Computing Contrast on Conceptual Spaces

Giovanni Sileno, Isabelle Bloch, Jamal Atif, Jean-Louis Dessalles

3 July 2018, Workshop on Artificial Intelligence and Cognition (AIC) @ Palermo

giovanni.sileno@telecom-paristech.fr
“small” problem

The standard theory of conceptual spaces insists on lexical meaning: linguistic marks are associated to regions.
“small” problem

The standard theory of *conceptual spaces* insists on *lexical meaning*: linguistic marks are associated to regions.

→ *extensional* as the standard symbolic approach.
“small” problem

The standard theory of *conceptual spaces* insists on *lexical meaning*: linguistic marks are associated to regions.
→ *extensional* as the standard symbolic approach.

If *red*, or *green*, or *brown* correspond to regions in the color space...
“small” problem

The standard theory of *conceptual spaces* insists on *lexical meaning*: linguistic marks are associated to regions. → *extensional* as the standard symbolic approach.

If *red*, or *green*, or *brown* correspond to regions in the color space...

why do we say “*red dogs*” even if they are actually brown?
Predicates resulting from contrast

Alternative hypothesis [Dessalles2015]:
Predicates are generated *on the fly* after an operation of *contrast*.

\[ c = o - p \sim \text{“red”} \]

Predication follows principles of *descriptive pertinence*: *objects are determined by distinctive features*

Predicates resulting from contrast

Alternative hypothesis [Dessalles2015]:
Predicates are generated on the fly after an operation of contrast.

\[ c = o - p \sim \text{“red”} \]

- contrastor
- object (target)
- prototype (reference)

These dogs are “red dogs”:
- not because their color is red (they are brown),
- because they are more red than the dog prototype
Predicates resulting from contrast

In logic, usually: $above(a, b) \leftrightarrow below(b, a)$
Predicates resulting from contrast

In logic, usually: $\text{above}(a, b) \iff \text{below}(b, a)$

However, people don't say

"the table is below the apple."

"the board is above the leg."
Predicates resulting from contrast

In logic, usually: above\((a, b)\) \iff below\((b, a)\)

However, people don't say

"the table is below the apple."
"the board is above the leg."

If the contrastive hypothesis is correct,\[ c = a - b \sim \text{"above"} \]

superior in \textit{strength} to\[ c' = b - a \sim \text{"below"} \]
Directional relationships

We considered an existing method [Bloch2006] used in image processing to compute directional relative positions of visual entities (e.g. of biomedical images).

models of relations
for a point centered
in the origin

Directional relationships

We considered an existing method [Bloch2006] used in image processing to compute directional relative positions of visual entities (e.g. of biomedical images).
Directional relationships

We considered an existing method [Bloch2006] used in image processing to compute directional relative positions of visual entities (e.g. of biomedical images).
Directional relationships

We considered an existing method [Bloch2006] used in image processing to compute directional relative positions of visual entities (e.g. of biomedical images).
Directional relationships

We considered an existing method [Bloch2006] used in image processing to compute directional relative positions of visual entities (e.g. of biomedical images).

how much a is "above b"

operation scheme: \( a \sim b + "above" \)
Directional relationships

We considered an existing method [Bloch2006] used in image processing to compute directional relative positions of visual entities (e.g. of biomedical images).

how much a is “above b” inverse operation to contrast: merge

operation scheme: $a \sim b + \text{“above”}$
Directional relationships

We considered an existing method [Bloch2006] used in image processing to compute directional relative positions of visual entities (e.g., of biomedical images).

How much a is "above b"

Operation scheme: \[ a \sim b + \text{"above"} \]

Inverse operation to contrast: \textit{merge}

Alignment as overlap
Directional relationships

We considered an existing method [Bloch2006] used in image processing to compute directional relative positions of visual entities (e.g. of biomedical images).

Operation scheme: \(a \sim b + \text{“above”}\)

Inverse operation to contrast: \textit{merge}

\(\text{alignment as overlap}\)

\(\text{cf. with } o - p \sim \text{“red”}\)

How much \(a\) is \text{“above} \ b\)
Directional relationships

We considered an existing method [Bloch2006] used in image processing to compute directional relative positions of visual entities (e.g. of biomedical images).

If we settle upon contrast, we can categorize its output for relations!

“above b”
operation scheme: \( a \sim b + \text{“above”} \)
cf. with \( o - p \sim \text{“red”} \)
alignment as overlap
How does contrast work?
Computing contrast (1D)

- Consider coffees served in a bar. Intuitively, whether a coffee is qualified as being *hot* or *cold* depends mostly on what the speaker expects of coffees served at bars, rather than a specific absolute temperature.

\[ c = o - p \sim \text{“hot”} \]
Computing contrast (1D)

- Consider coffees served in a bar. Intuitively, whether a coffee is qualified as being \textit{hot} or \textit{cold} depends mostly on what the speaker expects of coffees served at bars, rather than a specific absolute temperature.

- For simplicity, we represent objects on 1D (\textit{temperature}) with \textit{real coordinates}. 

\[ c = o - p \sim \text{"hot"} \]

\textit{contrastor} \hspace{1cm} \textit{object (target)} \hspace{1cm} \textit{prototype (reference)}
Computing contrast (1D)

\[ c = o - p \sim \text{"hot"} \]

- Because prototypes are defined together with a concept region, let us consider some regional information, for instance represented as an **egg-yolk** structure.
Computing contrast (1D)

\[ c = o - p \sim \text{“hot”} \]

- Because prototypes are defined together with a concept region, let us consider some regional information, for instance represented as an egg-yolk structure.

* For simplicity, we assume regions to be symmetric.
Computing contrast (1D)

\[ c = o - p \sim \text{“hot”} \]

- Because prototypes are defined together with a concept region, let us consider some regional information, for instance represented as an egg-yolk structure.
  - internal boundary (yolk) \( p \pm \sigma \) for typical elements of that category of objects (e.g. coffee served at bar).

* For simplicity, we assume regions to be symmetric.
Computing contrast (1D)

\[ c = o - p \sim \text{“hot”} \]

- Because prototypes are defined together with a concept region, let us consider some regional information, for instance represented as an *egg-yolk* structure.
  - internal boundary (*yolk*) \( p \pm \sigma \) for typical elements of that category of objects (e.g. coffee served at bar).
  - external boundary (*egg*) \( p \pm \rho \) for all elements directly associated to that category of objects.

* For simplicity, we assume regions to be symmetric.
Computing contrast (1D)

- Two required functions:
  - **centering** of target with respect to typical region
  - **scaling** to neutralize effects of scale (e.g. “hot coffee” vs “hot planet”)

\[ c = o - p \sim \text{“hot”} \]

* For simplicity, we assume regions to be symmetric.
Computing contrast (1D)

\[ C = \text{contrast}(o, \langle p, \sigma, \rho \rangle) = \bigcirc_{o-p} \frac{1}{\rho} = \bigcirc_{(o-p)/\rho}^{\sigma/\rho} \]

\( C \sim \text{“hot”} \)
Computing contrast (1D)

- As *contrastors* are extended objects, they might be compared to model categories represented as regions by measuring their *degree of overlap*:

\[
\text{strength}(r) = \frac{|C \cap M^{(r)}|}{|C|}
\]
Computing contrast (1D)

• As contrastors are extended objects, they might be compared to model categories represented as regions by measuring their degree of overlap:

\[
\text{strength}(r) = \frac{|C \cap M^{(r)}|}{|C|}
\]

• Most distinctive property: \(\arg \max_r |C \cap M^{(r)}|\)
Computing contrast (1D)

- Applying the previous computation, we can easily derive the **membership functions** of some general relations with respect to the objects of that category.
- For instance, by dividing the representational container in 3 equal parts, we have:
Computing contrast (1D)

- Applying the previous computation, we can easily derive the *membership functions* of some general relations with respect to the objects of that category.
- For instance, by dividing the representational container in 3 equal parts, we have:

\[
\begin{array}{cccccc}
\text{strength of } M_1 & \text{strength of } M_2 & \text{strength of } M_3 \\
\text{"cold"} & \text{"ok"} & \text{"hot"} \\
-\sigma & p-\rho/3 & +\sigma \\
-\sigma & p & -\sigma \\
-\sigma & p+\rho/3 & +\sigma \\
0
\end{array}
\]
Adaptation of parameters

- Given a certain category of objects and a certain dimension, parameters are chosen such that
  - $\sigma$ captures the *most typical* exemplars
  - $\rho$ covers *all* exemplars
Adaptation of parameters

- Given a certain category of objects and a certain dimension, parameters are chosen such as that
  - $\sigma$ captures the most typical exemplars
  - $\rho$ covers all exemplars

- Expected adaptive effects:
  - *relativization*: providing more contrastive exemplars, those which were highly contrastive before become less contrastive
Adaptation of parameters

- Given a certain category of objects and a certain dimension, parameters are chosen such that
  - $\sigma$ captures the most typical exemplars
  - $\rho$ covers all exemplars

- Expected adaptive effects:
  - *relativization*: providing more contrastive exemplars, those which were highly contrastive before become less contrastive
  - if pruning of exemplars holds, *hardening*: the concept region will recenter around the most recent elements.
Computing contrast (1D)

- The previous formulation might be extended to consider contrast between *two regions*, by utilizing *discretization* ($\lfloor \cdot \rfloor$ denotes the approximation to the nearest integer):

$$C = \text{contrast}_R(\langle t, \tau \rangle, \langle r, \sigma \rangle, \langle f, \rho \rangle) \approx \text{contrast} \left(\left\lfloor \frac{t}{2\tau} \right\rfloor, \left\lfloor \frac{r}{2\tau} \right\rfloor, \left\lfloor \frac{\sigma}{\tau} \right\rfloor, \left\lfloor \frac{\rho}{\tau} \right\rfloor\right)$$
Computing contrast (1D)

• The previous formulation might be extended to consider contrast between two regions, by utilizing discretization (denotes the approximation to the nearest integer):

\[ C = \text{contrast}_R(\langle t, \tau \rangle, \langle r, \sigma \rangle, \langle f, \rho \rangle) \approx \text{contrast} \left( \left\lfloor \frac{t}{2\tau} \right\rfloor, \left\lfloor \frac{r}{2\tau} \right\rfloor, \left\lfloor \frac{\sigma}{\tau} \right\rfloor, \left\lfloor \frac{\rho}{\tau} \right\rfloor \right) \]

• Simplification possible if the target region much smaller than the reference region...

• but in the other cases?
  - possible solution: aggregation of contrastors obtained by point-wise contrast
Computing contrast (>1D)

- Let us consider two 2D visual objects A and B (the two dimensions form a *Cartesian space*, and they are *not* perceptually independent).
- We can apply contrast iteratively for each point of A with respect to B, and then *aggregate* the resulting contrastors.
Computing contrast (>1D)

- Let us consider two 2D visual objects $A$ and $B$ (the two dimensions form a *Cartesian space*, and they are *not* perceptually independent).
- We can apply contrast iteratively for each point of $A$ with respect to $B$, and then *aggregate* the resulting contrastors.

\[
\mathcal{H}(A, B)(z) = \{a \in A, b \in B \mid a - b = z\}
\]
Computing contrast (>1D)

- Let us consider two 2D visual objects A and B (the two dimensions form a *Cartesian space*, and they are *not* perceptually independent).
- We can apply contrast iteratively for each point of A with respect to B, and then *aggregate* the resulting contrastors.

\[
\mathcal{H}(A, B)(z) = \{a \in A, b \in B \mid a - b = z\}
\]

\[
\mu_{A \Theta B}(z) = \frac{|\mathcal{H}(A, B)(z)|}{\max_w |\mathcal{H}(A, B)(w)|}
\]
Computing contrast (>1D)

- Let us consider two 2D visual objects A and B (the two dimensions form a *Cartesian space*, and they are *not* perceptually independent).
- We can apply contrast iteratively for each point of A with respect to B, and then aggregate the resulting contrastors.

\[ \mathcal{H}(A, B)(z) = \{ a \in A, b \in B \mid a - b = z \} \]

\[ \mu_{A \Theta B}(z) = \frac{|\mathcal{H}(A, B)(z)|}{\max_w |\mathcal{H}(A, B)(w)|} \]

*accumulation set*

*counting*

*normalization*
Computing contrast (>1D)

- Let us consider two 2D visual objects A and B (the two dimensions form a *Cartesian space*, and they are *not* perceptually independent).
- We can apply contrast iteratively for each point of A with respect to B, and then *aggregate* the resulting contrastors.

Work in progress: use of *erosion* to compute contrast!
Computing contrast (>1D)

- If dimensions are *perceptually independent*, we can apply contrast on each dimension separately:

\[
C = (C_1, \ldots, C_n) = (\text{contrast}(o_1, \langle p_1, \sigma_1, \rho_1 \rangle), \ldots, \text{contrast}(o_n, \langle p_n, \sigma_n, \rho_n \rangle))
\]
Computing contrast (>1D)

- If dimensions are *perceptually independent*, we can apply contrast on each dimension separately:

\[ C = (C_1, \ldots, C_n) = (\text{contrast}(o_1, \langle p_1, \sigma_1, \rho_1 \rangle), \ldots, \text{contrast}(o_n, \langle p_n, \sigma_n, \rho_n \rangle)) \]

- The result can be used to create a contrastive description of the object, i.e. its *most distinguishing* features.

- e.g. *apples* (as fruits):
  red, spherical, quite sugared
Computing contrast (>1D)

- Example: fruit domain from [Bechberger2017]:

<table>
<thead>
<tr>
<th>concept</th>
<th>hue</th>
<th>region</th>
<th>prototype (center)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>hue</td>
<td>roundness</td>
<td>sweetness</td>
</tr>
<tr>
<td>pear</td>
<td>0.50–0.70</td>
<td>0.40–0.60</td>
<td>0.35–0.45</td>
</tr>
<tr>
<td>orange</td>
<td>0.80–0.90</td>
<td>0.90–1.00</td>
<td>0.60–0.70</td>
</tr>
<tr>
<td>lemon</td>
<td>0.70–0.80</td>
<td>0.45–0.55</td>
<td>0.00–0.10</td>
</tr>
<tr>
<td>granny smith</td>
<td>0.55–0.60</td>
<td>0.70–0.80</td>
<td>0.35–0.45</td>
</tr>
<tr>
<td>apple (green type)</td>
<td>0.50–0.80</td>
<td>0.65–0.80</td>
<td>0.35–0.50</td>
</tr>
<tr>
<td>apple (yellow type)</td>
<td>0.65–0.85</td>
<td>0.65–0.80</td>
<td>0.40–0.55</td>
</tr>
<tr>
<td>apple (red type)</td>
<td>0.70–1.00</td>
<td>0.65–0.80</td>
<td>0.45–0.60</td>
</tr>
</tbody>
</table>

Computing contrast (>1D)

- Example: fruit domain from [Bechberger2017]:

<table>
<thead>
<tr>
<th>concept</th>
<th>hue</th>
<th>region</th>
<th>sweetness</th>
<th>prototype (center)</th>
<th>hue</th>
<th>roundness</th>
<th>sweetness</th>
</tr>
</thead>
<tbody>
<tr>
<td>pear</td>
<td>0.50–0.70</td>
<td>0.40–0.60</td>
<td>0.35–0.45</td>
<td>0.60</td>
<td>0.50</td>
<td>0.40</td>
<td></td>
</tr>
<tr>
<td>orange</td>
<td>0.80–0.90</td>
<td>0.90–1.00</td>
<td>0.60–0.70</td>
<td>0.80</td>
<td>0.95</td>
<td>0.65</td>
<td></td>
</tr>
<tr>
<td>lemon</td>
<td>0.70–0.80</td>
<td>0.45–0.55</td>
<td>0.00–0.10</td>
<td>0.75</td>
<td>0.50</td>
<td>0.05</td>
<td></td>
</tr>
<tr>
<td>granny smith</td>
<td>0.55–0.60</td>
<td>0.70–0.80</td>
<td>0.35–0.45</td>
<td>0.575</td>
<td>0.75</td>
<td>0.40</td>
<td></td>
</tr>
<tr>
<td>apple (green type)</td>
<td>0.50–0.80</td>
<td>0.65–0.80</td>
<td>0.35–0.50</td>
<td>0.65</td>
<td>0.725</td>
<td>0.425</td>
<td></td>
</tr>
<tr>
<td>apple (yellow type)</td>
<td>0.65–0.85</td>
<td>0.65–0.80</td>
<td>0.40–0.55</td>
<td>0.75</td>
<td>0.725</td>
<td>0.475</td>
<td></td>
</tr>
<tr>
<td>apple (red type)</td>
<td>0.70–1.00</td>
<td>0.65–0.80</td>
<td>0.45–0.60</td>
<td>0.85</td>
<td>0.725</td>
<td>0.525</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>concept</th>
<th>hue</th>
<th>region</th>
<th>sweetness</th>
<th>prototype (center/centroid)</th>
<th>hue</th>
<th>roundness</th>
<th>sweetness</th>
</tr>
</thead>
<tbody>
<tr>
<td>apple</td>
<td>0.50–1.00</td>
<td>0.65–0.80</td>
<td>0.35–0.60</td>
<td>0.75/0.75</td>
<td>0.725/0.725</td>
<td>0.475/0.475</td>
<td></td>
</tr>
<tr>
<td>fruit</td>
<td>0.50–1.00</td>
<td>0.40–1.00</td>
<td>0.00–0.70</td>
<td>0.75/0.72</td>
<td>0.70/0.70</td>
<td>0.35/0.42</td>
<td></td>
</tr>
</tbody>
</table>

Computing contrast (>1D)

- Example: fruit domain from [Bechberger2017]:

<table>
<thead>
<tr>
<th>concept</th>
<th>hue</th>
<th>region roundness</th>
<th>sweetness</th>
<th>prototype (center)</th>
<th>hue</th>
<th>roundness</th>
<th>sweetness</th>
</tr>
</thead>
<tbody>
<tr>
<td>pear</td>
<td>0.50–0.70</td>
<td>0.40–0.60</td>
<td>0.35–0.45</td>
<td>0.60</td>
<td>0.50</td>
<td>0.40</td>
<td></td>
</tr>
<tr>
<td>orange</td>
<td>0.80–0.90</td>
<td>0.90–1.00</td>
<td>0.60–0.70</td>
<td>0.80</td>
<td>0.95</td>
<td>0.65</td>
<td></td>
</tr>
<tr>
<td>lemon</td>
<td>0.70–0.80</td>
<td>0.45–0.55</td>
<td>0.00–0.10</td>
<td>0.75</td>
<td>0.50</td>
<td>0.05</td>
<td></td>
</tr>
<tr>
<td>granny smith</td>
<td>0.55–0.60</td>
<td>0.70–0.80</td>
<td>0.35–0.45</td>
<td>0.575</td>
<td>0.75</td>
<td>0.40</td>
<td></td>
</tr>
<tr>
<td>apple (green type)</td>
<td>0.50–0.80</td>
<td>0.65–0.80</td>
<td>0.35–0.50</td>
<td>0.65</td>
<td>0.725</td>
<td>0.425</td>
<td></td>
</tr>
<tr>
<td>apple (yellow type)</td>
<td>0.65–0.85</td>
<td>0.65–0.80</td>
<td>0.40–0.55</td>
<td>0.75</td>
<td>0.725</td>
<td>0.475</td>
<td></td>
</tr>
<tr>
<td>apple (red type)</td>
<td>0.70–1.00</td>
<td>0.65–0.80</td>
<td>0.45–0.60</td>
<td>0.85</td>
<td>0.725</td>
<td>0.525</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>concept</th>
<th>hue</th>
<th>region roundness</th>
<th>sweetness</th>
<th>prototype (center/centroid)</th>
<th>hue</th>
<th>roundness</th>
<th>sweetness</th>
</tr>
</thead>
<tbody>
<tr>
<td>apple</td>
<td>0.50–1.00</td>
<td>0.65–0.80</td>
<td>0.35–0.60</td>
<td>0.75/0.75</td>
<td>0.725/0.725</td>
<td>0.475/0.475</td>
<td></td>
</tr>
<tr>
<td>fruit</td>
<td>0.50–1.00</td>
<td>0.40–1.00</td>
<td>0.00–0.70</td>
<td>0.75/0.72</td>
<td>0.70/0.70</td>
<td>0.35/0.42</td>
<td></td>
</tr>
</tbody>
</table>

Container regions can be used as basis to extract distinctive features
Computing contrast (>1D)

- For instance, by contrasting each fruit concept with the aggregate “fruit” concept (using discretization, taking $\sigma = 0.5\rho$), we obtain the following contrastors (centers):

<table>
<thead>
<tr>
<th>concept</th>
<th>hue</th>
<th>roundness</th>
<th>sweetness</th>
<th>red</th>
<th>green</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>pear</td>
<td>-0.6</td>
<td>-0.7</td>
<td>0.1</td>
<td>-0.3</td>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>orange</td>
<td>0.4</td>
<td>0.8</td>
<td>0.4</td>
<td>1.0</td>
<td>0.0</td>
<td>-1.0</td>
</tr>
<tr>
<td>lemon</td>
<td>0.0</td>
<td>-0.7</td>
<td>-0.4</td>
<td>0.8</td>
<td>0.8</td>
<td>-1.0</td>
</tr>
<tr>
<td>apple</td>
<td>0.0</td>
<td>0.1</td>
<td>0.2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>
Computing contrast (>1D)

- For instance, by contrasting each fruit concept with the aggregate “fruit” concept (using discretization, taking $\sigma = 0.5\rho$), we obtain the following contrastors (centers):

<table>
<thead>
<tr>
<th>concept</th>
<th>hue</th>
<th>roundness</th>
<th>sweetness</th>
<th>red</th>
<th>green</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>pear</td>
<td>-0.6</td>
<td>-0.7</td>
<td>0.1</td>
<td>-0.3</td>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>orange</td>
<td>0.4</td>
<td>0.8</td>
<td>0.4</td>
<td>1.0</td>
<td>0.0</td>
<td>-1.0</td>
</tr>
<tr>
<td>lemon</td>
<td>0.0</td>
<td>-0.7</td>
<td>-0.4</td>
<td>0.8</td>
<td>0.8</td>
<td>-1.0</td>
</tr>
<tr>
<td>apple</td>
<td>0.0</td>
<td>0.1</td>
<td>0.2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

In principle, a similar output could provide the weights of features in forming a certain concept → basis for object categorization.
Individuation and concept formation

- We believe discriminatory aspects might be crucial not only for *individuation*, but also for the *formation of concepts*.
  - this is aligned with recent empirical experiences [Ben-Yosef2018] showing the fundamental role of the spatial organization of visual elements in *object recognition* tasks.

Conclusion

- By referring to a **contrast** mechanism:
  - *membership functions* become derived objects,
  - references and frames provide a natural **contextualization**,
  - *modifier-head* concept combinations are directly implemented (no need of *contrast classes*).

Conclusion

- By referring to a **contrast** mechanism:
  - *membership functions* become derived objects,
  - references and frames provide a natural *contextualization*,
  - *modifier-head* concept combinations are directly implemented (no need of *contrast classes*),
  - problems with geometric axioms in relation to *similarity judgments* (symmetry, triangle inequality, minimality, diagnosticity effect) are easily explained [Sileno2017]

Conclusion

- By referring to a **contrast** mechanism:
  - *membership functions* become derived objects,
  - references and frames provide a natural *contextualization*,
  - *modifier-head* concept combinations are directly implemented (no need of *contrast classes*),

- Future research track:
  - contrast is defined in duality with merge,
  - merge produces order relations between concepts
  - the resulting lattice is a space of concepts

*Do conceptual spaces emerge from contrastive functions?*