Semantic Space of Logic

Orges Leka (firstname.lastname@gmail.com)

9. Oktober 2024 in Limburg

This paper presents a novel framework for constructing a semantic geometry of logic using Reproducing Kernel Hilbert Spaces (RKHS). It develops a geometric structure to measure similarities between entities in a set X by utilizing a positive semi-definite kernel. This structure facilitates the embedding of Boolean and Łukasiewicz logics within this geometric space, providing a novel perspective on traditional logical operations and paradoxes such as the Liar Paradox.

The research introduces mathematical formalism for projecting semantic vectors and explores the relationships between conceptual and semantic spaces through the application of the cosine kernel. The paper also demonstrates how logical operations, such as conjunction and implication, can be represented as geometric operations in RKHS. Through practical examples, the paper showcases potential applications in natural language processing and artificial intelligence, where combining geometric and logical reasoning enhances both understanding and computational efficiency.

Overall, this paper sets the groundwork for further exploration into the intersections of geometry, logic, and semantics, offering promising directions for future theoretical and applied research in complex information systems.

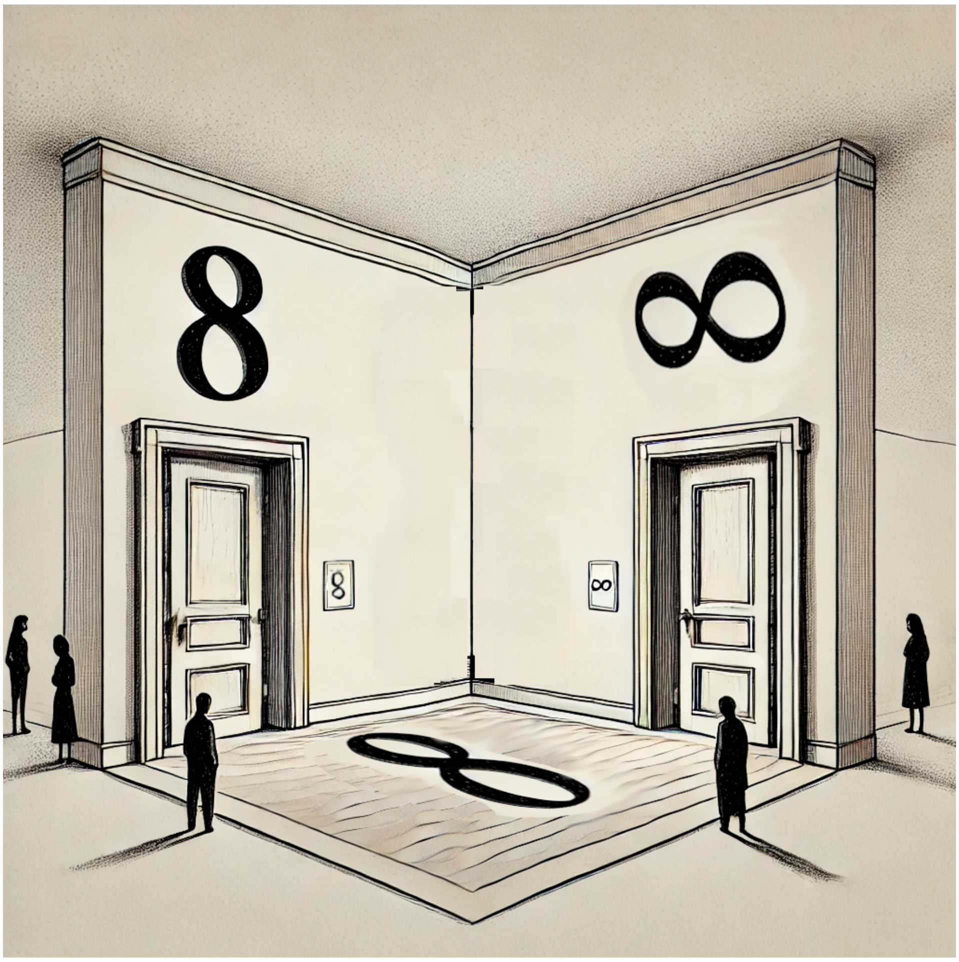

Abbildung 1: Semantic space of logic

Content

1 Introduction and aim

Disclaimer: This writing has been written by the author in dialogue with a chat-assistant to correct the wording and input new impulses. All errors which remain, are in sole responsibility of the author, so it would be nice, if an error is found to report it to the author via email, so it can be corrected.

This paper aims to initiate the development of a semantic geometry of logic by utilizing the theoretical framework of Reproducing Kernel Hilbert Spaces (RKHS) [7]. I am thankful to Prof. Saburou Saitoh for the correspondence with suggestions and discussions around the topic of RKHS. To anchor this abstract concept within practical applications, we provide a series of examples. We begin by considering a set X, which can represent various entities such as words, real-world objects, abstract objects, or natural numbers. For this set, we assume that we are given a positive semi-definite kernel function k, where $k : X \times X \to [-1, 1]$, satisfying $k(x, x) = 1$ for all $x \in X$.

The kernel trick, a widely used method in machine learning, allows efficient computations in infinite-dimensional spaces while employing only finite resources. This method has been notably applied in support vector machines for solving classification problems. Implicit in this approach is a geometric structure that underlies the computations. To the best of our knowledge, the first formal exploration of a geometry for logic was undertaken by David Miller and Jonathan Westphal in their works 'A Geometry of Logic' and 'Logic of Vectors'. I am thankful to David Miller for pointing to the articles by Thomas Mormann: 'Geometry of Logic and Truth Approximation'. In R. Festa, A. Aliseda, & J. Peijnenburg, editors (2005), pp. 429–452. Confirmation, Empirical Progress, and Truth Approximation. Amsterdam & Atlanta: Editions Rodopi B.V.

'Truthlikeness for Theories on Countable Languages'. In I.C. Jarvie, K.M. Milford, & D.W. Miller, editors (2006), pp.3-15. Karl Popper. A Centenary Assessment. Volume

III, Science. Aldershot & Burlington VT: Ashgate. Paperback edition. London: College Publications.

In both articles a similar route is taken as is being developed in this writing.

2 Definitions and first properties

2.1 Motivation:

The main motivation for these definitions comes from the following situation:

Let X be a finite set and $k : X \times X \to \mathbb{R}$ be a positive semi-definite, symmetric function with $k(x,x) = 1$ and $-1 \leq k(x,y) \leq 1$. Then, by the Moore-Aronszajn theorem, or since X is finite, Cholesky-decomposition will do, there exists a map $\phi : X \to H$, where H is a Hilbert space, such that

$$k(x,y) = \langle \phi(x), \phi(y) \rangle.$$

Now, let $w \in X$, so that $1 = k(w,w) = |\phi(w)|^2 = |\phi(w)|$. Then for each $x \in X$, we have:

$$k(w,x)\phi(w) = \langle \phi(w), \phi(x) \rangle \phi(w) = t\phi(w) =: \alpha,$$

where I have set $t := k(w,x)$, and thus $-1 \leq t \leq 1$. I call the resulting vector $t\phi(w) = \alpha$, which is an element of $G_w := \{t\phi(w) \mid -1 \leq t \leq 1\}$.

The following definitions are inspired by Lukasiewicz logic and the idea is to represent a perspective in logic by a unit vector $\phi(w)$ and projections of other unit vectors give rise to a logic like the Lukasiewicz logic but which depends on the perspective vector $\phi(w)$:

For a set X and $w \in X$ such that $\phi(w) \in H$ where H is a Hilbert space and such that$|\phi(w)| = 1$ we define $G_w := \{t\phi(w) \mid -1 \leq t \leq 1\}$.

For $\alpha, \beta \in G_w$ we define:

$$\alpha \wedge \beta := \min(\langle \phi(w), \alpha \rangle, \langle \phi(w), \beta \rangle)\phi(w)$$

$$\alpha \vee \beta := \max(\langle \phi(w), \alpha \rangle, \langle \phi(w), \beta \rangle)\phi(w)$$

$$\alpha \to \beta := \min(1, 1 + \langle \phi(w), \beta \rangle - \langle \phi(w), \alpha \rangle)\phi(w)$$

$$\alpha \leftrightarrow \beta := (\alpha \to \beta) \wedge (\beta \to \alpha)$$

$$\neg\alpha := -\alpha$$

$$\mu(\alpha) = \begin{cases} T & \text{if } \langle \phi(w), \alpha \rangle > 0, \\ I & \text{if } \langle \phi(w), \alpha \rangle = 0, \\ F & \text{if } \langle \phi(w), \alpha \rangle < 0. \end{cases}$$

In the case of $\alpha := \langle \phi(w), \phi(x) \rangle \phi(w) = k(w,x)\phi(w)$ and $\beta := \langle \phi(w), \phi(y) \rangle \phi(w) = k(w,y)\phi(w)$, we have, due to

$$\begin{aligned} \langle \phi(w), \alpha \rangle &= \langle \phi(w), \langle \phi(w), \phi(x) \rangle \phi(w) \rangle \\ &= \langle \phi(w), \phi(w) \rangle \cdot \langle \phi(w), \phi(x) \rangle = 1 \cdot \langle \phi(w), \phi(x) \rangle \\ &= \langle \phi(w), \phi(x) \rangle \end{aligned}$$

that

$$\begin{aligned} \alpha \wedge \beta &= \min(\langle \phi(w), \alpha \rangle, \langle \phi(w), \beta \rangle)\phi(w) \\ &= \min(\langle \phi(w), \phi(x) \rangle, \langle \phi(w), \phi(y) \rangle)\phi(w) = \min(k(w,x), k(w,y))\phi(w) \end{aligned}$$

which is how we define $x \wedge y \operatorname{rel} w$.

We have the following properties:

1. Double Negation

Statement:

$$\neg(\neg\alpha) = \alpha$$

Proof: Given that $\neg\alpha = -\alpha$, applying negation twice yields:

$$\neg(\neg\alpha) = \neg(-\alpha) = -(-\alpha) = \alpha$$

Hence, double negation holds.

2. De Morgan's Laws

First Law:

$$\neg(\alpha \wedge \beta) = \neg\alpha \vee \neg\beta$$

Proof:

$$\neg(\alpha \wedge \beta) = -\min(\langle \phi(w), \alpha \rangle, \langle \phi(w), \beta \rangle)\phi(w)$$

$$\neg\alpha \vee \neg\beta = \max(\langle \phi(w), -\alpha \rangle, \langle \phi(w), -\beta \rangle)\phi(w) = \max(-\langle \phi(w), \alpha \rangle, -\langle \phi(w), \beta \rangle)\phi(w)$$

Since $\max(-a, -b) = -\min(a, b)$, the equality holds.

Second Law:

$$\neg(\alpha \vee \beta) = \neg\alpha \wedge \neg\beta$$

Proof: Similar reasoning as above shows that:

$$\neg(\alpha \vee \beta) = -\max(\langle \phi(w), \alpha \rangle, \langle \phi(w), \beta \rangle)\phi(w)$$

$$\neg\alpha \wedge \neg\beta = \min(\langle \phi(w), -\alpha \rangle, \langle \phi(w), -\beta \rangle)\phi(w) = \min(-\langle \phi(w), \alpha \rangle, -\langle \phi(w), \beta \rangle)\phi(w)$$

Since $\min(-a, -b) = -\max(a, b)$, the equality holds.

3. Associativity and Commutativity

Associativity: Both $\wedge$ and $\vee$ are associative because the min and max functions are associative.

$$\alpha \wedge (\beta \wedge \gamma) = (\alpha \wedge \beta) \wedge \gamma$$

$$\alpha \vee (\beta \vee \gamma) = (\alpha \vee \beta) \vee \gamma$$

Commutativity: Both $\wedge$ and $\vee$ are commutative.

$$\alpha \wedge \beta = \beta \wedge \alpha$$

$$\alpha \vee \beta = \beta \vee \alpha$$

4. Distributivity

Distributive Laws:

$$\alpha \wedge (\beta \vee \gamma) = (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$$

$$\alpha \vee (\beta \wedge \gamma) = (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$$

Proof: These follow from the distributive properties of min and max:

$$\min(a, \max(b, c)) = \max(\min(a, b), \min(a, c))$$

$$\max(a, \min(b, c)) = \min(\max(a, b), \max(a, c))$$

5. Idempotency

Statements:

$$\alpha \wedge \alpha = \alpha$$

$$\alpha \vee \alpha = \alpha$$

Proof: Since $\min(a, a) = a$ and $\max(a, a) = a$, idempotency holds.

6. Law of Included Middle

Statement:

$$\mu(\alpha \vee \neg\alpha) = T \text{ or } = I$$

Explanation: $\alpha \vee \neg\alpha = \max(\langle \phi(w), \alpha \rangle, \langle \phi(w), -\alpha \rangle)\phi(w) = |\langle \phi(w), \alpha \rangle| \phi(w)$. Therefore, $\mu(\alpha \vee \neg\alpha)$ maps to T if $\langle \phi(w), \alpha \rangle \neq 0$ and I if $\langle \phi(w), \alpha \rangle = 0$. This is similar to the classical law (Law of excluded middle) but includes an intermediate state I which stands for 'neutral / indeterminate / unknown'.

7. Biconditional

We have:

$$\alpha \leftrightarrow \beta = (1 - |\langle \phi(w), \alpha - \beta \rangle|)\phi(w)$$

Proof: Let $\alpha = t\phi(w)$, $\beta = t'\phi(w)$. Then we have:

$$\alpha \leftrightarrow \beta = \min(1, 1 + t' - t)\phi(w) \wedge \min(1, 1 + t - t')\phi(w)$$
$$= \min(\min(1, 1 + t' - t), \min(1, 1 + t - t'))\phi(w) = (1 - |t - t'|)\phi(w)$$
$$= (1 - |\langle \phi(w), \alpha - \beta \rangle|)\phi(w)$$

8. Tolerance relation on G_w

Statement: The relation $\equiv$ defined through

$$\alpha \equiv \beta :\iff \mu(\alpha \leftrightarrow \beta) = T$$

is symmetric and reflexive, hence a tolerance relation on G_w. **Proof:** Reflexive: By '7. Biconditional' we have:

$$\alpha \leftrightarrow \alpha = (1 - |\langle \phi(w), \alpha - \alpha \rangle|)\phi(w) = \phi(w)$$

hence $\mu(\alpha \leftrightarrow \alpha) = T$ so $\alpha \equiv \alpha$.
Symmetry: We have by '7. Biconditional':

$$\alpha \leftrightarrow \beta = (1 - |\langle \phi(w), \alpha - \beta \rangle|)\phi(w)$$
$$= (1 - |-\langle \phi(w), -\alpha + \beta \rangle|)\phi(w)$$
$$= (1 - |\langle \phi(w), \beta - \alpha \rangle|)\phi(w) = \beta \leftrightarrow \alpha$$

hence $T = \mu(\alpha \leftrightarrow \beta) = \mu(\beta \leftrightarrow \alpha)$ so $\alpha \equiv \beta \implies \beta \equiv \alpha$.

Comment

The relation is in general not transitive.

9. Law of Non-Contradiction

Statement:

$$\mu(\alpha \wedge \neg\alpha) \neq T$$

Proof: Compute $\alpha \wedge \neg\alpha$:

$$\alpha \wedge \neg\alpha = \min(\langle \phi(w), \alpha \rangle, \langle \phi(w), -\alpha \rangle)\phi(w)$$

Let $t = \langle \phi(w), \alpha \rangle$. Then:

$$\alpha \wedge \neg\alpha = \min(t, -t)\phi(w) = -|t|\phi(w)$$

Thus:

$$\mu(\alpha \wedge \neg\alpha) = \begin{cases} F & \text{if } t \neq 0, \\ I & \text{if } t = 0. \end{cases}$$

In both cases, $\mu(\alpha \wedge \neg\alpha) \neq T$. Therefore, the conjunction of a statement and its negation never maps to true, aligning with the Law of Non-Contradiction.

10. Contraposition

Statement:

$$\alpha \to \beta = \neg\beta \to \neg\alpha$$

Proof: Using the definition of implication:

$$\alpha \to \beta = \min(1, 1 + \langle \phi(w), \beta \rangle - \langle \phi(w), \alpha \rangle)\phi(w)$$

$$\begin{aligned} \neg\beta \to \neg\alpha &= \min(1, 1 + \langle \phi(w), \neg\alpha \rangle - \langle \phi(w), \neg\beta \rangle)\phi(w) \\ &= \min(1, 1 + \langle \phi(w), -\alpha \rangle - \langle \phi(w), -\beta \rangle)\phi(w) \\ &= \min(1, 1 - \langle \phi(w), \alpha \rangle + \langle \phi(w), \beta \rangle)\phi(w) \\ &= \min(1, 1 + \langle \phi(w), \beta \rangle - \langle \phi(w), \alpha \rangle)\phi(w) \\ &= \alpha \to \beta \end{aligned}$$

Hence, $\alpha \to \beta = \neg\beta \to \neg\alpha$, demonstrating that contraposition holds in this framework.

11. Law of Identity

Statement:

$$\mu(\alpha \to \alpha) = T$$

Proof: Using the definition of implication:

$$\alpha \to \alpha = \min(1, 1 + \langle \phi(w), \alpha \rangle - \langle \phi(w), \alpha \rangle)\phi(w) = \min(1, 1)\phi(w) = \phi(w)$$

Thus:

$$\mu(\alpha \to \alpha) = \mu(\phi(w)) = T \quad (\text{since } \langle \phi(w), \phi(w) \rangle = 1 > 0)$$

Hence, the implication of any statement with itself always maps to true, analogous to the Law of Identity in classical logic.

12. Disjunctive Syllogism

Statement: If $\mu(\alpha \vee \beta) = T$ and $\mu(\neg\alpha) = T$, then $\mu(\beta) = T$.

Proof: Given $\mu(\alpha \vee \beta) = T$, we have:

$$\max(\langle \phi(w), \alpha \rangle, \langle \phi(w), \beta \rangle) > 0$$

And $\mu(\neg\alpha) = T$ implies:

$$\langle \phi(w), \neg\alpha \rangle = \langle \phi(w), -\alpha \rangle = -\langle \phi(w), \alpha \rangle > 0 \implies \langle \phi(w), \alpha \rangle < 0$$

Substituting into the first condition:

$$\max(\langle \phi(w), \alpha \rangle, \langle \phi(w), \beta \rangle) = \max(\text{a negative number}, \langle \phi(w), \beta \rangle) > 0$$

This implies:

$$\langle \phi(w), \beta \rangle > 0 \implies \mu(\beta) = T$$

Hence, 'Disjunctive Syllogism' holds in this framework.

For $x, y, z \in X$, w such that $|\phi(w)| = 1$ and $\alpha := k(w,x)\phi(w), \beta := k(w,y)\phi(w), \gamma := k(w,z)\phi(w)$, the preceding definitons become, (please see the section 'Proofs' before the 'Appendix with SageMath/Python' code):

$$\pi_w(x) = \frac{\langle \phi(w), \phi(x) \rangle}{\langle \phi(w), \phi(w) \rangle} \phi(w) =^{1=<\phi(w),\phi(w)>} \langle \phi(w), \phi(x) \rangle \, \phi(w) = k(w,x)\phi(w) \tag{1}$$

For this last equation, we write:

$$x \operatorname{rel} w := x \text{ (relative to } w) := \pi_w(x) = k(w,x)\phi(w) \tag{2}$$

Now we can imagine what the meaning of 'I have a different perspective' means. We simply associate the projection of the semantic vector of x to some other perspective vector w. Then 'changing perspective' simply means to change the perspective vector from w to say $\hat{w}$.

Let us define

$$(x \wedge y) \operatorname{rel} w := \min(k(w,x), k(w,y))\phi(w) \tag{3}$$

$$(x \vee y) \operatorname{rel} w := \max(k(w,x), k(w,y))\phi(w) \tag{4}$$

$$(\neg x) \operatorname{rel} w := -\pi_w(x) \tag{5}$$

and following Lukasiewicz:

$$(x \to y) \operatorname{rel} w := \min(1, 1 + k(w,y) - k(w,x))\phi(w) \tag{6}$$

$$(x \leftrightarrow y) \operatorname{rel} w := (x \equiv y) \operatorname{rel} w := (x \rightarrow y) \wedge (y \rightarrow x) \operatorname{rel} w \tag{7}$$

We have the de Morgan rules, double negation, contraposition:

$$(\neg x) \wedge (\neg y) \operatorname{rel} w = \neg(x \vee y) \operatorname{rel} w \tag{8}$$

$$(\neg x) \vee (\neg y) \operatorname{rel} w = \neg(x \wedge y) \operatorname{rel} w \tag{9}$$

$$(\neg(\neg x)) \operatorname{rel} w = x \operatorname{rel} w \tag{10}$$

$$(x \rightarrow y) \operatorname{rel} w = ((\neg y) \rightarrow (\neg x)) \operatorname{rel} w \tag{11}$$

Let us define three cases of truth values: T, F, I which should be translated to True, False, Indeterminate. For all $h \in$ RKHS space H, we have:

$$\mu(h) = \begin{cases} T & \text{if } \langle \phi(w), h \rangle > 0, \\ I & \text{if } \langle \phi(w), h \rangle = 0, \\ F & \text{if } \langle \phi(w), h \rangle < 0. \end{cases} \tag{12}$$

We have a version of 'Modus ponens':

$$\text{If } \mu(x \operatorname{rel} w) = T \text{ and } \mu(x \rightarrow y \operatorname{rel} w) = T \text{ then } \mu((x \wedge (x \rightarrow y)) \rightarrow y) \operatorname{rel} w) = T \tag{13}$$

3 Separable semantic space

A finite semantic space $S = (X, k)$ with labels $y_i = \pm 1$ for each $x_i \in X, 1 \leq i \leq n$ is called separable by the y_i if there exist a w such that $\hat{X} := X \cup \{w\}$, $\hat{k} : \hat{X} \times \hat{X} \rightarrow \mathbb{R}$ is a positive semi-definite function on $\hat{X}$ which extends k, that is: $\hat{k}(x, y) = k(x, y) \forall x, y \in X$ and such that $\hat{S} := (\hat{X}, \hat{k})$ is a semantic space, such that the following holds:

$$\forall 1 \leq i \leq n : \operatorname{sign}(\hat{k}(x_i, w)) = y_i \in \{\pm 1\}$$

If $S = (X, k)$ is separable by $Y = \{y_i | 1 \leq i \leq n\}$, then there exists a w such that:

$$\forall 1 \leq i \leq n : \mu(x_i \operatorname{rel} w) = T, \text{ if } y_i > 0, \text{ else } F$$

The proof, consists in the definition of μ and because the sign can, by separability assumption, take only values ± 1.

4 From separable semantic space to Boolean algebras

First we begin with a few properties which are easy to prove: Let $S = (X, k)$ be a semantic space and $\phi(w) \neq 0$ be a perspective vector. Then the following are immediate from the definitions:

- $\pi_w(w) = \frac{k(w,w)}{|\phi(w)|^2} = \frac{1}{1}\phi(w) = \phi(w)$
- $w \operatorname{rel} w = \phi(w)$
- $(w \wedge x) \operatorname{rel} w = x \operatorname{rel} w$
- $(w \vee x) \operatorname{rel} w = w \operatorname{rel} w$
- $(\neg w) \operatorname{rel} w = -\pi_w(w) = -\phi(w)$
- $((\neg w) \wedge x) \operatorname{rel} w = (\neg w) \operatorname{rel} w$
- $((\neg w) \vee x) \operatorname{rel} w = x \operatorname{rel} w$

For instance with $\mathbf{1} \operatorname{rel} w := w \operatorname{rel} w$ and $\mathbf{0} \operatorname{rel} w := (\neg w) \operatorname{rel} w$, we have:

$$(\mathbf{1} \wedge x) \operatorname{rel} w = x \operatorname{rel} w \tag{14}$$

because,

$$\begin{aligned}(\mathbf{1} \wedge x) \operatorname{rel} w = (w \wedge x) \operatorname{rel} w = \\ = \min(k(w,w), k(w,x))\phi(x) = \min(1, k(w,x))\phi(w) = \\ = k(w,x)\phi(w) = x \operatorname{rel} w\end{aligned}$$

and similarliy

$$(\mathbf{0} \vee x) \operatorname{rel} w = w \operatorname{rel} w \tag{15}$$

because,

$$\begin{aligned}(\mathbf{0} \vee x) \operatorname{rel} w = ((\neg w) \vee x) \operatorname{rel} w = \\ = \max(k(w, \neg w), k(w,x))\phi(x) = \max(-k(w,w), k(w,x))\phi(w) = \\ = \max(-1, k(w,x))\phi(w) = k(w,x)\phi(w) = x \operatorname{rel} w\end{aligned}$$

Let now $S = (X, k)$ be a separable space by $Y := \{y_i | 1 \leq i \leq |X| = n\}$ and let $\phi(w)$ be a perspective vector such that:

$$\forall 1 \leq i \leq n : \operatorname{sign}(\langle \phi(x_i), \phi(w) \rangle = \operatorname{sign}(k(x_i, w)) = \pm 1 =^! y_i$$

Then the semantic space $\hat{S} := (\hat{X}, \hat{k})$ with $\hat{X} := X \cup \{w, \neg w\}$ give rise to a Boolean algebra $A = (\hat{X}, \mathbf{1} = w, \mathbf{0} = \neg w, \wedge, \vee)$.

Proof: First we recall the definiton of Boolean algebra:

A **Boolean algebra** is a set A, equipped with two binary operations $\wedge$ (called "meet" or "and"), $\vee$ (called "join" or "or"), a unary operation $\neg$ (called "complement" or "not") and two elements 0 and 1 in A (called "bottom" and "top", or "least" and "greatest" element, also denoted by the symbols $\perp$ and $\top$, respectively), such that for all elements a, b, and c of A, the following axioms hold:

$\mathbf{a} \vee (\mathbf{b} \vee \mathbf{c}) = (\mathbf{a} \vee \mathbf{b}) \vee \mathbf{c}$	$\mathbf{a} \wedge (\mathbf{b} \wedge \mathbf{c}) = (\mathbf{a} \wedge \mathbf{b}) \wedge \mathbf{c}$	**associativity**
$\mathbf{a} \vee \mathbf{b} = \mathbf{b} \vee \mathbf{a}$	$\mathbf{a} \wedge \mathbf{b} = \mathbf{b} \wedge \mathbf{a}$	**commutativity**
$\mathbf{a} \vee (\mathbf{a} \wedge \mathbf{b}) = \mathbf{a}$	$\mathbf{a} \wedge (\mathbf{a} \vee \mathbf{b}) = \mathbf{a}$	**absorption**
$\mathbf{a} \vee \mathbf{0} = \mathbf{a}$	$\mathbf{a} \wedge \mathbf{1} = \mathbf{a}$	**identity**
$\mathbf{a} \vee (\mathbf{b} \wedge \mathbf{c}) = (\mathbf{a} \vee \mathbf{b}) \wedge (\mathbf{a} \vee \mathbf{c})$	$\mathbf{a} \wedge (\mathbf{b} \vee \mathbf{c}) = (\mathbf{a} \wedge \mathbf{b}) \vee (\mathbf{a} \wedge \mathbf{c})$	**distributivity**
$\mathbf{a} \vee \neg\mathbf{a} = \mathbf{1}$	$\mathbf{a} \wedge \neg\mathbf{a} = \mathbf{0}$	**complements**

Then we proceed: Commutativity, associativity are true because $\min, \max$ are commutative and associative. Absorption and distributivity are true, beacuse $\{\min, \max\}$ over the real numbers is a distributive lattice. We prove the complements, as the identity has been proven for all semantic spaces:

$$(x \vee (\neg x)) \operatorname{rel} w = \max(k(w,x) - k(w,x))\phi(w) =$$
$$|k(w,x)|\phi(w) =^{\text{separable}} |\pm 1|\phi(w) =$$
$$\phi(w) = w \operatorname{rel} w = \mathbf{1} \operatorname{rel} w$$

Similarily:

$$(x \wedge (\neg x)) \operatorname{rel} w = \min(k(w,x) - k(w,x))\phi(w) =$$
$$-|k(w,x)|\phi(w) =^{\text{separable}} -|\pm 1|\phi(w) =$$
$$-\phi(w) = (\neg w) \operatorname{rel} w = \mathbf{0} \operatorname{rel} w$$

4.1 Construction of separable semantic spaces

Idea: Given a finite set X with a positive definite kernel on X, such that $S = (X, k)$ is a semantic space, then the Gram matrix is positive definite and so invertible. Let now $Y = \{y_i = \pm 1 | 1 \leq i \leq |X| = n\}$ be 'any' labeling of the x_i. Then we can find an extension $\hat{k} : \hat{X} \times \hat{X} \to \mathbb{R}$ of k, where $\hat{X} := X \cup w, w \notin X$ such that, there exist $c_i \in \mathbb{R}$ with:

$$\forall 1 \leq i \leq n : y_i = \operatorname{sign}(\left[\sum_{j=1}^{n} c_j y_j k(x_i, x_j)\right]) \tag{16}$$

This would allow us to define:

$$\hat{k}(w, x_i) := \sum_{j=1}^{n} c_j y_j k(x_i, x_j)$$

and so we would get a separable semantic space:

$$\forall 1 \leq i \leq n : y_i = \text{sign}(\hat{k}(w, x_i))$$

But because the Gram matrix $G = (k(x_i, x_j))_{i,j}$ is invertible, the first equation with the c_j can be solved by setting:

$$c := G^{-1} y$$

where $c = (c_1, \cdots, c_n), y = (y_1, \cdots, y_n)$. The same strategy has been used to show, that 'primes are linearly separable': Are primes linearly separable?

To make things more concrete, here is a semantic space with n elements, which is separable for every Y:

- $X = \{1, \cdots, n\}$
- $k(i, j) = \frac{\gcd(i,j)^2}{ij}$
- Let $Y = \{y_i = \pm 1 | 1 \leq i \leq n\}$ be any set on n elements consisting of ± 1.

Then set

$$c := G^{-1} y$$

and put

$$\hat{k}_0(w, i) := \sum_{j=1}^{n} c_j y_j k(i, j)$$

$$\hat{k}_0(w, w) := \sum_{1 \leq i,j \leq n} c_i y_j c_j y_j k(i, j)$$

$$\hat{k}_0(i, j) := k(i, j) \ \forall 1 \leq i, j \leq n$$

where $G = (k(i, j))_{1 \leq i,j \leq n}$ is the Gram matrix, which is shown below, to have non-zero determinant. We have to normalize the kernel:

$$\hat{k}(w, i) := \frac{\hat{k}_0(w, i)}{\sqrt{\hat{k}_0(w, w) k(i, i)}}$$

$$\hat{k}(w, w) := 1$$

By the discussion above this method produces a semantic space S which given Y is separable for Y. This concludes also the proof, that Boolean algebras can be constructed using semantic spaces.

5 Semantic spaces of logic from finite groups

It is possible to construct to a finite group G a semantic space $S := (G, k)$ where the elements of $X = G$ are the group elements and k is a positive semidefinite kernel on the group. Here is the construction of the kernel:

Let G be a finite group with $n = |G|$ elements. By Cayley's theorem for finite groups, we have an injective homomorphism of groups:

$$\pi : G \to S_n, g \mapsto \pi(g) \tag{17}$$

where each group element g is mapped to the permutation of the symmetric group S_n on $n = |G|$ elements, wich it generates by left multiplication:

$$\pi(g) : G \to G, x \mapsto g \cdot x$$

We want to associate to each group element g a matrix and then use the Frobenius inner product to define a positive semi-definite function k on the group G as follows:

It is known , see for instance "The Kendall and Mallows Kernels for Permutations"by Yunlong Jiao and Jean-Philippe Vert, that the Kendall-tau function can be made to a positive semi-definite kernel k for permutations.

Let $\mathbf{1}_{\{x\}}$ be the indicator function, which is $= 1$ if the boolean variable x is true and 0 if the boolean variable x is false, and let:

$$\phi : S_n \to M_n(\mathbb{R}), \sigma \mapsto (\mathbf{1}_{\{\sigma(i)>\sigma(j)\}} - \mathbf{1}_{\{\sigma(i)<\sigma(j)\}})_{1\leq i,j\leq n} \tag{18}$$

where $M_n(\mathbb{R})$ denotes $n \times n$ matrices over $\mathbb{R}$.

The embedding from the finite group G to $M_n(\mathbb{R})$ is then given by:

$$\psi : G \to M_n(\mathbb{R}), g \mapsto \frac{1}{\sqrt{n(n-1)}} \cdot \phi(\pi(g)) \tag{19}$$

We use the Frobenius inner product on $M_n(\mathbb{R})$, which is given by:

$$\langle A, B\rangle := \mathrm{tr}(A \cdot B^T) \tag{20}$$

to define a positive semi-definite, symmetric function k on G:

$$k : G \times G \to \mathbb{R}, (g, h) \mapsto \mathrm{tr}(\psi(g) \cdot \psi(h)^T) \tag{21}$$

The kernel k can be normalized to take value between -1 and 1 and since $k(g, g)$ is constant for all $g \in G$, we can normalize k to take values $k(g, g) = 1$ and so we get a semantic space of logic given a finite group G.

It remains in each case to specify a perspective vector w.

5.1 Some boolean algebras from elementary abelian groups of order 2^m

Conjecture: Taking as a group G an elementary abelian group of order 2^m and doing the construction with the kernel, and taking as perspective vector $w = e \in G$ the neutral element of G, then we get a boolean algebra as defined above.

Here are some examples:

5.1.1 $|G| = 2$

Tabelle 1: Gram Matrix K

1	-1
-1	1

Tabelle 2: Truth values of vectors

[1, 2]	T
[2, 1]	F

Tabelle 3: Truth Table for AND (Perspective w = ())

	()	(1,2)
()	T	F
(1,2)	F	F

Tabelle 4: Truth Table for OR (Perspective w = ())

	()	(1,2)
()	T	T
(1,2)	T	F

Tabelle 5: Truth Table for IMPLIES (Perspective w = ())

	()	(1,2)
()	T	T
(1,2)	F	T

Tabelle 6: Truth Table for IFF (Perspective w = ())

	()	(1,2)
()	T	F
(1,2)	F	T

5.1.2 $G =$ Klein-Four group

Tabelle 7: Gram Matrix K

1	1/3	-1/3	-1
1/3	1	-1	-1/3
-1/3	-1	1	1/3
-1	-1/3	1/3	1

We interpret the value 1/3 of the projection $k(w,x)\phi(w)$ of x in direction of the perspective vector $\phi(w)$ as being 33 percent = 1/3 sure , that the vector x has truth value 'True'.

We interpret the value $-1/3$ of the projection $k(w,x)\phi(w)$ of x in direction of the perspective vector $\phi(w)$ as being 33 percent = 1/3 sure , that the vector x has truth value 'False'.

If one is forced to take the values 'True','Indeterminate', 'False' (T, I, F), then one can take the sign of the projection. Otherwise, one can interpret the real value of the projection on the perspective vector as a 'fuzzy' version of 'True','False'. The 'Indeterminate' value can occur only when the projection is $= 0$, so there is not much nuanced interpretation there.

Tabelle 8: Truth values of vectors (Perspective w = ())

[1, 2, 3, 4]	T
[2, 1, 4, 3]	T
[3, 4, 1, 2]	F
[4, 3, 2, 1]	F

Tabelle 9: Truth Table for AND (Perspective w = ())

	()	(3,4)	(1,2)	(1,2)(3,4)
()	T	T	F	F
(3,4)	T	T	F	F
(1,2)	F	F	F	F
(1,2)(3,4)	F	F	F	F

Tabelle 10: Truth Table for OR (Perspective w = ())

	()	(3,4)	(1,2)	(1,2)(3,4)
()	T	T	T	T
(3,4)	T	T	T	T
(1,2)	T	T	F	F
(1,2)(3,4)	T	T	F	F

Tabelle 11: Truth Table for IMPLIES (Perspective w = ())

	()	(3,4)	(1,2)	(1,2)(3,4)
()	T	T	T	T
(3,4)	T	T	T	T
(1,2)	F	T	T	T
(1,2)(3,4)	F	F	T	T

Tabelle 12: Truth Table for IFF (Perspective w = ())

	()	(3,4)	(1,2)	(1,2)(3,4)
()	T	T	F	F
(3,4)	T	T	T	F
(1,2)	F	T	T	T
(1,2)(3,4)	F	F	T	T

5.1.3 $|G| = 8$ and G is elementary abelian

Tabelle 13: Gram Matrix K

1	5/7	3/7	1/7	-1/7	-3/7	-5/7	-1
5/7	1	1/7	3/7	-3/7	-1/7	-1	-5/7
3/7	1/7	1	5/7	-5/7	-1	-1/7	-3/7
1/7	3/7	5/7	1	-1	-5/7	-3/7	-1/7
-1/7	-3/7	-5/7	-1	1	5/7	3/7	1/7
-3/7	-1/7	-1	-5/7	5/7	1	1/7	3/7
-5/7	-1	-1/7	-3/7	3/7	1/7	1	5/7
-1	-5/7	-3/7	-1/7	1/7	3/7	5/7	1

Tabelle 14: Truth values of vectors (Perspective w = ())

[1, 2, 3, 4, 5, 6, 7, 8]	T
[2, 1, 4, 3, 6, 5, 8, 7]	T
[3, 4, 1, 2, 7, 8, 5, 6]	T
[4, 3, 2, 1, 8, 7, 6, 5]	T
[5, 6, 7, 8, 1, 2, 3, 4]	F
[6, 5, 8, 7, 2, 1, 4, 3]	F
[7, 8, 5, 6, 3, 4, 1, 2]	F
[8, 7, 6, 5, 4, 3, 2, 1]	F

Tabelle 15: Truth Table for AND (Perspective w = ())

	()	(5,6)	(3,4)	(3,4)(5,6)	(1,2)	(1,2)(5,6)	(1,2)(3,4)	(1,2)(3,4)(5,6)
()	T	T	T	T	F	F	F	F
(5,6)	T	T	T	T	F	F	F	F
(3,4)	T	T	T	T	F	F	F	F
(3,4)(5,6)	T	T	T	T	F	F	F	F
(1,2)	F	F	F	F	F	F	F	F
(1,2)(5,6)	F	F	F	F	F	F	F	F
(1,2)(3,4)	F	F	F	F	F	F	F	F
(1,2)(3,4)(5,6)	F	F	F	F	F	F	F	F

Tabelle 16: Truth Table for OR (Perspective w = ())

	()	(5,6)	(3,4)	(3,4)(5,6)	(1,2)	(1,2)(5,6)	(1,2)(3,4)	(1,2)(3,4)(5,6)
()	T	T	T	T	T	T	T	T
(5,6)	T	T	T	T	T	T	T	T
(3,4)	T	T	T	T	T	T	T	T
(3,4)(5,6)	T	T	T	T	T	T	T	T
(1,2)	T	T	T	T	F	F	F	F
(1,2)(5,6)	T	T	T	T	F	F	F	F
(1,2)(3,4)	T	T	T	T	F	F	F	F
(1,2)(3,4)(5,6)	T	T	T	T	F	F	F	F

Tabelle 17: Truth Table for IMPLIES (Perspective w = ())

	()	(5,6)	(3,4)	(3,4)(5,6)	(1,2)	(1,2)(5,6)	(1,2)(3,4)	(1,2)(3,4)(5,6)
()	T	T	T	T	T	T	T	T
(5,6)	T	T	T	T	T	T	T	T
(3,4)	T	T	T	T	T	T	T	T
(3,4)(5,6)	T	T	T	T	T	T	T	T
(1,2)	F	T	T	T	T	T	T	T
(1,2)(5,6)	F	F	T	T	T	T	T	T
(1,2)(3,4)	F	F	F	T	T	T	T	T
(1,2)(3,4)(5,6)	F	F	F	F	T	T	T	T

Tabelle 18: Truth Table for IFF (Perspective w = ())

	()	(5,6)	(3,4)	(3,4)(5,6)	(1,2)	(1,2)(5,6)	(1,2)(3,4)	(1,2)(3,4)(5,6)
()	T	T	T	T	F	F	F	F
(5,6)	T	T	T	T	T	F	F	F
(3,4)	T	T	T	T	T	T	F	F
(3,4)(5,6)	T	T	T	T	T	T	T	F
(1,2)	F	T	T	T	T	T	T	T
(1,2)(5,6)	F	F	T	T	T	T	T	T
(1,2)(3,4)	F	F	F	T	T	T	T	T
(1,2)(3,4)(5,6)	F	F	F	F	T	T	T	T

5.2 Some new logic tables based on finite groups

Here we present some new logic tables based on finite groups of order 4 and cyclic:

5.2.1 $G = C_4$ is cyclic of order 4

Tabelle 19: Gram Matrix K

1	0	-1/3	0
0	1	0	-1/3
-1/3	0	1	0
0	-1/3	0	1

Tabelle 20: Truth values of vectors (Perspective w = ())

[1, 2, 3, 4]	T
[2, 3, 4, 1]	I
[3, 4, 1, 2]	F
[4, 1, 2, 3]	I

Tabelle 21: Truth Table for AND (Perspective w = ())

	()	(1,2,3,4)	(1,3)(2,4)	(1,4,3,2)
()	T	I	F	I
(1,2,3,4)	I	I	F	I
(1,3)(2,4)	F	F	F	F
(1,4,3,2)	I	I	F	I

Tabelle 22: Truth Table for OR (Perspective w = ())

	()	(1,2,3,4)	(1,3)(2,4)	(1,4,3,2)
()	T	T	T	T
(1,2,3,4)	T	I	I	I
(1,3)(2,4)	T	I	F	I
(1,4,3,2)	T	I	I	I

Tabelle 23: Truth Table for IMPLIES (Perspective w = ())

	()	(1,2,3,4)	(1,3)(2,4)	(1,4,3,2)
()	T	T	T	T
(1,2,3,4)	I	T	T	T
(1,3)(2,4)	F	T	T	T
(1,4,3,2)	I	T	T	T

Tabelle 24: Truth Table for IFF (Perspective w = ())

	()	(1,2,3,4)	(1,3)(2,4)	(1,4,3,2)
()	T	I	F	I
(1,2,3,4)	I	T	T	T
(1,3)(2,4)	F	T	T	T
(1,4,3,2)	I	T	T	T

6 The Liar Paradox in Semantic Spaces

The Liar Paradox is a classic illustration of self-reference and inconsistency in logic, often summarized by the statement, 'This statement is false.'. If we assume the statement is true, then it must be false as it claims. Conversely, if we assume it is false, then it would paradoxically be true. This paradox highlights the difficulties in dealing with self-referential statements in formal systems. It is known, that using multiple valued logic, one can give the statement 'This statement is false.' an indeterminate logical value different from true or false. What we want to illustrate here, is that it is possible to find a framework, where both the classical paradox and the new interpretation coexist logically possible by altering the perspective.

To illustrate how semantic perspectives can resolve or interpret this paradox differently, let's consider an example involving vectors:

We will be interpreting the Liar paradox like this: We are searching for a sentence or better let us call it formula L whose negation $\neg L$ is equal to L: $\neg L = L$.

The basic idea is to change the perspective or point of view w, so that the truth value or even the formula itself satisfy: $\neg L = L$ (first interpretation) or in terms of truth values $\mu(\neg L) = \mu(L)$ (second interpretation).

Let $X := \{e_1, -e_1, e_2, -e_2\}$,where $e_1 = (0, 1)$, $e_2 = (1, 0)$ are standard basis vectors in $\mathbb{R}^2$ with the usual inner product $k(x, y) = \langle\phi(x), \phi(y)\rangle = \langle x, y\rangle$ and $\phi(x) = x$ given by the identity map . We chose $L = (0, 1) = e_1$ and $w = (1, 0) = e_2$. With this choice, the negation of L relative to w is $\neg L \operatorname{rel} w$ is given by $-L \operatorname{rel} w$ and with this choice, w is perpendicular to L and $-L$, hence this perspective might be suited for 'resolving' the paradox as described above, since it is not in the same subspace spanned by L and $-L$. We compute: $k(w, L) = 0 \implies \mu(L \operatorname{rel} w) = I$ and also $k(w, \neg L) = -k(w, L) = 0 \implies \mu(\neg L \operatorname{rel} w) = I$. Hence

$$L \operatorname{rel} w = \pi_w(L) = k(w, L)\phi(w) = 0$$

$$0 = -k(w, L)\phi(w) = k(w, \neg L)\phi(w) = \pi_w(\neg L) = (\neg L) \operatorname{rel} w$$

and the first interpretation of the Liar paradox is satisfied, and also

$$\mu(L \operatorname{rel} w) = I = \mu(\neg L \operatorname{rel} w)$$

and so the second interpretation of the Liar paradox is satisfied.

If we change the perspective w to $w := L = (0, 1)$ or $w := -L = (0, -1)$ then we get in the first case:

$$k(w, L) = 1 \implies k(w, -L) = -1 \implies \mu(L\,\mathrm{w}) = T \neq F = \mu(\neg L\,\mathrm{w})$$

hence the second interpretation of the Liar paradox can not be satisfied and so, also the first interpretation can not be satisfied.

In the case of $w := -L$ we get again that the Liar paradox can not be satisfied:

$$k(w, L) = -1 \implies k(w, -L) = 1 \implies \mu(L\,\mathrm{w}) = F \neq T = \mu(\neg L\,\mathrm{w})$$

Comment: It seems that one can give different logical meanings to the Liar paradox through the change of the perspective. If one insists that the perspective must lie in the same subspace generated by L and $-L$, then the Liar paradox can not be satisfied as shown above. If one allows a perspective perpendicular to the subspace generated by L and $-L$, then the Liar paradox can be satisfied in some sense, as shown above. Usually in mathematics, it is often the case, that, while some equation might not be solvable in some set or space or mathematical structure such as ring or field, it can be solved in a larger space encompassing the original space as a substructure: $x^2 = -1$ is not solvable in $\mathbb{R}$ but is solvable in $\mathbb{C}$ which contains $\mathbb{R}$, for example.

In fact we have the following, which are not difficult to show: Let $|\phi(w)| = 1 = |\phi(L)|$. Than we have

- $L \operatorname{rel} w = (\neg L) \operatorname{rel} w$
- $\iff \dim(H) > 1$ and $k(L, w) = 0$
- $\iff \dim(H) > 1$ and $\mu(L \operatorname{rel} w) = I$
- $\iff \dim(H) > 1$ and $\mu((\neg L) \operatorname{rel} w) = I$
- $\iff \dim(H) > 1$ and $\mu((\neg L) \operatorname{rel} w) = \mu(L \operatorname{rel} w)$

7 Conceptual Spaces and Semantic Spaces

Conceptual spaces and semantic spaces are both geometric frameworks used to model concepts and their relationships, but they originate from different disciplines and have distinct features. This section explores these two frameworks, their connections through the cosine kernel, and highlights the incorporation of logic in semantic spaces, which is not inherent in conceptual spaces.

7.1 Conceptual Spaces

A *conceptual space* is a geometric structure where each dimension represents a specific quality or property relevant to the concepts being modeled [2]. Concepts are represented as points or regions within this space, and the distance between points reflects the dissimilarity between concepts. The conceptual space framework allows for the modeling of concepts based on their properties and the natural relationships between them. It emphasizes the use of geometric notions like distance and direction to capture conceptual similarity and difference.

7.2 Semantic Spaces

A *semantic space* as defined in this note is often a high-dimensional vector space, such as a Reproducing Kernel Hilbert Space (RKHS) [7], where meanings or concepts are represented as vectors. In semantic spaces, similarity between concepts is measured using inner products or kernel functions.

One distinctive feature of semantic spaces is the incorporation of logical operations. Logic in semantic spaces is facilitated through the use of kernel functions and the geometric interpretation of logical connectives. This allows for operations such as conjunction, disjunction, and implication to be defined in terms of vector operations, which is not inherent in the traditional framework of conceptual spaces.

7.3 Connecting Conceptual Spaces and Semantic Spaces via the Cosine Kernel

The cosine kernel serves as a bridge between conceptual spaces and semantic spaces by providing a way to embed the geometric structure of a conceptual space into a semantic space while preserving similarity relationships.

7.3.1 The Cosine Kernel

The *cosine kernel* is defined for vectors $x, y \in \mathbb{R}^n$ as:

$$k(x, y) = \frac{\langle x, y \rangle}{\|x\| \|y\|}, \tag{22}$$

where $\langle x, y \rangle$ denotes the standard inner product, and $\|x\|$ is the Euclidean norm of x. The cosine kernel measures the cosine of the angle between x and y, providing a normalized similarity measure that depends solely on the orientation of the vectors, not their magnitude.

7.3.2 Properties of the Cosine Kernel

The cosine kernel possesses key properties that make it suitable for embedding conceptual spaces into semantic spaces:

Theorem 7.1. *For all non-zero vectors $x, y \in \mathbb{R}^n$, the cosine kernel satisfies:*

1. $-1 \leq k(x, y) \leq 1$.
2. $k(x, x) = 1$.
3. $k(x, y)$ *is positive semi-definite.*

Beweis. **1. Boundedness ($-1 \leq k(x, y) \leq 1$)**

By the Cauchy-Schwarz inequality, for any vectors x, y:

$$|\langle x, y \rangle| \leq \|x\| \|y\|.$$

Dividing both sides by $\|x\| \|y\|$ (since $x, y \neq 0$, the norms are positive):

$$\left| \frac{\langle x, y \rangle}{\|x\| \|y\|} \right| \leq 1,$$

which implies:

$$-1 \leq k(x, y) \leq 1.$$

2. Self-Similarity ($k(x, x) = 1$)
Computing $k(x, x)$:

$$k(x, x) = \frac{\langle x, x \rangle}{\|x\| \|x\|} = \frac{\|x\|^2}{\|x\|^2} = 1.$$

3. Positive Semi-Definiteness
For any finite set $\{x_i\}_{i=1}^m \subset \mathbb{R}^n \setminus \{0\}$ and real coefficients α_i, consider:

$$\begin{aligned}\sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j k(x_i, x_j) &= \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j \frac{\langle x_i, x_j \rangle}{\|x_i\| \|x_j\|} \\ &= \left\| \sum_{i=1}^m \alpha_i \frac{x_i}{\|x_i\|} \right\|^2 \geq 0.\end{aligned}$$

Since the squared norm is always non-negative, $k(x, y)$ is a positive semi-definite kernel. □

7.3.3 Embedding into a Reproducing Kernel Hilbert Space

By virtue of being a positive semi-definite kernel, the cosine kernel defines a Reproducing Kernel Hilbert Space H with an associated feature map $\phi : \mathbb{R}^n \setminus \{0\} \to H$ such that:

$$k(x, y) = \langle \phi(x), \phi(y) \rangle_H. \tag{23}$$

This embedding preserves the geometric relationships from the conceptual space in the semantic space.

7.4 Differences Highlighted: Logic in Semantic Spaces

While both conceptual and semantic spaces model concepts geometrically, a key difference lies in the incorporation of logic within semantic spaces. Semantic spaces, especially when structured as an RKHS, allow for the definition of logical operations using vector operations and kernel functions. Logical connectives such as conjunction ($\wedge$), disjunction ($\vee$), and implication ($\rightarrow$) can be modeled geometrically.

For example, given vectors representing propositions, one can define logical operations as:

$$\begin{aligned}
\text{Conjunction: } & x \wedge y = \min\{k(w,x), k(w,y)\} \cdot w, \\
\text{Disjunction: } & x \vee y = \max\{k(w,x), k(w,y)\} \cdot w, \\
\text{Negation: } & \neg x = -x,
\end{aligned}$$

where w is a perspective vector in the semantic space.

This incorporation of logic is not inherent in conceptual spaces, which primarily focus on the representation of concepts and their similarities or differences based on properties. Semantic spaces extend this by enabling logical reasoning and operations within the geometric framework.

7.5 Implications and Applications

The ability to perform logical operations in semantic spaces opens up possibilities for applications in natural language processing, knowledge representation, and artificial intelligence. It allows for the combination of geometric representations of meaning with formal logic, facilitating more sophisticated reasoning about concepts.

By embedding conceptual spaces into semantic spaces using the cosine kernel, we can leverage the strengths of both frameworks: the intuitive geometric modeling of concepts and the formal logical operations available in semantic spaces.

8 Examples and possible applications

8.1 Example from a dataset of 'Conceptual spaces'-community

Her is an example inspired by research in philosophy and cognitive science on **Conceptual Spaces** (see for instance the paper of Antti Hautamäki, 'A Perspectivist approach to conceptual spaces' [3]), which are similar to the 'semantic spaces':

The dataset includes properties of animals such as the scaled number of legs and intelligence.

Gram Matrix Here is the Gram matrix G constructed from the dataset after scaling and computing the cosine kernel:

	Legs	Skin Cover	Weight	Intelligence	Speed
Legs	1.00	0.28	0.54	0.43	−0.46
Skin Cover	0.28	1.00	−0.22	0.12	0.43
Weight	0.54	−0.22	1.00	0.76	−0.85
Intelligence	0.43	0.12	0.76	1.00	−0.56
Speed	−0.46	0.43	−0.85	−0.56	1.00

We can perform a Cholesky decomposition on this Gram matrix to find an embedding $\phi(x)$ such that:

$$G_{x,y} = k(x, y) = \langle \phi(x), \phi(y) \rangle$$

where $X = \{\text{Legs}, \text{Skin Cover}, \text{Weight}, \text{Intelligence}, \text{Speed}\}$.

Logical Formulas and Interpretations Let's select **Intelligence** as our perspective vector w and compute the projections of each property onto w using the reproducing property:

$$\text{proj}(w, x) = k(w, x)$$

Computed Projections:

- **Legs**:
$$\text{proj}(w, \text{Legs}) = k(\text{Intelligence}, \text{Legs}) = 0.43$$
- **Skin Cover**:
$$\text{proj}(w, \text{Skin Cover}) = k(\text{Intelligence}, \text{Skin Cover}) = 0.12$$
- **Weight**:
$$\text{proj}(w, \text{Weight}) = k(\text{Intelligence}, \text{Weight}) = 0.76$$
- **Intelligence**:
$$\text{proj}(w, \text{Intelligence}) = k(\text{Intelligence}, \text{Intelligence}) = 1.00$$
- **Speed**:
$$\text{proj}(w, \text{Speed}) = k(\text{Intelligence}, \text{Speed}) = -0.56$$

We interpret these projections as measures of similarity or "degrees of truth" relative to the perspective of intelligence.

Defined Logical Formulas 1. **Formula 1**: Intelligence $\to a$

- **Meaning**: If an animal is intelligent, then property a holds.
- **Implementation**:
$$\text{Implies}(\text{proj}(w, \text{Intelligence}), \text{proj}(w, a))$$

2. **Formula 2**: Legs $\wedge\, a$

- **Meaning**: The animal has legs **and** property a holds.
- **Implementation**:
$$\text{And}(\text{proj}(w, \text{Legs}), \text{proj}(w, a))$$

3. **Formula 3**: Intelligence $\leftrightarrow$ (Speed $\wedge\, a$)

- **Meaning**: The animal is intelligent **if and only if** it is fast **and** property a holds.
- **Implementation**:
$$\text{Iff}(\text{proj}(w, \text{Intelligence}), \text{And}(\text{proj}(w, \text{Speed}), \text{proj}(w, a)))$$

Evaluation and Interpretation of Formulas We evaluate each formula for all properties $a \in X$.

Formula 1: Intelligence $\to a$ For each property a:

- **Legs**:
$$\text{Implies}(1.00, 0.43) = \min(1, 1 + 0.43 - 1.00) = 0.43$$
Interpreted as **True**.
- **Skin Cover**:
$$\text{Implies}(1.00, 0.12) = \min(1, 1 + 0.12 - 1.00) = 0.12$$
Interpreted as **True**.
- **Weight**:
$$\text{Implies}(1.00, 0.76) = \min(1, 1 + 0.76 - 1.00) = 0.76$$
Interpreted as **True**.
- **Intelligence**:
$$\text{Implies}(1.00, 1.00) = \min(1, 1 + 1.00 - 1.00) = 1.00$$
Interpreted as **True**.
- **Speed**:
$$\text{Implies}(1.00, -0.56) = \min(1, 1 - 0.56 - 1.00) = -0.56$$
Interpreted as **False**.

Interpretation:

- **Legs, Skin Cover, Weight, Intelligence**: There's a positive implication from intelligence to these properties, meaning that higher intelligence is associated with these traits.
- **Speed**: The implication is false, indicating that higher intelligence does not imply higher speed—in fact, they are negatively correlated.

Formula 2: Legs $\wedge a$ For each property a:

- **Legs**:
$$\text{And}(0.43, 0.43) = \min(0.43, 0.43) = 0.43$$
Interpreted as **True**.

- **Skin Cover**:
$$\text{And}(0.43, 0.12) = \min(0.43, 0.12) = 0.12$$
Interpreted as **True**.

- **Weight**:
$$\text{And}(0.43, 0.76) = \min(0.43, 0.76) = 0.43$$
Interpreted as **True**.

- **Intelligence**:
$$\text{And}(0.43, 1.00) = \min(0.43, 1.00) = 0.43$$
Interpreted as **True**.

- **Speed**:
$$\text{And}(0.43, -0.56) = \min(0.43, -0.56) = -0.56$$
Interpreted as **False**.

Interpretation:

- **Legs, Skin Cover, Weight, Intelligence**: The conjunction is true, suggesting that having legs is positively associated with these properties.
- **Speed**: The conjunction is false due to the negative correlation between legs and speed.

Formula 3: Intelligence $\leftrightarrow$ (**Speed** $\wedge\, a$) For each property a:

1. Compute $\text{And}(\text{proj}(w, \text{Speed}), \text{proj}(w, a))$:
   - **Legs**:
   $$\text{And}(-0.56, 0.43) = \min(-0.56, 0.43) = -0.56$$
   - **Skin Cover**:
   $$\text{And}(-0.56, 0.12) = \min(-0.56, 0.12) = -0.56$$
   - **Weight**:
   $$\text{And}(-0.56, 0.76) = \min(-0.56, 0.76) = -0.56$$
   - **Intelligence**:
   $$\text{And}(-0.56, 1.00) = \min(-0.56, 1.00) = -0.56$$
   - **Speed**:
   $$\text{And}(-0.56, -0.56) = \min(-0.56, -0.56) = -0.56$$

2. Compute $\text{Iff}(1.00, \text{And}(\text{proj}(w, \text{Speed}), \text{proj}(w, a)))$:
   - For all a:

$$\begin{aligned}
\text{Iff}(1.00, -0.56) &= \min\left(\text{Implies}(1.00, -0.56),\ \text{Implies}(-0.56, 1.00)\right) \\
\text{Implies}(1.00, -0.56) &= \min(1, 1 - 0.56 - 1.00) = -0.56 \\
\text{Implies}(-0.56, 1.00) &= \min(1, 1 + 1.00 + 0.56) = 1.00 \\
\text{Iff}(1.00, -0.56) &= \min(-0.56, 1.00) = -0.56
\end{aligned}$$

   Interpreted as **False**.

Interpretation:

- The equivalence is false across all properties. This reflects that intelligence is not equivalent to the conjunction of speed and any other property, emphasizing the negative relationship between intelligence and speed.

Analysis with Weight as Perspective Similarly, we can perform the analysis using **Weight** as the perspective vector w. The projections are computed as:

$$\text{proj}(w, x) = k(w, x)$$

Computed Projections:

- **Legs**:
$$\text{proj}(w, \text{Legs}) = k(\text{Weight}, \text{Legs}) = 0.54$$
- **Skin Cover**:
$$\text{proj}(w, \text{Skin Cover}) = k(\text{Weight}, \text{Skin Cover}) = -0.22$$
- **Weight**:
$$\text{proj}(w, \text{Weight}) = k(\text{Weight}, \text{Weight}) = 1.00$$
- **Intelligence**:
$$\text{proj}(w, \text{Intelligence}) = k(\text{Weight}, \text{Intelligence}) = 0.76$$
- **Speed**:
$$\text{proj}(w, \text{Speed}) = k(\text{Weight}, \text{Speed}) = -0.85$$

Formula 1: Weight $\to a$ For each property a:

- **Legs**:
$$\text{Implies}(1.00, 0.54) = \min(1, 1 + 0.54 - 1.00) = 0.54$$
Interpreted as **True**.

- **Skin Cover**:
$$\text{Implies}(1.00, -0.22) = \min(1, 1 - 0.22 - 1.00) = -0.22$$
Interpreted as **False**.

- **Weight**:
$$\text{Implies}(1.00, 1.00) = \min(1, 1 + 1.00 - 1.00) = 1.00$$
Interpreted as **True**.

- **Intelligence**:
$$\text{Implies}(1.00, 0.76) = \min(1, 1 + 0.76 - 1.00) = 0.76$$
Interpreted as **True**.

- **Speed**:
$$\text{Implies}(1.00, -0.85) = \min(1, 1 - 0.85 - 1.00) = -0.85$$
Interpreted as **False**.

Interpretation:

- **Legs, Weight, Intelligence**: The implication is true, suggesting that higher weight is associated with these properties.
- **Skin Cover, Speed**: The implication is false, indicating that higher weight does not imply these properties.

Formula 2: Legs $\land a$ For each property a:

- **Legs**:
$$\text{And}(0.54, 0.54) = \min(0.54, 0.54) = 0.54$$
Interpreted as **True**.

- **Skin Cover**:
$$\text{And}(0.54, -0.22) = \min(0.54, -0.22) = -0.22$$
Interpreted as **False**.

- **Weight**:
$$\text{And}(0.54, 1.00) = \min(0.54, 1.00) = 0.54$$
Interpreted as **True**.

- **Intelligence**:
$$\text{And}(0.54, 0.76) = \min(0.54, 0.76) = 0.54$$
Interpreted as **True**.

- **Speed**:
$$\text{And}(0.54, -0.85) = \min(0.54, -0.85) = -0.85$$
Interpreted as **False**.

Interpretation:

- **Legs, Weight, Intelligence**: The conjunction is true, suggesting a positive association from the perspective of weight.
- **Skin Cover, Speed**: The conjunction is false due to negative projections.

Formula 3: Weight $\leftrightarrow$ (Speed $\wedge a$) For each property a:

1. Compute $\text{And}(\text{proj}(w, \text{Speed}), \text{proj}(w, a))$:
   - **Legs**:
   $$\text{And}(-0.85, 0.54) = \min(-0.85, 0.54) = -0.85$$
   - **Skin Cover**:
   $$\text{And}(-0.85, -0.22) = \min(-0.85, -0.22) = -0.85$$
   - **Weight**:
   $$\text{And}(-0.85, 1.00) = \min(-0.85, 1.00) = -0.85$$
   - **Intelligence**:
   $$\text{And}(-0.85, 0.76) = \min(-0.85, 0.76) = -0.85$$
   - **Speed**:
   $$\text{And}(-0.85, -0.85) = \min(-0.85, -0.85) = -0.85$$
2. Compute $\text{Iff}(1.00, \text{And}(\text{proj}(w, \text{Speed}), \text{proj}(w, a)))$:

- For all a:

$$\begin{aligned}
\text{Iff}(1.00, -0.85) &= \min\left(\text{Implies}(1.00, -0.85),\ \text{Implies}(-0.85, 1.00)\right) \\
\text{Implies}(1.00, -0.85) &= \min(1, 1 - 0.85 - 1.00) = -0.85 \\
\text{Implies}(-0.85, 1.00) &= \min(1, 1 + 1.00 + 0.85) = 1.00 \\
\text{Iff}(1.00, -0.85) &= \min(-0.85, 1.00) = -0.85
\end{aligned}$$

  Interpreted as **False**.

Interpretation:

- The equivalence is false across all properties, indicating that weight is not equivalent to the conjunction of speed and any other property. This emphasizes the negative relationship between weight and speed.

8.2 Quantitative truth querying

Suppose that we have defined a semantic space, have a given perspective w. Then we can form formulas from the element of the semantic space and ask for: How true is the formula? ($-1 = 100$ percent false, $0 =$ Indeterminate, $+1 = 100$ percent True given the kernel k and the perspective w.) Here is a toy example:

Let $w = 2$ and consider the set $X = \{2, 3, \frac{1}{2}\}$.

The projections of each element in X onto w using the projection function are calculated as follows:

$$p_2 = \text{proj}(w, 2) = 1.0, \quad p_3 = \text{proj}(w, 3) = 0, \quad p_{\frac{1}{2}} = \text{proj}\left(w, \frac{1}{2}\right) = -1.0$$

We define several logical formulas that operate on these projections:

- Formula 1 : $(\neg a \wedge (b \vee c)) \rightarrow \neg(a \vee b)$
- Formula 2 : $(\neg a \wedge (b \vee c)) \vee \neg(a \vee b)$
- Formula 3 : $a \wedge b$
- Formula 4 : $a \vee b$

Applying these formulas to $a = p_2, b = p_3$, and $c = p_{\frac{1}{2}}$, we obtain:

$$\begin{aligned}
&\text{Formula 1 Output} : 1, \\
&\text{Formula 2 Output} : -1.0, \\
&\text{Formula 3 Output} : 0, \\
&\text{Formula 4 Output} : 1.0.
\end{aligned}$$

Converting these outputs to truth values:

$$\begin{aligned}&\text{'T' for } 1,\\&\text{'F' for } -1.0,\\&\text{'I' for } 0,\\&\text{'T' for } 1.0.\end{aligned}$$

So one might say:
$(\neg 2 \wedge (3 \vee \frac{1}{2})) \rightarrow \neg(2 \vee 3)$ is a 100 percent true formula (under k and relative to w). $(\neg 2 \wedge (3 \vee \frac{1}{2})) \vee \neg(2 \vee 3)$ is a 100 percent false formula (under k and relative to w). $2 \wedge 3$ is neither true nor false, hence indeterminate (under k and relative to w). $2 \vee 3$ ia a 100 percent true formual (under k and relative to w).

One possible semidefinite kernel k for the natural numbers where the formulas above make sense is given by the kernel k_0:

$$k_0(a,b) := \sum_{p \mid \mathrm{GCD}(a,b), p \text{ prime}} v_p(a) \cdot v_p(b) \tag{24}$$

where $\mathrm{GCD}(a/b, c/d) = \frac{\gcd(ad,bc)}{bd}$ is the gcd function extended to $\mathbb{Q}_{>0}$ by Scott Beslin and Grant Boudreaux in the paper 'Extending greatest common divisors across the rationals' [1] and $v_p(n)$ is the valuation of n to the prime p.

Then the normalized kernel is defined as:

$$k(a,b) := \frac{k_0(a,b)}{\sqrt{k_0(a,a)k_0(b,b)}} \tag{25}$$

So this example illustrates how one can query the semantic space of logic for given formulas and the quantitative truthfulness of the formula.

8.3 Logical consequences of truth voting / beliefs

Suppose one is given a finite set $X = \{x_1, \cdots, x_n\}$ of everyday sentences, which one can argue about if they are true or not. Suppose further that some kind of voting is being done to determine the truthfulness of x_i: either true $y_i := +1$ or false $y_i := -1$. As the last ingredient, we require a positive definite kernel k on X which measure some sort of similarity between the sentences x_i in such a way, that the semantic space $S = (X, k)$ is separable for the voted Y. Hence there exists a w such that:

$$\mathrm{sign}(k(x_i, w)) = y_i$$

(The method to find such a w might be, for example a support vector machines classifier, and can be easily implemented in python in the module scikit-learn [11] for instance.)

Now the participants of the voting might be interested, once the truthfulness / beliefs is voted for and defined, on logical consequences of this definition:

For instance they might want to know if a given formula F formed by $\equiv, \rightarrow, \vee, \wedge, \neg$ and x_i-s, such as for example:

$x_1 \wedge x_2 \vee (\neg(x_3 \rightarrow (x_4 \equiv x_5)))$

is true or not and to what quantitative degree it is true / believed to be true or not. Here the participants have chosen one w and do not change it any more.

By the example application given above, this can be done using the definitions in the semantic space.

8.4 Generating sequence of tokens

Let $S = (X, k)$ be a semantic space and $x_1, x_2, \cdots, x_r$ be a sequence of elements of X. We want to be able to express the following idea as a formula: Starting from x_1 we deduce in a sequence of r steps that x_r must be true. Therefore we interpret this sequence as the following formula, which basically says, I start with x_1 as being true, then I keep on adding that from $x_1 \leftrightarrow x_i$, until we arrive at x_r:

$$F = x_1 \wedge (x_1 \leftrightarrow x_2) \wedge (x_1 \leftrightarrow x_3) \wedge \ldots \wedge (x_1 \leftrightarrow x_r) \tag{26}$$

Having this formula F, we can now compute given say the perspective $w = x_1$ how likely F is:

A negative value indicates that this formula, hence sequence must be false, a value near 0 indicates that this sequence is neither true nor false, and a value significantly greater then 0, near 1, indicates that this sequence is true from the w perspective.

This could help with generative models, where one would start with $F = x_1$ and keep adding x such that $F' := F \wedge (x_1 \rightarrow x)$ has the largest value among the $x \in X$.

9 A semantic space of logic on natural numbers

The function $k(a, b) = \frac{\gcd(a,b)^2}{ab}$ is a p.d. kernel similarity on the natural numbers. By the theory of Reproducing Kernel Hilbert space, there exists a Hilbert space H and an embedding of the natural numbers $\phi : \mathbf{N} \rightarrow H$ such that:

$$k(a, b) = \langle \phi(a), \phi(b) \rangle$$

In practice, given finitely many natural number $a_1, \cdots, a_r$ we can compute such an embedding for those numbers using the Cholesky decomposition of the Gram matrix $G = (k(a_i, a_j))_{i,j} = CC^T$ and then assigning to a_i the i-th column vector of C.

In the given specific case however, we can give a direct embedding for all numbers:

Let e_d be the d-th standard-basis vector in the Hilbert space $H = l_2(\mathbb{N})$. Let $h(n) = J_2(n)$ be the second Jordan totient function. Define:

$$\phi(n) = \frac{1}{n} \sum_{d|n} \sqrt{h(d)} e_d$$

.

Then we have:

$$\langle \phi(a), \phi(b) \rangle = \frac{\gcd(a,b)^2}{ab} =: k(a,b)$$

The vectors $\phi(a_i)$ are linearly independent for each finite set $a_1, \cdots, a_n$ of natural numbers, since

$$\det(G_n) = \prod_{i=1}^{n} \frac{h(a_i)}{a_i^2}$$

is not zero, where G_n denotes the Gram matrix.

10 Satisfiability of formulas in semantic spaces of logic

In everyday logic, when faced with a problem, it helps to change the perspective to solve the problem. This idea can be made precise in semantic space of logic. Here the word 'problem' is translated to 'a formula which should be satisfied' and changing perspective means 'changing the perspective vector'.

10.1 Problem Formulation

Consider a semantic space $S = (X, k)$, where $|X|$ is finite, and we have a set of formulas $f_1, \ldots, f_r$, each involving logical operations such as AND, OR, NOT, IMPLIES, and IFF, and elements from X. The problem is as follows:

Problem 1: Find a perspective $\phi(w) \in H$ such that, for all $i = 1, \ldots, r$:

$$\mu(f_i) \neq F \quad \text{(False)}$$

In other words, find a perspective w such that all formulas are satisfied as either T (True) or I (Indeterminate), but none evaluate to F (False).

Stated as above it is easy to find a perspective vector w which evaluates each formula to I, by chosing w perpendicular ($k(w, x_i) = 0$) to each x_i, but this is an impractical solution, as every formula is evaluated to 'indeterminate'. One might try to add additional restrictions to this problem, such as trying to maximize the number of formulas which are evaluated to 'True'. For instance if $X = \{x\}$ and $f_1 = x, f_2 = \neg x$ then the only solution is to find a w perpendicular to x , $k(w, x) = 0$, and so the two formulas become 'indeterminate' evaluated through w. So this problem setting might be useful, when it is allowed to have contradictory formulas f_i, but one wants to satisfy as much of the remaining formulas as possible while evaluting the contradictions to 'indeterminate'.

Problem 2: Find a perspective $\phi(w) \in H$ such that, for all $i = 1, \ldots, r$:

$$\mu(f_i) = T \quad \text{(True)}$$

or conclude that no such perspective vector $\phi(w)$ exists.

In other words, find a perspective w such that all formulas are satisfied as T (True), but none evaluate to F (False) or I (Indeterminate). This problem setting is similar to the 'Boolean satisfiability problem', but here, one changes the evaluation of the x_i through a change in the perspective vector w.

Comment: One simple algorithm for the second problem to find a solution where all the x_i have constant truth value say $c = k(w, x_i)$ and the Gram matrix is invertible, so that the kernel is positive definite, is the following method: Suppose f is a formula in x_l. First assume that c is an unknown, which we will determine later, and assume that w is a placeholder-symbol:

- Let $(\alpha_1, \ldots, \alpha_n) := c \cdot G^{-1} \cdot (1, \ldots, 1)$
- Then $\forall i = 1, \ldots, n$: $k(w, x_i) = c = \text{const.}$
- Choose one after the other $c_1 = 0.5, c_2 = 0, c_3 = -0.5$ and accordingly w_1, w_2, w_3 and compute $\mu_{w_j}(f(x_l) \,\mathrm{rel}\, w_j)$. If it equals $= T$ then set $w := w_j$, and the second problem is solvable.
- Otherwise the 2. problem is not solvable for constant $c = k(w, x_i)$ as it gives I or F.

11 Further ideas

This section expands upon some speculative applications and extensions of the semantic space framework. Initial explorations might focus on:

- Interpreting the kernel function $k(w, x)$ as a measure of correlation, a concept that requires further investigation, but is immediate, since the Gram matrix G is symmetric, positive semi-definite, with ones on the diagonal $G_{x,x} = k(x, x) = 1$ and $-1 \leq G_{x,y} = k(x, y) \leq 1 \forall x, y \in X$.
- The application of this framework to SVM classification. In SVM, one typically defines a semantic space $S = (X, k)$ along with binary labels for elements in X to find a perspective w that classifies new points $\hat{x}$ correctly. This methodology could enhance the determination of truth values in a semantic vector space, with perspectives w defined such that they simultaneously satisfy multiple truth conditions.
- The assignment of the value 'I = indeterminate' to points x on the SVM hyperplane that divides classes, providing a nuanced treatment of boundary cases in classification.
- Considering elements of the semantic space X as 'atoms' and True elements as 'axioms', with formulas defined recursively using logical operators. This structure allows the exploration of the consequences of various sets of axioms under changes in perspective.

- Exploring the possibility of time-varying logic by allowing the perspective vector w to vary with time t, thereby creating a dynamic logical structure over a static semantic space $S = (X, k)$.

A semantic space as defined in this note is a finite set X with a symmetric, positive semi-definite function $k : X \times X \to \mathbb{R}$ such that:

$$k(x, x) = 1 \forall x \in X, -1 \leq k(x, y) \leq 1 \forall x, y \in X$$

A correlation matrix R has the following characteristic properties: symmetric, positive semi-definite, $R_{i,i} = 1 \forall i = 1, \ldots, n$, $|R_{i,j}| \leq 1 \forall i, j$

Thus, we can consider the Gram matrix $G = (k(x, y))_{x,y \in X}$ in a semantic space as a correlation matrix. This allows us to, consider individual entries $G_{x,y}$ in the Gram matrix as correlation values $\rho_{x,y} := G_{x,y}$ and perhaps apply a statistical test which tests the hypothesis: $\rho_{x,y} = 0$. To apply the test, it must first be clarified how many degrees of freedom are available, and whether the underlying random variables are pairwise bivariate normally distributed. The test size with $df = m - 2$ degrees of freedom, where m is the 'number of individuals' would be:

$$t = \frac{\rho_{x,y}}{\sqrt{\frac{1-\rho_{x,y}^2}{m-2}}}$$

If one is given a positive-definite Gram matrix, then one can perform Cholesky decomposition or KPCA to obtain vectors $\phi(x), x \in X$ which could be interpreted as random vectors with the property that:

$$\langle \phi(x), \phi(y) \rangle = k(x, y) = \rho_{x,y}$$

The dimension d of these vectors would then correspond to the number of individuals m and in Cholesky decomposition or also in KPCA this number would be greatest with $m = n$. Thus, the degrees of freedom would be $df = n - 2 = |X| - 2$.

For every Gram matrix, there can be different vectors that realize this matrix.

The question that arises is, whether there is a method to find the vectors $(\phi(x), \phi(y))$ so that they have pairwise bivariate normally distributed entries. Then one could apply the t-test from statistics with confidence, without ever having to practically find these vectors.

12 Proofs

Let H be an RKHS to the kernel k. Let $B := \{h \in H||h| \leq 1\}$ and for a w with $|\phi(w)| = 1$ let $G_w := \{t\phi(w)| - 1 \leq t \leq 1\}$.

12.1 Projection onto the Perspective Vector

Equation (1): For $h \in H$ with $|h| \leq 1$ we define $\pi_w(h)$ as the projection of h on $\phi(w)$:

$$\pi_w(h) := \langle \phi(w), h \rangle \, \phi(w)$$

This is a mapping $\pi_w : B \to G_w$ and we write also $h \operatorname{rel} w := \pi_w(h) = \langle \phi(w), h \rangle \, \phi(w)$

For $x \in X$ we define $\pi_w(x)$ as the projection of $\phi(x)$ on $\phi(w)$ and this becomes with the reproducing property $k(x, w) = \langle \phi(x), \phi(w) \rangle$ equal to:

$$\pi_w(x) = \langle \phi(w), \phi(x) \rangle \, \phi(w) = k(w, x)\phi(w)$$

Proof:

In the Hilbert space H, the projection of a vector h onto a vector $\phi(w)$ is given by:

$$\pi_w(h) = \frac{\langle \phi(w), h \rangle}{\langle \phi(w), \phi(w) \rangle} \phi(w)$$

And because $\langle \phi(w), \phi(w) \rangle = k(w, w) = 1$, it follows that:

$$\pi_w(x) = \langle \phi(w), h \rangle \, \phi(w)$$

This shows that the projection of h onto $\phi(w)$ is described by the given equation.

12.2 Relative representation of x with respect to w

Equation (2):

For $h \in H$ with $|h| \leq 1$ we define as above:

$$h \operatorname{rel}(w) := \pi_w(h) = \langle \phi(w), h \rangle \, \phi(w)$$

For $x \in X$ we define:

$$x \operatorname{rel}(w) := \phi(x) \operatorname{rel}(w) = \pi_w(\phi(x)) = k(w, x)\phi(w)$$

Proof:

This is a direct definition based on the previous equation. Since the projection of x onto w is given by $\pi_w(x) = k(w, x)\phi(w)$, we define x relative to w precisely as this projection. (The notation $x \operatorname{rel}(w)$ should be thought of as a convenient way of writing $\phi(x) \operatorname{rel}(w)$.)

12.3 Comment

In what follows, we are going to define functions

$$\Box : G_w \times G_w \to G_w$$

and for convenience we will write:

$$h\Box h' \operatorname{rel} w := (h' \operatorname{rel} w)\Box(h' \operatorname{rel} w) := \cdots$$

where $\cdots$ will be defined shortly below.
We have the following:

$$(h \operatorname{rel} w) \operatorname{rel} w = h \operatorname{rel} w$$

Proof: We have

$$\pi_w(\pi_w(h)) = \langle \phi(w), \pi_w(h) \rangle \, \phi(w) = \langle \phi(w), \langle \phi(w), h \rangle \, \phi(w) \rangle \, \phi(w) = \langle \phi(w), h \rangle \, \phi(w) = \pi_w(h)$$

The shortcut notation with $\Box$ and this last observation, allows us to write for instance some formulas like:

$$(h_1 \wedge (\neg h_2 \vee h_3) \to (h_4)) \operatorname{rel} w$$

without having to write:

$$((h_1(\operatorname{rel} w) \wedge (\neg h_2 \operatorname{rel} w \vee h_3 \operatorname{rel} w)) \operatorname{rel} w \to (h_4 \operatorname{rel} w) \operatorname{rel} w) \operatorname{rel} w$$

which really is not readable any more.

12.4 Definition of the logical AND operation

Equation (3):
The mapping $\wedge$ is defined as:

$$\wedge : G_w \times G_w \to G_w$$

through

$$(t\phi(w), t'\phi(w)) \mapsto \min(t, t')\phi(w)$$

For $h, h' \in B$ we get, with $t = \langle \phi(w), h \rangle$ and $t' = \langle \phi(w), h' \rangle$:

$$h \wedge h' \operatorname{rel}(w) := h \operatorname{rel}(w) \wedge h' \operatorname{rel}(w) = \min(\langle \phi(w), h \rangle, \langle \phi(w), h' \rangle)\phi(w)$$

for $x, y \in X$ this becomes (under the identification of $x \operatorname{rel}(w) = \phi(x) \operatorname{rel} w$)

$$x \wedge y \operatorname{rel}(w) := \min(k(w, x), k(w, y))\phi(w)$$

Proof:
This equation defines the logical AND operation in the geometric context. The idea is that the minimal similarity (given by the kernel) between x and w, as well as y and w, represents the 'common' element. Multiplying by $\phi(w)$ projects the result back into the Hilbert space in the direction of w.

12.5 Definition of the logical OR operation

Equation (4):
The mapping $\vee$ is defined as:

$$\vee : G_w \times G_w \to G_w$$

through

$$(t\phi(w), t'\phi(w)) \mapsto \max(t, t')\phi(w)$$

For $h, h' \in B$ we get, with $t = \langle \phi(w), h \rangle$ and $t' = \langle \phi(w), h' \rangle$:

$$h \vee h' \operatorname{rel}(w) := h \operatorname{rel}(w) \wedge h' \operatorname{rel}(w) := \max(\langle \phi(w), h \rangle, \langle \phi(w), h' \rangle)\phi(w)$$

for $x, y \in X$ this becomes (under the identification of $x \operatorname{rel}(w) = \phi(x) \operatorname{rel} w$)

$$x \vee y \operatorname{rel}(w) := \max(k(w, x), k(w, y))\phi(w)$$

12.6 Definition of negation

Equation (5):
The negation $\neg$ is defined as:

$$\neg : G_w \to G_w$$

through:

$$t\phi(w) \mapsto -t\phi(w)$$

for $h \in B$ we get with $t = \langle \phi(w), h \rangle$ the following:

$$\neg(h \operatorname{rel} w) = (-h) \operatorname{rel} w$$

For $x \in X$ this becomes:

$$\neg x = -\pi_w(x) = -k(w, x)\phi(w) = -(x \operatorname{rel} w)$$

Proof:
The negation of a vector x is defined by reflecting its projection across the origin. Since $\pi_w(x) = k(w, x)\phi(w)$, the negation is simply $-k(w, x)\phi(w)$.

12.7 Definition of implication

Equation (6):
The implication is defined as

$$\rightarrow : G_w \times G_w \rightarrow G_w$$

through:

$$t\phi(w) \rightarrow t'\phi(w) := \min(1, 1 + t' - t)\phi(w)$$

For $x, y \in X$, this becomes:

$$x \rightarrow y \, \mathrm{rel}(w) := \min(1, 1 + k(w, y) - k(w, x))\phi(w)$$

Comment:
In Łukasiewicz logic, the implication $x \rightarrow y$ is defined by $\min(1, 1 - v(x) + v(y))$, where $v(x)$ is the truth value of x. In the geometric context, $v(x)$ corresponds to the inner product $k(w, x)$. Therefore, we obtain the given equation.

12.8 Definition of equivalence

Equation (7):
The equivalence is defined as

$$\leftrightarrow : G_w \times G_w \rightarrow G_w$$

through:

$$t\phi(w) \leftrightarrow t'\phi(w) := (t\phi(w) \rightarrow t'\phi(w)) \wedge (t'\phi(w) \rightarrow t\phi(w))$$

For $x, y \in X$ this becomes:

$$x \leftrightarrow y \, \mathrm{rel}(w) := (x \rightarrow y) \wedge (y \rightarrow x) \, \mathrm{rel}(w)$$

Comment:
Logical equivalence is traditionally defined as a combination of two implications. Here, this is geometrically implemented by computing both implications and then taking the minimal similarity (see the AND operation).

12.9 De Morgan's laws

For $x, y \in X$ we have:
Equation (8):

$$(\neg x) \wedge (\neg y) = \neg(x \vee y) \text{ rel } (w)$$

Equation (9):

$$(\neg x) \vee (\neg y) = \neg(x \wedge y) \text{ rel } (w)$$

Proof:
We demonstrate this exemplarily for Equation (8):
Left-hand side:

$$(\neg x) \wedge (\neg y) = \min(k(w, \neg x), k(w, \neg y))\phi(w)$$

Since $\neg x = -\pi_w(x)$ and $k(w, \neg x) = \langle \phi(w), -\pi_w(x) \rangle = -\langle \phi(w), \pi_w(x) \rangle = -k(w, x)$, it follows that:

$$(\neg x) \wedge (\neg y) = \min(-k(w, x), -k(w, y))\phi(w) = -\max(k(w, x), k(w, y))\phi(w)$$

Right-hand side:

$$\neg(x \vee y) = -(\max(k(w, x), k(w, y))\phi(w)) = -\max(k(w, x), k(w, y))\phi(w)$$

Thus, the left-hand and right-hand sides are equal.
The proof for Equation (9) proceeds analogously.

12.10 Double negation

Equation (10): For all $x \in X$:

$$\neg(\neg x) = x \text{ rel}(w)$$

Proof:
Calculating the double negation:

$$\neg(\neg x) = -\pi_w(\neg x) = -k(w, \neg x)\phi(w)$$

Since $k(w, \neg x) = -k(w, x)$, it follows that:

$$\neg(\neg x) = -(-k(w, x)\phi(w)) = k(w, x)\phi(w) = \pi_w(x)$$

Since $\pi_w(x) = x \operatorname{rel} w$, the equation holds.

12.11 Contraposition

Equation (11): For all $x, y \in X$:

$$x \to y \operatorname{rel}(w) = (\neg y) \to (\neg x) \operatorname{rel}(w)$$

Proof:
We show that both sides are identical:
Calculating the left-hand side:

$$x \to y\,\mathrm{rel}(w) = \min(1, 1 + k(w,y) - k(w,x))\phi(w)$$

Calculating the right-hand side:

$$(\neg y) \to (\neg x) = \min(1, 1 + k(w,\neg x) - k(w,\neg y))\phi(w)$$

Since $k(w,\neg x) = -k(w,x)$, we have:

$$k(w,\neg x) = -k(w,x) \quad \text{and} \quad k(w,\neg y) = -k(w,y)$$

and we get:

$$\min(1, 1 + k(w,\neg x) - k(w,\neg y))\phi(w) = \min(1, 1 - k(w,x) - (-k(w,y)))\phi(w)$$

which is equal to

$$\min(1, 1 + k(w,y) - k(w,x))\phi(w) = x \to y\,\mathrm{rel}(w)$$

Thus, both expressions are identical, and the equation is proven.

12.12 Definition of truth assignment

Equation (12): For all $(h\,\mathrm{rel}\,w) \in G_w$, we have:

$$\mu(h\,\mathrm{rel}\,w) = \begin{cases} T & \text{if } \langle \phi(w), h \rangle > 0, \\ I & \text{if } \langle \phi(w), h \rangle = 0, \\ F & \text{if } \langle \phi(w), h \rangle < 0. \end{cases}$$

Proof:

This is translates the value of the inner product $\langle \phi(w), h \rangle$ into a truth value. It is a definition within the model and does not require further proof, but a comment: For $h = x\,\mathrm{rel}\,w = \pi_w(x)$ we get $\langle \phi(w), h \rangle = k(w,x)$, so the truth value of elements of $x \in X$ depends on the kernel value $k(w,x)$ of x with the perspective w.

12.13 Modus ponens

Equation (13):

$$\text{If } \mu(x\,\mathrm{rel}\,w) = T \text{ and } \mu(x \to y\,\mathrm{rel}\,w) = T, \text{ then } \mu((x \wedge (x \to y)) \to y\,\mathrm{rel}\,w) = T$$

Proof:

Given:

- $\mu(x \operatorname{rel} w) = T$:

  This means that the inner product $\langle \phi(w), x \operatorname{rel} w \rangle > 0$.

- $\mu(x \to y \operatorname{rel} w) = T$:

  This means that $\langle \phi(w), x \to y \operatorname{rel} w \rangle > 0$.

Our Goal

Show that $\mu\left((x \wedge (x \to y)) \to y \operatorname{rel} w\right) = T$, i.e., the inner product $\langle \phi(w), (x \wedge (x \to y)) \to y \operatorname{rel} w \rangle > 0$.

Step 1: Understand the Given Conditions

From the definitions:

$$x \operatorname{rel} w = k(w, x)\phi(w)$$

$$x \to y \operatorname{rel} w = v\phi(w), \quad \text{where } v = \min\left(1,\ 1 + k(w, y) - k(w, x)\right)$$

Given that $\mu(x \operatorname{rel} w) = T$, we have:

$$\langle \phi(w), x \operatorname{rel} w \rangle = k(w, x) > 0$$

So, $k(w, x) = s > 0$.
Given that $\mu(x \to y \operatorname{rel} w) = T$, we have:

$$\langle \phi(w), x \to y \operatorname{rel} w \rangle = v > 0$$

Where:

$$v = \min\left(1,\ 1 + k(w, y) - k(w, x)\right) = \min\left(1,\ 1 + r - s\right)$$

Let $k(w, y) = r$.
Since $v > 0$, it follows that:

$$1 + r - s > 0 \implies r > s - 1$$

Step 2: Compute $x \wedge (x \to y) \operatorname{rel} w$

From the definition of logical AND:

$$x \wedge (x \to y) \operatorname{rel} w = \min\left(k(w, x),\ v\right) \phi(w) = \min\left(s,\ v\right) \phi(w)$$

Let:

$$t = \min(s,\ v)$$

So:

$$x \wedge (x \to y) \operatorname{rel} w = t\phi(w)$$

Since both $s > 0$ and $v > 0$, it follows that $t > 0$.

Step 3: Compute $(x \wedge (x \to y)) \to y \operatorname{rel} w$

From the definition of implication:

$$(x \wedge (x \to y)) \to y \operatorname{rel} w = \min(1,\ 1 + k(w, y) - t)\,\phi(w) = \min(1,\ 1 + r - t)\,\phi(w)$$

We need to show that:

$$\langle \phi(w), (x \wedge (x \to y)) \to y \operatorname{rel} w \rangle > 0$$

Compute the inner product:

$$\langle \phi(w), (x \wedge (x \to y)) \to y \operatorname{rel} w \rangle = \min(1,\ 1 + r - t)$$

Step 4: Show that $\min(1,\ 1 + r - t) > 0$

Since $t = \min(s,\ v)$, $t \leq s$ and $t \leq v$.
Given that $s > 0$ and $v > 0$, $t > 0$.
Case 1: If $t = s$

$$1 + r - t = 1 + r - s$$

Since $v = \min(1, 1 + r - s)$ and $v > 0$, it implies $1 + r - s > 0$.
Thus:

$$1 + r - t = 1 + r - s > 0$$

Case 2: If $t = v$
Then $t \leq v \leq 1$, and since $v = 1 + r - s \leq 1$, $1 + r - s \leq 1$.
So:

$$1 + r - t \geq 1 + r - v = 1 + r - (1 + r - s) = s > 0$$

Conclusion

In both cases, $1 + r - t > 0$. Therefore:

$$\langle \phi(w), (x \wedge (x \to y)) \to y \,\mathrm{rel}\, w \rangle = \min(1,\ 1 + r - t) > 0$$

So:

$$\mu\left((x \wedge (x \to y)) \to y \,\mathrm{rel}\, w\right) = T$$

13 Appendix, Sagemath / Python code

```
def kk0(a,b):
    return min(a,b)/max(a,b)

def kk1(a,b):
    return gcd(a,b)**2/(a*b)

def kk2(a,b):
    return sign(gcd(a,b))*sigma(gcd(a,b))/(sign(a)*sigma(a)*sign(b)*
       sigma(b))

def kkL(a,b):
    return a.dot_product(b)/(a.dot_product(a)*b.dot_product(b))

def mu(x):
    return sign(x)

def proj(w,x):
    return kk(w,x)

def And(x,y):
    return min(x,y)

def Or(x,y):
    return max(x,y)

def Implies(x,y):
    return And(1,1+y-x)

def Iff(x,y):
    return And(Implies(x,y),Implies(y,x))

def Not(x):
    return -x

def gram(rr):
    return matrix([[kk(a,b) for a in rr] for b in rr])
```

```
def Table(func,w,rr,printMu = True):
    if printMu:
        return matrix([[mu(func(proj(w,x),proj(w,y))) for x in rr] for
            y in rr])
    else:
        return matrix([[func(proj(w,x),proj(w,y)) for x in rr] for y in
            rr])

def deMorgan(w,rr):
    one = all([And(Not(proj(w,x)),Not(proj(w,y)))==Not(Or(proj(w,x),
        proj(w,y))) for x in rr for y in rr])
    two = all([Or(Not(proj(w,x)),Not(proj(w,y)))==Not(And(proj(w,x),
        proj(w,y))) for x in rr for y in rr])
    return one and two

def contraposition(w,rr):
    return all([Implies(proj(w,x),proj(w,y))==Implies(Not(proj(w,y)),
        Not(proj(w,x))) for x in rr for y in rr])

def doubleNegation(w,rr):
    return all([proj(w,x)==Not(Not(proj(w,x))) for x in rr])

def modusPonens(w,rr):
    return all([mu(Implies(And(proj(w,a),Implies(proj(w,a),proj(w,b))),
        proj(w,b)))==1 for a in rr for b in rr if mu(proj(w,a))==1 and
        mu(Implies(proj(w,a),proj(w,b)))==1])




# Number logic:
#rr = [1,2,3,4,5,6]
#w = 1
#kk = kk2

# Number log-logic:
def kkl(a,b):
    x = sum([valuation(a,p)*valuation(b,p) for p in prime_divisors(gcd(
        a,b))])
    y1 = sqrt(sum([valuation(a,p)**2 for p in prime_divisors(a)]))
    y2 = sqrt(sum([valuation(b,p)**2 for p in prime_divisors(b)]))
    return x/(y1*y2)


rr = [1/3,1/2,2,3]
ww = [2,3]
kk = kkl

# Simplex logic:
def ee(n,N):
    return vector([1*(k==n) for k in range(1,N+1)])

```

```
N=3
rr = [ee(n,N) for n in range(1,N+1)]
v = 1/N*sum(rr)
w = v*1/v.norm()
rr.extend([-ee(n,N) for n in range(1,N+1)])

ww = [w]
kk = kkL

# Lukasiewicz logic:
e1 = vector([1,0])
e0 = vector([0,1])

rr = [-e1,e0,e1]
ww = [e0,e1,-e1]
kk = kkL


print(latex(gram(rr)))

def checks(w,rr):

    pm = False
    print("deMorgan-Rules are satisfied:")
    print(deMorgan(w,rr))

    print("contraposition satisfied:")
    print(contraposition(w,rr))

    print("doubleNegation:")
    print(doubleNegation(w,rr))

    print("modusPonens:")
    print(modusPonens(w,rr))

    print("Not")
    for x in rr:
        print(proj(w,x),Not(proj(w,x)))
        #print(mu(proj(w,x)), mu(Not(proj(w,x))))

    print("And")
    print(latex(Table(And,w,rr,printMu=pm)))

    print("Or")
    print(latex(Table(Or,w,rr,printMu=pm)))

    print("Implies")
    print(latex(Table(Implies,w,rr,printMu=pm)))

    print("Iff")
    print(latex(Table(Iff,w,rr,printMu=pm)))

```

```
for w in ww:
    print("w = ",latex(w))
    checks(w,rr)
#for n in range(1,21):
#    rr = list(range(-n,0))
#    rr.extend(range(1,n+1))
#    print(n,rr)
#    print(gram(rr).rank())
```

For the groups computation, the following code is used:

```
groups60 = [
(['()'],'group_id_1-1'),
(['(1,2)'],'group_id_2-1'),
(['(1,2,3)'],'group_id_3-1'),
(['(1,2,3,4)'],'group_id_4-1'),
(['(3,4)', '(1,2)'],'group_id_4-2'),
(['(1,2,3,4,5)'],'group_id_5-1'),
(['(1,2)(3,6)(4,5)', '(1,3,5)(2,4,6)'],'group_id_6-1'),
(['(3,4,5)', '(1,2)'],'group_id_6-2'),
(['(1,2,3,4,5,6,7)'],'group_id_7-1'),
(['(1,2,3,4,5,6,7,8)'],'group_id_8-1'),
(['(3,4,5,6)', '(1,2)'],'group_id_8-2')]

def inv(perm):
    n = len(perm)
    return set([(i,j) for i in range(1,n+1) for j in range(1,n+1) if i
        < j and perm[i-1]>perm[j-1]])

def lt(p1,p2):
    return inv(p1).issubset(inv(p2))



def regularPermutationsInSymmetricGroup(finiteGroup):
    from sage.matrix.operation_table import OperationTable
    G = finiteGroup
    O = OperationTable(G,operator.mul,names="elements")
    #print(latex(O.table()))
    ll = [ Permutation([xx +1 for xx in x]) for x in O.table()]
    return ll


def embeddInMatrixSpace(sigma,normalized=True,floatIt=True):
    """sigma = Permutation"""
    n = len(sigma)
    m = matrix([[0 if not floatIt else float(0.0) for i in range(n)]
        for j in range(n)])
    for i in range(n):
        for j in range(n):
            m[i,j] = 1*(sigma(i+1)>sigma(j+1))-1*(sigma(i+1)<sigma(j+1)
                )
```

```
    if normalized:
        return 1/sqrt(n*(n-1))*m if not floatIt else float(1/sqrt(n*(n-1)))*m
    else:
        return m

def kk(p1,p2):
    m1 = embeddInMatrixSpace(p1,normalized=True,floatIt=True)
    m2 = embeddInMatrixSpace(p2,normalized=True,floatIt=True)
    return (m1*m2.transpose()).trace()

def grammat(group):
    perms = regularPermutationsInSymmetricGroup(group)
    return matrix([[kk(p1,p2) for p1 in perms] for p2 in perms])

def mu(x):
    return sign(x)

def proj(w, x):
    return kk(w, x)

def And(x, y):
    return min(x, y)

def Or(x, y):
    return max(x, y)

def Implies(x, y):
    return min(1, 1 + y - x)

def Iff(x, y):
    return min(Implies(x, y), Implies(y, x))

def Not(x):
    return -x

# Table function
def Table(func, w, perms,showTruthValues=False,wIsAlreadyVector=False):
    table = []
    n = len(perms)
    for y in perms:
        row = []
        for x in perms:
            if not wIsAlreadyVector:
                value = func(proj(w, x), proj(w, y))
            else:
                vX = embeddInMatrixSpace(x,normalized=True,floatIt=True)
                vY = embeddInMatrixSpace(y,normalized=True,floatIt=True)
                #print(x,vX)
                #print(y,vY)
```

```
                value = func((matrix(w)*vX.transpose()).trace(),(matrix
                    (w)*vY.transpose()).trace())
            truth_value = (value)
            if (truth_value) > 10**-3 and showTruthValues:
                row.append('T')
            elif (truth_value) < -10**-3 and showTruthValues:
                row.append('F')
            elif showTruthValues:
                row.append('I')
            elif not showTruthValues:
                row.append(value)
        table.append(row)
    return table



numberOfObservedGroups=7
for gens,name in groups60[1:numberOfObservedGroups]:

    group = PermutationGroup(gens)
    #criteria = (group.order()==8 and group.is_elementary_abelian())
    #criteria = group.order()==3
    criteria = group.order()==4 and group.is_cyclic()
    #criteria = group.order()==2
    if not criteria :
        continue
    print(name)
    K = grammat(group)

    perms = regularPermutationsInSymmetricGroup(group)

    elements = list(group)
    perm_dict = dict(zip(elements, perms))

    # Compute the Gram matrix
    print("Gram Matrix K:")
    print(K)

    continue

    # Perspectives
    perspectives = [x for x in elements]


    # Logical Operations
    operations = {'AND': And, 'OR': Or, 'IMPLIES': Implies, 'IFF': Iff}

    # Compute and display tables for each perspective
    for w_label in perspectives:
        w = perm_dict[w_label]
        print(f"\nPerspective w = {w_label}")
        print("truth values of vectors:")
```

```
        for x in perms:
            val = (proj(w,x))
            print(x, "= w" if w==x else "   ", "F" if val < -10**-3 else "T" if val > 10**-3 else "I")
            #print(x, "= w" if w==x else "   ", val)
        showTruthValues = True
        for op_name, op_func in operations.items():
            table = Table(op_func, w, perms,showTruthValues=showTruthValues)
            print(f"\nTruth Table for {op_name} (Perspective w = {w_label}):")
            print(' ' * 5 + ' '.join([str(e).rjust(5) for e in elements]))
            for e_row, row in zip(elements, table):
                if not showTruthValues:
                    print(str(e_row).rjust(5) + ' '.join([str(round(val,3)).rjust(5) for val in row]))
                else:
                    print(str(e_row).rjust(5) + ' '.join([str(val).rjust(5) for val in row]))
```

Literatur

[1] Beslin, Scott, and Grant Boudreaux. *Extending Greatest Common Divisors Across the Rationals*. Australian Senior Mathematics Journal, 27(2), 55-62, 2013.

[2] Gärdenfors, Peter. *Conceptual Spaces: The Geometry of Thought*. MIT Press, 2000.

[3] Antti Hautamäki, *A Perspectivist Approach to Conceptual Spaces*, Journal or Conference Name (if available), Year (if available).

[4] Miller, David, and Jonathan Westphal. *A Geometry of Logic*.

[5] Mormann, Thomas. *Geometry of Logic and Truth Approximation*. In R. Festa, A. Aliseda, & J. Peijnenburg, editors (2005), pp. 429–452. *Confirmation, Empirical Progress, and Truth Approximation*. Amsterdam & Atlanta: Editions Rodopi B.V.

[6] Mormann, Thomas. *Truthlikeness for Theories on Countable Languages*. In I.C. Jarvie, K.M. Milford, & D.W. Miller, editors (2006), pp. 3-15. *Karl Popper. A Centenary Assessment. Volume III, Science*. Aldershot & Burlington VT: Ashgate. Paperback edition. London: College Publications.

[7] Saitoh, Saburou. *Theory of Reproducing Kernels and Applications*. Longman Scientific & Technical, 1988.

[8] Schölkopf, Bernhard, and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002.

[9] Y. Jiao, and J.-P. Vert. "The Kendall and Mallows Kernels for Permutations.Ïn F. R. Bach, and D. M. Blei (Eds.), Proceedings of the 32nd International Conference on Machine Learning (ICML 2015), volume 37 of JMLR Workshop and Conference Proceedings, pages 1935-1944. JMLR.org, 2015. Available at: `https://proceedings.mlr.press/v37/jiao15.html`.

[10] The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 9.3)*, 2021. Available at: `https://www.sagemath.org`.

[11] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, É. *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12, pp. 2825-2830, 2011. Available at: `https://scikit-learn.org`.