On Linear Separability of Sequences and Structures

Alessandro Sperduti

Dipartimento di Matematica Pura ed Applicata

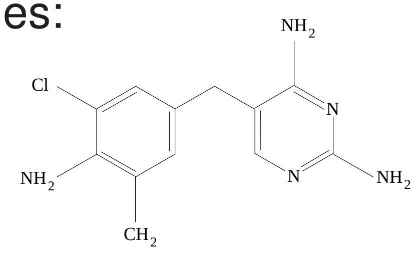
Università di Padova

Structured data

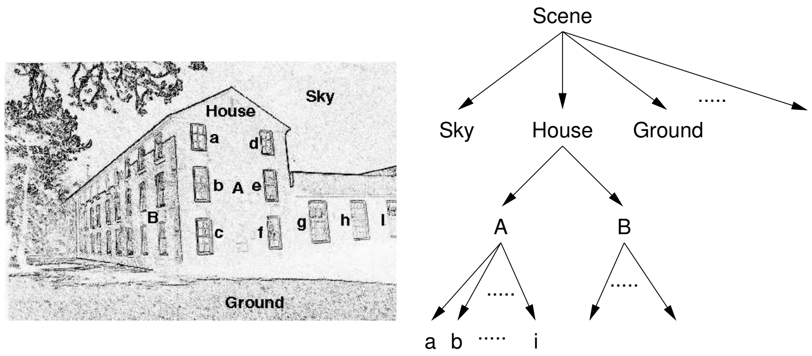
In some learning domains, data cannot be represented with poor data structures such as records

Some examples:

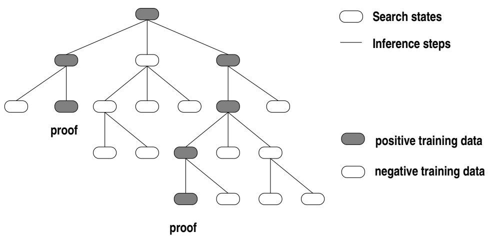
- Chemistry



- Pattern recognition



- Heuristics for searching in AI



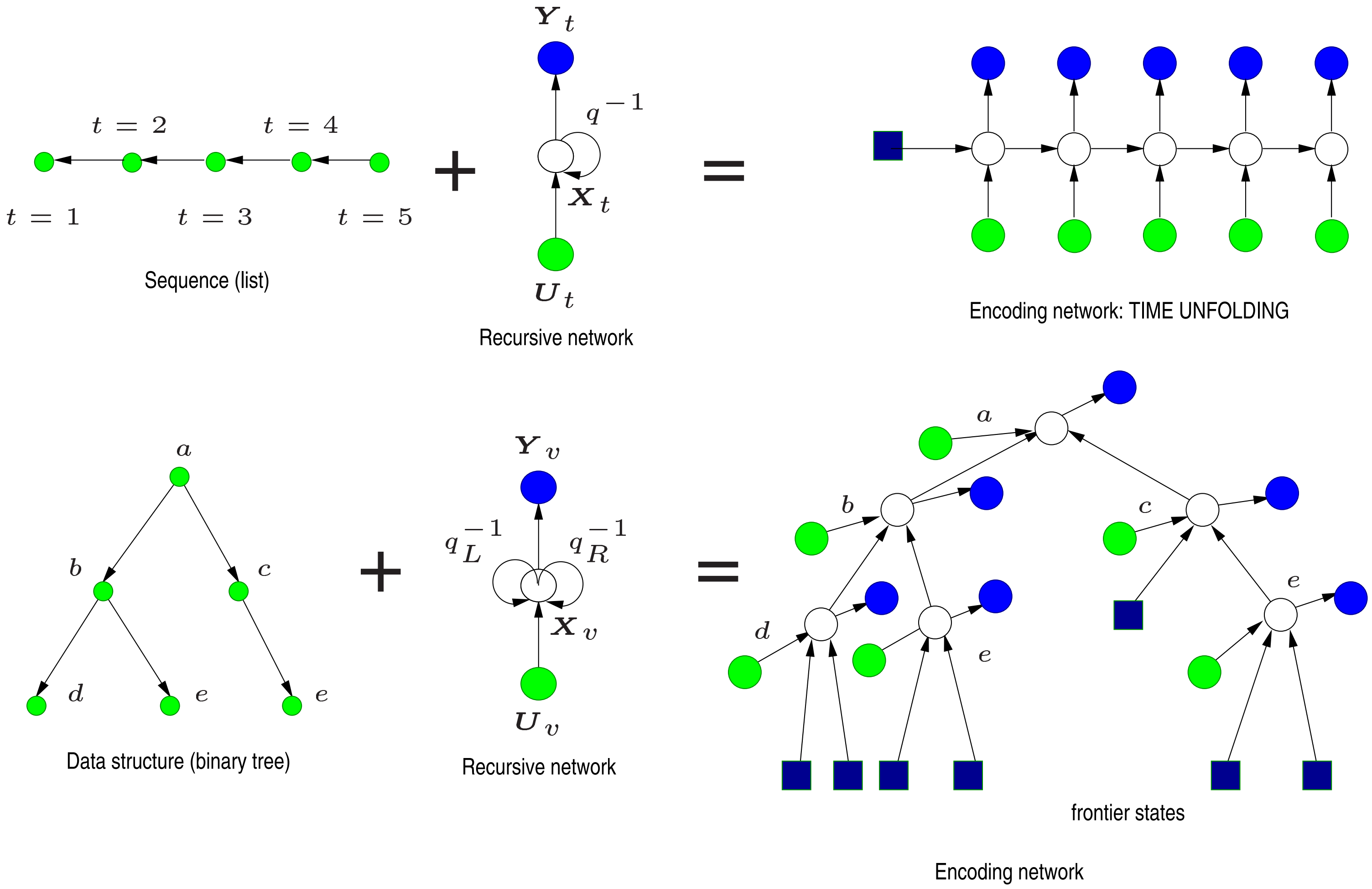
Recursive Artificial Neural Networks

Recursive Neural Networks have been defined to directly deal with structured data

MODEL TYPE

DATA TYPE	Deterministic	Stochastic
STATIC records, fixed-size arrays	*Feedforward ANNs*	*Mixture models*
SEQUENTIAL serially ordered entities	*Recurrent ANNs*	*HMMs, IOHMMs*
STRUCTURAL relations among domain variables	*Recursive ANNs*	*Recursive HMMs*

Data structure + Recursive net = Encoding net

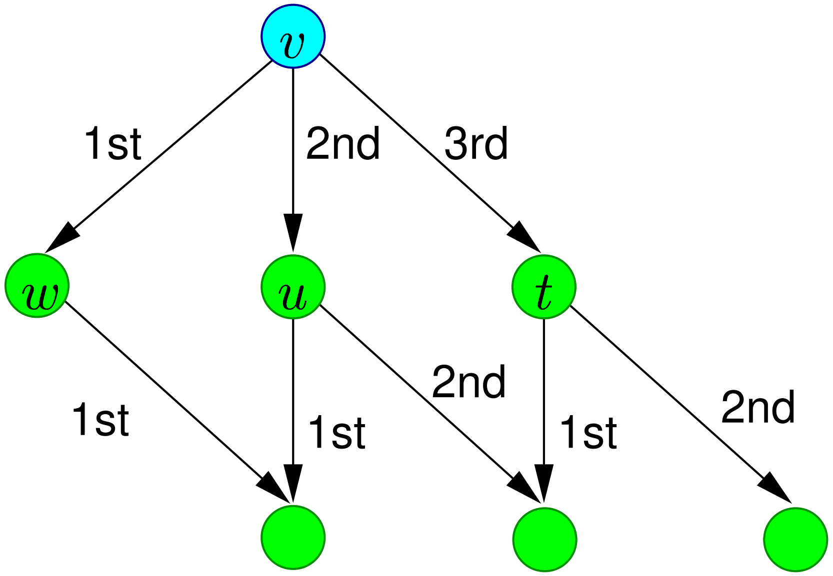


Linear Separability for Sequences and Structures: Motivations

- Historically the concept of linear separability for a set of vectors has been very important since it allowed to identify classification problems which are easy to solve
- Unfortunately this concept cannot be directly applied to sequences and structures due to the higher complexity of data
- The aim of this work is to try to extend linear separability for sequences and structures
- This should allow the characterization of classification in structured domains which are “easy” to solve

Structured Data

Directed ordered acyclic graphs (DOAGs): for each vertex v, a total order $\prec$ is defined on the edges leaving from v.



Moreover:

- vertexes are marked by *labels*, either categorical or numerical (in general, real-valued vectors);
- for each vertex v, outdegree$(v) \leq c$ (bounded outdegree)
- each structure should posses a *supersource* s from where all the other vertexes can be reached

Theoretical Model

Given a node v of a structure $\mathcal{U}$, define the net input of a linear recursive neuron as

$$net(v) = \boldsymbol{W}^t \boldsymbol{u}_v + \sum_{j=1}^{c} \hat{w}_j net(ch_j[v]),$$

where

- $\boldsymbol{u}_v$ is the input label attached to vertex v, and $ch_j[v]$ is the j-th children of v;
- $\boldsymbol{W}$ are the weights associated to the input label;
- $\hat{w}_j$ are the weights on the recursive connections.

The output of the neuron for $\mathcal{U}$ is then computed on the *supersource* s as

$$o(\mathcal{U}) = sgn(net(s)) = \begin{cases} 1 & \text{if } net(s) \geq 0 \\ -1 & \text{otherwise.} \end{cases}$$

Linear Separability for Sequences

Let consider a sequence $\mathcal{S} \equiv (s_0, \ldots, s_{n-1})$ of length n:

$$net(s_i) = \boldsymbol{W}^t \boldsymbol{u}_{s_i} + \hat{w}_1 net(s_{i-1}), \;\; i = 0, \ldots, n-1$$

where

- $\boldsymbol{u}_{s_i}$ is the label attached to item s_i;
- by definition, $net(s_{-1}) = 0$.

The output for $\mathcal{S}$ can be written as

$$o(\mathcal{S}) = sgn(\boldsymbol{W}^t \underbrace{\sum_{i=0}^{n-1} \boldsymbol{u}_{s_{n-1-i}} (\hat{w}_1)^i}_{\boldsymbol{X}_{(\mathcal{S},\hat{w}_1)}}) \, .$$

Linear Separability for Sequences

The sequence $\mathcal{S}$ is encoded as a polynomial of the recursive connection $\hat{w}_1$:

$$\mathcal{S} \xrightarrow{Poly(\hat{w}_1)} \boldsymbol{X}(\mathcal{S}, \hat{w}_1)$$

So, given a set of sequences $\boldsymbol{S} = \{\mathcal{S}_1, \ldots, \mathcal{S}_m\}$, the application of the linear recursive neuron transforms each sequence in a vector in the label space:

$$\begin{array}{ccc} \mathcal{S}_1 & & \boldsymbol{X}(\mathcal{S}_1, \hat{w}_1) \\ \mathcal{S}_2 & & \boldsymbol{X}(\mathcal{S}_2, \hat{w}_1) \\ \vdots & \xrightarrow{Poly(\hat{w}_1)} & \vdots \\ \mathcal{S}_{m-1} & & \boldsymbol{X}(\mathcal{S}_{m-1}, \hat{w}_1) \\ \mathcal{S}_m & & \boldsymbol{X}(\mathcal{S}_m, \hat{w}_1) \end{array}$$

Linear Separability for Sequences

$$Sequence\ Space \xrightarrow{Poly(\hat{w}_1)} Label\ Space$$

Thus we can define linear separability in the Sequence Space by using the concept of linear separability in the Label Space:

Definition [Linear Separability for Sequences]
A set of sequences $\boldsymbol{S} = \{\mathcal{S}_1, \ldots, \mathcal{S}_m\}$ is *linearly separable* (l.s.) if there exists a real number $\bar{w}$ such that the set $\mathcal{X}(\boldsymbol{S}, \bar{w}) = \{\boldsymbol{X}(\mathcal{S}_1, \bar{w}), \ldots, \boldsymbol{X}(\mathcal{S}_m, \bar{w})\}$ is *linearly separable*.

Linear Separability for Sequences

So, it is not hard to reformulate some well known results for set of vectors

Theorem [Linearly Nonseparable Vectors]
A set of vectors $\{Y_1, \ldots, Y_k\}$, $Y_i \in \mathbb{R}^d$, is not l.s. if and only if there exists a positive linear combination of the vectors that equals $\mathbf{0}$, i.e., there exists $b_i \geq 0,\ 1 \leq i \leq k$, such that $\sum_{i=1}^{k} b_i Y_i = \mathbf{0}$, and $b_j > 0$ for some $j,\ 1 \leq j \leq k$.

in the context of sequences:

Theorem [Linearly Nonseparable Sequences]
A set of sequences $\boldsymbol{S} = \{\mathcal{S}_1, \ldots, \mathcal{S}_m\}$ is not l.s. *if and only if* $\forall w \in \mathbb{R}$ there exists a set of nonnegative numbers $b_i,\ i = 1, \ldots, m$, not all null, such that

$$\sum_{i=1}^{m} b_i \boldsymbol{X}(\mathcal{S}_i, w) = \mathbf{0}, \quad \text{where} \quad \boldsymbol{X}(\mathcal{S}_i, w) = \sum_{j=0}^{l_i-1} \boldsymbol{u}_{l_i-1-j}^{\mathcal{S}_i}(w)^j\ .$$

Linear Separability for Sequences

If we rewrite equation

$$\sum_{i=1}^{m} b_i \boldsymbol{X}(\mathcal{S}_i, w) = \mathbf{0}$$

in a more compact form as

$$\sum_{j=0}^{M-1} (\boldsymbol{L}_{M-1-j}\boldsymbol{b})(w)^j = \mathbf{0}$$

where M is the maximum length of the sequences in $\boldsymbol{S}$, $\boldsymbol{b}^t = [b_1, \ldots, b_m]$, and the matrices $\boldsymbol{L}_j$, $0 \leq j < M$, are defined as

$$\boldsymbol{L}_j = \left[\begin{array}{ccc} \boldsymbol{\lambda}_j^{\mathcal{S}_1}, & \ldots & , \boldsymbol{\lambda}_j^{\mathcal{S}_m} \end{array} \right] \text{where for } q = 1, \ldots, m, \; \boldsymbol{\lambda}_j^{\mathcal{S}_q} = \begin{cases} \boldsymbol{u}_j^{\mathcal{S}_q} & \text{if } j < l_q \\ \mathbf{0} & \text{if } j \geq l_q \end{cases}$$

Linear Separability for Sequences

Then we can state sufficient conditions for non l.s.

Theorem

If $\exists \boldsymbol{b} \in \mathbb{R}^m$ not null, and with the non null components all of the same sign, such that $\forall j,\ \boldsymbol{b} \in kernel(\boldsymbol{L}_j)$, then $\boldsymbol{S}$ is not l.s.

True because, since every $kernel(\boldsymbol{L}_j)$ contains the vector $\boldsymbol{b}$, then either $\boldsymbol{b}$ or $-\boldsymbol{b}$ will satisfy the non-separability conditions regardless of the value of w.

Linear Separability for Sequences

It is also possible to state sufficient conditions for l.s.

Theorem

$\boldsymbol{S}$ is l.s. if one of the following conditions are satisfied:

i) the columns of $\boldsymbol{L}_0$ are linearly separable;

(proof: it is possible to find a w which “maintains” the sign produced by the first label of each sequence)

ii) the set of vectors $\sigma_{\mathcal{S}_i} = \sum_{j=0}^{l_i-1} \boldsymbol{u}_j^{\mathcal{S}_i}$ are linearly separable;

(proof: $\rightarrow$ just set $w = 1$)

iii) the set $Tails = \{\boldsymbol{u}_{l_1-1}^{\mathcal{S}_1}, \boldsymbol{u}_{l_2-1}^{\mathcal{S}_2}, \ldots, \boldsymbol{u}_{l_m-1}^{\mathcal{S}_m}\}$ is linearly separable;

(proof: $\rightarrow$ just set $w = 0$)

Linear Separability for Structures

When considering a structure $\mathcal{U}$ with maximum outdegree c, the linear recursive neuron "encodes" the structure in a polynomial in c variables:

$$o(\mathcal{U}) = o(s) = sgn(\boldsymbol{W}^t \underbrace{\sum_{v \in \mathrm{vert}(\mathcal{U})} \boldsymbol{u}_v \sum_{p \in Path_{\mathcal{U}}(v)} \prod_{j=1}^{n} (\hat{w}_j)^{\rho_j(p)}}_{\boldsymbol{\Xi}(\mathcal{U}, \hat{w}_1, \ldots, \hat{w}_c)}),$$

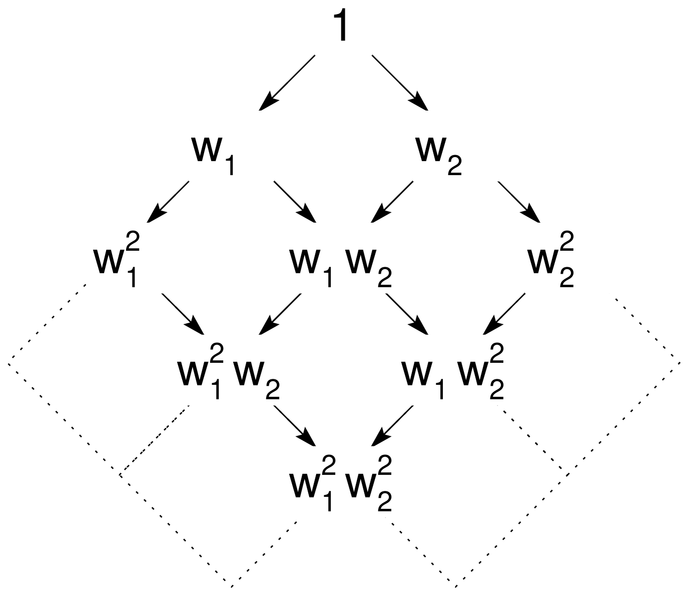
where

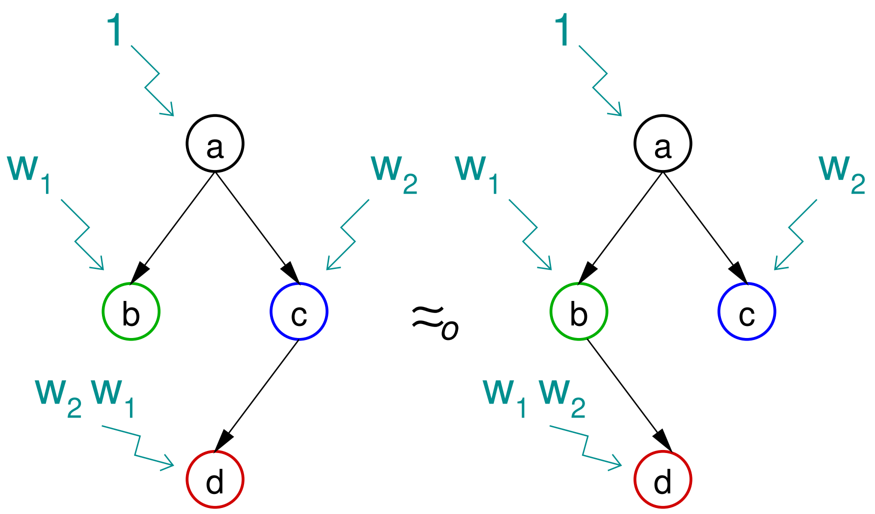
- $\boldsymbol{u}_v$ is the label attached to vertex v;
- s is the supersource of $\mathcal{U}$;
- $Path_{\mathcal{U}}(v)$ is the set of paths within $\mathcal{U}$ from s to v;
- $\rho_j(p)$ is the number of occurrences within path p of the jth pointer.

Linear Separability for Structures

In (Frasconi, Gori, Sperduti, TNN 1998) it was observed that in recursive neural networks distinct (related) DAGs may get the same output (a *collision* occur), regardless of the value associated to the weights of the network.

For a linear recursive neuron collisions are even more frequent:



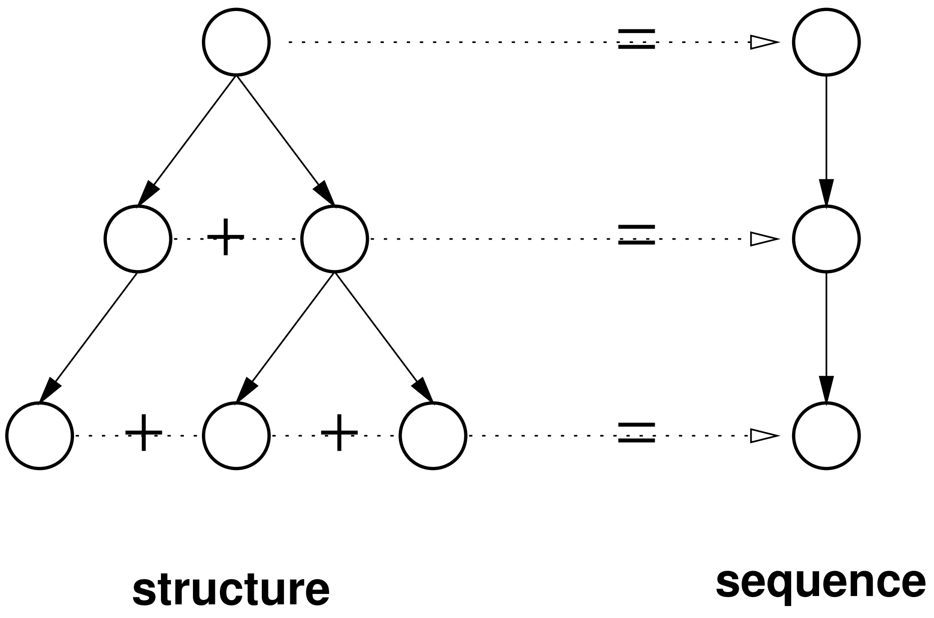


It is readily clear that structures that collide cannot be linearly separable.

Linear Separability for Structures

Almost all the results for l.s. of sequences can easily be extended to structures.

Specifically, results for sequences can be used to derive results for structures by transforming the structures in sequences, e.g. $(\hat{w}_1 = \hat{w}_2 = \cdots = \hat{w}_c = w)$



Conclusions

- Given a specific theoretical model we where able to extend the concept of linear separability for sequences and structures
- Some sufficient conditions which are not very difficult to verify have been defined for (non-)linear separability
- We still lack an algorithm (Perceptron-like ?) to efficiently find a separation when a set (of structures) is linearly separable. An algorithm can be defined when the labels are real values (just look at the roots of the polynomials), however we do not have an algorithm able to cover the more general case of real-valued vectors.