Structured Learning

Hidden Markov Support Vector Machines for NLP tasks

Giuseppe Castellucci
castellucci@ing.uniroma2.it
Web Mining & Retrieval a.a. 2013/2014

Outline

- Structured Learning
- SVM-HMM
- Task modeling examples
  - Part of Speech tagging
  - Named Entity Recognition and Classification

Structured Learning

- Learning algorithms so far in the course
  - Classification of "simple" outputs
- Structured Learning
  - Classification of "complex" outputs
  - Such as *sequences* or *trees*
- In general,
  - Learn dependencies between *arbitrary* input and *arbitrary* outputs

A Structured Learning framework

- Learn a **w**-parameterized function $f(\mathbf{x}) = \arg\max_{\mathbf{y}\in\mathcal{Y}} F(\mathbf{x}, \mathbf{y}; \mathbf{w})$
- Where F(**x**,**y**;**w**) is linear in some combined feature representation of inputs and outputs Φ

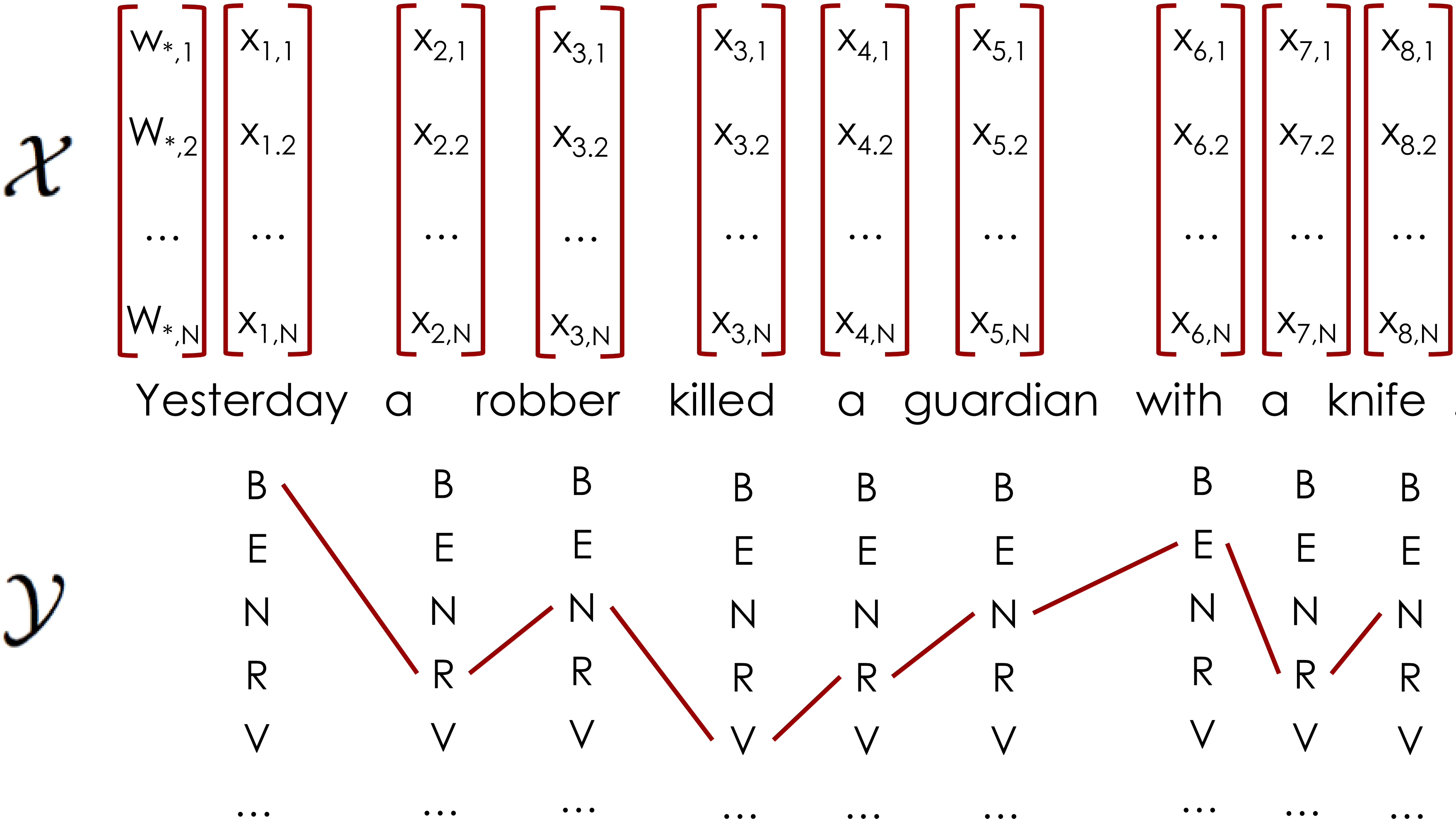
  $F(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle$

- In particular, Φ(**x**,**y**) is responsible of extracting features jointly from input-output pairs
  - Dependency between **x** and **y** can be fully explained only by jointly looking at some property of **x** and **y**
  - Even more true if **y** has an internal structure

SVM-HMM

- Learn a function whose
  - Input is a **sequence** of observation
  - Output is a **sequence** of labels
- Sequence related problems in NLP
  - Part-Of-Speech tagging
  - Named-Entity Recognition and Classification
  - Chunking
  - Semantic Role Labeling
- Why?
  - Generative models
  - Discriminative models

SVM-HMM: the idea



SVM-HMM in Structured Learning

- Learn a discriminative model isomorphic to a *k*-order Hidden Markov Model

  Input: feature vectors $\mathbf{x} = (x_1 \dots x_l) \in \mathcal{X}$
  Output: label sequence $\mathbf{y} = (y_1 \dots y_l) \in \mathcal{Y}$

- In SVM-HMM $\Phi(\mathbf{x},\mathbf{y})$ represents
  - interaction between observations and classes
    - **Emissions** in HMM terminology
  - interaction between adjacent classes
    - **Transitions** in HMM terminology

SVM-HMM classification

$$y = \arg\max_y \{ \sum_{i=1\ldots l} [\sum_{j=1\ldots k} (x_i \cdot w_{y_{i-j} \ldots y_i}) + \Phi_{tr}(y_{i-j}, \ldots, y_i) \cdot w_{tr}] \}$$

Given a story of lenght *k*

Emissions

Transitions

- The Cutting-Plane algorithm is applied to estimate ***w***
- The *Viterbi* algorithm is used to output the best sequence explaining an observation

Sequence Labeling with SVM-HMM

- SVM-HMM represents both
  - Generative models (Hidden Markov Model)
  - Discriminative models (Support Vector Machine)
- In NLP
  - Treat a sentence as a sequence
  - Ideal to take into account contextual information
  - To find the best solution for the entire sequence
- How to model NLP related problems?
  - Two examples: POS Tagging and NERC

Part-of-Speech tagging

- Task: Assign to each token in a sentence the correct grammatical category
- POS tagging can be modeled as a sequential tagging task
  - Linguistic information can be acquired by annotated examples
- We could classify each word without contextual information, i.e. ignoring other words in the sentence
  - It can work for not ambiguous cases: "the" "often"
  - … but the context is *crucial* to classify a word like "run"

Modeling

- An HMM model:
  - The sentence is a **SEQUENCE**
  - Words (represented through a set of features) are our **OBSERVATIONS**
  - HMM **STATES** are mapped into POS tags
  - The **transition probability** is estimated from the training set
  - SVM classifier are used to estimate the **emission probability**
  - The solution is estimated by applying the **Viterbi** algorithm

Feature Engineering

- The better feature representation the better will be the performance
  - Feature engineering (for each token)
    - Contextual (*k* words before and after the target word using *Padding*)
    - The word *prefix* and *suffix*
    - Boolean indicators of: IsTheFirstWord, ContainsNumbers, StartsWithCapital,ContainsSymbols,isAllNumbers
    - Dictionary Information, e.g. morphology (if available)
  - Feature post-processing
    - Normalization
    - Do not mix features!
  - E.g. *Ieri Giuseppe Castellucci era al parco*.

BEGIN_1 BEGIN_0 Ie ri Ieri FirstWord NotContainsNumbers StartsCapital NotContainsSymbol NotAllNumbers
BEGIN_0 Ieri Gi pe Giuseppe NotFirstWord NotContainsNumbers StartsCapital NotContainSymbol NotAllNumbers
.
.
.

Setup	System	TA	UWTA
Open	RevNLT	**97.68**	95.21
	Best System1	97.03	**95.30**
Close	RevNLT	**96.93**	93.39
	Best System2	96.91	**93.81**

Results

- Evaluation
  - Token based accuracy
- Italian performances on the EVALITA 2009 task
  - EVALITA is a campaign to evaluate systems on the Italian language
- Experimental setup
  - Training dataset: 108874 words in 3719 sentences
  - Development dataset: 5021 words in 147 sentences
  - Test dataset: 5066 words in 147 sentences
  - In development and test 17% of unknown words
  - 37 classes
  - Open and Close evaluation refer to the possibility to use external resources

Named Entity Recognition and Classification

- Task: Find and classify entities in a sentence
  - Classify w.r.t. predefined classes, as PERSON, LOCATION, ORGANIZATION, etc…
- We can model it as a labeling task
  - Linguistic information can be acquired by annotated examples
- Again, assign to each token in a sentence a specific class

Modeling

- An HMM model:
  - The sentence is a **SEQUENCE**
  - Words (represented through a set of features) are **OBSERVATIONS**
  - HMM **STATES** are mapped into Named Entities, e.g. PER,LOC,X
  - **Transition probabilities** estimated from the training set
  - SVM classifiers used to estimate the **emission probability**
  - The solution computed by the **Viterbi** algorithm

Multi-word entities

- Named Entities are also multi-word expressions
  - Yesterday **Giuseppe Castellucci** was happy.
- How to manage multi-word expression in SVM-HMM?
  - First solution is to label each token with a class
  - Yesterday/X Giuseppe/PER Castellucci/PER was/X happy/X /.
- What if an entity directly follows an entity of the same class?
  - Ideas?

IOB notation

- Discriminate from the **B**egin, the **I**nside or the **O**utside of an entity for each class
  - Yesterday/O Giuseppe/B-PER Castellucci/I-PER was/O happy/O ./O
- If entities are consecutive
  - discriminate with B-* tags
- Two possible approaches
  - Cascade of two classifiers (locate entities and then classify w.r.t. classes)
  - A single classifier (jointly classifies the boundaries and the classes)

Feature Engineering

- Same as Part Of Speech tagging + the Part-Of-Speech of a token
  - For each token,
    - Contextual (k words before and after the target word)
    - The word prefix and suffix
    - Boolean indicators of: IsTheFirstWord, ContainsNumbers, StartsWithCapital,ContainsSymbols,isAllNumbers
    - Dictionary Information, e.g. morphology information
    - Part-Of-Speech
  - Again, feature post-processing
    - Normalization
    - Do not mix features!

Results

- Evaluation
  - Entity-based Precision,Recall and F1
- Experimental setup
  - Evalita 2009 NER task
  - Training dataset: 11410 entities in 11227 sentences
  - Test dataset: 4966 entities in 4136 sentences
  - 4 classes: Person, Location, Organization and GeoPoliticalEntity
- Accuracy: ≈76 F1. Best in Evalita ≈82 F1

How to use SVM^{HMM}

- Download:
  - http://download.joachims.org/svm_hmm/current/svm_hmm.tar.gz
- Compile (make)
- Learn: svm_hmm_learn -c <C> **--t <ORDER_T> -e 0.1 -e 1** training_input.dat modelfile.dat
  - -c: Typical SVM parameter C trading-off slack vs. magnitude of the weight-vector (1, 10, 100, 10^3, 10^4 depends by the training set size).
  - --t: Order of dependencies of transitions in HMM (1,2 o 3)
- Classify: svm_hmm_classify test_input.dat modelfile.dat classify.tags

SVM^{HMM} input

class Sent_id Feature vector

Comment

...
4 qid:1 1:1 2:1 51:1 247:1 2675:1 # four
12 qid:1 58:1 84:1 197:1 250:1 433:1 1145:1 2677:1 # score
3 qid:1 8:1 83:1 88:1 202:1 363:1 364:1 438:1 1147:1 # and
4 qid:1 16:1 47:1 87:1 135:1 197:1 365:1 366:1 # seven
15 qid:1 30:1 49:1 142:1 197:1 202:1 387:1 # years
8 qid:1 39:1 83:1 202:1 267:1 392:1 # ago
20 qid:1 83:1 87:1 247:1 269:1 2675:1 2676:1 # our
.....
21 qid:2 5:1 83:1 576:1 923:1 1379:1 1469:1 # now
19 qid:2 23:1 84:1 87:1 577:1 926:1 1383:1 1470:1 # we
30 qid:2 26:1 83:1 84:1 88:1 433:1 578:1 627:1 # are
29 qid:2 7:1 8:1 9:1 87:1 88:1 438:1 628:1 1077:1 3377:1 # engaged
8 qid:2 15:1 16:1 17:1 23:1 47:1 185:1 1082:1 3381:1 # in
...
8 qid:3 23:1 47:1 48:1 87:1 219:1 1621:1 # on
7 qid:3 3:1 26:1 49:1 50:1 459:1 # a
9 qid:3 5:1 197:1 217:1 460:1 519:1 1535:1 1536:1 1537:1 # great
12 qid:3 8:1 109:1 202:1 219:1 522:1 531:1 1538:1 1539:1 1540:1 # battlefield

Sparse notation

References

- Altun et Al. *Hidden Markov Support Vector Machines*, ICML 2003
- I. Tsochantaridis, T. Hofmann, T. Joachims, Y. Altun. *Support Vector Machine Learning for Interdependent and Structured Output Spaces.* International Conference on Machine Learning (ICML), 2004.
- I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, Large Margin Methods for Structured and Interdependent Output Variables, Journal of Machine Learning Research (JMLR), 6(Sep):1453-1484, 2005.
- T. Joachims, T. Finley, Chun-Nam Yu, Cutting-Plane Training of Structural SVMs, Machine Learning Journal, 77(1):27-59, 2009.
- Danilo Croce, Giuseppe Castellucci, Emanuele Bastianelli. Structured Learning for Semantic Role Labeling, Intelligenza Artificiale, 6(2),163-176, 2012
- http://www.cs.cornell.edu/people/tj/svm_light/svm_struct.html
- http://www.cs.cornell.edu/people/tj/svm_light/svm_hmm.html