Kernel-based Discriminative Learning Algorithms for Labeling Sequences, Trees, and Graphs

Hisashi Kashima
Yuta Tsuboi
IBM Research
Tokyo Research Laboratory
{hkashima, yutat}@jp.ibm.com

Outline

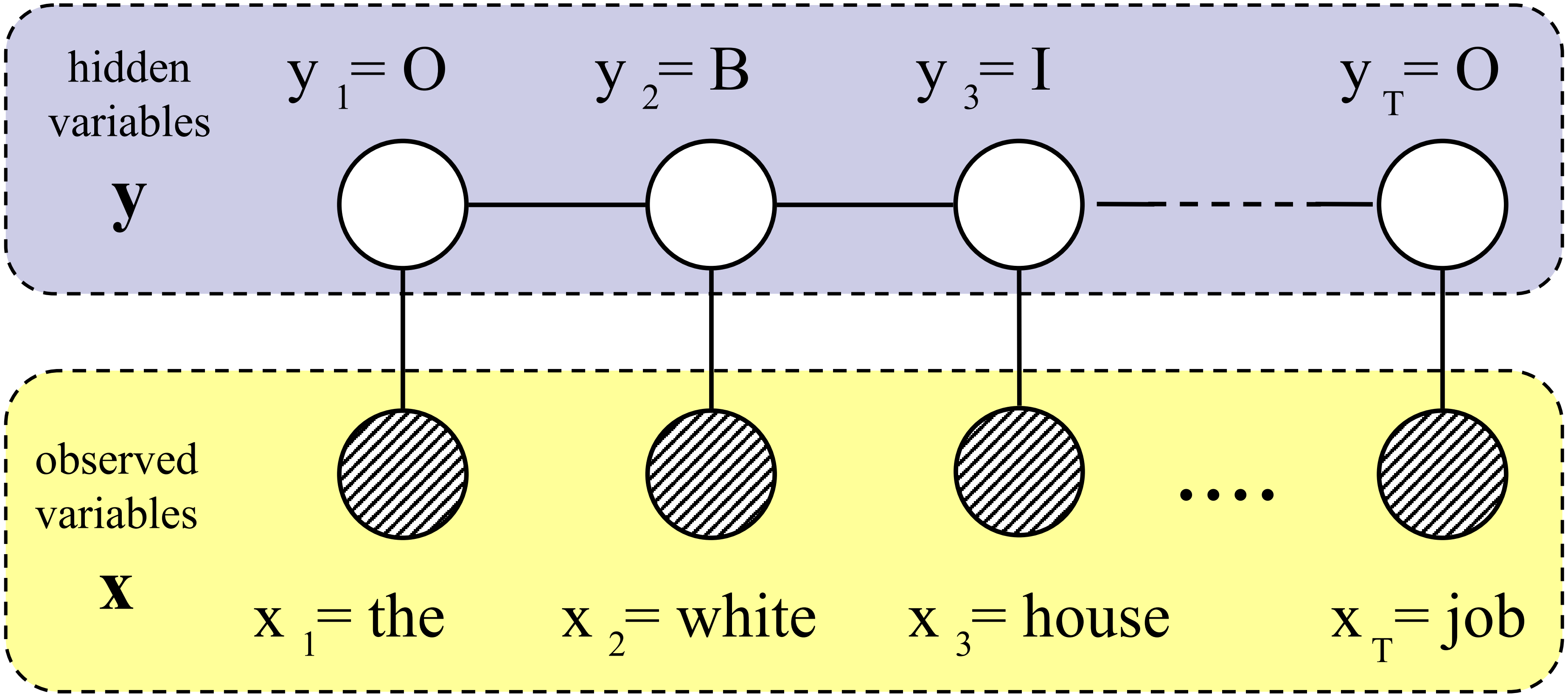
- Definition of labeling problem for structured data
- Hidden Markov (HM) perceptron
- Marginalized labeling perceptron
- Experiments on information extraction problems

Labeling Problems

- Mapping M from observed variables **x** to hidden variables **y**
  - $M : \Sigma_x^{|\mathbf{x}|} \rightarrow \Sigma_y^{|\mathbf{x}|}$ where |**x**|=|**y**|
- Training data are pairs of (**x**, **y**)
  - $(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), (\mathbf{x}^{(2)}, \mathbf{y}^{(2)}), (\mathbf{x}^{(3)}, \mathbf{y}^{(3)}), \ldots$
    where $|\mathbf{x}^{(i)}| = |\mathbf{y}^{(i)}|$
- Can be seen in many areas
  - Natural language processing
  - Bioinformatics
  - Web mining

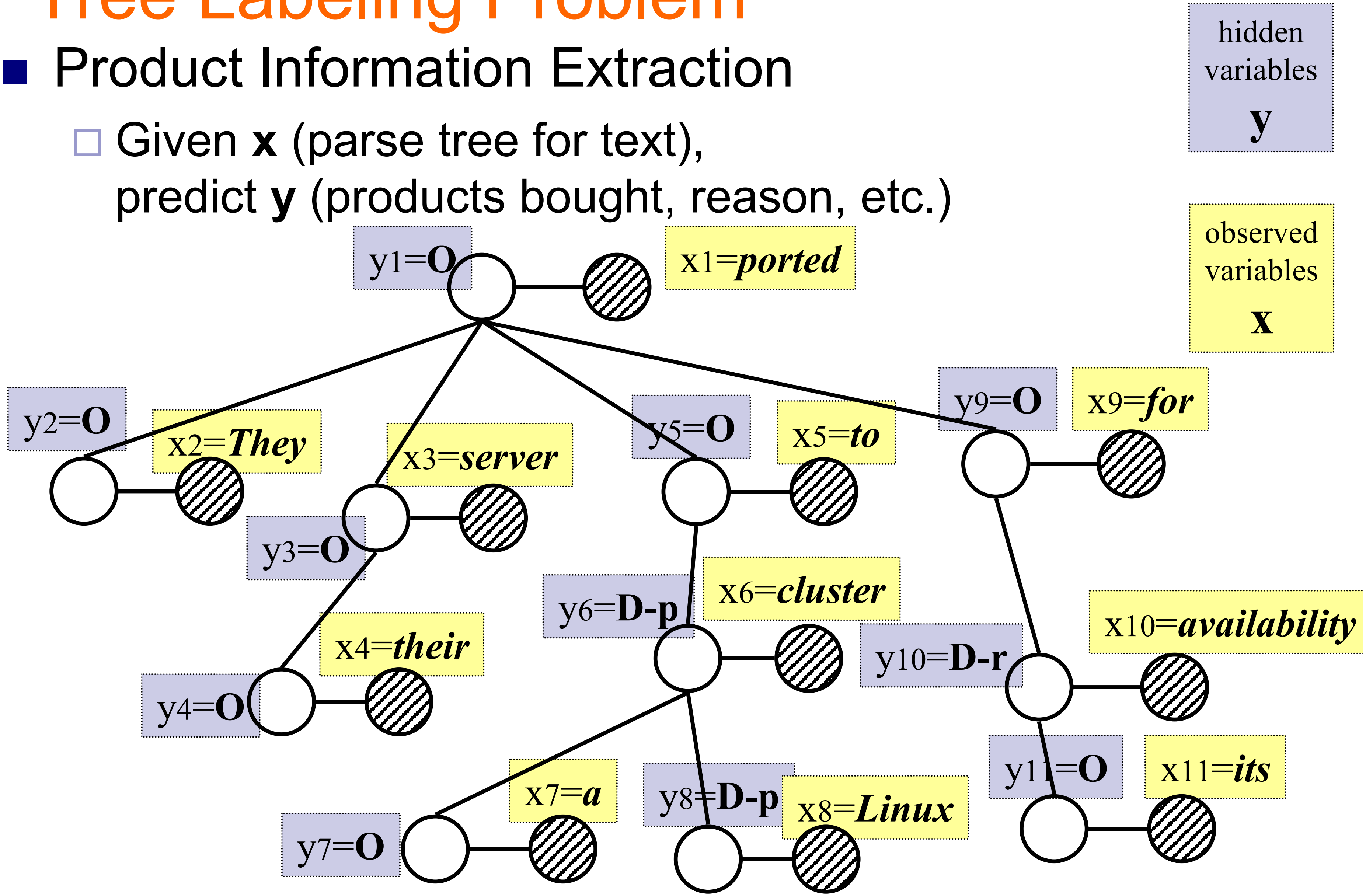
Sequence Labeling Problem

- Named Entity Extraction
  - Given **x** (sequence of words), predict **y** (organization name, person name,…)



Tree Labeling Problem

- Product Information Extraction
  - Given **x** (parse tree for text), predict **y** (products bought, reason, etc.)



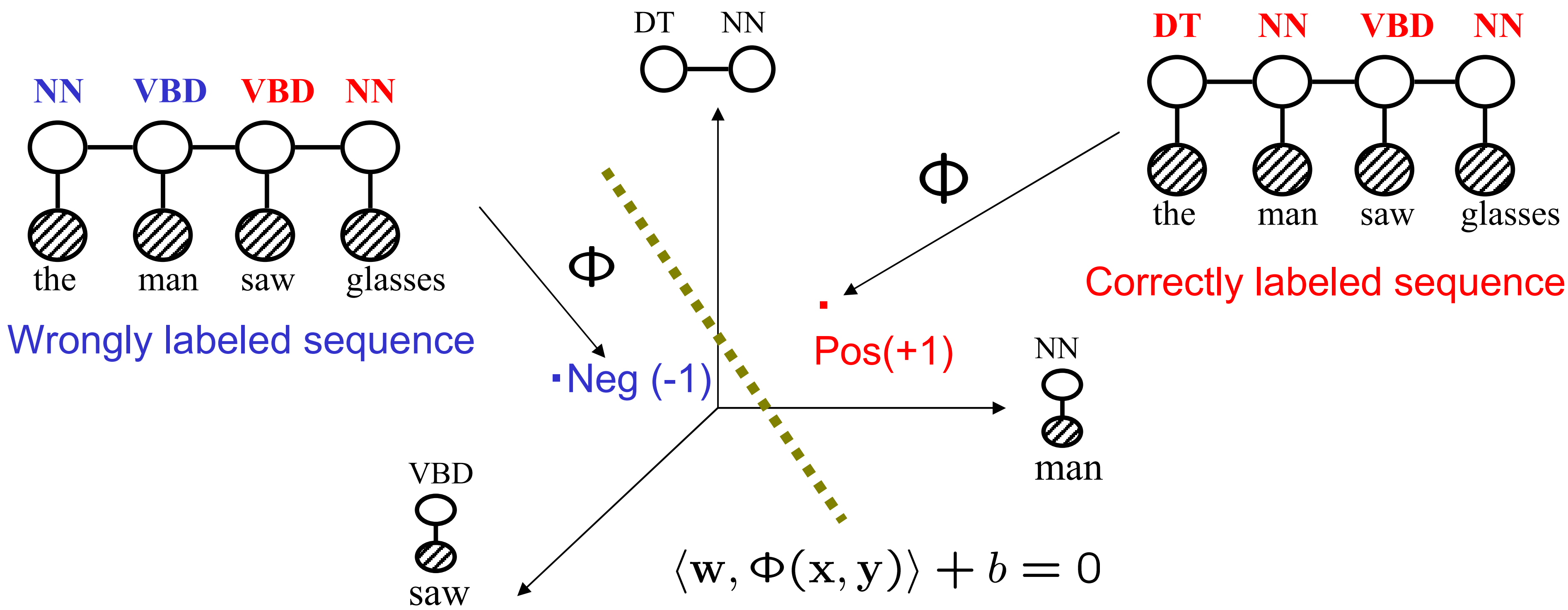
"They ported their server to a linux cluster for its availability"

HM(Hidden Markov)-Perceptron [Collins, 2002]

- Conditional models P(**y**|**x**) for labeling problems
  - Maximum Entropy Markov Model (MEMM) [McCallum et al, 2000]
  - Conditional Random Field (CRF) [Lafferty et al, 2001]
  - HM-SVM [Altun et al, 2003]
- Allows overlapping features
  - Prefix/suffix
  - Contains upper/lower case
  - Contains numbers
  - …
- Efficient alternative to CRF (Conditional Random Field)
  - Online algorithm

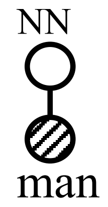
HM(Hidden Markov)-Perceptron [Collins, 2002]

- Reduces labeling problems to binary classification
  - $\Sigma_x^{|\mathbf{x}|} \times \Sigma_y^{|\mathbf{x}|} \to \{+1, -1\}$
- A labeled sequence (**x**,**y**) is mapped into a feature space
  - As a feature vector $\Phi(\mathbf{x}, \mathbf{y})$
  - Positive examples = sequences with correctly labeled **y**
  - Negative examples = sequences with wrongly labeled **y**

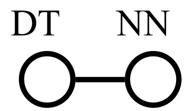


Feature Space for Labeled Sequences

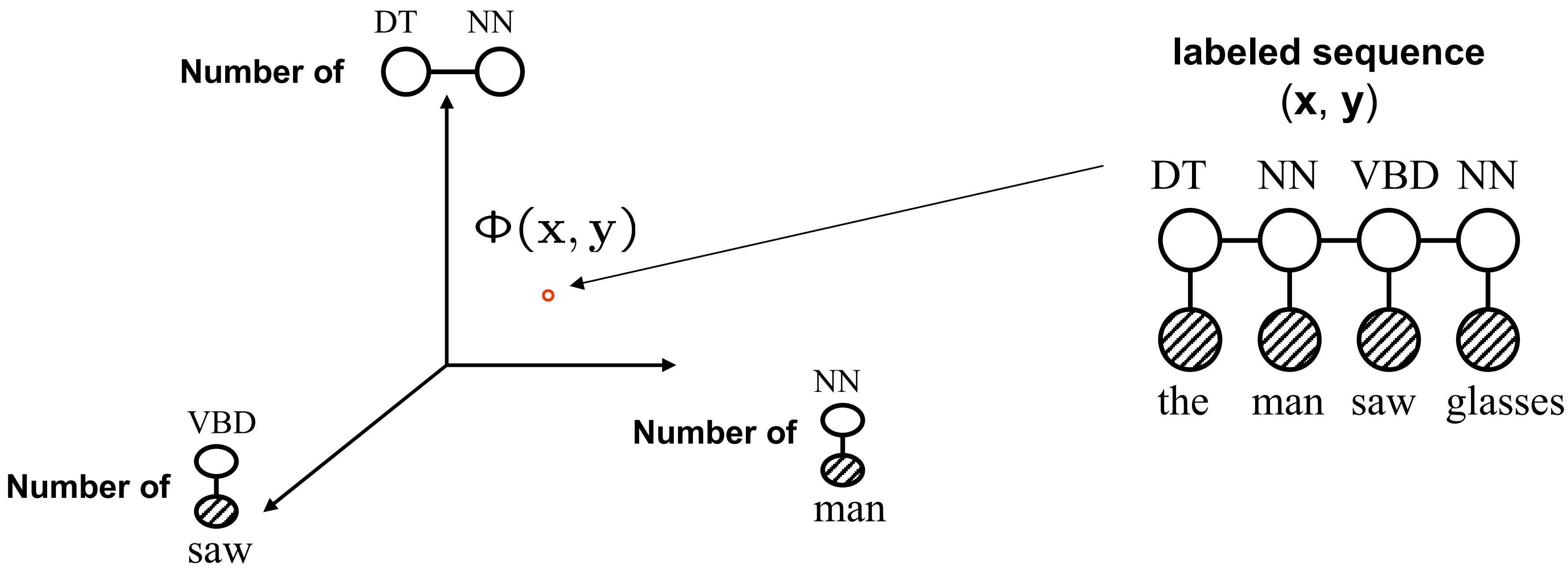
- Two types of features
  - Pairs of a hidden variable and an observed variable (corresponding to emitting probabilities in HMM)



  - Pairs of two consecutive hidden variables (corresponding to transition probabilities in HMM)



- Feature vector representation $\Phi(\mathbf{x},\mathbf{y})$ is constructed by using the number of times each feature appears in labeled sequence (**x**,**y**)



Algorithm of HM-Perceptron

- Prediction
  - Given **x**, output the prediction of **y** with the highest score

$$\hat{\mathbf{y}}(\mathbf{x}) = \underset{\mathbf{y}\in\Sigma_y{}^{|\mathbf{x}|}}{\mathrm{argmax}} \langle \mathbf{w}, \Phi(\mathbf{x},\mathbf{y})\rangle \ \text{(primal form)}$$

$$= \underset{\mathbf{y}\in\Sigma_y{}^{|\mathbf{x}|}}{\mathrm{argmax}} \sum_{j=1}^{|Example|} \sum_{\tilde{\mathbf{y}}\in\Sigma_y{}^{|\mathbf{x}|}} \alpha_j(\tilde{\mathbf{y}})\langle \Phi(\mathbf{x}^{(j)},\tilde{\mathbf{y}}), \Phi(\mathbf{x},\mathbf{y})\rangle \ \text{(dual form)}$$

Among all possible ways of labeling for **y**

kernels

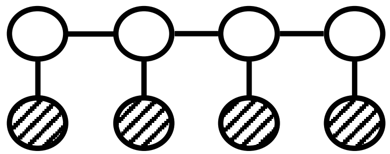
- Update
  - When the prediction is wrong,

If $\hat{\mathbf{y}}^{(i)} \neq \mathbf{y}^{(i)}$, (prediction for the i-th example is wrong)

- $\mathbf{w}^{new} = \mathbf{w}^{old} + \Phi(\mathbf{x}^{(i)},\mathbf{y}^{(i)}) - \Phi(\mathbf{x}^{(i)},\hat{\mathbf{y}}^{(i)})$ (primal)
- $$\left.\begin{array}{l} \bullet\ \alpha_i^{new}(\mathbf{y}^{(i)}) = \alpha_i^{old}(\mathbf{y}^{(i)}) + 1 \\ \bullet\ \alpha_i^{new}(\hat{\mathbf{y}}^{(i)}) = \alpha_i^{old}(\hat{\mathbf{y}}^{(i)}) - 1, \end{array}\right\} \ \textbf{(dual)}$$

HM-Perceptron with Long Features

- **Longer features** to incorporate wider contexts
  - idioms, motifs,…



- Computational complexity of prediction

$$\begin{aligned}\hat{\mathbf{y}}(\mathbf{x}) &= \operatorname*{argmax}_{\mathbf{y}\in\Sigma_y{}^{|\mathbf{x}|}} \langle \mathbf{w}, \Phi(\mathbf{x},\mathbf{y})\rangle \ \text{(primal form)} \\ &= \operatorname*{argmax}_{\mathbf{y}\in\Sigma_y{}^{|\mathbf{x}|}} \sum_{j=1}^{|Example|} \sum_{\tilde{\mathbf{y}}\in\Sigma_y{}^{|\mathbf{x}|}} \alpha_j(\tilde{\mathbf{y}})\langle \Phi(\mathbf{x}^{(j)},\tilde{\mathbf{y}}), \Phi(\mathbf{x},\mathbf{y})\rangle \ \text{(dual form)}\end{aligned}$$

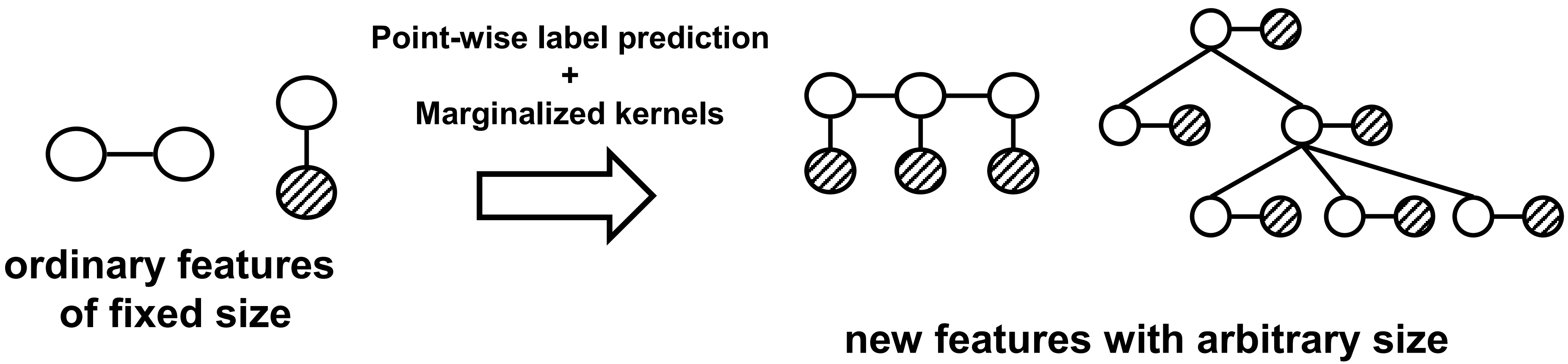
depends **exponentially** on the max number of hidden variables contained in features (even for the dual form)

  - Based on Viterbi algorithm

- Cannot incorporate arbitrary long features
  - Obstacle to apply kernels such as string kernels, tree kernels, graph kernels, …

Marginalized Labeling Perceptron
(Proposed Method)

- Can incorporate arbitrary size features with polynomial time complexity
  - Point-wise label prediction [Kakade et al. ,2002]
    - No need to use Viterbi
      since each hidden variable is predicted independently
  - Dual representation allows to use (Marginalized) kernels for structured data



Marginalized Labeling Perceptron (primal)

- Marginalized Labeling Perceptron

$$\hat{y}_t(\mathbf{x}) = \operatorname*{argmax}_{\tilde{y}_t \in \Sigma_y} \left\langle \mathbf{w}, \sum_{\mathbf{y}: y_t = \tilde{y}_t} P(\mathbf{y}|\mathbf{x}) \Phi(\mathbf{x}, \mathbf{y}) \right\rangle$$

Point-wise prediction (no Viterbi !)

Marginalized feature vector <u>with fixed label at *t*</u> (incorporates all candidates)

Feature vector of arbitrary size features

Prior distribution with small size features (e.g. HMM, MEMM, CRF, HM-Perceptron…)

- HM-Perceptron

$$\hat{\mathbf{y}}(\mathbf{x}) = \operatorname*{argmax}_{\mathbf{y} \in {\Sigma_y}^{|\mathbf{x}|}} \langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle$$

Find the best hidden sequence among all possible ways of labeling for *y* (needs Viterbi !)

Marginalized Labeling Perceptron (dual)

- Dual representation
  - Kernel methods to handle arbitrary size features efficiently
- Prediction

$$\hat{y}_t(\mathbf{x}) = \underset{\tilde{y}_t \in \Sigma_y}{\mathrm{argmax}} \sum_{j=1}^{|Examples|} \sum_{\tau=1}^{|\mathbf{x}^{(j)}|} \sum_{\tilde{y}_\tau \in \Sigma_y} \alpha_{j\tau}(\tilde{y}_\tau) \underbrace{K(\mathbf{x}^{(j)}, \mathbf{x}, \tau, t, \tilde{y}_\tau, \tilde{y}_t)}_{\text{Marginalized Kernel}}$$

  - Marginalized kernel

$$K(\mathbf{x}, \mathbf{x}', t, \tau, \tilde{y}_t, \tilde{y}'_\tau) = \sum_{\mathbf{y}:y_t=\tilde{y}_t} \sum_{\mathbf{y}':y'_\tau=\tilde{y}'_\tau} P(\mathbf{y}|\mathbf{x}) P(\mathbf{y}'|\mathbf{x}') \langle \Phi(\mathbf{x}, \mathbf{y}), \Phi(\mathbf{x}', \mathbf{y}') \rangle$$

Positions whose labels are fixed

fixed labels at *t* & *t*

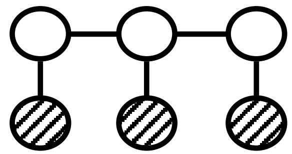
Marginalize over all possible ways of labeling of y & y' with fixed labels at *t* & τ

Kernel between two labeled structures

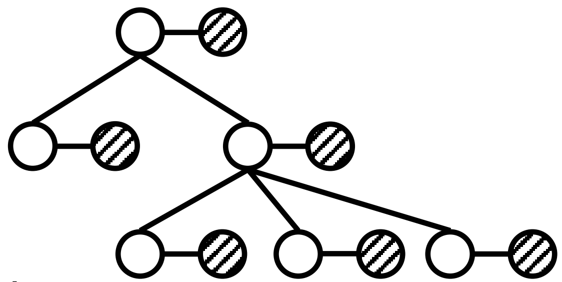
- Update
  - $\alpha_{it}^{new}(y_t^{(i)}) = \alpha_{it}^{old}(y_t^{(i)}) + 1$
  - $\alpha_{it}^{new}(\hat{\mathbf{y}}^{(i)}) = \alpha_{it}^{old}(\hat{y}_t^{(i)}) - 1$

Marginalized Kernels for Labeling Structured Data

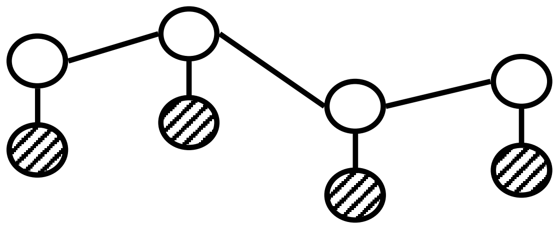
- $K(\mathbf{x},\mathbf{x}',t,\tau,\tilde{y}_t,\tilde{y}'_\tau) = \sum_{\mathbf{y}:y_t=\tilde{y}_t}\sum_{\mathbf{y}':y'_\tau=\tilde{y}'_\tau} P(\mathbf{y}|\mathbf{x})P(\mathbf{y}'|\mathbf{x}')\langle\Phi(\mathbf{x},\mathbf{y}),\Phi(\mathbf{x}',\mathbf{y}')\rangle$
- 3 kernels
  - Sequence labeling
    - Sequence features



  - Ordered tree labeling
    - Tree features



  - Directed acyclic graph (DAG) labeling
    - Path features



Arbitrary size features

Computation of Marginalized Kernels

- Given two sets of observed variables **x** & **x'**
  - Compute kernels for all pairs of positions *t* & *τ*
  - Kernel decomposition for efficient computation

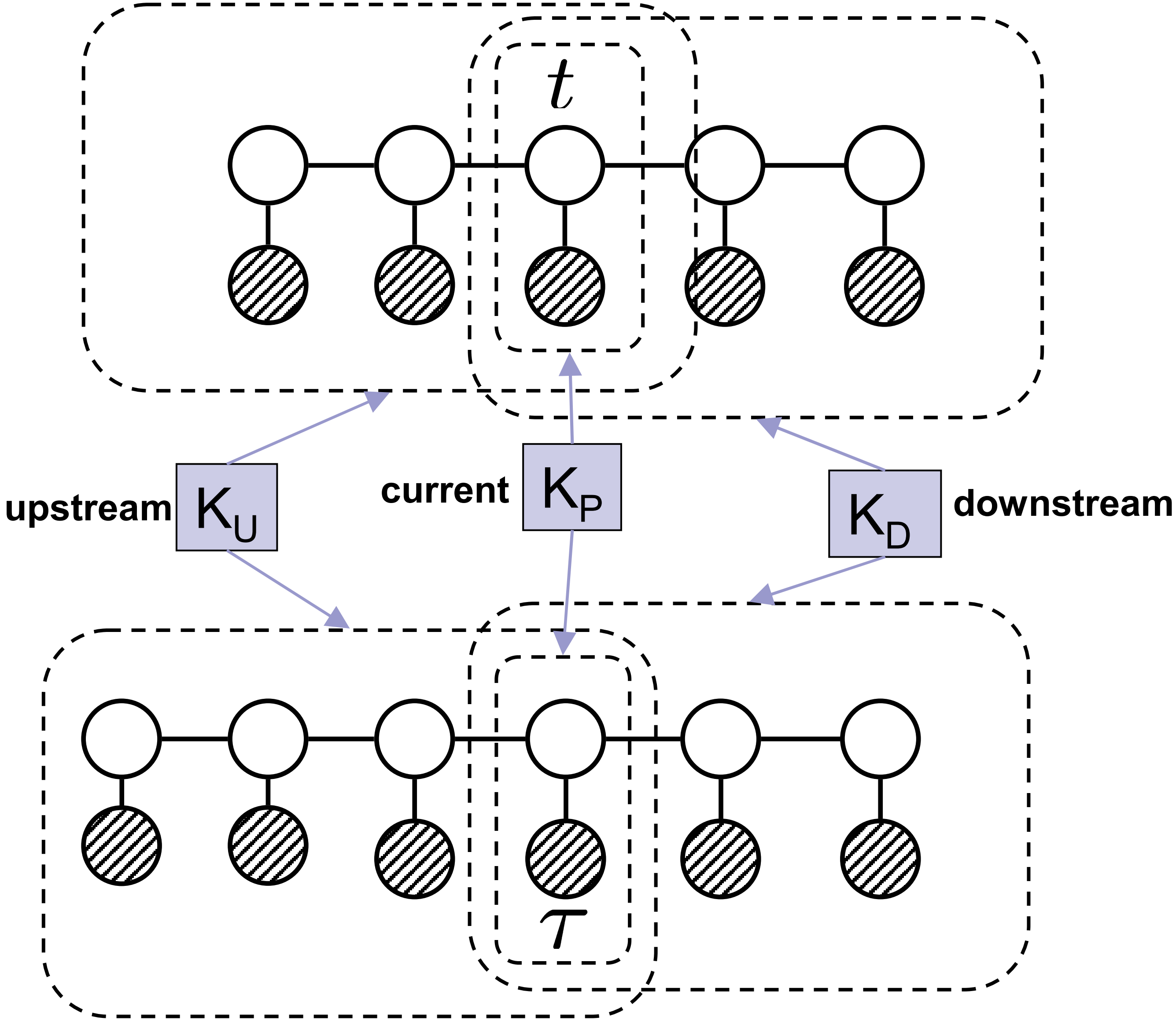
$$
\begin{aligned}
K(\mathbf{x},\mathbf{x}',t,\tau,\tilde{y}_t,\tilde{y}'_\tau) &= \sum_{\mathbf{y}:y_t=\tilde{y}_t}\ \sum_{\mathbf{y}':y'_\tau=\tilde{y}'_\tau} P(\mathbf{y}|\mathbf{x})P(\mathbf{y}'|\mathbf{x}')\langle\Phi(\mathbf{x},\mathbf{y}),\Phi(\mathbf{x}',\mathbf{y}')\rangle \\
&= \underbrace{K_U(\mathbf{x},\mathbf{x}',t,\tau)}_{\text{up-stream kernel}} \cdot \underbrace{K_P(\mathbf{x},\mathbf{x}',t,\tau,\tilde{y}_t,\tilde{y}'_\tau)}_{\text{current position kernel}} \cdot \underbrace{K_D(\mathbf{x},\mathbf{x}',t,\tau)}_{\text{down-stream kernel}}
\end{aligned}
$$

Each kernel is recursively computed by dynamic programming

  - Computational Complexity O(|x| |x'|)

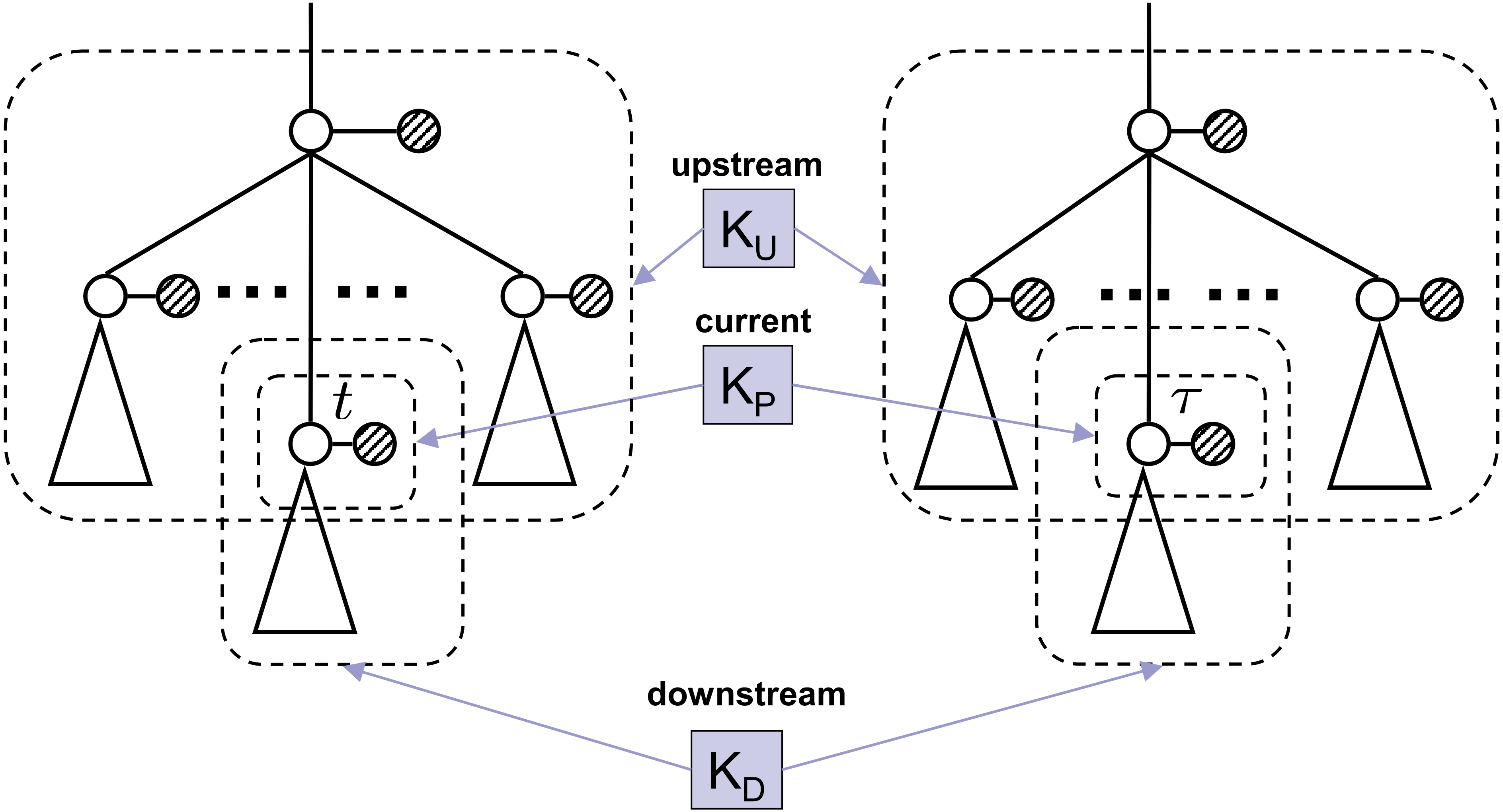
Kernel Decomposition (Sequences)

- **Analogous to Forward-Backward algorithm for HMM**

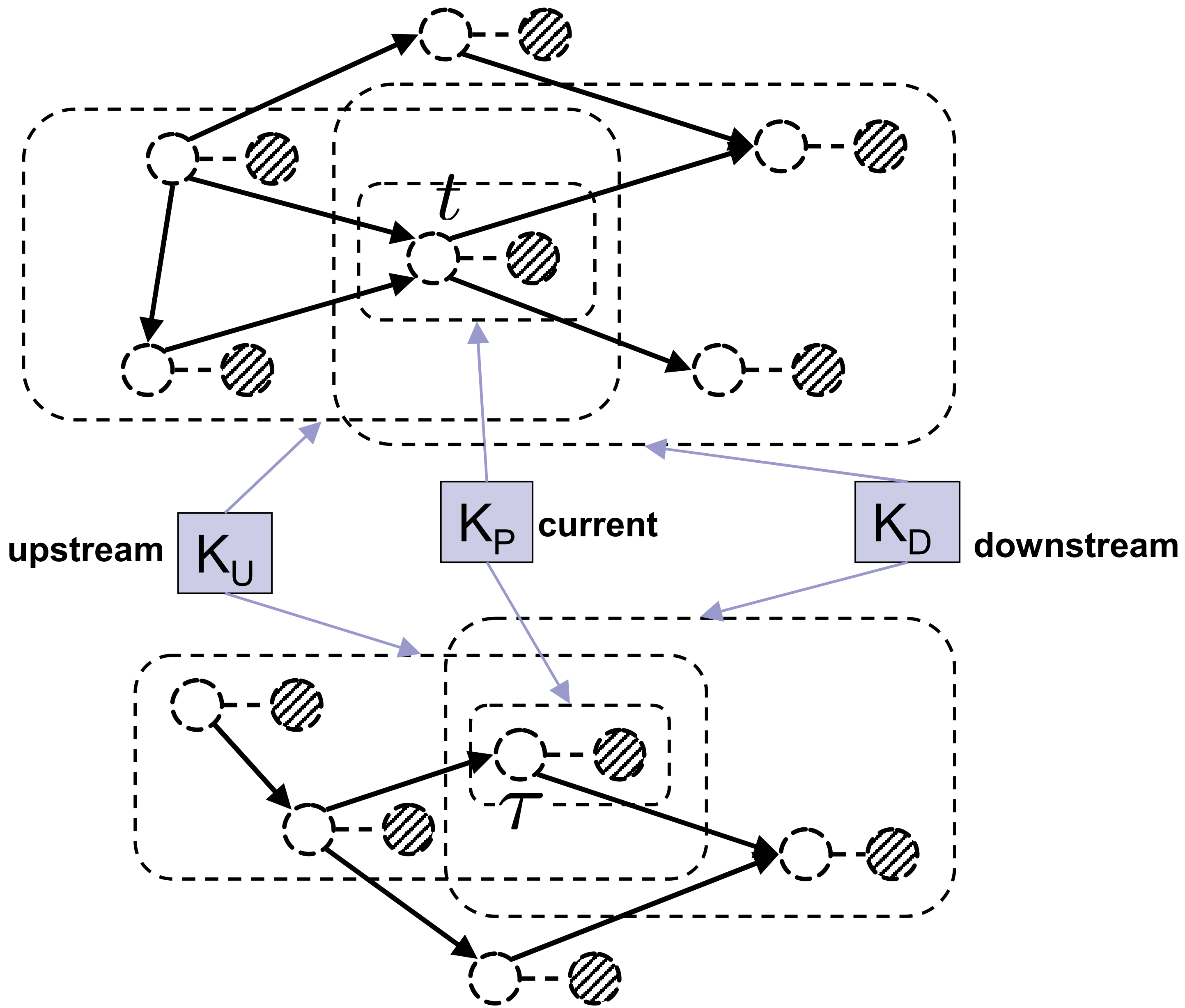


Kernel Decomposition (Ordered Trees)

- **Analogous to Inside-Outside algorithm for probabilistic CFG**



Kernel Decomposition (DAG)



Experiments: Named Entity Recognition

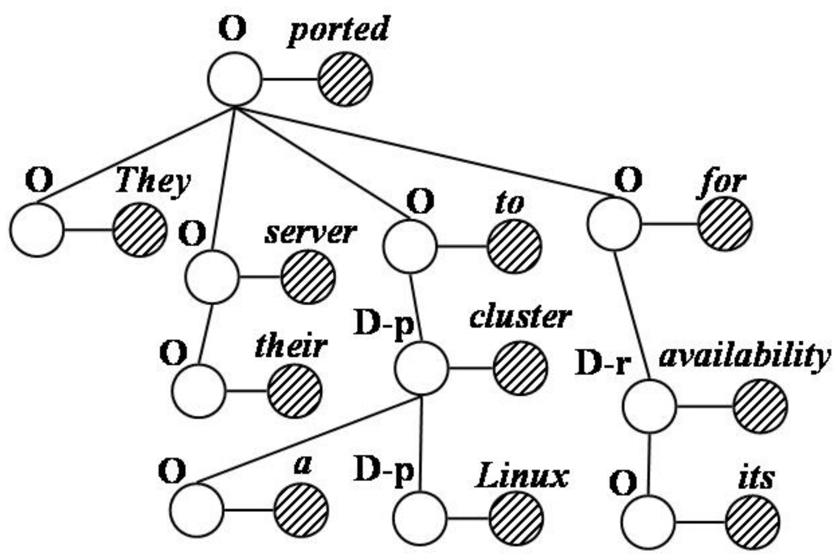
- 300 sentences (8,541 tokens)
  - Sequence labeling problem
  - Special Session of CoNLL2002 on NER
  - 9 labels to indicate *person name*, *organization name*, *place*, …
- Word Features (S2 feature in [Altun et al. 2003])
  - Word
  - Spelling Features
    - prefix and suffix
    - upper/lower case
    - contains dot
    - …

  with 2 degree polynomial kernel
- Comparison of two methods
  - HM-Perceptron
    - window size = 3
  - Marginalized labeling perceptron with sequence kernel

Experiments: Product Usage Information Extraction

- 184 sentences (3,570 tokens)
  - a sales log written by sales representatives (written in Japanese)
  - 12 labels to indicate *product name*, *company name*, *reason*,…
- Word Features
  - word
  - part-of-speech,
  - base form
  - character type features
    alphabet, numbers, Hiragana, Katakana

  with 2 degree polynomial kernel



- Comparison of 3 methods
  - HM-Perceptron (sequence data)
    - window size = 3
  - Marginalized labeling perceptron with sequence kernel (sequence data)
  - Marginalized labeling perceptron with tree kernel (ordered tree data)
    - Parse trees are constructed by Japanese Statistical Parser [Kanayama et al. 2000]

Results (3-fold cross validation)

- Uniform distributions as the priors for marginalized kernels
- Named entity recognition

	Accuracy	Precision	Recall	F1
HM-Perceptron	82.9% (7.4)	21.8% (9.0)	15.6% (4.5)	17.2 (4.0)
Sequence Kernel	**88.4**% (3.9)	**52.3**% (17.5)	**19.3**% (2.2)	**27.9** (5.1)

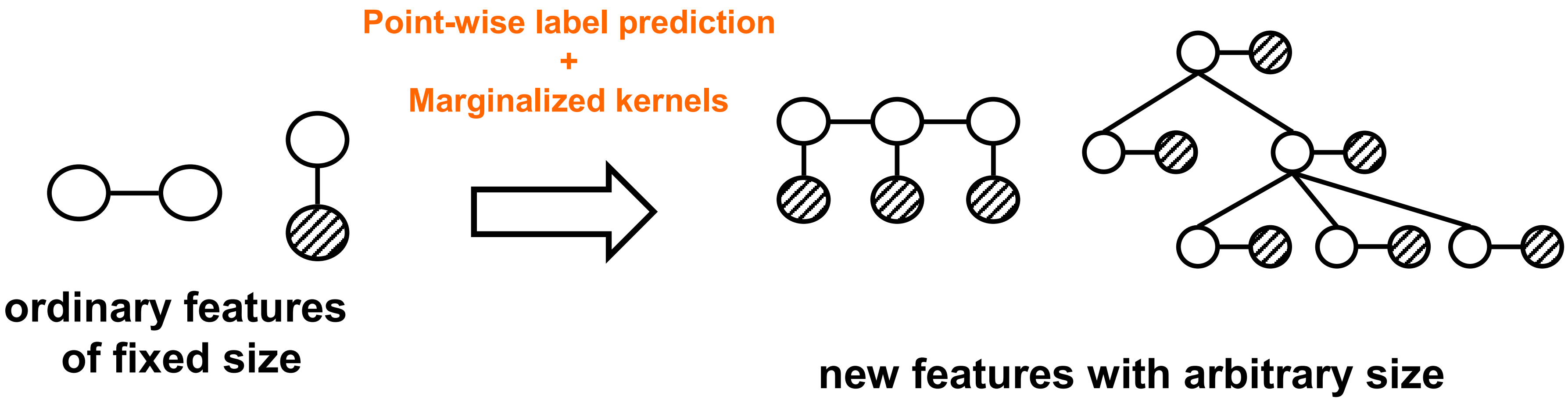
- Product usage information extraction

	Accuracy	Precision	Recall	F1
HM-Perceptron	87.7%(2.0)	40.0%(16.5)	29.0%(11.4)	30.7(9.3)
Sequence Kernel	88.4% (1.4)	45.7% (10.0)	**35.2**% (17.7)	36.7(5.8)
Tree Kernel	**89.8**% (1.2)	**51.5**% (5.2)	32.3% (17.4)	**37.9** (12.1)

(standard deviation)

Conclusion

- Marginalized labeling perceptron
  - Labeling learning algorithm for general structured data
    - Sequences, trees, graphs, …
  - Can handle features with arbitrarily many hidden variables by using
    - Point-wise label prediction
    - Marginalized kernels



Future Work

- Large margin classifiers (e.g. HM-SVM)
- More complex priors (e.g. CRF, MEMM)