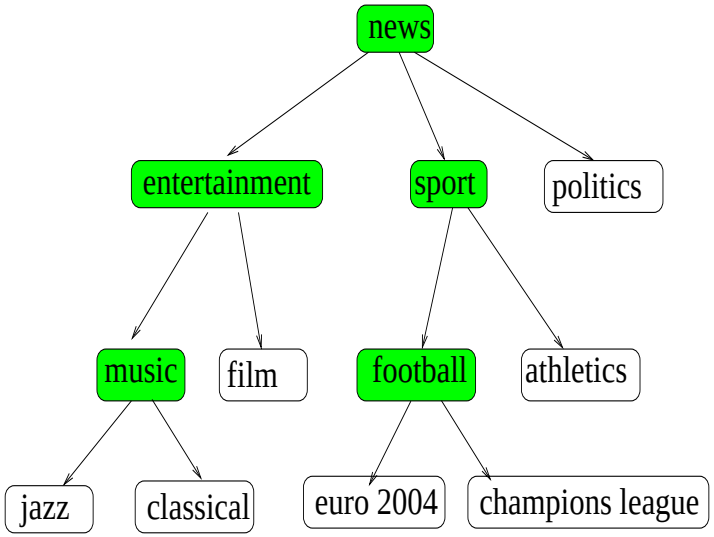
Learning Hierarchical Multi-Category Text Classification Models

Juho Rousu
Department of Computer Science
University of Helsinki, Finland

Craig Saunders, Sandor Szedmak and John Shawe-Taylor
Electronics and Computer Science
University of Southampton, UK

Hierarchical Multilabel Classification: union of partial paths model

Goal: Given document x, and hierachy $T = (V, E)$, predict multilabel $\mathbf{y} \in \{+1, -1\}^k$ where the positive microlabels y_i form a union of partial paths in T




BBC News | ENTERTAINMENT | Football pundit accuses Posh

Front Page
World
UK
UK Politics
Business
Sci/Tech
Health
Education
Sport
Entertainment
New Music
Releases
Talking Point
In Depth
Audio/Video

Saturday, 8 January, 2000, 15:02 GMT

Football pundit accuses Posh

David and Victoria Beckham are permanently in the public eye

"No arrests were made because there was no written evidence"
real 28k

Football Focus' Mark Lawrenson "He lives a kind of pop star life"
real 28k

BBC football pundit Mark Lawrenson has accused David Beckham and his pop star wife Victoria of "courting publicity".

SPECIAL REPORT
World Club Championships

Lawrenson, an analyst on BBC1's Football Focus, spoke out during a discussion about Beckham's sending off in Thursday's World Club Championship match.

Frequently used learning strategies for hierarchies

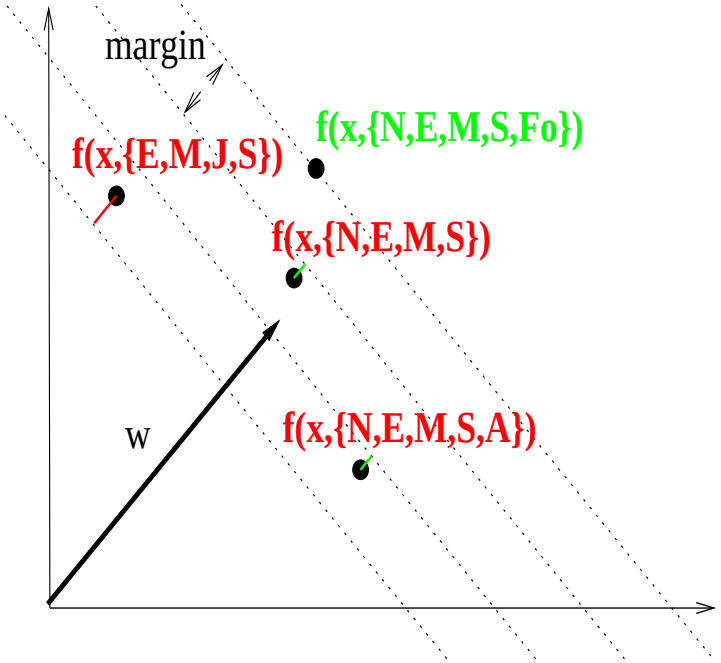
- **Flatten the hierarchy**: Learn each microlabel independently with classification learner of your choice
  - Computationally relatively inexpensive
  - Does not make use of the dependencies between the microlabels
- **Hierarchical training**: Train a node j with examples $(x, \mathbf{y})$ that belong to the parent, i.e. $y_{pa(j)} = 1$.
  - Some of the microlabel dependencies are learned.
  - However, training data fragments towards the leaves, hence estimation becomes less reliable
  - Model is not explicitly trained in terms of a loss function for the hierarchy.

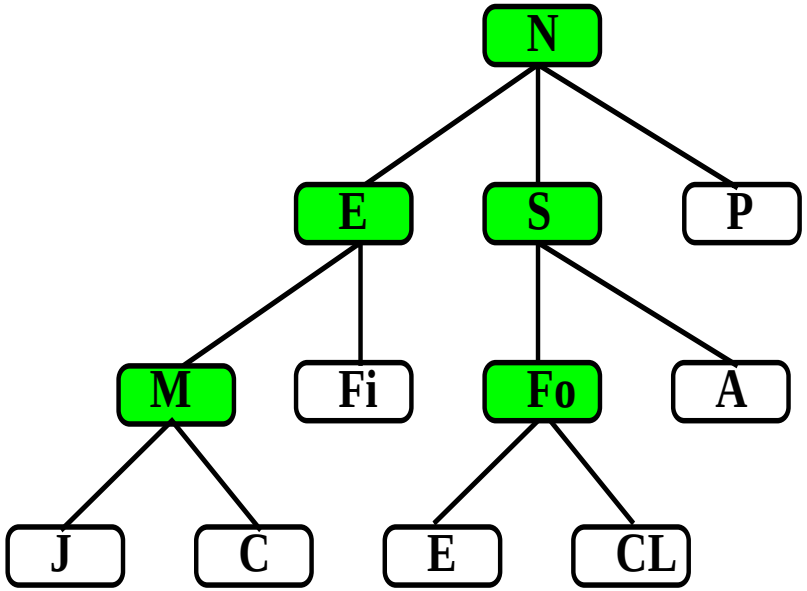
We wish to improve on these approaches...

Max-margin Structured output approach (Taskar et al., 2004; Tsochantaridis et al., 2004; ...)

Goal:

- Separate the correct multilabel from the incorrect ones by a large margin.
- Let the targeted margin scale proportionally to the loss of the multilabel
- Allow slack for non-separability of data





Loss functions for hierarchies

Consider a true multilabel $\mathbf{y} = (y_1, \ldots, y_k) \in \{+1, -1\}^k$, and a predicted one $\hat{\mathbf{y}} = (\hat{y}_1, \ldots, \hat{y}_k)$. Many choices:

- **Zero-one loss**: $\ell_{0/1}(\mathbf{y}, \hat{\mathbf{y}}) = [\![\mathbf{y} \neq \hat{\mathbf{y}}]\!]$; treats all incorrect multilabels alike
- **Symmetric difference loss**: $\ell_\Delta(\mathbf{y}, \hat{\mathbf{y}}) = \sum_j [\![y_j \neq \hat{y}_j]\!]$; counts incorrect microlabels.

Neither of the above takes the hierarchy into account. These do:

- **Hierarchical loss** (Cesa-Bianchi et al. 2004):
  $\ell_H(\mathbf{y}, \hat{\mathbf{y}}) = \sum_j c_j [\![y_j \neq \hat{y}_j \;\; \& \;\; y_k = \hat{y}_k \forall k \in ancestors(j)]\!]$; the first mistake along a path is penalized
- **Simplified hierarchical loss**:
  $\ell_{\tilde{H}}(\mathbf{y}, \hat{\mathbf{y}}) = \sum_j c_j [\![y_j \neq \hat{y}_j \;\; \& \;\; y_{parent(j)} = \hat{y}_{parent(j)}]\!]$; mistake in the child is penalized if the parent was correct.

Optimization problem

Primal form:

$$\min_{\mathbf{w},\boldsymbol{\xi}\geq 0} \frac{1}{2}\left|\left|\mathbf{w}\right|\right|^2 + C\sum_{i=1}^{m}\xi_i$$

$$\text{s.t. } \mathbf{w}^T\left(\phi(\mathbf{x}_i,\mathbf{y}_i) - \phi(\mathbf{x}_i,\mathbf{y})\right) \geq \boldsymbol{\ell}(\mathbf{y}_i,\mathbf{y}) - \xi_i, \forall i, \mathbf{y}\in\{+1,-1\}^k$$

Dual:

$$\max_{\boldsymbol{\alpha}>0} \sum_{i,\mathbf{y}} \alpha(x_i,\mathbf{y})\ell(\mathbf{y}_i,\mathbf{y}) - \frac{1}{2}\sum_{x_i,\mathbf{y}}\sum_{x_i',\mathbf{y}'}\alpha(x_i,\mathbf{y})^T K(x_i,\mathbf{y};x_i',\mathbf{y}')\alpha(x_i',\mathbf{y}')$$

$$\text{s.t.} \sum_{\mathbf{y}} \alpha(x_i,\mathbf{y}) \leq C, \forall i$$

- Exponential number (in size of the hierarchy) of primal constraints and dual variables, one per pseudo-example $(x_i,\mathbf{y})$
- Cannot be solved in this form for realistic-sized datasets, many approaches to make the model tractable (Taskar et al., 2004, 2005; Tshochantaridis et al. 2004)

Marginalized problem

A polynomial-sized problem can be obtained by marginalization (c.f. Taskar *et al.* , 2004), if the loss function and the feature representation is chosen suitably.

Our choices:

- Edge-marginals of dual variables : $\mu_e(x, \mathbf{y}_e) = \sum_{\mathbf{u}|\mathbf{u}_e=\mathbf{y}_e} \alpha(x, \mathbf{u})$
- Loss function decomposable by the edges: $\ell(\mathbf{y}, \mathbf{y}') = \sum_e \ell(\mathbf{y}_e, \mathbf{y}'_e)$; symmetric difference loss and simplified hierarchical loss apply
- Kernel decomposable by the edges: $K(x, \mathbf{y}; x', \mathbf{y}') = \sum_e K_e(x, \mathbf{y}_e; x, \mathbf{y}'_e)$; requires a feature vector with a block for each edge; we repeat $\phi(x)$ (bag-of-words or substring spectrum) for each edge-labeling $\boldsymbol{\phi}_e(x, \mathbf{y}_e) = (\llbracket \mathbf{y}_e = \mathbf{u}_e \rrbracket \boldsymbol{\phi}(x))_{\mathbf{u}_e \in \{+1,-1\}^2}$

Marginalized problem

The optimization problem gets the form:

$$\max_{\boldsymbol{\mu}>0} \sum_{e\in E} \boldsymbol{\mu}_e^T \boldsymbol{\ell}_e - \frac{1}{2} \sum_{e\in E} \boldsymbol{\mu}_e^T K_e \boldsymbol{\mu}_e$$

$$\text{s.t} \sum_{y,y'} \mu_e(i,y,y') \leq C, \forall i, e \in E,$$

$$\sum_{y'} \mu_e(i,y',y) = \sum_{y'} \mu_{e'}(i,y,y'),\ \forall i, \forall y, (e,e') : e = parent(e'),$$

Notes:

- Marginal consistency constraints need to be inserted to make the problem correspond to the original problem.
- Despite polynomial-size, the problem still is too large to solve with off-the-shelf QP methods, and it does not decompose easily.
- However: prediction (solving $\text{argmax}_{\mathbf{y}} \mathbf{w}^T \phi(x, \mathbf{y})$) is efficient using dynamic-programming inference to process the hierarchy

Efficient optimization

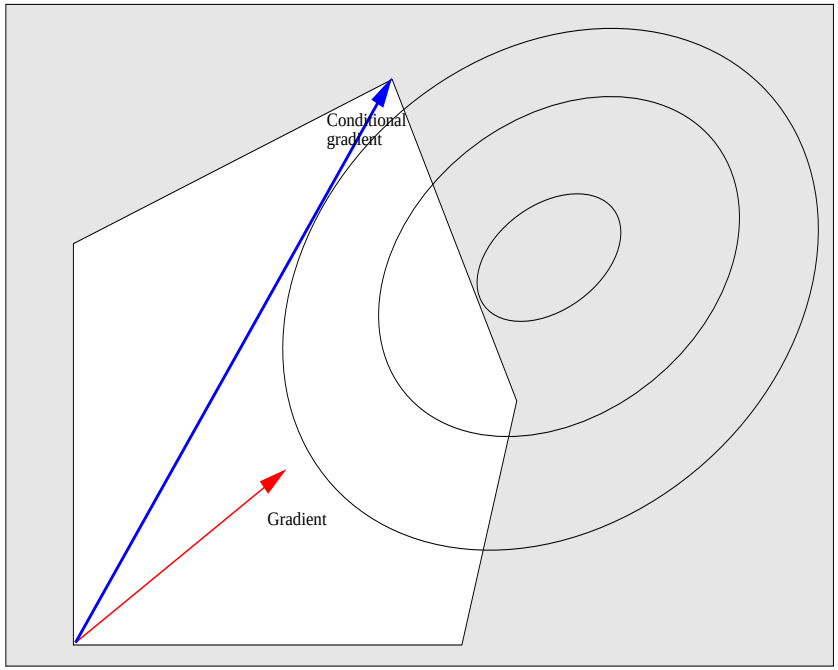
We use Conditional Gradient Descent to optimize the marginalized dual problem
Ingredients:

- Iterative gradient search in the feasible set
- Update direction is the highest feasible point assuming current gradient; found by solving a constrained linear program: $\max_{\boldsymbol{\mu} \in \mathcal{F}} (\boldsymbol{\ell} - K\boldsymbol{\mu}_0)^T \boldsymbol{\mu}$
- Allows to decompose training to single example subspaces: updates within single-example subspaces can be done independently, after initialization.

Conditional Gradient-based training

The algorithm:

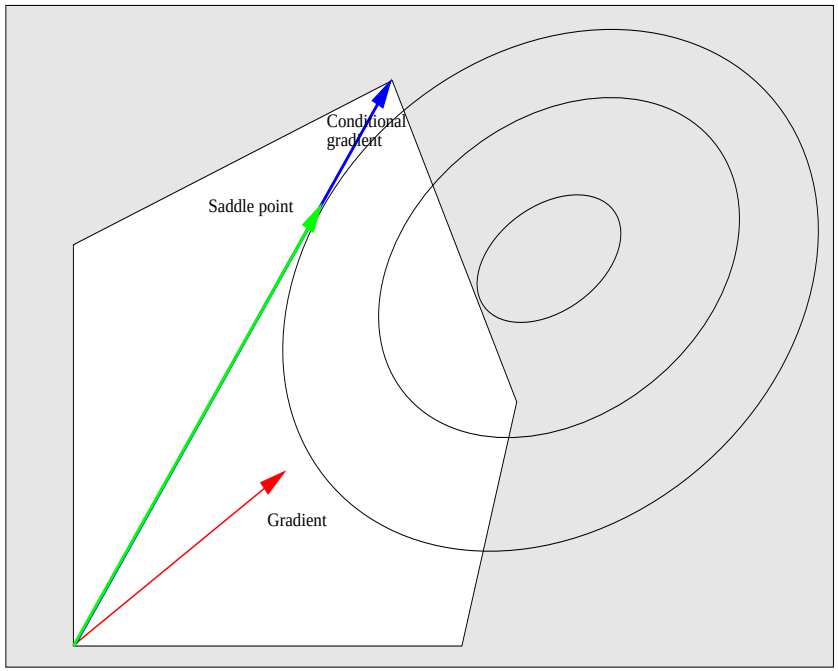
- Solve the feasible max-gradient direction



Conditional Gradient-based training

The algorithm:

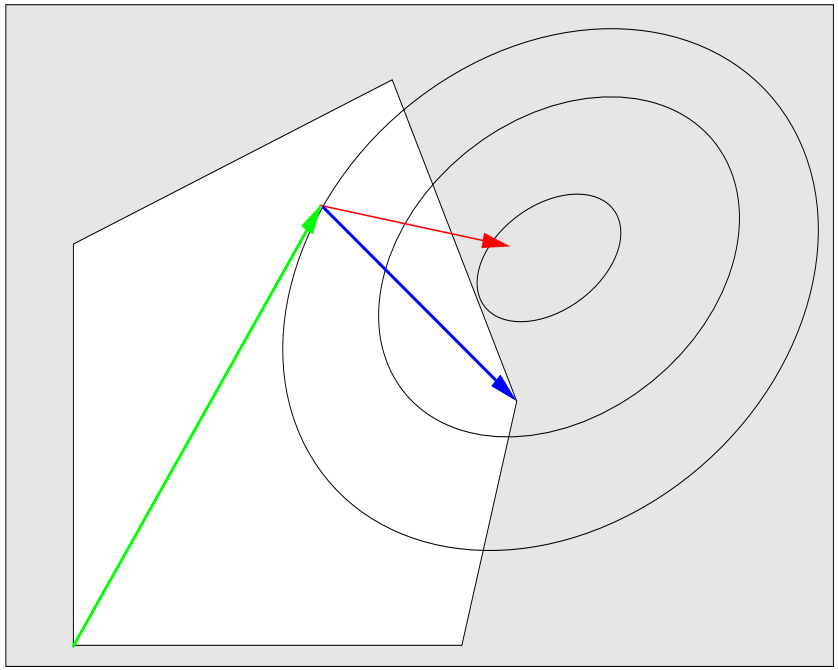
- Solve the feasible max-gradient direction
- Find saddle point along the direction



Conditional Gradient-based training

The algorithm:

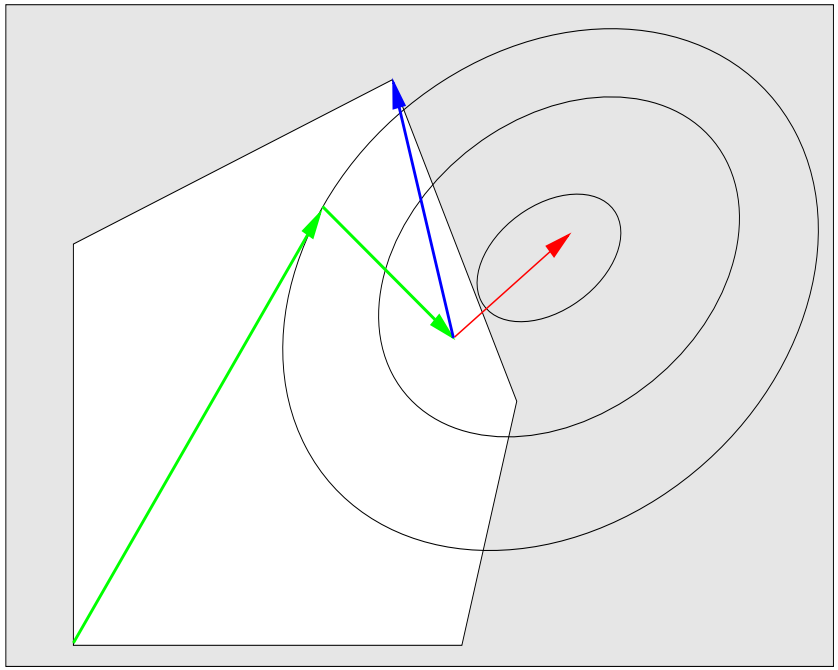
- Solve the feasible max-gradient direction
- Find saddle point along the direction
- Step to the saddle point and repeat



Conditional Gradient-based training

The algorithm:

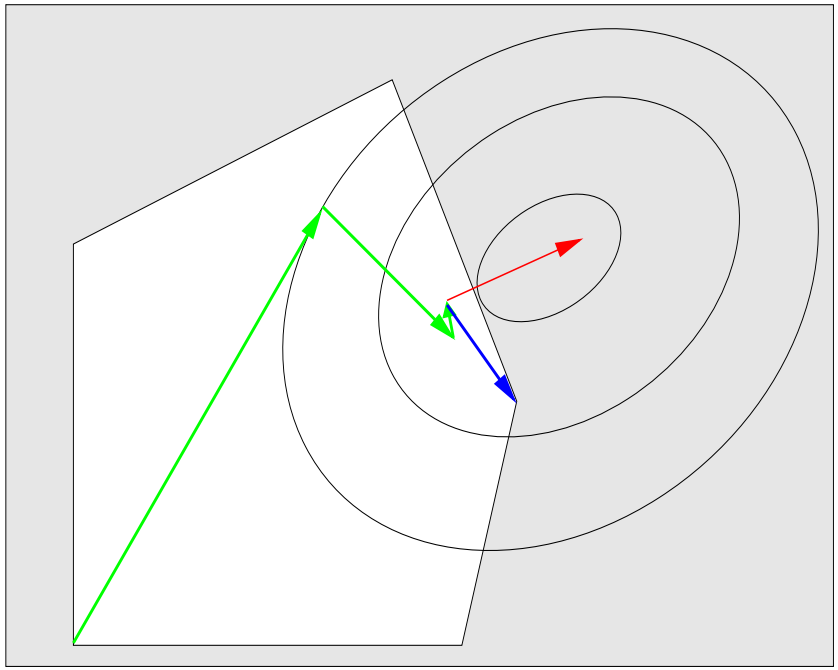
- Solve the feasible max-gradient direction
- Find saddle point along the direction
- Step to the saddle point and repeat



Conditional Gradient-based training

The algorithm:

- Solve the feasible max-gradient direction
- Find saddle point along the direction
- Step to the saddle point and repeat



Experiments

Datasets:

- Reuters Corpus Volume 1 ('CCAT' family), 34 microlabels, maximum tree depth 3, bag-of-words with TFIDF wieghting, 2500 documents were used for training and 5000 for testing.
- WIPO-alpha patent dataset (D section), 188 microlabels, maximum tree depth 4, 1372 documents for training, 358 for testing.

Algorithms:

- Our algorithm: H-M^3 ('Hierarchical Maximum Margin Markov')
- Comparison: Flat SVM, hierarchically trained SVM, hierarchical regularized least squares algorithm (Cesa-Bianchi et al. 2004)
- Implementation in MATLAB 7, LIPSOL solver used in the gradient ascent
- Tests run on a high-end Pentium PC with 1GB RAM

Microlabel prediction quality: whole tree

Prediction loss, precision, recall and F1 values obtained using different learning algorithms on Reuter's (left) and WIPO-alpha data (right).

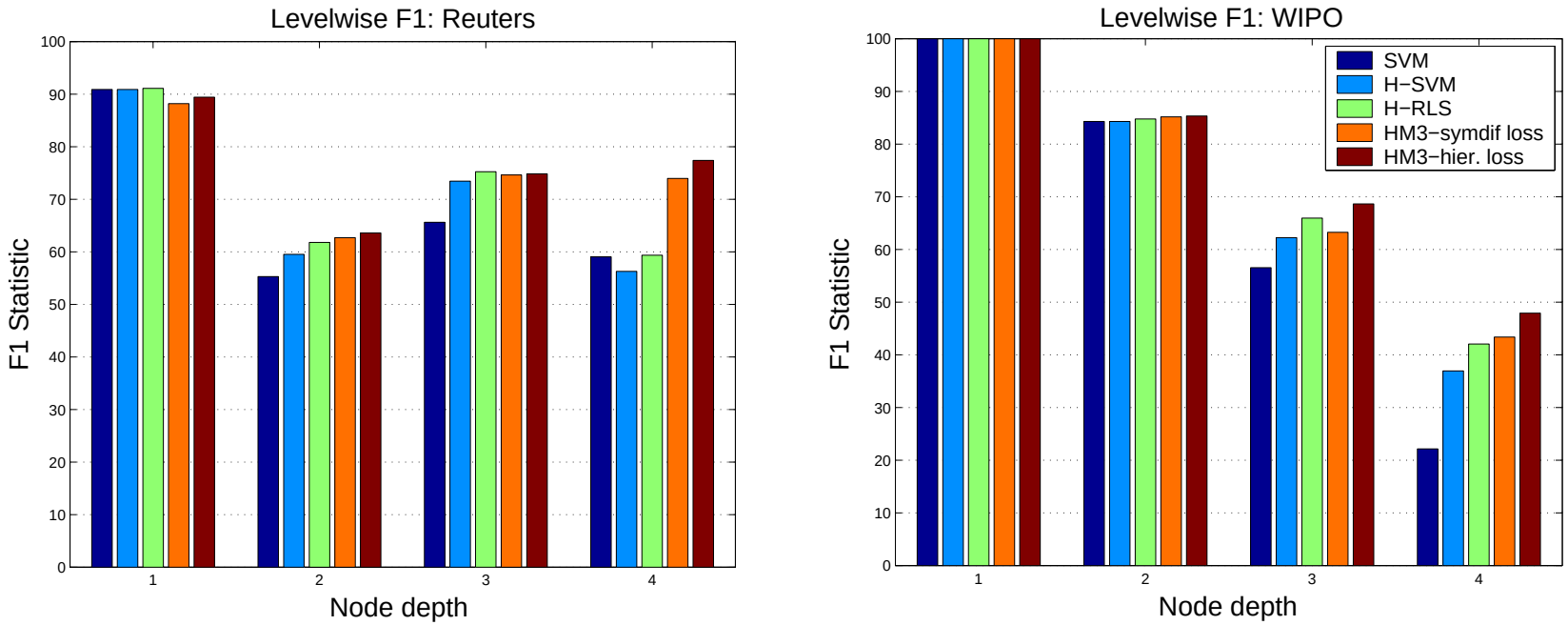
Alg.	$\ell_{0/1}$	P	R	F1
SVM	32.9	**94.6**	58.4	72.2
H-SVM	29.8	92.3	63.4	75.1
H-RLS	28.1	91.5	65.4	**76.3**
H-M^3-ℓ_Δ	***27.1***	*91.0*	*64.1*	*75.2*
H-M^3-$\ell_{\tilde{H}}$	*27.9*	*85.4*	***68.3***	*75.9*

Alg.	$\ell_{0/1}$	P	R	F1
SVM	87.2	**93.1**	58.2	71.6
H-SVM	76.2	90.3	63.3	74.4
H-RLS	72.1	88.5	66.4	75.9
H-M^3-ℓ_Δ	*70.9*	*90.3*	*65.3*	*75.8*
H-M^3-$\ell_{\tilde{H}}$	***65.0***	*84.1*	***70.6***	***76.7***

Flat SVM obtains highest precision but the lowest recall and F1. H-M^3 obtains the best $\ell_{0/1}$ loss and recall. The F1 values are similar for all hierarchical methods

Levelwise F1

F1 statistics computed for each node depth separately for Reuters (left) and WIPO (right)

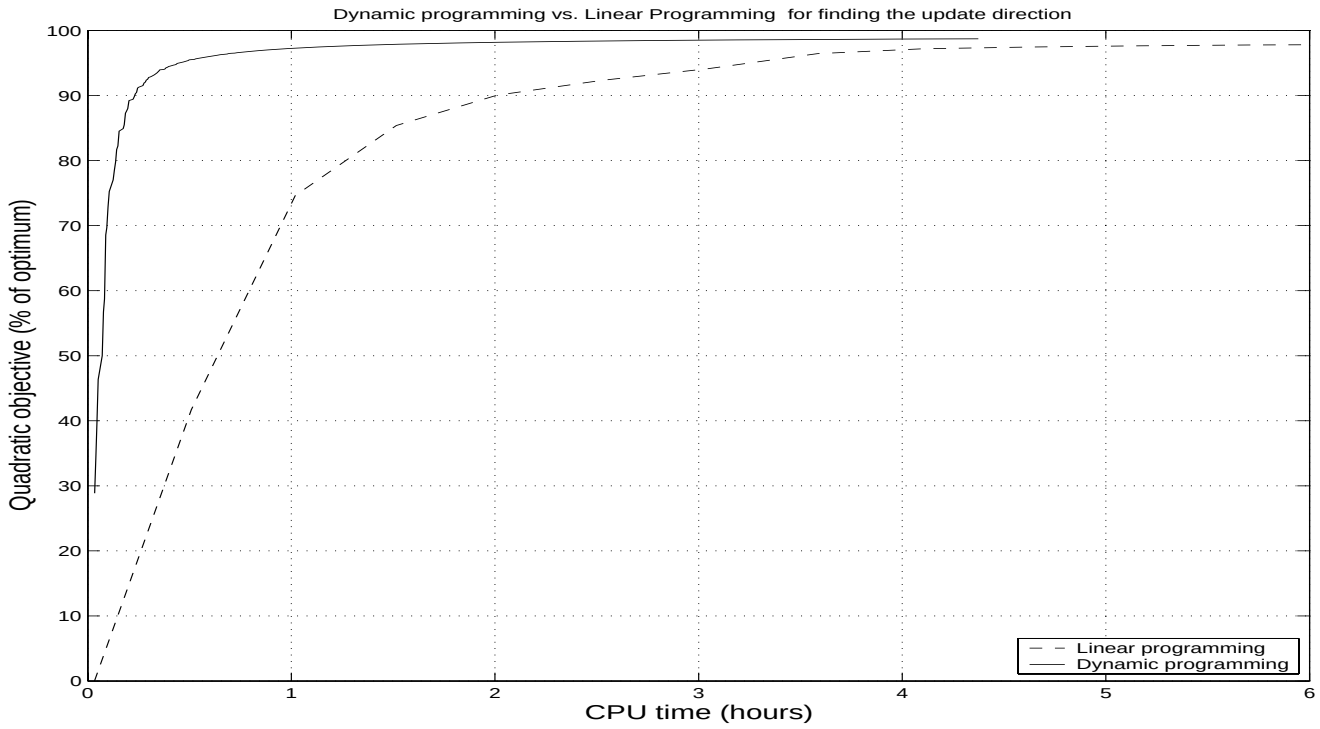


Flat SVM is poor in recalling deep nodes, H-M^3-$\ell_{\tilde{H}}$ is the best prediction method in the leaves.

Optimization efficiency

Optimization efficiency on WIPO dataset (1372 training examples, 188 microlabels) on a 3GHZ Pentium 4, 1GB main memory

LP = update directions via linear programming DP = update directions via dynamic programmming [not in the paper]



Conclusions

- We presented an kernel-based approach for hierarchical text classification when documents can belong to more than one category at a time
- Utilizing the dependency structure of microlabels in a Markovian way leads to improved prediction accuracy on deep hierarchies
- Optimization is made feasible by utilizing decomposition of the original problem and making incremental conditional gradient search in the subproblems.
- Tractable optimization for medium-sized datasets (thousands of examples hundreds of microlabels)