  - multiplicative combination of every input and every output feature

- $f(\mathbf{x}, y; \mathbf{w}) = \langle \mathbf{w}, \Phi(\mathbf{x}, y) = \sum_j \phi_j^{\mathcal{Y}}(y) \langle \mathbf{w}_j, \Phi^{\mathcal{X}}(\mathbf{x}) \rangle$
- **Inner products** can be computed efficiently

$$\langle \Phi(\mathbf{x}, y), \Phi(\mathbf{x}', y') \rangle = \langle \Phi^{\mathcal{X}}(\mathbf{x}), \Phi^{\mathcal{X}}(\mathbf{x}') \rangle \cdot \langle \Phi^{\mathcal{Y}}(y), \Phi^{\mathcal{Y}}(y') \rangle$$

  - no need to explicitly perform tensor product expansion (if learning algorithm can work with inner products/kernels)

• I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun, *Support Vector Machine Learning for Interdependent and Structured Output Spaces*, ICML 2004.

Sequence Feature Maps

- Joint maps for sequences $(\mathbf{x}, y) = ((x_1, \ldots, x_T), (y_1, \ldots, y_T))$

$$\Phi(\mathbf{x}, y) = \sum_{t=1}^{T} \Phi^t(\mathbf{x}, y)$$

  - extract features locally at position t
  - sum up contributions
  - e.g. counts over local indicator functions (binary features)
- Then:

$$\langle \mathbf{w}, \Phi(\mathbf{x}, y) \rangle = \sum_{t=1}^{T} \langle \mathbf{w}, \Phi^t(\mathbf{x}, y) \rangle$$

Reproducing Kernel Hilbert Spaces

RKHS

$$\mathcal{H}_k \equiv \left\{ f : f(\cdot) = \sum_{z \in \mathcal{S}} \alpha_z k(\cdot, z),\ \mathcal{S} \subseteq \mathcal{X} \times \mathcal{Y},\ \|f\| < \infty \right\}$$

compatibility functions over inputs and outputs

expansion coefficients

generating **kernel**

$$k : (\mathcal{X} \times \mathcal{Y})^2 \to \mathbb{R}$$

inner product/norm

$$f, g \in \mathcal{H}_k,\ f(\cdot) = \sum_{z \in \mathcal{S}} \alpha_z k(\cdot, z),\ g(\cdot) = \sum_{z' \in \mathcal{S}'} \beta_{z'} k(\cdot, z'),$$

$$\langle f, g \rangle \equiv \sum_{z \in \mathcal{S}} \sum_{z' \in \mathcal{S}'} \alpha_z \beta_{z'} k(z, z'), \quad \|f\|^2 \equiv \langle f, f \rangle$$

RKHS and Finite-dimensional Vector Spaces

Special case: kernels defined via inner products

$$k((x,y),(x',y')) = \langle \Phi(x,y), \Phi(x',y') \rangle, \quad \Phi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$$

$$\Rightarrow \quad \mathcal{H}_k = \{ f(\cdot) : \langle w, \Phi(\cdot) \rangle \}$$

linear functions in some finite dimensional feature representation as special case

Tensor product feature maps

$$\begin{aligned} k((x,y),(x',y')) &= k_{\mathcal{X}}(x,x') \cdot k_{\mathcal{Y}}(y,y') \\ &\underset{\substack{\text{finite-dim} \\ \text{case}}}{=} \langle \Phi_{\mathcal{X}}(x) \otimes \Phi_{\mathcal{Y}}(y), \Phi_{\mathcal{X}}(x') \otimes \Phi_{\mathcal{Y}}(y') \rangle \end{aligned}$$

Kernels serve two purposes: flexible and efficient feature extraction, efficient combination of input/output features

4 Statistical Inference for Structured Prediction

Regularization & Representer Theorem

Combine risk function with appropriate **regularization functional** (Hilbert space norm)

$$\hat{f}(\mathcal{S}) = \operatorname*{argmin}_{f \in \mathcal{H}} C(f; \mathcal{S}) + \Omega(\|f\|_{\mathcal{H}})$$

training sample

stabilizer (penalize norm)

Representer Theorem

Denote by $\mathcal{H}$ an RKHS on $\mathcal{X} \times \mathcal{Y}$ with kernel k and let $\mathcal{S} = ((x_i, y_i))_{i \in [n]}$. Furthermore let $C(f; \mathcal{S})$ be a functional depending on f only via its values on the augmented sample $\mathcal{S}' \equiv \{(x_i, y) : (x_i, y_i) \in \mathcal{S}\}$. Let Ω be a strictly monotonically increasing function. Then the solution of the optimization problem $\hat{f}(\mathcal{S}) \equiv \operatorname{argmin}_{f \in \mathcal{H}} C(f; \mathcal{S}') + \Omega(\|f\|_{\mathcal{H}})$ can be written as:

$$\hat{f}(\cdot) = \sum_{i=1}^{n} \sum_{y \in \mathcal{Y}} \beta_{iy} k(\cdot, (x_i, y))$$

- B. Schölkopf, R. Herbrich, A. Smola, and R. C. Wiliamson, *A Generalized Representer Theorem*, In Proceedings of the Annual Conference on Computational Learning Theory, pages 416-426, 2001.
- T. Hofmann, B. Schölkopf, and A. Smola, *A Tutorial Review of RKHS Methods in Machine Learning*, *Annals of Statistics, 2008*

Exponential Family Conditional Models

In order to derive learning/inference algorithms, a **probabilistic interpretation of compatibility functions** is fruitful.

Define family of **conditional distributions**

$$p(y|x;f) = \exp\left[f(x,y) - g_f(x)\right]$$

higher compatibility, higher conditional probability

log partition function

$$g_f(x) := \log \sum_{y \in \mathcal{Y}} \exp\left[f(x,y)\right]$$

$$\begin{aligned} p(y|x;f) &= \exp\left[\langle f, k(\cdot,(x,y))\rangle - g_f(x)\right] \\ &= \exp\left[\langle w, \Phi(x,y)\rangle - g_f(x)\right] \end{aligned}$$

canonical parameters

sufficient statistics (given x)

- B. O. Koopman. On distributions admitting a sufficient statistic. Trans. of the American Math. Society, 39, 1936.
- M. K. Murray and J.W. Rice. Differential geometry and statistics. Chapman and Hall, 1993.

Risk Functions: Log-Loss

(Negative) **Conditional log-likelihood**

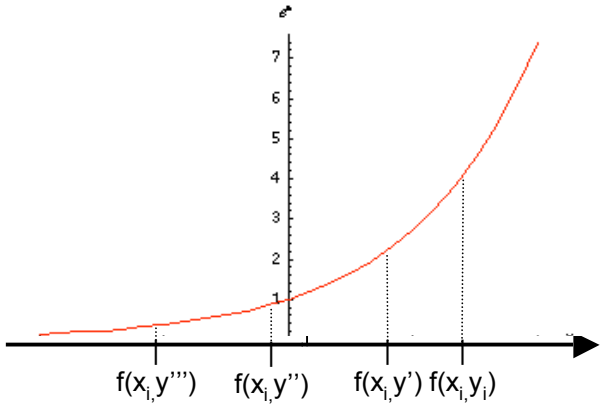
$$g_f(x_i) := \log \sum_{y \in \mathcal{Y}} \exp\left[f(x_i, y)\right]$$

$$C^{cl}(f; \mathcal{S}) = -\sum_{i=1}^{n} \log p(y_i | x_i, f) = -\sum_{i=1}^{n} \left[f(x_i, y_i) - g_f(x_i)\right]$$

maximize compatibility on training pairs

minimize log-partition function

Conditional Random Field
(CRF)



- J. Lafferty, F. Pereira, and A. McCallum. *Conditional random fields: Probabilistic models for segmenting and labeling sequence data*. ICML, 2001
- S. Kakade, Y.W. Teh, and S. Roweis. *An alternative objective function for Markovian fields*. ICML, 2002.
- Y. Altun. M. Johnson, and T. Hofmann, *Investigating Loss Functions and Optimization Methods for Discriminative Learning of Label Sequences*, EMNLP, 2003.

Learning CRFs

Parametric case

$$C^{cl}(w;\mathcal{S}) = -\sum_{i=1}^{n}\left[\langle w, \Phi(x_i, y_i)\rangle - \log \sum_{y} \exp\langle w, \Phi(x_i, y)\rangle\right]$$

Gradient-based Methods

$$\nabla_w C^{cl}(w;\mathcal{S}) = \sum_{i=1}^{n} \mathbf{E}\left[\Phi(x_i, Y)\right] - \sum_{i=1}^{n} \Phi(x_i, y_i)$$

expectation over outputs under current conditional model

exact inference (e.g. forward-backward, junction tree) or approximate inference

Numerical optimization

- Iterative scaling (not recommended)
- Preconditioned conjugate gradient
- Limited Memory Quasi-Newton (BFGS)
- Bayesian CRFs (avoid overfitting)

- J. Lafferty, F. Pereira, and A. McCallum. *Conditional random fields: Probabilistic models for segmenting and labeling sequence data*. ICML, 2001
- H. M. Wallach. Efficient training of conditional random fields. Master's thesis, University of Edinburgh, 2002.
- Y. Qi, M. Szummer, T. Minka, *Bayesian Conditional Random Fields*, AISTATS, 2005.

Regularization & Representer Theorem

Representer Theorem (again)

$$\hat{f}(\cdot) = \sum_{i=1}^{n} \sum_{y \in \mathcal{Y}} \beta_{iy} k(\cdot, (x_i, y))$$

sum can be huge, if output space is combinatorial
- **sparseness is crucial**

Risk Functions: Hinge Risk

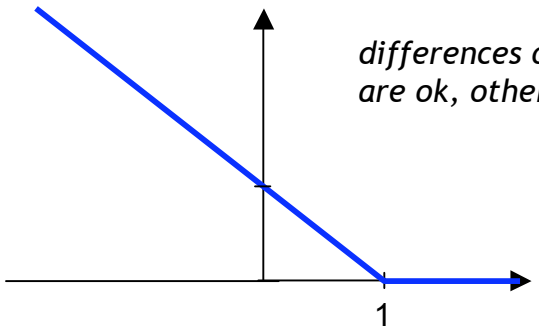
Minimum Log-Odds

$$r(x,y;f) := \log \frac{p(y|x;f)}{\max_{y' \neq y} p(y'|x;f)} = \underbrace{f(x,y) - \max_{y' \neq y} f(x,y')}$$

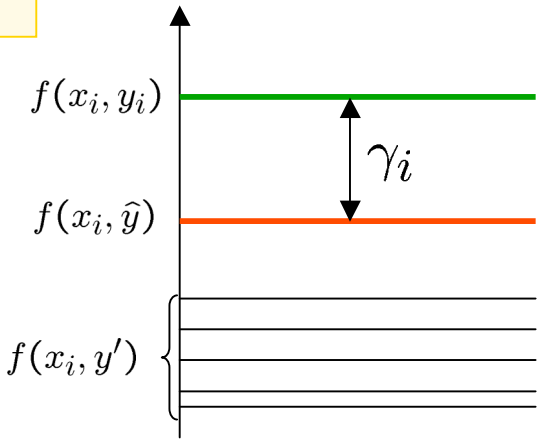
difference between compatibility of correct vs. best incorrect output

Hinge Loss

$$C^{mm}(f;\mathcal{S}) = \sum_{i=1}^{n} \max\{0, 1 - r(x_i,y_i;f)\}$$



differences of 1 and greater are ok, otherwise linear penalty



SVMstruct: Problem Formulation

Minimize Hinge loss = structured Support Vector Machines

Primal QP

$$\min_{w,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$$

$$\text{s.t. } \langle w, \Phi(x_i, y_i) - \Phi(x_i, y)\rangle \geq 1 - \xi_i, \forall i, \ \forall y \neq y_i$$

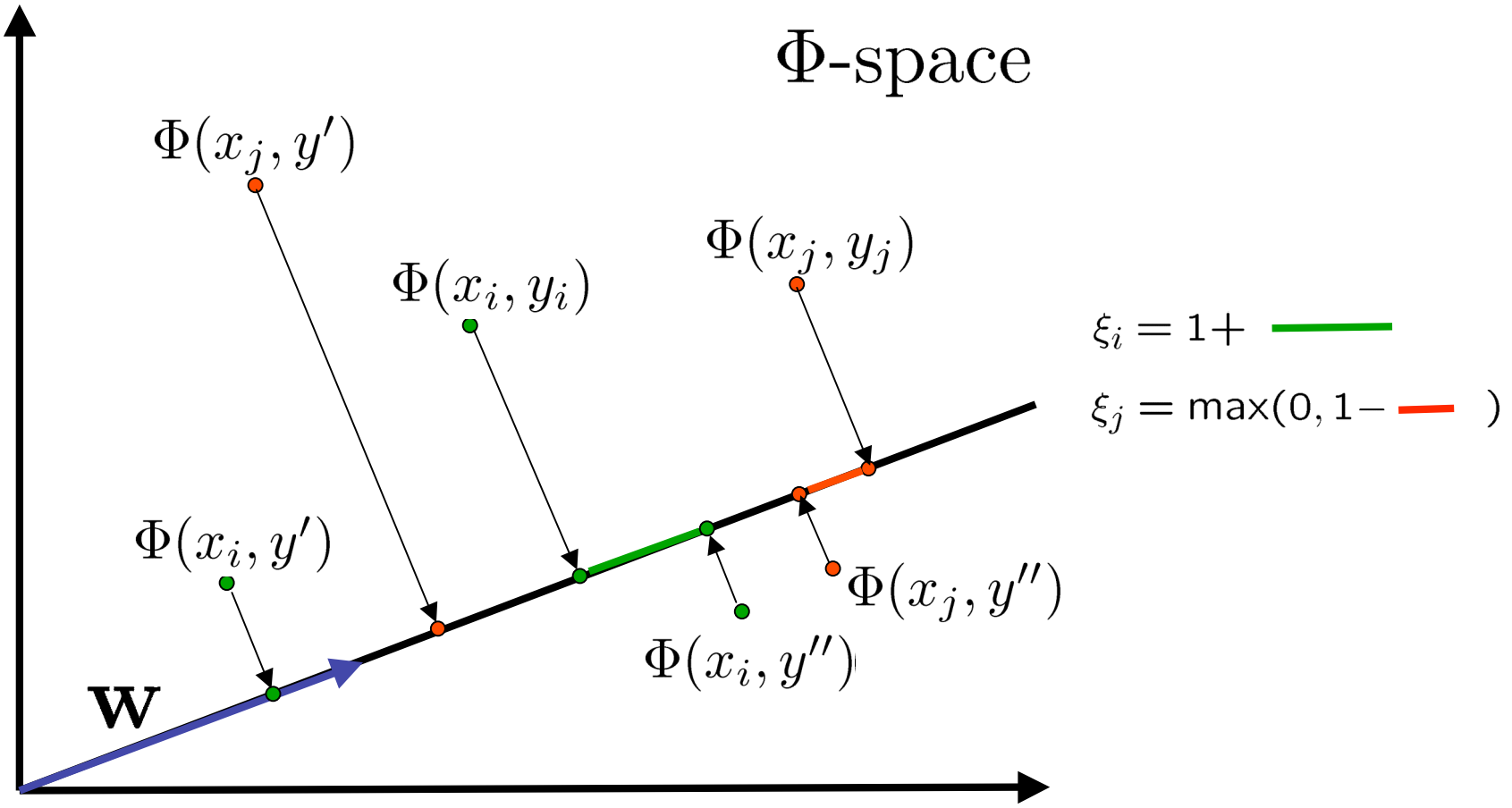
$$\xi_i \geq 0, \quad \forall i$$

$$f(\cdot) = \langle w, \Phi(\cdot)\rangle$$

number of constraints = cardinality of output space

- I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun, *Support Vector Machine Learning for Interdependent and Structured Output Spaces*, ICML 2004.

Soft Margin: Illustration



SVMstruct: Dual Formulation

Dual problem formulation

Dual QP

$$\max_{\alpha} \; -\frac{1}{2} \sum_{i,j=1}^{n} \sum_{\substack{y \neq y_i \\ y' \neq y_j}} \alpha_{iy} \alpha_{jy'} k'((x_i, y), (x_j, y')) + \sum_{i=1}^{n} \sum_{y \neq y_i} \alpha_{iy}$$

$$\text{s.t. } \alpha_{iy} \geq 0, \quad \sum_{y \neq y_i} \alpha_{iy} \leq C$$

$$f(\cdot) = \sum_{i,y} \beta_{iy} k(\cdot, (x_i, y))$$

Block structure:
Independent constraints
for every training example

$$\beta_{iy} = \begin{cases} -\alpha_{iy} & \text{if } y \neq y_i \\ \sum_{y \neq y_i} \alpha_{iy} & \text{otherwise} \end{cases}$$

SVMstruct: Algorithm

1: initialize $S = \emptyset$, $\mathbf{w} = 0$, $\xi = 0$
2: **repeat**
3: **for** all training examples (x_i, y_i) **do**
4: compute $\hat{y} = \text{argmax}_{y \neq y_i} \langle \mathbf{w}, \Phi(x_i, y) \rangle$
5: **if** $1 - \langle \mathbf{w}, \delta\Phi_i(\hat{y}) \rangle > \xi_i + \epsilon$ **then**
6: $S_i \leftarrow S_i \cup \{\hat{y}\}, \quad S \leftarrow S \cup \{i\hat{y}\}$
7: optimize w.r.t. $(\mathbf{w}, \xi)$ over S (or S_i)
8: **end if**
9: **end for**
10: **until** S has not changed

Constraint generation

Adding constraint

Re-optimization

$$\delta\Phi_i(\hat{y}) \equiv \Phi(x_i, y_i) - \Phi(x_i, y)$$

SVMstruct: Variable Selection

- **Incrementally add constraints** (sequential strengthening of QP)
- Corresponds to variable selection in the dual QP
- Sequentially **optimize dual over selected variables**

- Select maximally violating constraints (or at least ϵ-violating constraints) – requires black box search to find best or second best outputs

Theorem:

After selecting at most $t \leq \frac{2nCR}{\epsilon^2}$ ϵ-violating constraints no remaining constraint is violated by more than ϵ.

- I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, *Large Margin Methods for Structured and Interdependent Output Variables*, JMLR 2005.
- http://www.cs.cornell.edu/People/tj/svm_light/svm_struct.html

SVMstruct: Proof of theorem (sketch)

- Dual objective is upper bounded by primal, which is upper bounded by nC (zero weight vector).
- We show that each variable selections for an ε-violated constraint will increase the dual objective by some minimum (bounded away from zero).
- This limits the number of sequentially ε-violated constraints.

Lemma 10 *Let J be a symmetric, positive semi-definite matrix, and define a concave objective in α*

$$\Theta(\alpha) = -\frac{1}{2}\alpha' J\alpha + \langle \mathbf{h}, \alpha \rangle ,$$

which we assume to be bounded from above. Assume that a solution α^o and an optimization direction η is given such that $\langle \nabla\Theta(\alpha^o), \eta \rangle > 0$. Then optimizing Θ starting from α^o along the chosen direction η will increase the objective by

$$\max_{\beta>0} \{\Theta(\alpha^o + \beta\eta)\} - \Theta(\alpha^o) = \frac{1}{2}\frac{\langle \nabla\Theta(\alpha^o), \eta \rangle^2}{\eta' J \eta} > 0 .$$

Proposition 15 ($\mathrm{SVM}_2^{\triangle m}$) *For $\mathrm{SVM}_2^{\triangle m}$ step 10 in Algorithm 1 the improvement $\delta\Theta$ of the dual objective is lower bounded by*

$$\delta\Theta \geq \frac{1}{2}\frac{\varepsilon^2}{R_i^2 + \frac{n}{C}} , \quad \text{where} \quad R_i = \max_{\mathbf{y}} \|\delta\Psi_i(\mathbf{y})\| .$$

SVMstruct: Support Outputs

- SVM: sparseness = select some examples
- SVMstruct: sparseness = select some outputs for some examples

- Example: named entity tagging (Spanish corpus)

```
PP ESTUDIA YA PROYECTO LEY TV REGIONAL REMITIDO POR LA JUNTA MERIDA ( EFE ) .
O  N       N  N        M   m  m        N        N   N  O     L      N O   N N
```

```
ONNNMmmNNNOLNONN
--------M-------
------N---------
----------P-----
----N-----------
N---P-----------
-----------o----
```

Support Vector Sequences
(only difference to correct sequence shown)

- Y.Altun, I. Tsochantaridis, T. Hofmann, *Hidden Markov Support Vector Machines*, ICML, 2003

5 Loss Functions

Problem-Specific Loss Functions

Problem-specific prediction loss functions

$$\triangle : Y \times Y \rightarrow \mathbb{R},$$
$$\triangle(y, y') = \text{"loss for predicting } y' \text{ while truth is } y\text{"}$$

This is important to model quality of predictions in domain/application-dependent way.

Example: Hamming loss

$$y = (0, 1, 1, 0, 1, 1, 0, 0, 1, 0)$$
$$y' = (0, 1, 1, 1, 1, 0, 0, 0, 1, 1)$$
$$|y - y'| = (0, 0, 0, 1, 0, 1, 0, 0, 0, 1) \Rightarrow \triangle(y, y') = 3$$

Margin Re-scaling

We are requiring different margins for different outputs

$$\langle \mathbf{w}, \Phi(\mathbf{x}_i, y_i) - \Phi(\mathbf{x}_i, y) \rangle \geq \triangle(y_i, y) - \xi_i$$

This results in a modified Hinge loss

$$C_{\triangle}^{sm}(f; \mathcal{S}) = \sum_{i=1}^{n} \max\{0, \max_{y \neq y_i} [\triangle(y_i, y) - (f(x_i, y_i) - f(x_i, y))]\}$$

- B. Taskar, C. Guestrin and D. Koller, Maximum Margin Markov Networks, NIPS 2003.

SVMstruct: Dual for Margin Re-scaling

Dual problem formulation

Dual QP

$$\max_{\alpha} \; -\frac{1}{2} \sum_{i,j=1}^{n} \sum_{\substack{y \neq y_i \\ y' \neq y_j}} \alpha_{iy} \alpha_{jy'} k'((x_i, y), (x_j, y')) + \sum_{i=1}^{n} \sum_{y \neq y_i} \alpha_{iy}$$

$$\text{s.t. } \alpha_{iy} \geq 0, \quad \sum_{y \neq y_i} \frac{\alpha_{iy}}{\triangle(y_i, y)} \leq C$$

Minor modification in bound constraints

Black Box: Constraint Selection

Without margin re-scaling, we need a blackbox to solve:

$$\hat{y}_i = \arg\max_{y \neq y_i} f(\mathbf{x}_i, y)$$

With margin re-scaling, this becomes:

$$\hat{y}_i = \arg\max_{y \neq y_i} [\triangle(y_i, y) - f(\mathbf{x}_i, y_i) + f(\mathbf{x}_i, y)]$$

Requirements:

- Search (argmax) needs to be feasible
- Second best output may be required
- Loss function needs to decompose in similar manner

Many tractable use cases: dynamic programming, min-cut, etc.

6 Compositionality and Graphical Models

Compositionality

Additive decomposition of compatibility function over "parts"

$$f(x,y) = \underbrace{\sum_{c}}_{\text{additivity over parts}} \underbrace{f_c(x_c, y_c)}_{\text{dependency on subset of variables}}$$

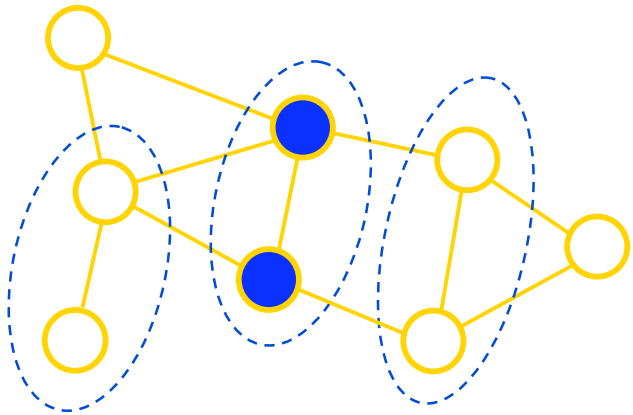
Induces decompositionality of kernel functions (*proof elementary*)

$$k(z,z') = \sum_{c,d} k_{cd}(z_c, z'_d), \quad z = (x,y),\ z' = (x',y')$$

- T. Hofmann, B. Schölkopf, and A. Smola, *A Tutorial Review of RKHS Methods in Machine Learning*, *Annals of Statistics*, *2008*

Markov Networks

Markov Networks



conditional independence between variables corresponds to **graph separation**

$$P(x) = \frac{1}{Z} \prod_c \Psi_c(x_c)$$

product over **clique** potential functions (Hamersley-Clifford)

$$= \exp\left[\sum_c f_c(x_c) - \log Z\right]$$

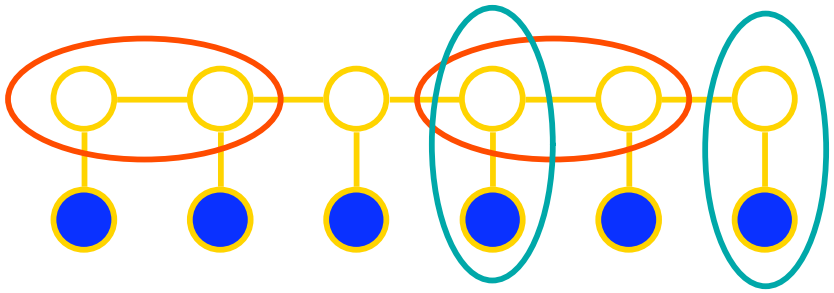
Conditional independence graph

A **conditional independence graph** (or Markov network) is an undirected graph $\mathcal{G} = (X, E)$ such that for any pair of variables $(X_i, X_j) \notin E$ if and only if $X_i \perp\!\!\!\perp X_j | X - \{X_i, X_j\}$.

- distributions with full support pairwise = local = global Markov property

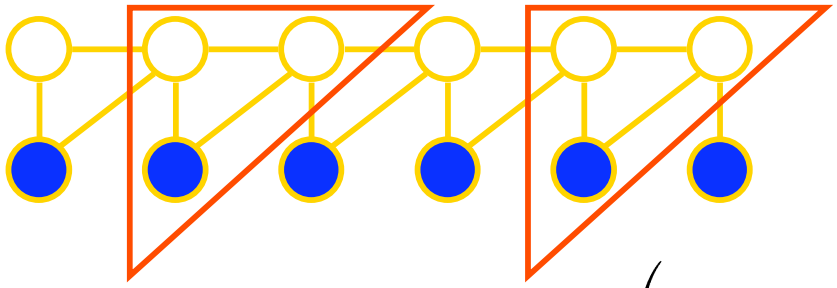
Markov Network Kernels: Example

Label sequence learning with HMM structure



$$k_{cd}\left(\begin{array}{c}\text{y}\\\text{y'}\end{array},\begin{array}{c}\text{u}\\\text{u'}\end{array}\right)=\begin{cases}1, & \text{if } y=u \text{ and } y'=u'\\ 0, & \text{otherwise}\end{cases}$$

Overlapping input features



$$k_{\bar{c}\bar{d}}\left(\begin{array}{c}\text{y}\\\text{x}\end{array},\begin{array}{c}\text{u}\\\text{x'}\end{array}\right)=\begin{cases}k(x,x'), & \text{if } y=u\\ 0, & \text{otherwise}\end{cases}$$

$$k_{\tilde{c}\tilde{d}}\left(\begin{array}{c}\text{y}\quad\text{y'}\\\text{x}\end{array},\begin{array}{c}\text{u}\quad\text{u'}\\\text{x'}\end{array}\right)=\begin{cases}k(x,x'), & \text{if } y=u \text{ and } y'=u'\\ 0, & \text{otherwise}\end{cases}$$

Representer Theorem over Cliques

- Standard representer theorem yields expansion with $|\mathcal{S}| \cdot |\mathcal{Y}|$ terms

problematic

- SVMstruct: sparseness due to Hinge loss properties
- Logistic loss: no sparseness, too many parameters

Clique-based Sparse Approximation

For decomposable RKHS:

$$f(z) = \sum_{i=1}^{n} \sum_{c} \sum_{y_c} \beta^i_{cy_c} \sum_{d} k_{cd}((x_{ic}, y_c), z_d)$$

much fewer variables (further: parameter tying)

- B. Taskar, C. Guestrin and D. Koller, Maximum Margin Markov Networks, NIPS 2003.
- J. Lafferty, X. Zhu, and Y. Liu, *Kernel conditional random fields: Representation and clique selection*, ICML, 2004.
- Y. Altun, A. Smola, and T. Hofmann, *Exponential Families for Conditional Random Fields, UAI 2004*.
- T. Hofmann, B. Schölkopf, and A. Smola, *A Tutorial Review of RKHS Methods in Machine Learning, Annals of Statistics, 2008*

SVMstruct with Approximate Inference

- **Undergenerating** methods $$y' = \arg\max_{y \in \bar{\mathcal{Y}} \subset \mathcal{Y}} f(\mathbf{x}, y) \approx \arg\max_{y \in \mathcal{Y}} f(\mathbf{x}, y) = y^*$$
  - Examples: Greedy, loopy belief propagation
  - Theoretical analysis: convergence, bounded (but not arbitrary small) overall approximation quality, if quality of approximation can be bounded
$$\rho\left[\langle \mathbf{w}, \Phi(\mathbf{x}, y^*)\rangle + \Delta(y, y^*)\right] \leq \left[\langle \mathbf{w}, \Phi(\mathbf{x}, y')\rangle + \Delta(y, y')\right]$$

- **Overgenerating** methods $$y' = \arg\max_{y \in \bar{\mathcal{Y}} \supset \mathcal{Y}} f(\mathbf{x}, y) \approx \arg\max_{y \in \mathcal{Y}} f(\mathbf{x}, y) = y^*$$
  - Examples: LP relaxation (leading to solutions over {0,1/2,1}) [Boros & Hammer 2002] & Quadratic pseudo-Boolean optimization using graph cuts [Kolmogorov & Rother 2004]
  - Use fractional numbers (1/2) in constraints -> overly constrained problem, but feasible solution will be found

- T. Finley and T. Joachims. Training structural SVMs when exact inference is intractable. ICML 2008.

SVMstruct with Approximate Inference

- Evaluation on multi-labeled classification = Multiple coupled/overlapping binary classification problems
- Dependencies modeled as a binary MRF
- Relaxation methods (overgenerating) work best for training.
- Adding fractional constraints during training helps avoid fractional solutions in inference.

Table 2. Multi-labeling loss on six datasets. Results are grouped by dataset. Rows indicate separation oracle method. Columns indicate classification inference method.

	GREEDY	LBP	COMBINE	EXACT	LPROG	GREEDY	LBP	COMBINE	EXACT	LPROG
	SCENE DATASET					MEDIAMILL DATASET				
GREEDY	10.67±.28	10.74±.28	10.67±.28	10.67±.28	10.67±.28	23.39±.16	25.66±.17	24.32±.17	24.92±.17	27.05±.18
LBP	10.45±.27	10.54±.27	10.45±.27	10.42±.27	10.49±.27	22.83±.16	22.83±.16	22.83±.16	22.83±.16	22.83±.16
COMBINE	10.72±.28	11.78±.30	10.72±.28	10.77±.28	11.20±.29	19.56±.14	20.12±.15	19.72±.14	19.82±.14	20.23±.15
EXACT	10.08±.26	10.33±.27	10.08±.26	10.06±.26	10.20±.26	19.07±.14	27.23±.18	19.08±.14	18.75±.14	36.83±.21
LPROG	10.55±.27	10.49±.27	10.49±.27	10.49±.27	10.49±.27	18.50±.14	18.26±.14	18.26±.14	18.21±.14	18.29±.14
	YEAST DATASET					SYNTH1 DATASET				
GREEDY	21.62±.56	21.77±.56	21.58±.56	21.62±.56	24.42±.61	8.86±.08	8.86±.08	8.86±.08	8.86±.08	8.86±.08
LBP	24.32±.61	24.32±.61	24.32±.61	24.32±.61	24.32±.61	13.94±.12	13.94±.12	13.94±.12	13.94±.12	13.94±.12
COMBINE	22.33±.57	37.24±.77	22.32±.57	21.82±.56	42.72±.81	8.86±.08	8.86±.08	8.86±.08	8.86±.08	8.86±.08
EXACT	23.38±.59	21.99±.57	21.06±.55	20.23±.53	45.90±.82	6.89±.06	6.86±.06	6.86±.06	6.86±.06	6.86±.06
LPROG	20.47±.54	20.45±.54	20.47±.54	20.48±.54	20.49±.54	8.94±.08	8.94±.08	8.94±.08	8.94±.08	8.94±.08
	REUTERS DATASET					SYNTH2 DATASET				
GREEDY	5.32±.09	13.38±.21	5.06±.09	5.42±.09	16.98±.26	7.27±.07	27.92±.20	7.27±.07	7.28±.07	19.03±.15
LBP	15.80±.25	15.80±.25	15.80±.25	15.80±.25	15.80±.25	10.00±.09	10.00±.09	10.00±.09	10.00±.09	10.00±.09
COMBINE	4.90±.09	4.57±.08	4.53±.08	4.49±.08	4.55±.08	7.90±.07	26.39±.19	7.90±.07	7.90±.07	18.11±.15
EXACT	6.36±.11	5.54±.10	5.67±.10	5.59±.10	5.62±.10	7.04±.07	25.71±.19	7.04±.07	7.04±.07	17.80±.15
LPROG	6.73±.12	6.41±.11	6.38±.11	6.38±.11	6.38±.11	5.83±.05	6.63±.06	5.83±.05	5.83±.05	6.29±.06

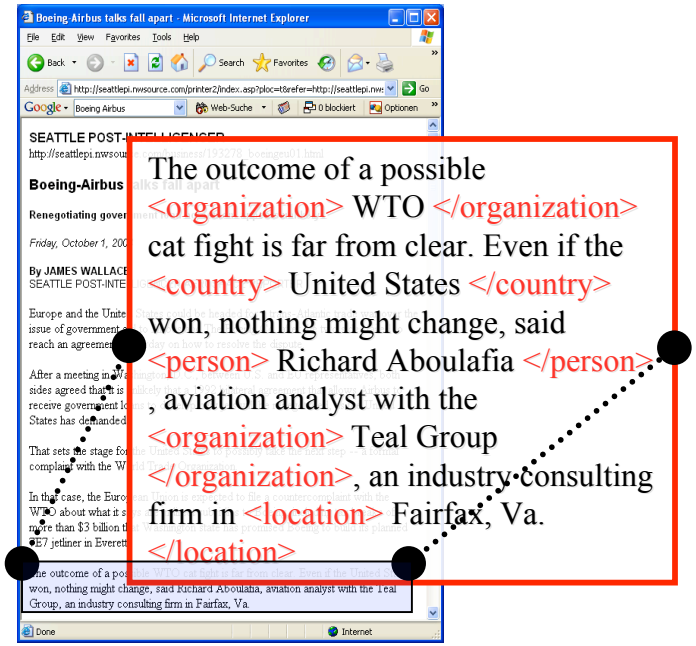
- T. Finley and T. Joachims. Training structural SVMs when exact inference is intractable. ICML 2008.

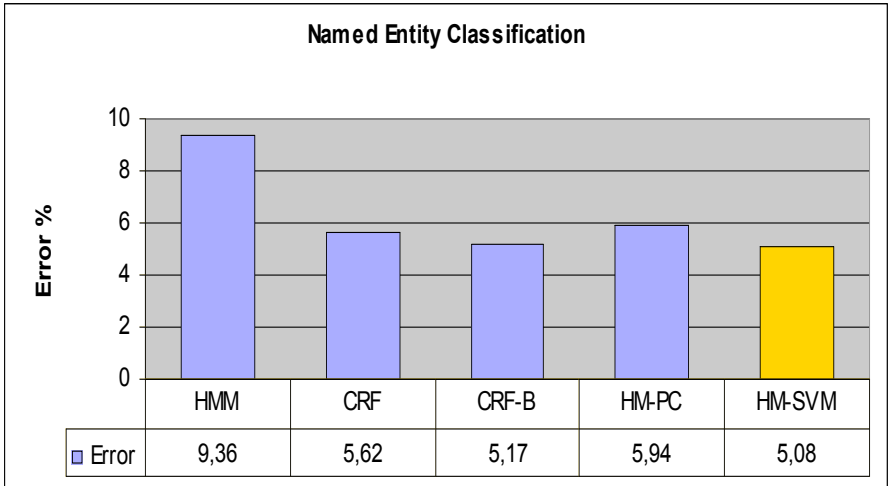
7 Applications

7.1 Named Entity Classification

Application: Named Entity Classification

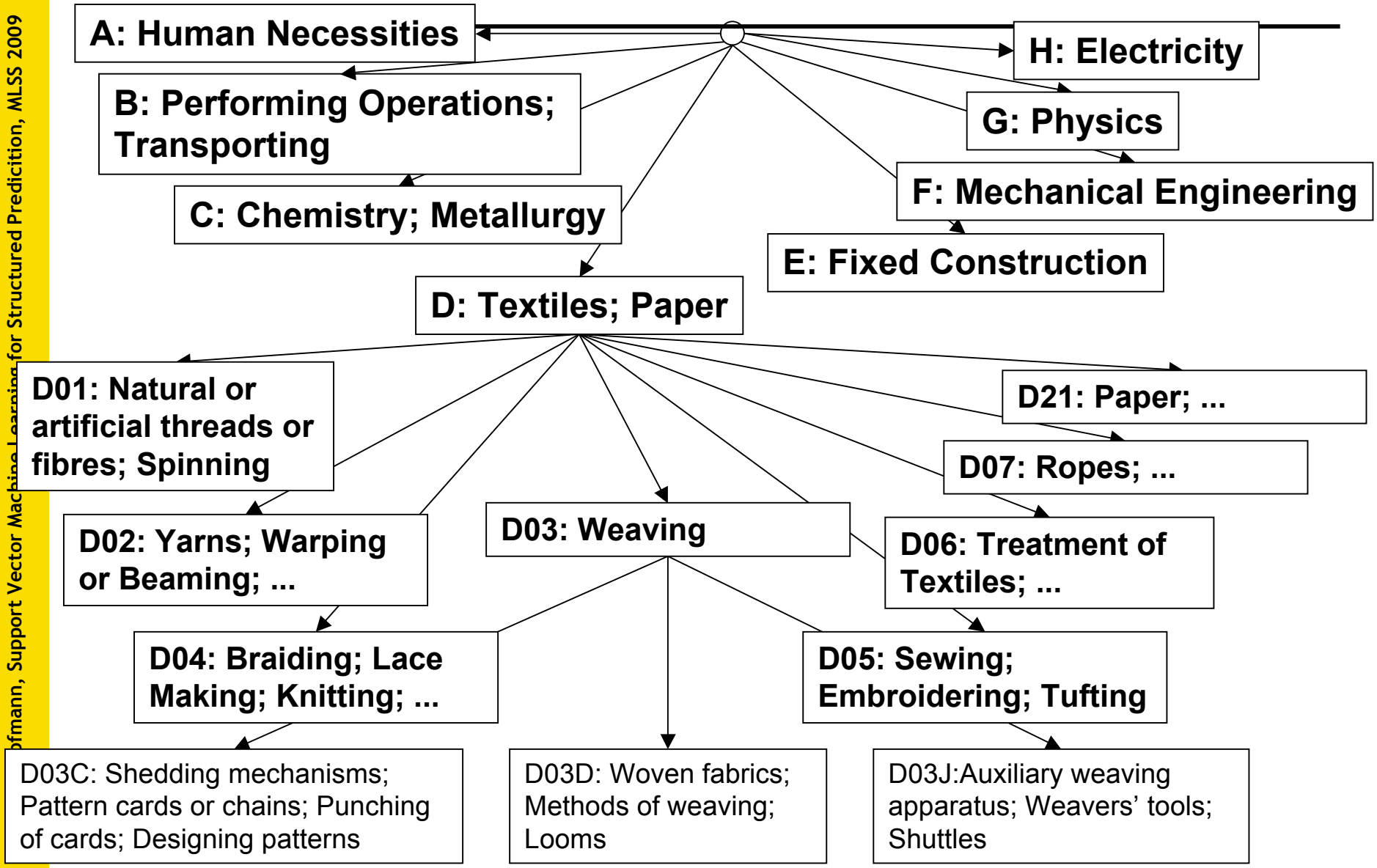
- Goal: finding names in sentences
- Label set: beginning and continuation of **person**, **location**, **organization** and **miscellaneous** names and non-name
- Data set: Spanish newswire corpus, CoNLL 2002
- Input features: word identity and spelling properties of current, previous, and next observation





7.2 Hierarchical Classification

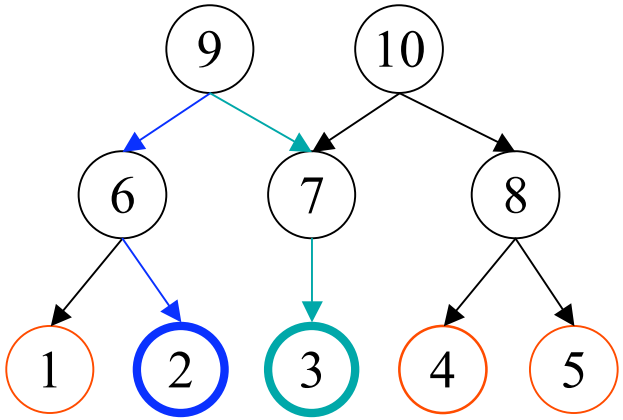
Taxonomies: IPC



- World Intellectual Property Organization. International patent classification. URL, 2001. http://www.wipo.int/classifications/en/

Class Attributes from Taxonomies

Hierarchical Classification with Taxonomies



$$\begin{matrix}\Phi_{\mathcal{Y}}(2) \\ \Phi_{\mathcal{XY}}(x,2)\end{matrix} = \begin{pmatrix}0\\1\\0\\0\\0\\1\\0\\0\\1\\0\end{pmatrix} \begin{pmatrix}\mathbf{0}\\ \Phi_{\mathcal{X}}(x)\\ \mathbf{0}\\ \mathbf{0}\\ \mathbf{0}\\ \Phi_{\mathcal{X}}(x)\\ \mathbf{0}\\ \mathbf{0}\\ \Phi_{\mathcal{X}}(x)\\ \mathbf{0}\end{pmatrix} \qquad \begin{matrix}\Phi_{\mathcal{Y}}(3) \\ \Phi_{\mathcal{XY}}(x',3)\end{matrix} = \begin{pmatrix}0\\0\\1\\0\\0\\0\\1\\0\\1\\0\end{pmatrix} \begin{pmatrix}\mathbf{0}\\ \mathbf{0}\\ \Phi_{\mathcal{X}}(x')\\ \mathbf{0}\\ \mathbf{0}\\ \mathbf{0}\\ \Phi_{\mathcal{X}}(x')\\ \mathbf{0}\\ \Phi_{\mathcal{X}}(x')\\ \mathbf{0}\end{pmatrix}$$

$$\langle \Phi_{\mathcal{XY}}(x,y), \Phi_{\mathcal{XY}}(x',y') \rangle = \underbrace{\langle \Phi_{\mathcal{Y}},(y), \Phi_{\mathcal{Y}}(y') \rangle}_{\text{similarity function over categories (generalization across categories, not just across inputs)}} \langle \Phi_{\mathcal{X}},(x), \Phi_{\mathcal{X}}(x') \rangle$$

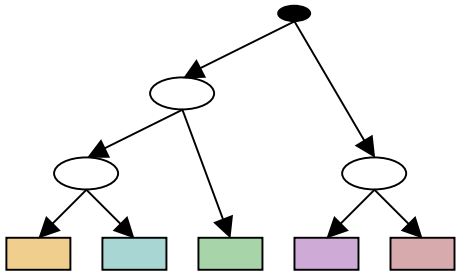
special case, multiclass = orthogonal class representation

- T. Hofmann, L. Cai, and M. Ciaramita, *Learning with Taxonomies: Classifying Documents and Words*, Workshop on Syntax, Semantics, and Statistics, NIPS, 2003

Hierarchical Classification

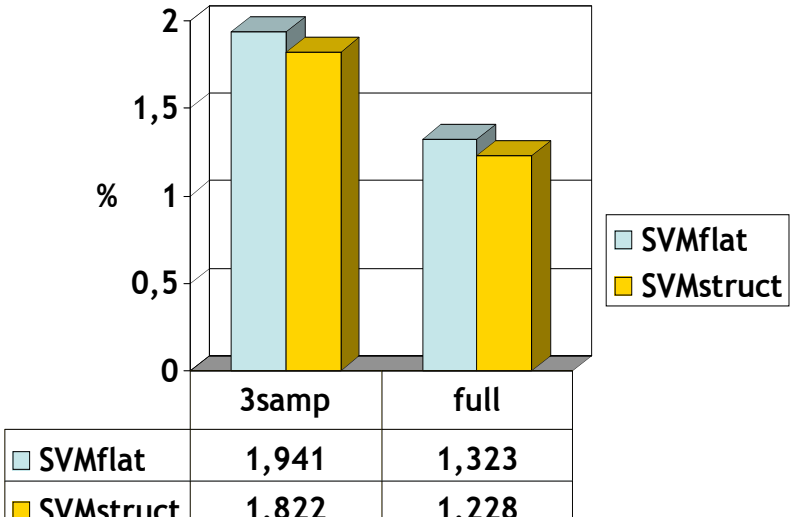
Model

- Input feature map: vector space representation of documents
- Class attributes from IPC taxonomy
- (Tree-specific loss function)
- *Soft-margin objective and SVMstruct – explained later!*



Experiments

- WIPO-alpha: patent database (2002)
- Based on IPC taxonomy
- Trained on different sections, full and subsampled (3 training examples per leaf category)



	3samp	full
SVMflat	1,941	1,323
SVMstruct	1,822	1,228

• L. Cai and T. Hofmann, *Hierarchical Document Categorization with Support Vector Machines, ACM CIKM, 2004.*

7.3 Binary Classification with Unbalanced Classes

Unbalanced Classes in Text Categorization

- Binary text categorization: only very few positive examples (<1%)
- Classification error = poor loss function (99%+ by trivial classification)
- More suitable error metrics:
  - **F1 score** = harmonic mean of precison and recall)
  - Precision-recall breakeven point
- Problem: **functions of sets of** (not individual) **predictions** -> structured classification

$$\triangle(y, \bar{y}) = 1 - F_1(y, \bar{y}), \quad y, \bar{y} \in \{-1, +1\}^n$$

- Representation (reduces to standard classification for 0/1 loss)

$$\Phi((\mathbf{x}_1, \ldots, \mathbf{x}_n), (y_1, \ldots, y_n)) = \sum_{i=1}^{n} y_i \Phi(\mathbf{x}_i)$$

- Constraint generation/prediction is efficient for metrics that depend on contingency table of prediction errors

- T. Joachims. A support vector method for multivariate performance measures. ICML 2005.

Experiments

- Comparison on 4 data sets
- Plain binary SVM classification vs. SVMs trained on specific loss

DATASET	METHOD	F_1	PRBEP	ROCAREA
REUTERS	STRUCT SVM	62.0	68.2	99.1
	SVM	56.1	65.7	98.6
ARXIV	STRUCT SVM	56.8	58.4	92.8
	SVM	49.6	57.9	92.7
OPTDIGITS	STRUCT SVM	92.5	92.7	99.4
	SVM	91.5	91.5	99.4
COVERTYPE	STRUCT SVM	73.8	72.1	94.6
	SVM	73.9	71.0	94.1

• T. Joachims. A support vector method for multivariate performance measures. ICML 2005.

7.4 Learning to Rank

Optimizing MAP with SVMstruct

- Loss function = 1 - MAP (mean average precision)
- **Inputs** q = search query
- **Outputs** y = rankings of documents/URLs (matrix of pairwise orderings)
- Training data: set of relevant and irrelevant documents per query

$$\Phi(q,y) = \frac{1}{|C^+||C^-|} \sum_{i \in C^+} \sum_{j \in C^-} y_{ij} \left[\phi(q,\mathbf{x}_i) - \phi(q,\mathbf{x}_j)\right]$$

- Efficient constraint generation: exploit fact that MAP only depends on label sequence

	TREC 9		TREC 10	
Model	MAP	W/L	MAP	W/L
$\text{SVM}^{\Delta}_{map}$	0.242	–	0.236	–
Best Func.	0.204	39/11 **	0.181	37/13 **
2nd Best	0.199	38/12 **	0.174	43/7 **
3rd Best	0.188	34/16 **	0.174	38/12 **

Comparison w/ Indri functions (Lemur project)

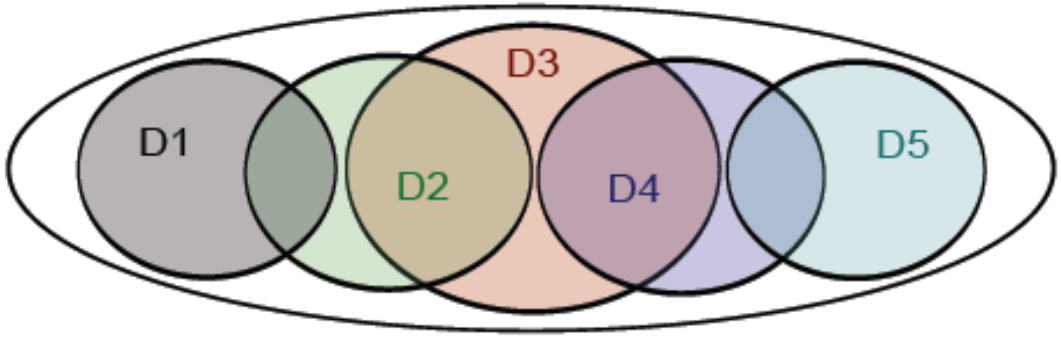
- Yisong Yue, T. Finley, F. Radlinski, T. Joachims, A Support Vector Method for Optimizing Average Precision. SIGIR 2007.

7.5 Optimizing Search Diversity

Search Result Diversity

- Traditionally:
  - Assess relevance of a document on ordinal scale (e.g. binary)
  - Quality of ranking = only depends on sequence of relevance scores (e.g. MAP, discounted cumulative gain)

- Challenge:
  - Relevance of document (at rank r) depends on other documents (at rank r'<r) = **relative relevance**
  - Novelty relative to other search result matters (diversity is good!)



- Effectively one needs to look at ranking and model inter-document dependencies = collective classification problem

Diversity via Topic Coverage

- Manually annotate results with **sub-topics** (for selected set of training queries)
- Prediction goal: find top n (=10) documents (=y) that maximize topic coverage, minimize weighted fraction of uncovered topics

$$\triangle(T, y), \qquad y \subset \{d_1, \ldots, d_m\}, \ |y| = n$$

- **Budgeted maximum coverage** problem -> (1-1/e) greedy approx.

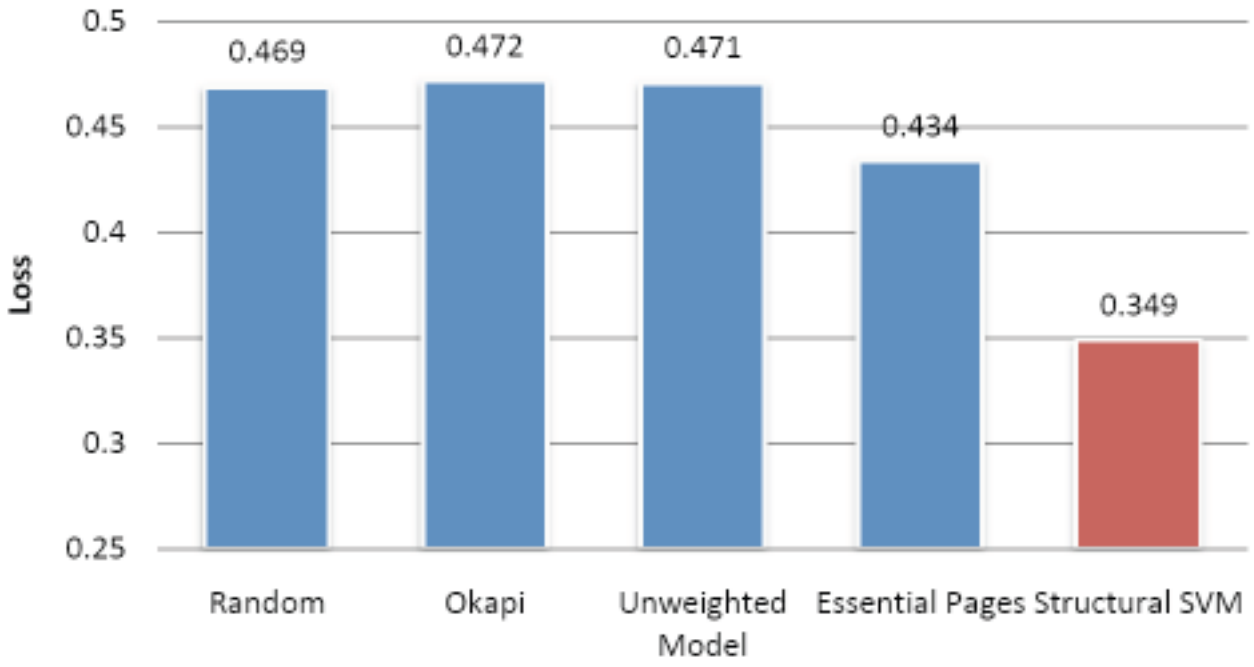
- How to generalize to new queries? Learn a **term weighting function**.

$$\phi(v, \mathbf{x}) = \begin{pmatrix} 1[v \text{ appears in at least } 15\% \text{ of } \mathbf{x}] \\ 1[v \text{ appears in at least } 35\% \text{ of } \mathbf{x}] \\ \ldots \end{pmatrix} \qquad \Psi(\mathbf{x}, \mathbf{y}) = \sum_{v \in V(\mathbf{y})} \phi(v, \mathbf{x}).$$

- T. Finley and T. Joachims. Training structural SVMs when exact inference is intractable. ICML 2008.
- Y. Yue and T. Joachims. Predicting diverse subsets using structural SVMs. ICML 2008

Experiments

- TREC 6-8 interactive track queries
- Fine-grained topic annotations
- Comparison with Okapi, unweighted word model, Essential Pages algorithm



- Y. Yue and T. Joachims. Predicting diverse subsets using structural SVMs. ICML 2008

Conclusions

- Generalization of SVMs to structured classification via **compatibility functions**
- **SVMstruct**: Hinge-loss induced sparseness, column generation approach to produce sequence of relaxations
  - efficient and highly generic approach
  - simple and theoretically well understood
  - alternative: clique-based representer theorem
- Wide range of **applications**
  - structured prediction problems
  - collective classification
  - many use cases in information retrieval