Basic Notions

Let X be a set of all possible programs in some language, and $x \in X$ be a program.

$n, m : X \to \mathbb{N}$, where
$n(x)$ is the total number of elements (e.g. variables) of a program x for which we are interested in inferring properties,
$m(x)$ is the total number of elements with known properties of a program x.

Let

$Labels_U$ denotes all possible values that a predicted property can take,

$Labels_K$ denotes a set of values that a known property can take,

$Labels = Labels_U \cup Labels_K$ denotes the set of all property values.

For a program x, the vector $z^x = \{z_1^x, ..., z_{m(x)}^x\}$ denotes the set of properties that are already known, where $z_i^x \in Labels_K$ for $i = 1, ..., m(x)$.

The notation $y = (y_1, ..., y_{n(x)})$ is used to denote a vector of predicted program properties, where $y \in Y$ and $Y = (Labels_U)^*$ in general.

Problem Definition

Let $D = \{\langle x^{(j)}, y^{(j)} \rangle\}_{j=1}^t$ denote the training data: t programs annotated with corresponding program properties.

Goal: learning a model that captures the conditional probability $Pr(y|x)$.

Prediction (MAP or Maximum a Posteriori query)
Given a new program x, find $y = \arg\max_{y' \in \Omega_x} Pr(y'|x)$,
where $\Omega_x \subseteq Y$ describes the set of possible assignments of properties y' for the program x.

Log-linear Conditional Random Fields (CRFs)

A model for the conditional probability of labels y given observations x is called **(log-linear) conditional random field**, if it is represented as:

$$Pr(y|x) = \frac{1}{Z(x)} \exp(score(y, x)),$$

- the *partition function*

$$Z(x) = \sum_{y \in \Omega_x} exp(score(y, x)),$$

which returns a real number depending only on the program x;

- $$score(y, x) = \sum_{i=1}^{k} w_i f_i(y, x) = w^T f(y, x),$$

  where f is a vector of *feature functions* $f_i : Y \times X \to \mathbb{R}$ and w is a vector of *weights* w_i.

Dependency Network

Let $Rels$ be the set of all element relations.

A multi-graph $G^x = \langle V^x, E^x \rangle$ is called a **dependency network** of the program x if

- $V^x = V_U^x \cup V_K^x$ denotes the set of program elements and consists of elements for which we would like to predict properties V_U^x and elements with known properties V_K^x;
- the set of edges $E^x \subseteq V^x \times V^x \times Rels$ denotes the fact that there is a relationship between two program elements and describes this relationship.

Feature Functions

Let $\{\psi_i\}_{i=1}^k$ be a set of **pairwise feature functions** s.t. $\psi_i : Labels \times Labels \times Rels \to \mathbb{R}$ scores a pair of program properties when they are related with the given relation.

$$\psi_{example}(l_1, l_2, e) = \begin{cases} 1 & \text{if } l_1 = \texttt{i} \text{ and } l_2 = \texttt{step} \text{ and } e = \texttt{L+=R} \\ 0 & \text{otherwise} \end{cases}$$

Let the assignment vector $A = (y, z^x)$ be a concatenation of the unknown properties y and the known properties z^x in x, and the property of the j'th element of vector A is accessed via A_j. Then the **feature function**[1] f_i is defined as:

$$f_i(y, x) = \sum_{\langle a,b,rel \rangle \in E^x} \psi_i\big((y, z^x)_a, (y, z^x)_b, rel\big).$$

Maximum a Posteriori (MAP) Inference in CRFs

$$y = \arg\max_{y' \in \Omega_x} Pr(y'|x)$$

$$\Updownarrow$$

$$y = \arg\max_{y' \in \Omega_x} score(y', x)$$

[1] feature functions are defined independently of the program being queried