Geometric Analogue of Holographic Reduced Representations

Agnieszka Patyk

Ph.D. Supervisor: prof. Marek Czachor

Faculty of Applied Physics and Mathematics, Gdańsk University of Technology, Poland
and
Centrum Leo Apostel (CLEA), Vrije Universiteit Brussel, Belgium

Grimma, 17–19 August 2008

Outline

- Background
  - Distributed representations
  - Previous architectures
- GA model
  - Binary parametrization of the geometric product
  - Cartan representation
  - Example
  - Signatures
- Test results
  - Recognition tests
  - Blade linearity
- Future work
- Computer program *CartanGA*

Distributed representations

Example

Distributed representations

Example

"green color"

+ "triangular shape"

= "green triangle"

Distributed representations

Example

"green color"

+ "triangular shape"

= "green triangle"

Idea

Distributed representations

Example

"green color"

+ "triangular shape"

= "green triangle"

Idea

- An opposite and an alternative of "traditional" data structures (lists, databases, etc...).

Distributed representations

Example

"green color"

+ "triangular shape"

= "green triangle"

Idea

- An opposite and an alternative of "traditional" data structures (lists, databases, etc...).
- Hinton (1986):

Distributed representations

Example

“green color”

+ “triangular shape”

= “green triangle”

Idea

- An opposite and an alternative of “traditional” data structures (lists, databases, etc...).
- Hinton (1986):
  - each concept is represented over a number of units (bytes)

Distributed representations

Example


"green color"


+ "triangular shape"


= "green triangle"

Idea

- An opposite and an alternative of "traditional" data structures (lists, databases, etc...).
- Hinton (1986):
  - each concept is represented over a number of units (bytes)
  - each unit participates in the representation of some number of concepts

Agnieszka Patyk, patyk@mif.pg.gda.pl

Distributed representations

Example

"green color"

+ "triangular shape"

= "green triangle"

Idea

- An opposite and an alternative of "traditional" data structures (lists, databases, etc...).
- Hinton (1986):
  - each concept is represented over a number of units (bytes)
  - each unit participates in the representation of some number of concepts
  - the size of a distributed representation is usually fixed

Distributed representations

Example

"green color"

$+$ "triangular shape"

$=$ "green triangle"

Idea

- An opposite and an alternative of "traditional" data structures (lists, databases, etc...).
- Hinton (1986):
  - each concept is represented over a number of units (bytes)
  - each unit participates in the representation of some number of concepts
  - the size of a distributed representation is usually fixed
  - the units have either binary or a continuous-space values

Distributed representations

Example

"green color"

+ "triangular shape"

= "green triangle"

Idea

- An opposite and an alternative of "traditional" data structures (lists, databases, etc...).
- Hinton (1986):
  - each concept is represented over a number of units (bytes)
  - each unit participates in the representation of some number of concepts
  - the size of a distributed representation is usually fixed
  - the units have either binary or a continuous-space values
  - data patterns are chosen as random vectors or matrices

Distributed representations

Example

"green color"

+ "triangular shape"

= "green triangle"

Idea

- An opposite and an alternative of "traditional" data structures (lists, databases, etc...).
- Hinton (1986):
    - each concept is represented over a number of units (bytes)
    - each unit participates in the representation of some number of concepts
    - the size of a distributed representation is usually fixed
    - the units have either binary or a continuous-space values
    - data patterns are chosen as random vectors or matrices
    - in most distributed representations only the overall pattern of activated units has a meaning (resemblance to a multi-superimposed photograph)

Distributed representations

Atomic objects and complex statements

Distributed representations

Atomic objects and complex statements

$$\underbrace{bite_{agent} \circledast Fido}_{chunk} + \underbrace{bite_{object} \circledast Pat}_{chunk} = \text{“Fido bit Pat”}$$

- roles: $bite_{agent}$, $bite_{object}$
- fillers: $Fido$, Pat

Distributed representations

Atomic objects and complex statements

$$\underbrace{bite_{agent} \circledast Fido}_{chunk} + \underbrace{bite_{object} \circledast Pat}_{chunk} = \text{"Fido bit Pat"}$$

- roles: $bite_{agent}$, $bite_{object}$
- fillers: $Fido$, Pat

$$name \circledast Pat + sex \circledast male + age \circledast 66 = \text{"PSmith"}$$

- roles: $name$, sex, age
- fillers: Pat, $male$, 66

Distributed representations

Atomic objects and complex statements

$$\underbrace{bite_{agent} \circledast Fido}_{chunk} + \underbrace{bite_{object} \circledast Pat}_{chunk} = \text{“Fido bit Pat”}$$

- roles: $bite_{agent}$, $bite_{object}$
- fillers: $Fido$, Pat

$$name \circledast Pat + sex \circledast male + age \circledast 66 = \text{“PSmith”}$$

- roles: $name$, sex, age
- fillers: Pat, $male$, 66

$$see_{agent} \circledast John + see_{object} \circledast \text{“}Fido\ bit\ Pat\text{”} = \text{“John saw Fido bit Pat”}$$

- Roles are always atomic objects but fillers may be complex statements.

Distributed representations

Atomic objects and complex statements

$$\underbrace{bite_{agent} \circledast Fido}_{chunk} + \underbrace{bite_{object} \circledast Pat}_{chunk} = \text{"Fido bit Pat"}$$

- roles: $bite_{agent}$, $bite_{object}$
- fillers: $Fido$, Pat

$$name \circledast Pat + sex \circledast male + age \circledast 66 = \text{"PSmith"}$$

- roles: $name$, sex, age
- fillers: Pat, $male$, 66

$$see_{agent} \circledast John + see_{object} \circledast \text{"}Fido\ bit\ Pat\text{"} = \text{"John saw Fido bit Pat"}$$

- Roles are always atomic objects but fillers may be complex statements.

Binding and chunking

$\circledast$ binding
$+$ superposition (also called *chunking*)

Distributed representations

Decoding

Distributed representations

Decoding

$\sharp$ decoding
x^{-1} (approximate/pseudo) inverse

Distributed representations

Decoding

$\sharp$ decoding
x^{-1} (approximate/pseudo) inverse

Examples

Distributed representations

Decoding

$\sharp$ decoding
x^{-1} (approximate/pseudo) inverse

Examples

"What is the name of PSmith?"

- $(name \circledast Pat + sex \circledast male + age \circledast 66) \sharp \, name^{-1} = Pat + noise \approx Pat$

Agnieszka Patyk, patyk@mif.pg.gda.pl

Distributed representations

Decoding

$\sharp$ decoding
x^{-1} (approximate/pseudo) inverse

Examples

"What is the name of PSmith?"

- $(name \circledast Pat + sex \circledast male + age \circledast 66) \sharp\, name^{-1} = Pat + noise \approx Pat$

"What did Fido do?"

- $(bite_{agent} \circledast Fido + bite_{object} \circledast Pat) \sharp\, Fido^{-1} = bite_{agent} + noise \approx bite_{agent}$

Distributed representations

Decoding

$\sharp$ decoding
x^{-1} (approximate/pseudo) inverse

Examples

"What is the name of PSmith?"

- $(name \circledast Pat + sex \circledast male + age \circledast 66) \sharp name^{-1} = Pat + noise \approx Pat$

"What did Fido do?"

- $(bite_{agent} \circledast Fido + bite_{object} \circledast Pat) \sharp Fido^{-1} = bite_{agent} + noise \approx bite_{agent}$

Clean-up memory

- An auto-associative collection of elements and statements (excluding single chunks) produced by the system.
- Given a noisy extracted vector such structure must be able to:
  - either recall the most similar item stored ...
  - ... or indicate, that no matching object had been found.
- We need a measure of similarity (in most models: scalar product, Hamming/Euclidean distance).

Distributed representations

Requirements/Problems

Agnieszka Patyk, patyk@mif.pg.gda.pl

Distributed representations

Requirements/Problems

- $\circledast, +$ - symmetric, preserving an equal "portion" of every component.

Distributed representations

Requirements/Problems

- $\circledast, +$ - symmetric, preserving an equal "portion" of every component.
- $\sharp$, inverse - easy/fast to compute.

Distributed representations

Requirements/Problems

- $\circledast, +$ - symmetric, preserving an equal "portion" of every component.
- $\sharp$, inverse - easy/fast to compute.
- Rules of composition and decomposition must be applicable to all elements of the domain.

Distributed representations

Requirements/Problems

- $\circledast, +$ - symmetric, preserving an equal "portion" of every component.
- $\sharp$, inverse - easy/fast to compute.
- Rules of composition and decomposition must be applicable to all elements of the domain.
- Fixed size of vectors irrespectively of the degree of complication.

Distributed representations

Requirements/Problems

- $\circledast, +$ - symmetric, preserving an equal "portion" of every component.
- $\sharp$, inverse - easy/fast to compute.
- Rules of composition and decomposition must be applicable to all elements of the domain.
- Fixed size of vectors irrespectively of the degree of complication.

Advantages

- Noise tolerance, good error correction properties.
- Storage efficiency.
- The number of superimposed patterns is not fixed.

Previous Architectures

Binary Spatter Code (Pentii Kanerva, 1997)

Previous Architectures

Binary Spatter Code (Pentii Kanerva, 1997)

- Data: n-bit binary vectors of great length ($n \approx 10000$)

Previous Architectures

Binary Spatter Code (Pentii Kanerva, 1997)

- Data: n-bit binary vectors of great length ($n \approx 10000$)
- Both $\circledast$ and $\sharp$ performed by XOR

Previous Architectures

Binary Spatter Code (Pentii Kanerva, 1997)

- Data: n-bit binary vectors of great length ($n \approx 10000$)
- Both $\circledast$ and $\sharp$ performed by XOR
- $+$ majority rule addition (with additional random vectors to break the tie)

Previous Architectures

Binary Spatter Code (Pentii Kanerva, 1997)

- Data: n-bit binary vectors of great length ($n \approx 10000$)
- Both $\circledast$ and $\sharp$ performed by XOR
- $+$ majority rule addition (with additional random vectors to break the tie)
- Similarity measure: Hamming distance

Previous Architectures

Binary Spatter Code (Pentii Kanerva, 1997)

- Data: n-bit binary vectors of great length ($n \approx 10000$)
- Both $\circledast$ and $\sharp$ performed by XOR
- $+$ majority rule addition (with additional random vectors to break the tie)
- Similarity measure: Hamming distance

Holographic Reduced Representation (Tony Plate, 1994, 2003)

Previous Architectures

Binary Spatter Code (Pentii Kanerva, 1997)

- Data: n-bit binary vectors of great length ($n \approx 10000$)
- Both $\circledast$ and $\sharp$ performed by XOR
- $+$ majority rule addition (with additional random vectors to break the tie)
- Similarity measure: Hamming distance

Holographic Reduced Representation (Tony Plate, 1994, 2003)

- Data: normalized n-byte vectors chosen from $N(0, 1/n)$

Previous Architectures

Binary Spatter Code (Pentii Kanerva, 1997)

- Data: n-bit binary vectors of great length ($n \approx 10000$)
- Both $\circledast$ and $\sharp$ performed by XOR
- $+$ majority rule addition (with additional random vectors to break the tie)
- Similarity measure: Hamming distance

Holographic Reduced Representation (Tony Plate, 1994, 2003)

- Data: normalized n-byte vectors chosen from $N(0, 1/n)$
- $\circledast$ performed by circular convolution

Previous Architectures

Binary Spatter Code (Pentii Kanerva, 1997)

- Data: n-bit binary vectors of great length ($n \approx 10000$)
- Both $\circledast$ and $\sharp$ performed by XOR
- $+$ majority rule addition (with additional random vectors to break the tie)
- Similarity measure: Hamming distance

Holographic Reduced Representation (Tony Plate, 1994, 2003)

- Data: normalized n-byte vectors chosen from $N(0, 1/n)$
- $\circledast$ performed by circular convolution
- $+$ "normalized" addition, e.g. $(bite + bite_{agent} \circledast Fido + bite_{object} \circledast Pat)/\sqrt{3}$

Previous Architectures

Binary Spatter Code (Pentii Kanerva, 1997)

- Data: n-bit binary vectors of great length ($n \approx 10000$)
- Both $\circledast$ and $\sharp$ performed by XOR
- $+$ majority rule addition (with additional random vectors to break the tie)
- Similarity measure: Hamming distance

Holographic Reduced Representation (Tony Plate, 1994, 2003)

- Data: normalized n-byte vectors chosen from $N(0, 1/n)$
- $\circledast$ performed by circular convolution
- $+$ "normalized" addition, e.g. $(bite + bite_{agent} \circledast Fido + bite_{object} \circledast Pat)/\sqrt{3}$
- Inverse: involution $x_i^* = x_{(-i) mod(n)}$

Previous Architectures

Binary Spatter Code (Pentii Kanerva, 1997)

- Data: n-bit binary vectors of great length ($n \approx 10000$)
- Both $\circledast$ and $\sharp$ performed by XOR
- $+$ majority rule addition (with additional random vectors to break the tie)
- Similarity measure: Hamming distance

Holographic Reduced Representation (Tony Plate, 1994, 2003)

- Data: normalized n-byte vectors chosen from $N(0, 1/n)$
- $\circledast$ performed by circular convolution
- $+$ "normalized" addition, e.g. $(bite + bite_{agent} \circledast Fido + bite_{object} \circledast Pat)/\sqrt{3}$
- Inverse: involution $x_i^* = x_{(-i) mod(n)}$
- $\sharp$ circular correlation (i.e. circular convolution with an involution), e.g: $\left((bite + bite_{agent} \circledast Fido + bite_{object} \circledast Pat)/\sqrt{3}\right) \circledast bite^*_{agent}$

Previous Architectures

Binary Spatter Code (Pentii Kanerva, 1997)

- Data: n-bit binary vectors of great length ($n \approx 10000$)
- Both $\circledast$ and $\sharp$ performed by XOR
- $+$ majority rule addition (with additional random vectors to break the tie)
- Similarity measure: Hamming distance

Holographic Reduced Representation (Tony Plate, 1994, 2003)

- Data: normalized n-byte vectors chosen from $N(0, 1/n)$
- $\circledast$ performed by circular convolution
- $+$ "normalized" addition, e.g. $(bite + bite_{agent} \circledast Fido + bite_{object} \circledast Pat)/\sqrt{3}$
- Inverse: involution $x_i^* = x_{(-i)mod(n)}$
- $\sharp$ circular correlation (i.e. circular convolution with an involution), e.g: $\left((bite + bite_{agent} \circledast Fido + bite_{object} \circledast Pat)/\sqrt{3}\right) \circledast bite^*_{agent}$
- Similarity measure: dot product

Binary parametrization of the geometric product

Binary parametrization

Binary parametrization of the geometric product

Binary parametrization

Let $\{x_1, \ldots, x_n\}$ be a string of bits and $\{e_1, \ldots, e_n\}$ be base vectors. Then

$$e_{x_1 \ldots x_n} = e_1^{x_1} \ldots e_n^{x_n}$$

where $e_k^0 = \mathbf{1}$.

Binary parametrization of the geometric product

Binary parametrization

Let $\{x_1, \ldots, x_n\}$ be a string of bits and $\{e_1, \ldots, e_n\}$ be base vectors. Then

$$e_{x_1 \ldots x_n} = e_1^{x_1} \ldots e_n^{x_n}$$

where $e_k^0 = \mathbf{1}$.
Example:

$$1 = e_{0\ldots0} \qquad e_1 = e_{10\ldots0} \qquad e_{235} = e_{011010\ldots0}$$

Binary parametrization of the geometric product

Binary parametrization

Let $\{x_1, \ldots, x_n\}$ be a string of bits and $\{e_1, \ldots, e_n\}$ be base vectors. Then

$$e_{x_1 \ldots x_n} = e_1^{x_1} \ldots e_n^{x_n}$$

where $e_k^0 = \mathbf{1}$.
Example:

$$1 = e_{0\ldots0} \qquad e_1 = e_{10\ldots0} \qquad e_{235} = e_{011010\ldots0}$$

Geometric product as a projective XOR representation

Binary parametrization of the geometric product

Binary parametrization

Let $\{x_1, \ldots, x_n\}$ be a string of bits and $\{e_1, \ldots, e_n\}$ be base vectors. Then

$$e_{x_1 \ldots x_n} = e_1^{x_1} \ldots e_n^{x_n}$$

where $e_k^0 = \mathbf{1}$.
Example:

$$1 = e_{0 \ldots 0} \qquad e_1 = e_{10 \ldots 0} \qquad e_{235} = e_{011010 \ldots 0}$$

Geometric product as a projective XOR representation

Example:

$$\begin{aligned} e_{12}e_1 &= e_{110\ldots0}e_{10\ldots0} = e_1e_2e_1 = -e_2e_1e_1 = -e_2 = -e_{010\ldots0} \\ &= -e_{(110\ldots0)\oplus(10\ldots0)} = (-1)^D e_{(110\ldots0)\oplus(10\ldots0)} \end{aligned}$$

Binary parametrization of the geometric product

Binary parametrization

Let $\{x_1, \dots, x_n\}$ be a string of bits and $\{e_1, \dots, e_n\}$ be base vectors. Then

$$e_{x_1 \dots x_n} = e_1^{x_1} \dots e_n^{x_n}$$

where $e_k^0 = \mathbf{1}$.
Example:

$$1 = e_{0\dots0} \qquad e_1 = e_{10\dots0} \qquad e_{235} = e_{011010\dots0}$$

Geometric product as a projective XOR representation

Example:

$$\begin{aligned} e_{12}e_1 &= e_{110\dots0}e_{10\dots0} = e_1 e_2 e_1 = -e_2 e_1 e_1 = -e_2 = -e_{010\dots0} \\ &= -e_{(110\dots0)\oplus(10\dots0)} = (-1)^D e_{(110\dots0)\oplus(10\dots0)} \end{aligned}$$

More formally, for two arbitrary strings of bits we have:

$$e_{A_1 \dots A_n} e_{B_1 \dots B_n} = (-1)^{\sum\limits_{k<l} B_k A_l} e_{(A_1 \dots A_n)\oplus(B_1 \dots B_n)}$$

Matrix representation

Pauli's matrices

$$\sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Matrix representation

Pauli's matrices

$$\sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Cartan representation

$$\begin{aligned} b_{2k} &= \underbrace{\sigma_1 \otimes \cdots \otimes \sigma_1}_{n-k} \otimes \sigma_2 \otimes \underbrace{1 \otimes \cdots \otimes 1}_{k-1}, \\ b_{2k-1} &= \underbrace{\sigma_1 \otimes \cdots \otimes \sigma_1}_{n-k} \otimes \sigma_3 \otimes \underbrace{1 \otimes \cdots \otimes 1}_{k-1}. \end{aligned}$$

Example

Representation

$PSmith = name \circledast Pat + sex \circledast male + age \circledast 66$

Example
Representation

$PSmith = name \circledast Pat + sex \circledast male + age \circledast 66$

$$Pat = c_{00100}$$

Example

Representation

$PSmith = name \circledast Pat + sex \circledast male + age \circledast 66$

$$Pat = c_{00100} = b_3$$

Example

Representation

$PSmith = name \circledast Pat + sex \circledast male + age \circledast 66$

$$Pat \quad = \quad c_{00100} = b_3 = \sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_3 \otimes 1$$

Example
Representation

$PSmith = name \circledast Pat + sex \circledast male + age \circledast 66$

$$
\begin{aligned}
Pat &= c_{00100} = b_3 = \sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_3 \otimes 1 \\
male &= c_{00111} = b_3 b_4 b_5 \\
&= (\sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_3 \otimes 1)(\sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_2 \otimes 1)(\sigma_1 \otimes \sigma_1 \otimes \sigma_3 \otimes 1 \otimes 1) \\
&= \sigma_1 \otimes \sigma_1 \otimes \sigma_3 \otimes (-i\sigma_1) \otimes 1
\end{aligned}
$$

Example

Representation

$PSmith = name \circledast Pat + sex \circledast male + age \circledast 66$

$$
\begin{aligned}
Pat &= c_{00100} = b_3 = \sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_3 \otimes 1 \\
male &= c_{00111} = b_3 b_4 b_5 \\
&= (\sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_3 \otimes 1)(\sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_2 \otimes 1)(\sigma_1 \otimes \sigma_1 \otimes \sigma_3 \otimes 1 \otimes 1) \\
&= \sigma_1 \otimes \sigma_1 \otimes \sigma_3 \otimes (-i\sigma_1) \otimes 1 \\
66 &= c_{11000} = b_1 b_2 = (\sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_3)(\sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_2) \\
&= 1 \otimes 1 \otimes 1 \otimes 1 \otimes (-i\sigma_1)
\end{aligned}
$$

Example

Representation

$$PSmith = name \circledast Pat + sex \circledast male + age \circledast 66$$

$$\begin{aligned}
Pat &= c_{00100} = b_3 = \sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_3 \otimes 1 \\
male &= c_{00111} = b_3 b_4 b_5 \\
&= (\sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_3 \otimes 1)(\sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_2 \otimes 1)(\sigma_1 \otimes \sigma_1 \otimes \sigma_3 \otimes 1 \otimes 1) \\
&= \sigma_1 \otimes \sigma_1 \otimes \sigma_3 \otimes (-i\sigma_1) \otimes 1 \\
66 &= c_{11000} = b_1 b_2 = (\sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_3)(\sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_2) \\
&= 1 \otimes 1 \otimes 1 \otimes 1 \otimes (-i\sigma_1) \\
name &= c_{00010} = b_4 = \sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_2 \otimes 1
\end{aligned}$$

Example

Representation

$PSmith = name \circledast Pat + sex \circledast male + age \circledast 66$

$$
\begin{aligned}
Pat &= c_{00100} = b_3 = \sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_3 \otimes 1 \\
male &= c_{00111} = b_3 b_4 b_5 \\
&= (\sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_3 \otimes 1)(\sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_2 \otimes 1)(\sigma_1 \otimes \sigma_1 \otimes \sigma_3 \otimes 1 \otimes 1) \\
&= \sigma_1 \otimes \sigma_1 \otimes \sigma_3 \otimes (-i\sigma_1) \otimes 1 \\
66 &= c_{11000} = b_1 b_2 = (\sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_3)(\sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_2) \\
&= 1 \otimes 1 \otimes 1 \otimes 1 \otimes (-i\sigma_1) \\
name &= c_{00010} = b_4 = \sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_2 \otimes 1 \\
sex &= c_{11100} = b_1 b_2 b_3 \\
&= (\sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_3)(\sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_2)(\sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_3 \otimes 1) \\
&= \sigma_1 \otimes \sigma_1 \otimes \sigma_3 \otimes 1 \otimes (-i\sigma_1)
\end{aligned}
$$

Example

Representation

$PSmith = name \circledast Pat + sex \circledast male + age \circledast 66$

$$
\begin{aligned}
Pat &= c_{00100} = b_3 = \sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_3 \otimes 1 \\
male &= c_{00111} = b_3 b_4 b_5 \\
&= (\sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_3 \otimes 1)(\sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_2 \otimes 1)(\sigma_1 \otimes \sigma_1 \otimes \sigma_3 \otimes 1 \otimes 1) \\
&= \sigma_1 \otimes \sigma_1 \otimes \sigma_3 \otimes (-i\sigma_1) \otimes 1 \\
66 &= c_{11000} = b_1 b_2 = (\sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_3)(\sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_2) \\
&= 1 \otimes 1 \otimes 1 \otimes 1 \otimes (-i\sigma_1) \\
name &= c_{00010} = b_4 = \sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_2 \otimes 1 \\
sex &= c_{11100} = b_1 b_2 b_3 \\
&= (\sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_3)(\sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_2)(\sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_3 \otimes 1) \\
&= \sigma_1 \otimes \sigma_1 \otimes \sigma_3 \otimes 1 \otimes (-i\sigma_1) \\
age &= c_{10001} = b_1 b_5 = (\sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_1 \otimes \sigma_3)(\sigma_1 \otimes \sigma_1 \otimes \sigma_3 \otimes 1 \otimes 1) \\
&= 1 \otimes 1 \otimes (-i\sigma_2) \otimes \sigma_1 \otimes \sigma_3
\end{aligned}
$$

Agnieszka Patyk, patyk@mif.pg.gda.pl

Example

Encoding and Decoding

PSmith

Example

Encoding and Decoding

PSmith

$$PSmith = name \circledast Pat + sex \circledast male + age \circledast 66$$

Example

Encoding and Decoding

PSmith

$$\begin{aligned} PSmith &= name \circledast Pat + sex \circledast male + age \circledast 66 \\ &= c_{00010}c_{00100} + c_{11100}c_{00111} + c_{10001}c_{11000} \end{aligned}$$

Example

Encoding and Decoding

PSmith

$$
\begin{aligned}
PSmith &= name \circledast Pat + sex \circledast male + age \circledast 66 \\
&= c_{00010} c_{00100} + c_{11100} c_{00111} + c_{10001} c_{11000} \\
&= -c_{00110} + c_{11011} + c_{01001}
\end{aligned}
$$

Example

Encoding and Decoding

PSmith

$$\begin{aligned} PSmith &= name \circledast Pat + sex \circledast male + age \circledast 66 \\ &= c_{00010}c_{00100} + c_{11100}c_{00111} + c_{10001}c_{11000} \\ &= -c_{00110} + c_{11011} + c_{01001} \end{aligned}$$

PSmith's name $= PSmith \sharp name^{-1} = PSmith \circledast name$

Example

Encoding and Decoding

PSmith

$$\begin{aligned} PSmith &= name \circledast Pat + sex \circledast male + age \circledast 66 \\ &= c_{00010}c_{00100} + c_{11100}c_{00111} + c_{10001}c_{11000} \\ &= -c_{00110} + c_{11011} + c_{01001} \end{aligned}$$

PSmith's name $= PSmith \sharp name^{-1} = PSmith \circledast name$

$$PSmith \circledast name = (-c_{00110} + c_{11011} + c_{01001})c_{00010}$$

Example

Encoding and Decoding

PSmith

$$\begin{aligned} PSmith &= name \circledast Pat + sex \circledast male + age \circledast 66 \\ &= c_{00010}c_{00100} + c_{11100}c_{00111} + c_{10001}c_{11000} \\ &= -c_{00110} + c_{11011} + c_{01001} \end{aligned}$$

PSmith's name $= PSmith \sharp name^{-1} = PSmith \circledast name$

$$\begin{aligned} PSmith \circledast name &= (-c_{00110} + c_{11011} + c_{01001})c_{00010} \\ &= -c_{00100} - c_{11001} - c_{01011} \end{aligned}$$

Example

Encoding and Decoding

PSmith

$$\begin{aligned} PSmith &= name \circledast Pat + sex \circledast male + age \circledast 66 \\ &= c_{00010}c_{00100} + c_{11100}c_{00111} + c_{10001}c_{11000} \\ &= -c_{00110} + c_{11011} + c_{01001} \end{aligned}$$

PSmith's name $= PSmith \sharp name^{-1} = PSmith \circledast name$

$$\begin{aligned} PSmith \circledast name &= (-c_{00110} + c_{11011} + c_{01001})c_{00010} \\ &= -c_{00100} - c_{11001} - c_{01011} \\ &= -Pat + noise = Pat' \end{aligned}$$

Example

Clean-up

Scalar product

Scalar (inner) product is performed by the means of matrix trace.

Example

Clean-up

Scalar product

Scalar (inner) product is performed by the means of matrix trace.

$$\begin{aligned}\langle Pat' | e_{Pat} \rangle &= Tr\Big((-c_{00100} - c_{11001} - c_{01011})c_{00100}\Big) \\ &= Tr\Big(-\mathbf{1} + c_{11101} - c_{01111}\Big) \\ &= -32 \neq 0\end{aligned}$$

Example

Clean-up

Scalar product

Scalar (inner) product is performed by the means of matrix trace.

$$
\begin{aligned}
\langle Pat' | e_{Pat} \rangle &= Tr\Big((-c_{00100} - c_{11001} - c_{01011}) c_{00100} \Big) \\
&= Tr\Big(-\mathbf{1} + c_{11101} - c_{01111} \Big) \\
&= -32 \neq 0 \\
\langle Pat' | e_{male} \rangle &= 0 \\
\langle Pat' | e_{66} \rangle &= 0 \\
\langle Pat' | e_{name} \rangle &= 0 \\
\langle Pat' | e_{sex} \rangle &= 0 \\
\langle Pat' | e_{age} \rangle &= 0 \\
\langle Pat' | PSmith \rangle &= 0
\end{aligned}
$$

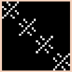
Signatures

Examples

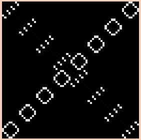
Signatures

Examples

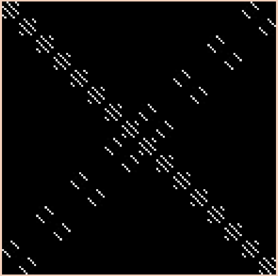
PSmith, $n = 5$



PSmith, $n = 6$



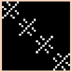
PSmith, $n = 7$



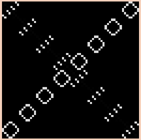
Signatures

Examples

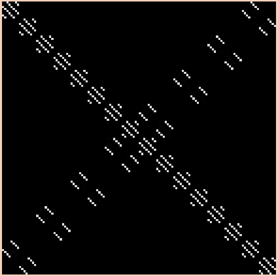
PSmith, $n = 5$



PSmith, $n = 6$



PSmith, $n = 7$



Signatures:

 LHS-signature

 RHS-signature

Agnieszka Patyk, patyk@mif.pg.gda.pl

Signatures

Dimensions

Signatures

Dimensions

- $2^{\lfloor \frac{n}{2} \rfloor}$ signatures on one of the diagonals.
- Each signature is of dimensions $2^{\lceil \frac{n}{2} \rceil} \times 2^{\lceil \frac{n}{2} \rceil}$.

Signatures

Dimensions

- $2^{\lfloor \frac{n}{2} \rfloor}$ signatures on one of the diagonals.
- Each signature is of dimensions $2^{\lceil \frac{n}{2} \rceil} \times 2^{\lceil \frac{n}{2} \rceil}$.

Cartan representation

$$
\begin{aligned}
b_{2k} &= \underbrace{\sigma_1 \otimes \cdots \otimes \sigma_1}_{at\ least\ \lfloor \frac{n}{2} \rfloor} \otimes \sigma_2 \otimes \underbrace{1 \otimes \cdots \otimes 1}_{k-1}, \\
b_{2k-1} &= \underbrace{\sigma_1 \otimes \cdots \otimes \sigma_1}_{at\ least\ \lfloor \frac{n}{2} \rfloor} \otimes \sigma_3 \otimes \underbrace{1 \otimes \cdots \otimes 1}_{k-1}.
\end{aligned}
$$

Recognition tests

Construction

Recognition tests

Construction

- Plate (HRR) construction, e.g. $eat + eat_{agent} \circledast Mark + eat_{object} \circledast theFish$

Recognition tests

Construction

- Plate (HRR) construction, e.g. $eat + eat_{agent} \circledast Mark + eat_{object} \circledast theFish$
- Agent-Object construction, e.g. $eat_{agent} \circledast Mark + eat_{object} \circledast theFish$

Recognition tests

Construction

- Plate (HRR) construction, e.g. $eat + eat_{agent} \circledast Mark + eat_{object} \circledast theFish$
- Agent-Object construction, e.g. $eat_{agent} \circledast Mark + eat_{object} \circledast theFish$

Vocabulary/Sentence set

Similar sentences, e.g:

- Fido bit Pat.
- Fido bit PSmith.
- Pat fled from Fido.
- PSmith fled from Fido.
- Fido bit PSmith causing PSmith to flee from Fido.
- Fido bit Pat causing Pat to flee from Fido.
- John saw that Fido bit PSmith causing PSmith to flee from Fido.
- John saw that Fido bit Pat causing Pat to flee from Fido.
- etc...

Altogether 42 atomic objects and 19 sentences.

Recognition tests

Agent-Object construction

Works better for:

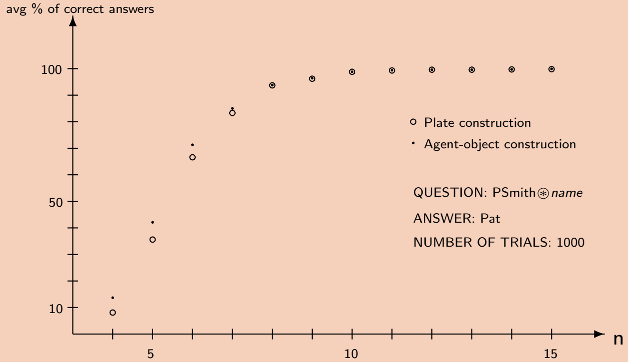
- simple sentences and simple answers

Recognition tests

Agent-Object construction

Works better for:

- simple sentences and simple answers



Recognition tests

Agent-Object construction

Works better for:

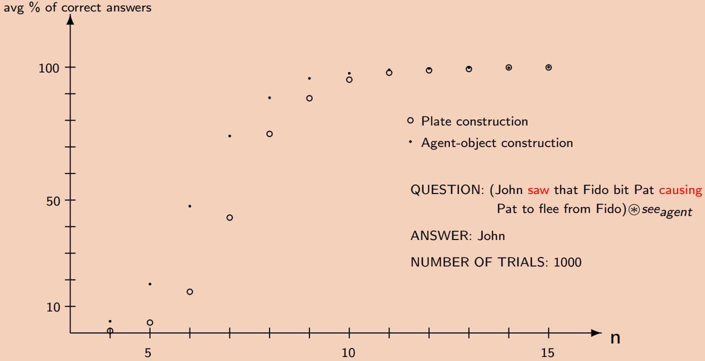
- simple sentences and simple answers,
- nested sentences, from which a rather unique information is to be derived.

Recognition tests

Agent-Object construction

Works better for:

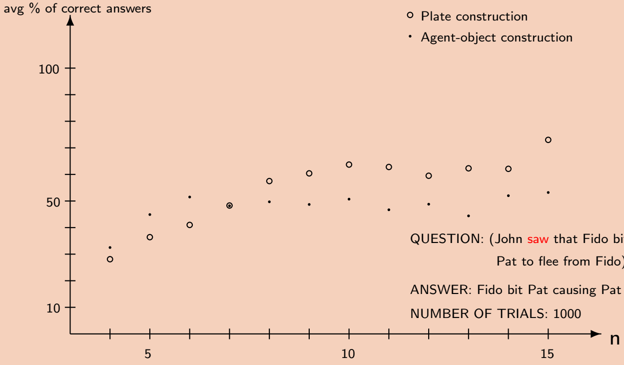
- simple sentences and simple answers,
- nested sentences, from which a rather unique information is to be derived.



Recognition tests

Plate (HRR) construction

Works better for nested sentences, from which a complex information needs to be derived.



QUESTION: (John saw that Fido bit Pat causing Pat to flee from Fido)$\circledast$ see_{object}

ANSWER: Fido bit Pat causing Pat to flee from Fido

NUMBER OF TRIALS: 1000

Blade linearity

Blade linearity

Problems

Blade linearity

Problems

- How often does the system produce identical blades representing atomic objects?

Blade linearity

Problems

- How often does the system produce identical blades representing atomic objects?
- How often does the system produce identical sentence chunks from different blades?

Blade linearity

Problems

- How often does the system produce identical blades representing atomic objects?
- How often does the system produce identical sentence chunks from different blades?
- How do the two above problems affect the number of (pseudo-) correct answers?

Blade linearity

Problems

- How often does the system produce identical blades representing atomic objects?
- How often does the system produce identical sentence chunks from different blades?
- How do the two above problems affect the number of (pseudo-) correct answers?

Assumption: ideal conditions

Blade linearity

Problems

- How often does the system produce identical blades representing atomic objects?
- How often does the system produce identical sentence chunks from different blades?
- How do the two above problems affect the number of (pseudo-) correct answers?

Assumption: ideal conditions

No two chunks of a sentence at any time are identical (up to a constant).

Blade linearity

Problems

- How often does the system produce identical blades representing atomic objects?
- How often does the system produce identical sentence chunks from different blades?
- How do the two above problems affect the number of (pseudo-) correct answers?

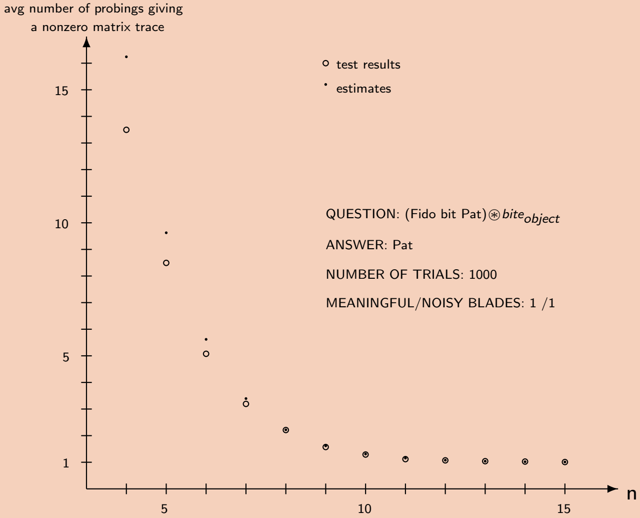
Assumption: ideal conditions

No two chunks of a sentence at any time are identical (up to a constant).

$$\langle A|C\rangle \neq 0 \equiv A \text{ and } C \text{ share a common blade}$$

Blade linearity

One meaningful blade, $L > 0$ noisy blades



QUESTION: (Fido bit Pat)$\circledast bite_{object}$

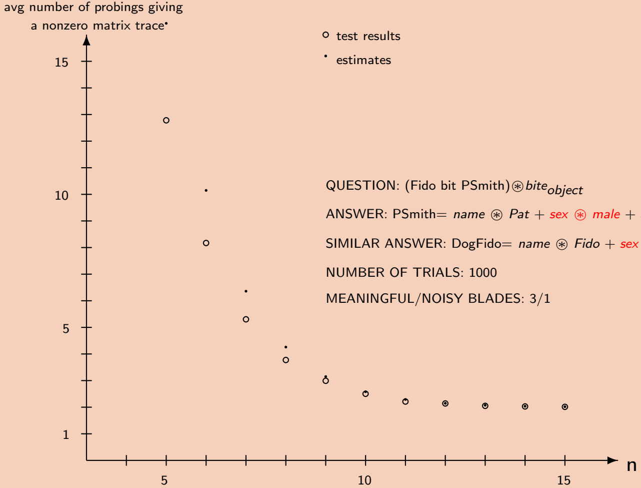
ANSWER: Pat

NUMBER OF TRIALS: 1000

MEANINGFUL/NOISY BLADES: 1 /1

Blade linearity

$K > 1$ meaningful blades, $L > 0$ noisy blades



QUESTION: (Fido bit PSmith)$\circledast bite_{object}$

ANSWER: PSmith= $name \circledast Pat + sex \circledast male + age \circledast 66$

SIMILAR ANSWER: DogFido= $name \circledast Fido + sex \circledast male + \ldots$

NUMBER OF TRIALS: 1000

MEANINGFUL/NOISY BLADES: 3/1

What's next?

Issues

What's next?

Issues

- Decide on construction (something between Plate and Agent-Object?).

What's next?

Issues

- Decide on construction (something between Plate and Agent-Object?).
- Tests for greater n.

What's next?

Issues

- Decide on construction (something between Plate and Agent-Object?).
- Tests for greater n.
- Scaling.

References

- M. Czachor, D. Aerts and B. De Moor. *On geometric-algebra representation of binary spatter codes.* preprint arXiv:cs/0610075 [cs.AI]. 2006.
- M. Czachor, D. Aerts and B. De Moor. *Geometric Analogue of Holographic Reduced Representation.* preprint arXiv:0710.2611. 2007.
- G. E. Hinton, J. L. McClelland and D. E. Rumelhart. *Distributed representations.* Parallel distributed processing: Explorations in the microstructure of cognition. vol. 1, 77109. The MIT Press, Cambridge, MA, 1986.
- P. Kanerva. *Binary spatter codes of ordered k-tuples.* Artificial Neural Networks ICANN Proceedings, Lecture Notes in Computer Science vol. 1112, pp. 869-873, C. von der Malsburg et al. (Eds.). Springer, Berlin, 1996.
- P. Kanerva. *Fully distributed representation.* Proc. 1997 Real World Computing Symposium (RWC97, Tokyo), pp. 358-365. Real World Computing Partnership, Tsukuba-City, Japan, 1997.
- T. Plate. *Holographic Reduced Representation: Distributed Representation for Cognitive Structures.* CSLI Publications, Stanford, 2003.