MDI 341: Structured output prediction

1 Introduction

Florence d'Alché

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

Patient records

- medical images
- biomedical signals
- results of medical exams
- symptoms measured by various sensors
- genotype
- transcriptomics

Client data

- documents, reports
- structured forms
- node in a social network

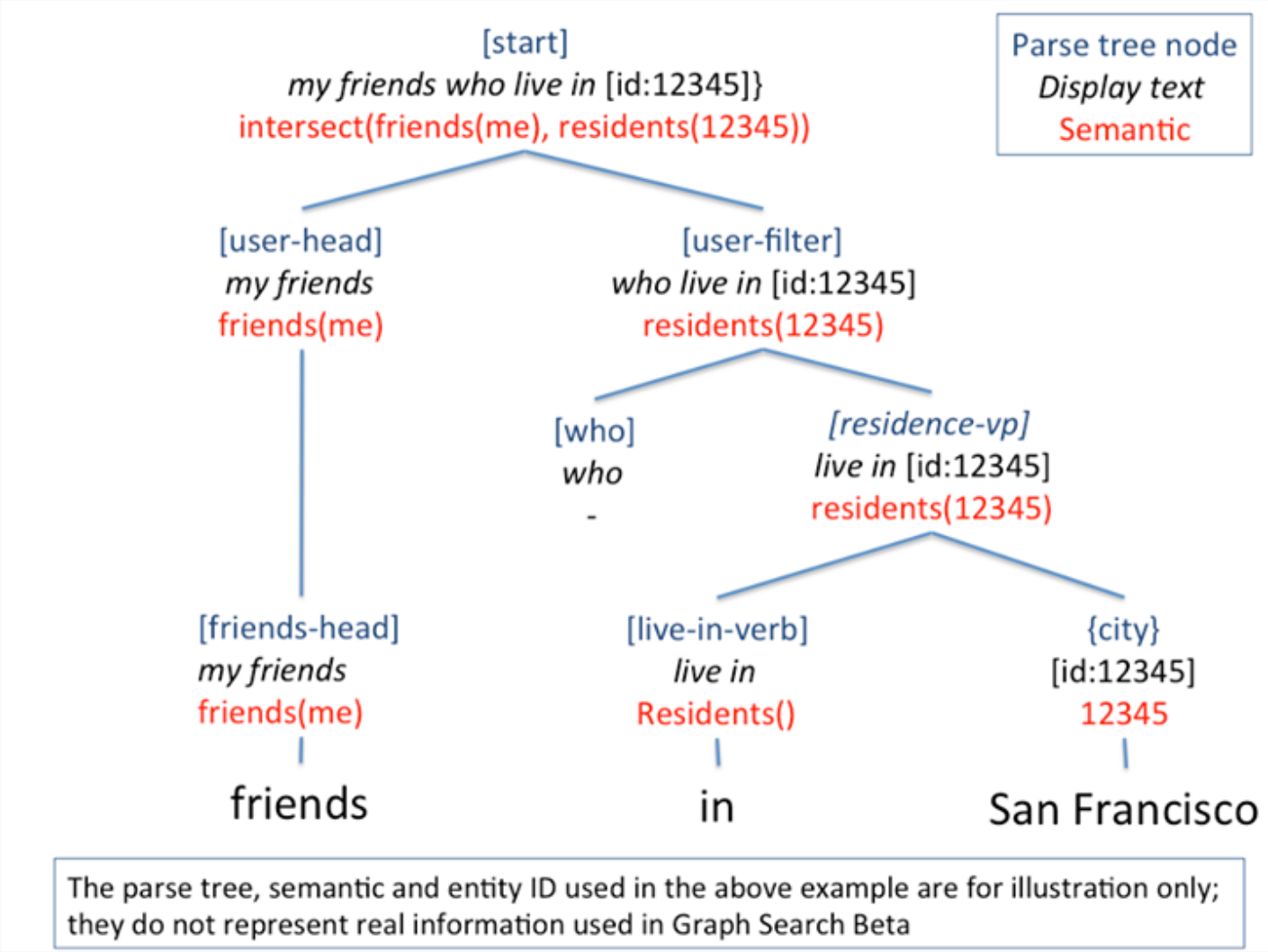
Example of structured data

```
{
  _id: <ObjectId1>,
  username: "123xyz",
  contact: {
             phone: "123-456-7890",
             email: "xyz@example.com"
           },
  access: {
            level: 5,
            group: "dev"
          }
}
```

Embedded sub-document

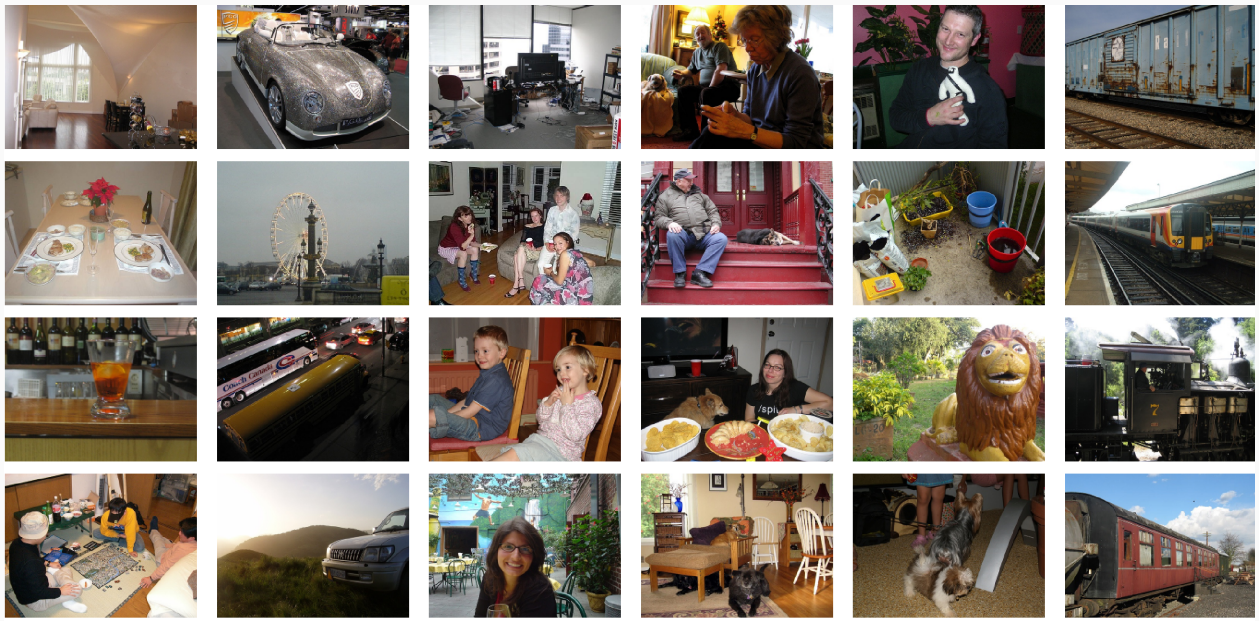
Embedded sub-document

Example of structured data



The parse tree, semantic and entity ID used in the above example are for illustration only; they do not represent real information used in Graph Search Beta

Unstructured data or ... implicitely structured data



Unstructured data or ... implicitely structured data

This method is naturally geared toward document-pivoted TC, since ranking the training documents for their similarity with the test document can be done once for all categories. For category-pivoted TC, one would need to store the document ranks for each test document, which is obviously clumsy; DPC is thus *de facto* the only reasonable way to use k-NN.

A number of different experiments (see Section 7.3) have shown k-NN to be quite effective. However, its most important drawback is its inefficiency at classification time: while, for example, with a linear classifier only a dot product needs to be computed to classify a test document, k-NN requires the entire training set to be ranked for similarity with the test document, which is much more expensive. This is a drawback of "lazy" learning methods, since they do not have a true training phase and thus defer all the computation to classification time.

6.9.1. Other Example-Based Techniques. Various example-based techniques have been used in the TC literature. For example, Cohen and Hirsh [1998] implemented an example-based classifier by extending standard relational DBMS technology with "similarity-based soft joins." In their WHIRL system they used the scoring function

$$CSV_i(d_j) = 1 - \prod_{d_z \in Tr_k(d_j)} (1 - RSV(d_j, d_z))^{\|\breve{\Phi}(d_z, c_i)\|}$$

as an alternative to (9), obtaining a small but statistically significant improvement over a version of WHIRL using (9). In their experiments this technique outperformed a number of other classifiers, such as a C4.5 decision tree classifier and the RIPPER CNF rule-based classifier.

A variant of the basic k-NN approach was proposed by Galavotti et al. [2000], who reinterpreted (9) by redefining

The difference from the original k-NN approach is that if a training document d_z similar to the test document d_j does not belong to c_i, this information is not discarded but weights *negatively* in the decision to classify d_j under c_i.

A combination of profile- and example-based methods was presented in Lam and Ho [1998]. In this work a k-NN system was fed *generalized instances* (GIs) in place of training documents. This approach may be seen as the result of

—clustering the training set, thus obtaining a set of clusters $K_i = \{k_{i1}, \ldots, k_{i|K_i|}\}$;

—building a profile $G(k_{iz})$ ("generalized instance") from the documents belonging to cluster k_{iz} by means of some algorithm for learning linear classifiers (e.g., Rocchio, WIDROW-HOFF);

—applying k-NN with profiles in place of training documents, that is, computing

$$\begin{aligned} CSV_i(d_j) &\stackrel{def}{=} \sum_{k_{iz} \in K_i} RSV(d_j, G(k_{iz})) \cdot \\ &\quad \frac{|\{d_j \in k_{iz} \mid \breve{\Phi}(d_j, c_i) = T\}|}{|\{d_j \in k_{iz}\}|} \cdot \\ &\quad \frac{|\{d_j \in k_{iz}\}|}{|Tr|} \\ &= \sum_{k_{iz} \in K_i} RSV(d_j, G(k_{iz})) \cdot \\ &\quad \frac{|\{d_j \in k_{iz} \mid \breve{\Phi}(d_j, c_i) = T\}|}{|Tr|}, \end{aligned} \tag{10}$$

where $\frac{|\{d_j \in k_{iz} \mid \breve{\Phi}(d_j, c_i) = T\}|}{|\{d_j \in k_{iz}\}|}$ represents the "degree" to which $G(k_{iz})$ is a positive instance of c_i, and $\frac{|\{d_j \in k_{iz}\}|}{|Tr|}$ represents its weight within the entire process.

This exploits the superior effectiveness (see Figure 3) of k-NN over linear classifiers while at the same time avoiding the sensitivity of k-NN to the presence of

Structured data

Definition
Data is said to be structured if it consists in several parts, and if, to describe fully the data, one need to describe the way the parts interact together.

Remark: *the relationship between parts can be fixed and known, and structured objects can be of fixed size (think about a form to fill) OR the kind of relationship can be known but the number of parts is not constant (think about a sentence).*

Supervised learning with structured data

- Prediction from structured data (input X is structured): not this course
- Structured output prediction (output Y is structured): this course

Classic supervised machine learning / structured output learning

Classic machine learning:

$$f : \mathcal{X} \to \mathbb{R} \tag{1}$$

Structured output learning :

$$f : \mathcal{X} \to \mathcal{Y} \tag{2}$$

$\mathcal{Y}$: set of structured objects

Sequence labeling

- *INPUT:* a sequence of tokens
- *OUTPUT:* a sequence of labels (one par token)

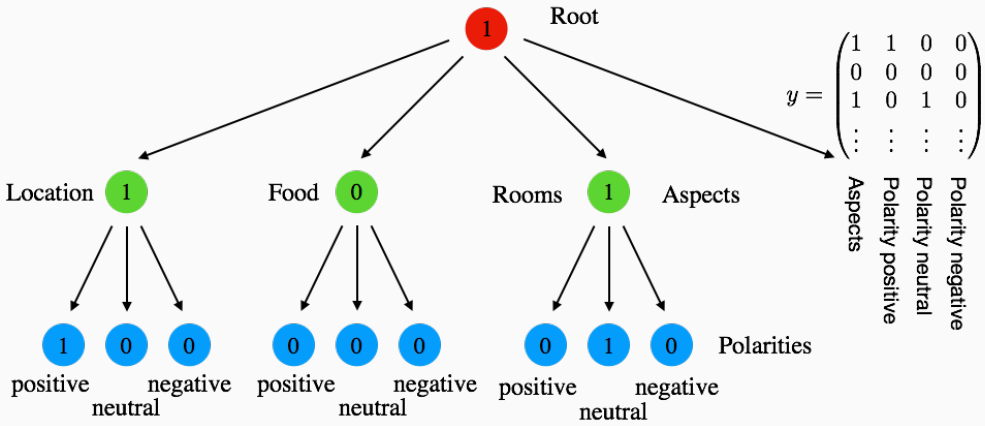
Example: Part-of-Speech tagging
Given a sentence, find parts-of-speech of all words.

The	Fed	raises	interest	rates
Determiner	Noun	Verb	Noun	Noun
Other possible tags in different contexts,	Verb (I *fed* the dog)		Verb (Poems don't *interest* me)	Verb (He rates movies online)

Opinion prediction

- Setup : we want to predict the labels of a known target graph structure (encoded by a tree).
- INPUT: hotel review
- OUTPUT : sentence level opinion annotations

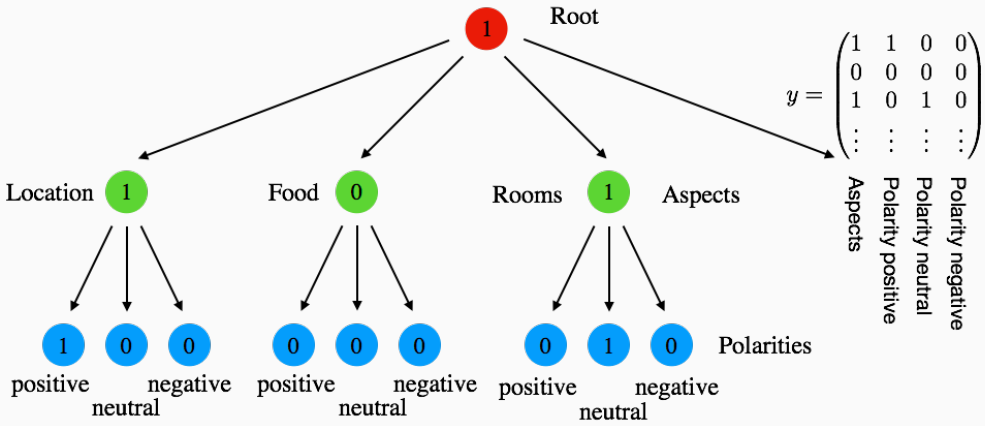
```
The room was ok,
nothing special, still
a perfect choice to
quickly join the main
places.
```



Opinion prediction

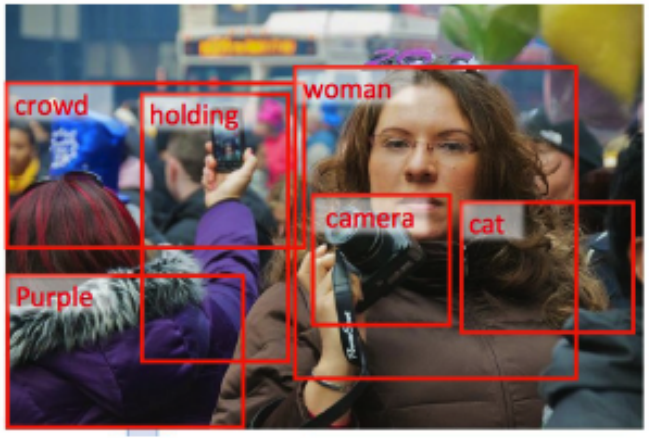
- Setup : we want to predict the labels of a known target graph structure (encoded by a tree).
- INPUT: hotel review
- OUTPUT : sentence level opinion annotations

The room was OK, nothing special, still a perfect choice to quickly join the main places.



Automatic image captioning

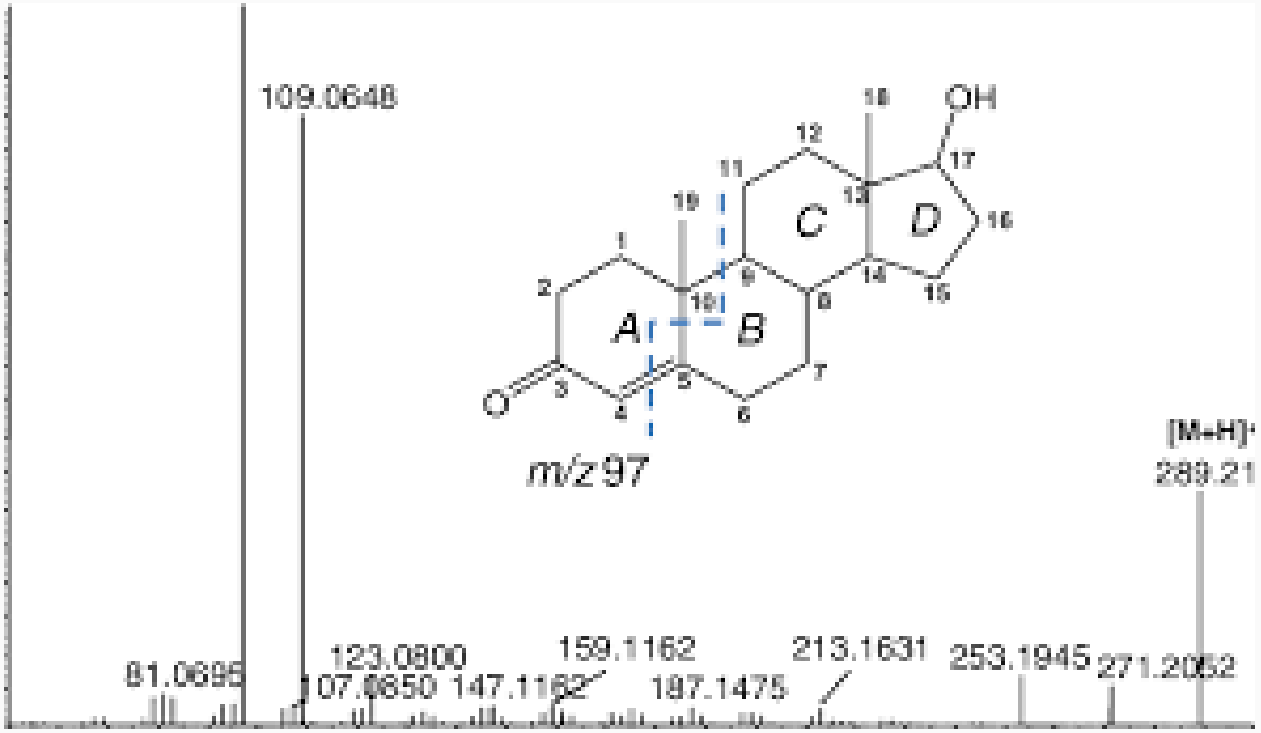
- INPUT: a raw image
- OUTPUT : a sentence (caption)



A woman holding a camera in the crowd

From mass spectrometry to metabolite

- INPUT: mass spectra
- OUTPUT : metabolite(s)



Difficulties of structured output prediction

- Outputs do not have the same size
- Need to compose several elements to make a prediction
- Interdependent outputs

Structured output prediction

Learning problem:
Given $\{(x_i, y_i), i = 1, \ldots n\}$ a i.i.d. sample drawn from a fixed but unknown joint probability distribution $P(X, Y)$, solve:

$$\min_f \mathbb{E}[\Delta(Y, f(X))]$$

Inherent difficulties of Structured output prediction

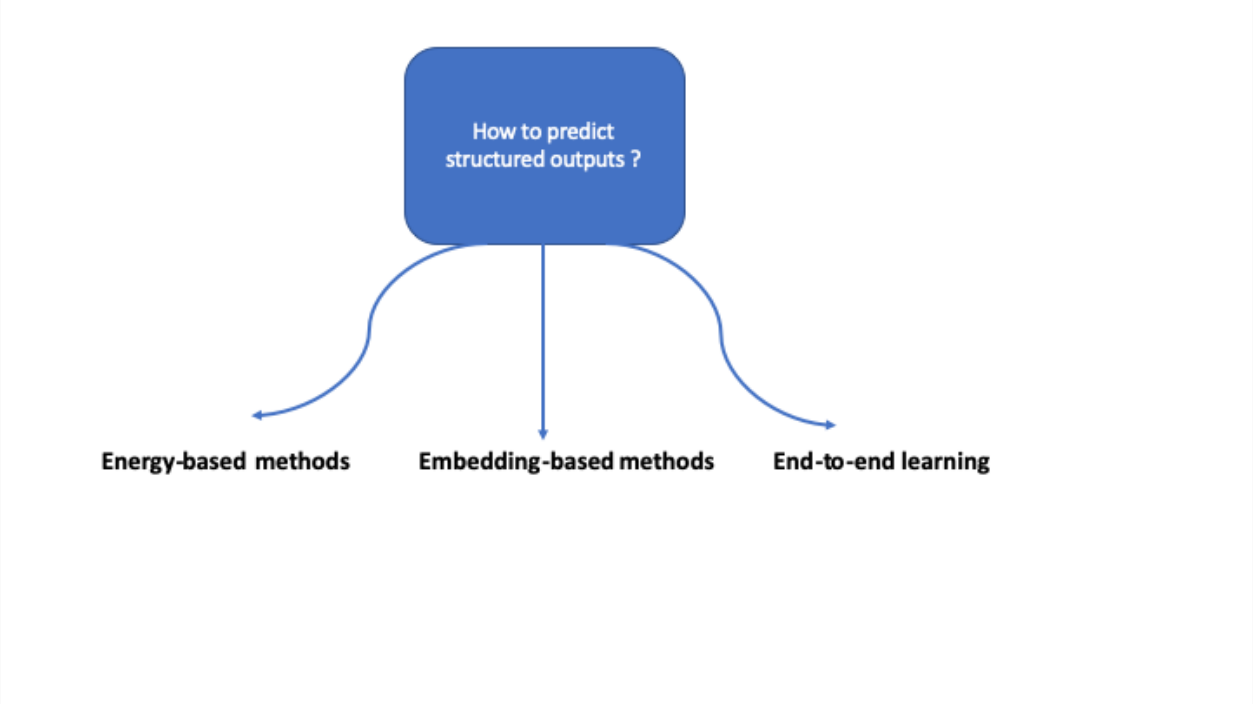
We will find the same usual suspects as in classic supervised learning

- How to represent data : features ?
- Which hypothesis space ?
- What loss ?
- How to solve the underlying optimization problem ?
- What kind of evaluation metrics ?

Structured output prediction: the main difficulty

The function f to be learned is from $\mathcal{X} \to \mathcal{Y}$ where $\mathcal{Y}$ is more complex than $\mathbb{R}$.

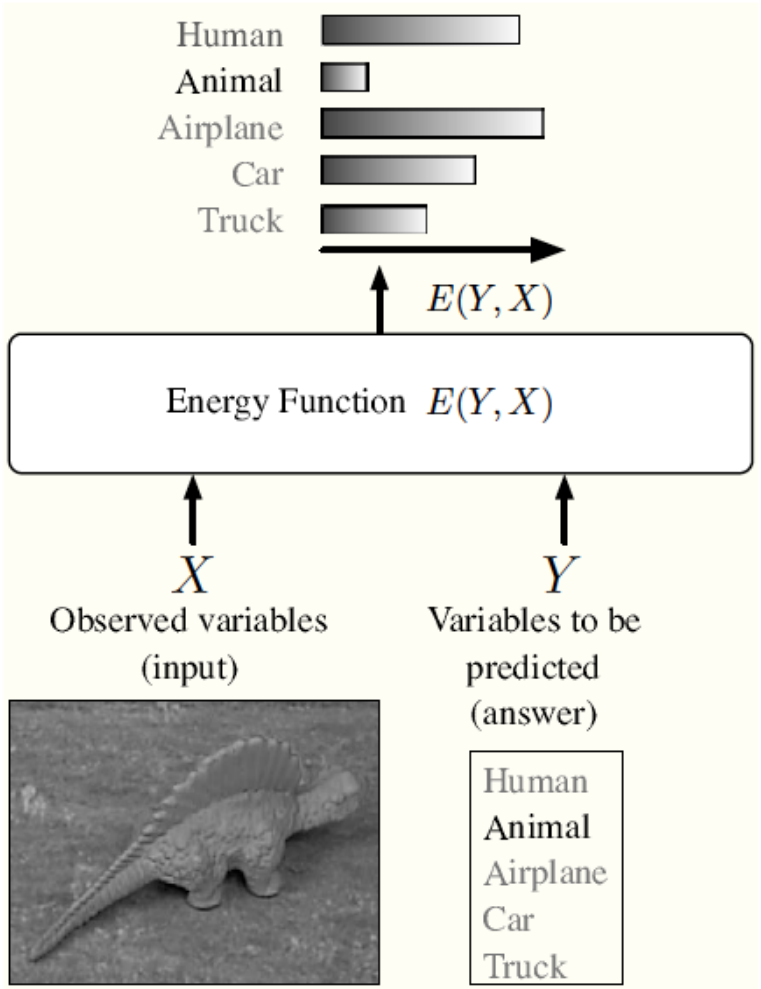
The discrete nature of $\mathcal{Y}$ and the absence of linear algebra on $\mathcal{Y}$ have pushed researchers to somehow **relax** this complex combinatorial problem into a continuous problem.



Energy-based learning

Idea: Capture dependencies between input and output variables by some energy function (or compatibility score).
Inference: setting the values of some variables (inputs) and finding the values of remaining ones by minimizing the associated energy.
Learning : finding an energy function that associates low energies to correct values of the remaining and high energy to incorrect ones.

Energy-based learning



See Le Cun et al. 2006 (tutorial).

Score-based / energy-based learning

In this course we adopt the vocabulary of compatibility score rather than senergy term:

Definition
Define g a score function that takes as inputs feature pairs on x and y:

$$f(x) = \arg\max_{y \in \mathcal{Y}} g(x, y) \tag{3}$$

$$g(x, y) \in \mathbb{R} \tag{4}$$

$$f(x) \in \mathcal{Y} \tag{5}$$

Learn f to minimize an expected risk $\mathbb{E}[\Delta(y, f(x))]$

$\Delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$: prediction losss that takes into account the structured nature of $\mathcal{Y}$'s objects.

Models in energy-based methods in general

Define g a **compatibility** score ($g(x, y) = -E(x, y)$) that takes as inputs feature pairs on x and y:

$$f(x) = \arg\max_{y \in \mathcal{Y}} g(x, y) \tag{6}$$

$$f(x) \in \mathcal{Y} \tag{7}$$

Examples of g:

$$\begin{aligned} g(x, y) &= w^T \phi(x, y) \\ g(x, y) &= P(Y = y|x) \\ g(x, y) &= -\|\phi(y) - h(x)\|^2 \\ g(x, y) &= \langle \phi(y), h(x) \rangle, \end{aligned}$$

where $h : \mathcal{X} \to \mathcal{F}_y$, where $\mathcal{F}_y$ is some Hilbert space.

Structured output learning

Learning problem:
Given $\{(x_i, y_i), i = 1, \ldots n\}$ a i.i.d. sample drawn from a fixed but unknown joint probability distribution $P(X, Y)$, solve:

$$\min_f \mathbb{E}[\Delta(Y, f(X))]$$

is converted into: Given $\{(x_i, y_i), i = 1, \ldots n\}$ a i.i.d. sample drawn from a fixed but unknown joint probability distribution $P(X, Y)$, solve:

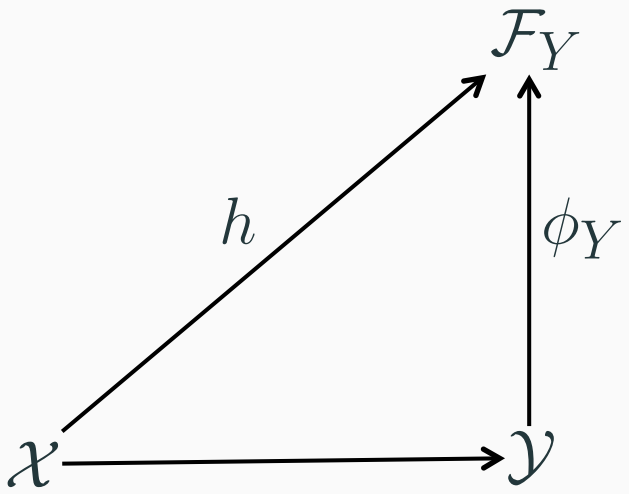
$$\min_g \mathbb{E}[\Delta(Y, \arg\min_y g(X, y)]$$

Prediction in energy-based methods

Difficulties

- Prediction itself is expensive
- Very often, the optimization problem to solve is NP-hard
- Learning means to take into account the cost of prediction

A second angle to structured output prediction



Regression with output embedding, surrrogate ergerssion
Take specifically $g(x, y) = -\|\phi(y) - h(x)\|^2$

A two-stage approach:

1. Learn h to minimize $\mathbb{E}[\|\phi(Y) - h(X)\|^2]$, i.e. learn h independently from the decoding phase. In practise,
   $\hat{h} = \arg\min \frac{1}{2n} \sum_i \|\phi(y_i) - h(x_i)\|^2 + \lambda\Omega(h)$
2. Solve the pre-image/decoding problem:

$$f(x) \quad = \quad \arg\min_{y \in \mathcal{Y}} \|\phi(y) - h(x)\|^2$$

Surrogate Regression

Learning is fast (cost of regression)

Difficulties

Prediction requires to solve a pre-image problem except in some cases like structured multiple outputs, link prediction ...

Predicted outputs are approximated

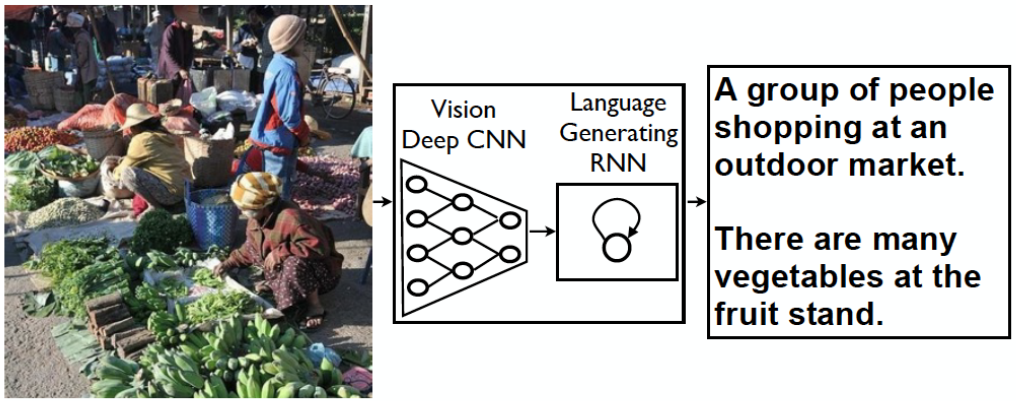
A third angle to structured output prediction: end-to-end learning

Build a model of the form:

$$f(x) = d \circ g(x)$$

Choose a way to represent elements of $\mathcal{Y}$ as vectors and define a loss $L(y, f(x))$ differentiable in its second variable.
d decodes $g(x)$ into a vectorial representation of elements of $\mathcal{Y}$.

Example of end-to-end learning for image captioning



Vinyals et al. 2015.

Important points

- (Nearly) Any predictive model can be used in the 3 approaches, including neural networks
- so the approaches only and mainly differ by the kind of relaxation that one wants to apply to $\mathcal{Y}$
- "nearly" because the third approach, end-to-end learning usually requires a combination of graphical probabilistic models and or neural networks and is surely **deep**.

We are going to emphasize the first family of approaches, which is the most flexible so far and the most known.