Linear Algebra

- We begin with recollections from the theory of linear algebra $\Rightarrow$ Fields. Vector Spaces. Algebras

- Recast linear information processing as operators in the algebra of endomorphisms of a vector space

Field ("Definition")

A field F is a set where a sum and a multiplication are defined

- Define numbers and the operations we perform on them $\Rightarrow$ Reals $F = \mathbb{R}$. Complexes $F = \mathbb{C}$
- There are two operations defined $\Rightarrow$ The sum and the product

Field (Definition)

A field F is a set with two binary operations: Addition $(+)$ and multiplication $(\cdot)$. Such that:

- $\Rightarrow$ Both are associative $\Rightarrow \alpha + (\beta + \gamma) = (\alpha + \beta) + \gamma$, and $\alpha \cdot (\beta \cdot \gamma) = (\alpha \cdot \beta) \cdot \gamma$
- $\Rightarrow$ Both are commutative $\Rightarrow \alpha + \beta = \beta + \alpha$, and $\alpha \cdot \beta = \beta \cdot \alpha$
- $\Rightarrow$ Both have identity elements $\Rightarrow \alpha + 0 = \alpha$, and $\alpha \cdot 1 = \alpha$
- $\Rightarrow$ Additive inverse $\Rightarrow$ For all $\alpha \in F$, there exists $-\alpha$ such that $\alpha + (-\alpha) = 0$
- $\Rightarrow$ Multiplicative inverse $\Rightarrow$ For all $\alpha \in F$, $\alpha \neq 0$, there exists α^{-1} such that $\alpha \cdot (\alpha^{-1}) = 1$
- $\Rightarrow$ Distributive property $\Rightarrow \alpha \cdot (\beta + \gamma) = (\alpha \cdot \beta) + (\alpha \cdot \gamma)$

Vector Space ("Definition")

A vector space M over the field F is a set whose elements can be added together and can be also multiplied by elements of the field F. A set of arrows.

- Define the signals we want to process $\Rightarrow$ Vectors in $\mathbb{R}^n$. Functions in $L_2([0,1])$. Sequences
- In addition to the field operations we add two more operations

$\Rightarrow$ The addition of signals and the multiplication of signals by scalars

Vector Space (Definition)

A vector space M over the field F is a set with two operations: Vector addition $(+)$ and scalar multiplication $(\times)$. Such that

- $\Rightarrow$ Vector addition is associative $\Rightarrow \mathsf{x} + (\mathsf{y} + \mathsf{z}) = (\mathsf{x} + \mathsf{y}) + \mathsf{z}$
- $\Rightarrow$ Vector addition is commutative $\Rightarrow \mathsf{x} + \mathsf{y} = \mathsf{y} + \mathsf{x}$
- $\Rightarrow$ Vector addition has an identity element $\Rightarrow \mathsf{x} + 0 = \mathsf{x}$
- $\Rightarrow$ Vector addition has an inverse $\Rightarrow$ For all $\mathsf{x} \in M$, there exists $-\mathsf{x}$ such that $\mathsf{x} + (-\mathsf{x}) = 0$

Vector Space (Definition)

A vector space M over the field F is a set with two operations: Vector addition $(+)$ and scalar multiplication $(\times)$. Such that

$\Rightarrow$ Scalar and field multiplication are compatible $\Rightarrow \alpha \times (\beta \times \mathrm{x}) = (\alpha \cdot \beta) \times \mathrm{x}$

$\Rightarrow$ Scalar multiplication has an identity element $\Rightarrow 1 \times \mathrm{x} = \mathrm{x}$

$\Rightarrow$ Distributive property w.r.t vector addition $\Rightarrow \alpha \times (\mathrm{x} + \mathrm{y}) = (\alpha \times \mathrm{x}) + (\alpha \times \mathrm{y})$

$\Rightarrow$ Distributive property w.r.t field addition $\Rightarrow (\alpha + \beta) \times \mathrm{x} = (\alpha \times \mathrm{x}) + (\beta \times \mathrm{x})$

Associative Algebra (Definition)

An associative algebra A is a vector space with a bilinear map $A \times A \to A$ mapping $(a, b) \to a * b$ and such that $(a * b) * c = a * (b * c)$.

- An algebra with unity is one with an identity element 1 such that $1 * a = a * 1 = a$
- The algebra is commutative if for all $a, b \in A$ we have $a * b = b * a$
- Add a fifth operation to define the linear transformation of a signal $\Rightarrow$ Endomorphisms

Signals and Vector Spaces

- Signals are the entities we want to process $\Rightarrow$ They are elements x of a vector space M

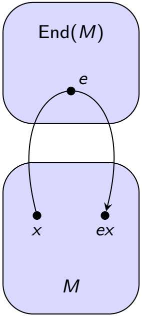
  $\Rightarrow$ We can add two signals $\Rightarrow y = x_1 + x_2$

  $\Rightarrow$ We can scale signals with elements of a field $\Rightarrow y = \alpha \times x$

- Set of vectors with n components $\Rightarrow M = \mathbb{R}^n \Rightarrow$ Graph signals. Or discrete signals
- Set of finite energy functions in $[0, 1] \Rightarrow M = L_2([0, 1]) \Rightarrow$ Graphon signals
- Sequences (discrete time). Functions in $\mathbb{R}$ (continuous time). Sequences with two indexes (images)

- An endomorphism e is a linear map from the vector space M into itself

$$e(\alpha_1 \times x_1 + \alpha_2 \times x_2) = (\alpha_1 \times ex_1) + (\alpha_2 \times ex_2)$$



- If $M = \mathbb{R}^n \Rightarrow$ Square matrix multiplications $\Rightarrow \mathrm{y} = \mathrm{Ex}$

- If $M = L_2([0,1]) \Rightarrow$ Linear functionals $\Rightarrow \mathrm{y}(u) = \int_0^1 E(u,v)\mathrm{x}(v)\,dv$

- The space of all the endomorphisms in M is End(M) $\Rightarrow$ All Matrices. Or all linear functionals

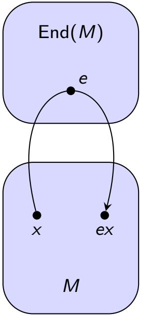
- The set End(M) of endomorphisms of a vector space M is (another) vector space

- The sum operation yields the endomorphism $e = e_1 + e_2$ defined as

$$ex = e_1 x + e_2 x$$

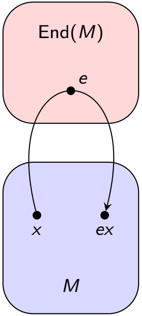
- Scalar multiplication yields the endomorphism $e' = \alpha e$ defined as

$$e'x = \alpha \times (ex)$$



- The set of square matrices of given dimension or the set of linear functionals is a vector space

- It has more structure, though $\Rightarrow$ $\text{End}(M)$ is also an associative algebra with unity
- Product $e = e_1 * e_2$ $\Rightarrow$ Composition of endomorphisms $\Rightarrow$ $ex = e_1(e_2 x)$
- The product of two matrices $\Rightarrow$ $\mathsf{E} = \mathsf{E}_1\mathsf{E}_2$
- Composed functional $\Rightarrow$ $\mathsf{y}(w) = \int_0^1 E_1(w,v)\left[\int_0^1 E_2(u,v)\mathsf{x}(v)dv\right]du$
- Linear Algebra $\Rightarrow$ Process signals (vectors) by composing linear maps (endomorphisms)



- An endomorphism is the set of all linear transformations that can be applied to a signal
- There is no structure in the space of endomorphisms $\Rightarrow$ Learning in $\text{End}(M)$ does not scale
- Introducing structure $\equiv$ Restricting the set of allowable endomorphisms

$\Rightarrow$ We accomplish this with an Algebra and a Representation to define sets of convolutional filters

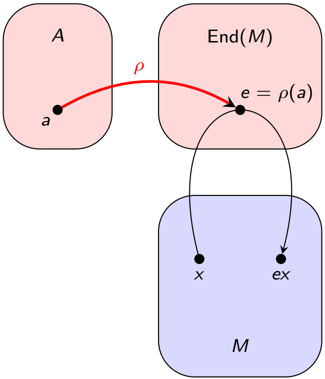
Algebraic Signal Processing

- The algebra of endomorphisms of a vector space does not leverage signal structure

- Convolutional filters use algebras and homomorphisms to restrict allowable linear transformations

- The signals we want to process live in a vector space M
- Linear processing with elements e of the algebra $\text{End}(M)$
- This is too general $\Rightarrow$ Restrict allowable operations

  $\Rightarrow$ To those that represent another (more restrictive) algebra
- Have to define homomorphisms and representations



Algebra Homomorphism

Let A and A' be two algebras. A homomorphism is a map $\rho : A \to A'$ that preserves the operations of A. I.e., for all $a, b \in A$ we have that

$\Rightarrow$ The homomorphism preserves the sum $\Rightarrow \rho(a + b) = \rho(a) +' \rho(b)$

$\Rightarrow$ The homomorphism preserves the product $\Rightarrow \rho(a * b) = \rho(a) *' \rho(b)$

$\Rightarrow$ The homomorphism preserves the scalar product $\Rightarrow \rho(\alpha \times a) = \alpha \times' \rho(a)$

- Doing operations on the algebra A is the same as carrying operations on the algebra A'

$\Rightarrow$ The converse need not be true. Algebra A' may be "richer." May have more elements

Representation

Consider an algebra A, a vector space M, and a homomorphism ρ from A to $\text{End}(M)$,

$$\rho : A \to \text{End}(M).$$

The pair (M, ρ) is said to be a representation of the associative algebra A

- Ties the abstract algebra A to concrete operations on signals that live in the vector space M
- We say that $a \in A$ is a filter $\Rightarrow$ The action of a on signal x produces the filtered signal $y = \rho(a)x$

- A polynomial over the field F is an expression of the form $\Rightarrow a = \sum_{k=0}^{K} a_k t^k$
- The coefficients a_k are elements of the field F. The sum $\sum_{k=0}^{K}$ and the powers t^k are just symbols.
- The Algebra of polynomials over F is the set of all polynomials along with the operations

Scalar multiplication

$$(\alpha \times a) = \sum_{k=0}^{K} (\alpha \cdot a_k) t^k$$

Vector sum

$$(a + b) = \sum_{k=0}^{K} (a_k + b_k) t^k$$

Algebra product

$$(a * b) = \sum_{k=0}^{K} \left[\sum_{j=0}^{k} a_j \cdot b_{k-j} \right] t^k$$

- Signals x in Vector space $M = \mathbb{R}^n$ $\Rightarrow$ Endomorphisms $\text{End}(M) = \mathbb{R}^{n \times n}$ (square matrices E)

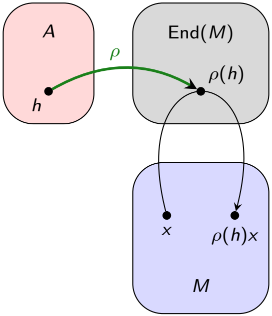
- Processing with E is too general $\Rightarrow$ Suppose that x is supported on a graph with shift operator S

- Define the homomorphism ρ from the algebra of polynomials to $\text{End}(M) = \mathbb{R}^{n \times n}$ in which we map

$$a = \sum_{k=0}^{K} a_k t^k \quad \rightarrow \quad \rho(a) = \sum_{k=0}^{K} a_k S^k$$

- Algebra of polynomials + Homomorphism ρ $\equiv$ Graph signal processing on shift operator S

- An Algebraic SP model is a triplet (A, M, ρ)
- A is an Algebra with unity where filters $h \in A$ live
  - $\Rightarrow$ It defines the rules of convolutional signal processing
- M is a vector space
  - $\Rightarrow$ The space containing the signals $\mathbf{x}$ we want to process
- ρ is a homomorphism from A to the endomorphisms of M
  - $\Rightarrow$ Instantiates the abstract filter h in the space $\text{End}(M)$
- Any $h \in A$ is a filter which operates on signals according to the homomorphism $\Rightarrow \mathbf{y} = \rho(h)\mathbf{x}$



Polynomials in an Algebra and Polynomial Functions

- Given an element of an algebra $a \in A$ and a set of coefficients $h_k \in F$, a polynomial is

$$p_A(a) = h_0 \times 1 + h_1 \times a + h_2 \times (a * a) + h_3 \times (a * a * a) + \ldots = \sum_k h_k \, a^k$$

- We know that $p_A(a) \in A$ because we start from $a \in A$ and use algebra operations throughout

- The element $p_A(a) \in A$ is generated from $a \in A$ using the operations of the algebra $(\times, +, *)$

- We use the algebra's operations $\Rightarrow$ If we change the algebra, the "same" polynomial is different

- For $a' \in A'$ the "same" polynomial performs different operations to generate $p_{A'}(a') \in A'$

$$p_{A'}(a') \;=\; h_0 \times' 1 \;+'\; h_1 \times' a \;+'\; h_2 \times' (a *' a) \;+'\; h_3 \times' (a *' a *' a) \;+'\; \ldots \;=\; \sum_k h_k\,(a')^k$$

- We use the operations $(\times', +', *')$ of the algebra A'.

- A related object is the polynomial function over the field F

- Consider given coefficients $h_k \in F$ and a variable $\lambda \in F$. The polynomial function p_F takes values

$$p_F(\lambda) = h_0 \cdot 1 + h_1 \cdot \lambda + h_2 \cdot (\lambda \cdot \lambda) + h_3 \cdot (\lambda \cdot \lambda \cdot \lambda) + \ldots = \sum_k h_k \lambda^k$$

- It is function of $\lambda \Rightarrow p_F : F \to F$. Defined in terms of the operations $(\cdot, +)$ of the field F.

- Resemblance to frequency responses is not coincidental

- Generalize to multiple elements $\Rightarrow$ For set of elements $\mathcal{A} = a_1 \ldots a_r \subseteq A$ define the polynomial

$$p_A(\mathcal{A}) = \sum_{k_1,\ldots k_r} h_{k_1,\ldots,k_r} a_1^{k_1} \ldots a_r^{k_r}$$

- Associated with the set of coefficients $h_{k_1,\ldots,k_r} \in F$. Algebra Operations. An element of the algebra

- For the same set of coefficients we define the polynomial function $p_F(\lambda_1, \ldots, \lambda_r) = p_F(\mathcal{L})$

$$p_F(\mathcal{L}) = \sum_{k_1,\ldots k_r} h_{k_1,\ldots,k_r} \lambda_1^{k_1} \ldots \lambda_r^{k_r}$$

- A different (simpler) object.. A function with variables $\lambda_i \in F$. Operations performed on the field F

Generators, Shift Operators, and Frequency Representations

- Algebraic Signal Processing is an abstraction of Convolutional Information Processing

- Three central components $\Rightarrow$ generators, shift operators, and frequency representations

Definition (Generators)

The set $\mathcal{G} \subseteq A$ generates A if all $h \in A$ can be represented as polynomials of the elements of $\mathcal{G}$,

$$h = \sum_{k_1,\dots k_r} h_{k_1,\dots,k_r} {g_1}^{k_1} \dots {g_r}^{k_r} = p_A(\mathcal{G})$$

- The elements $g \in \mathcal{G}$ are the generators of A. And $h = p_A(\mathcal{G})$ is the polynomial that generates h

  $\Rightarrow$ Filters can be built from the generating set using the operations of the algebra

- Given the algebra, the generators are given $\Rightarrow$ Filter h is completely specified by its coefficients

- The algebra of polynomials of a single variable t is generated by the polynomial $g = t$

$\Rightarrow$ Algebra elements are expressions $h = \sum_k h_k t^k$ $\Rightarrow$ They can be generated as $h = \sum_k h_k t^k$

- Algebra of polynomials of two variables x and y is generated by the polynomials $g_1 = x$ and $g_2 = y$

$\Rightarrow$ Algebra elements are expressions $h = \sum_k h_{kl} x^k y^l$ $\Rightarrow$ Can be generated as $h = \sum_k h_{kl} x^k y^l$

Definition (Shift Operators)

Let (M, ρ) be a representation of A and $\mathcal{G} \subseteq A$ a generator set of A. We say S is a shift operator if

$$\mathsf{S} = \rho(g), \quad \text{for some } g \in \mathcal{G}$$

The set $\mathcal{S} = \{\rho(g),\ g \in \mathcal{G}\}$ is the shift operator set of the representation (M, ρ) of algebra A.

- Generators g of Algebra A mapped to shift operators S in the space $\text{End}(M)$ of endomorphisms of M

Theorem (Filters as Polynomials on Shift Operators)

Let (M, ρ) be a representation of A with generators $g_i \in \mathcal{G}$ and shift operators $\mathbf{S}_i = \rho(g_i) \in \mathcal{S}$.

The representation $\rho(h)$ of filter h is a polynomial on the shift operator set,

$$h \;=\; p_A(\mathcal{G}) \;=\; \sum_{k_1,\ldots k_r} h_{k_1,\ldots,k_r}\, {g_1}^{k_1} \ldots {g_r}^{k_r} \quad \Rightarrow \quad \rho(h) \;=\; p_M(\mathcal{S}) \;=\; \sum_{k_1,\ldots k_r} h_{k_1,\ldots,k_r}\, {\mathbf{S}_1}^{k_1} \ldots {\mathbf{S}_r}^{k_r}$$

Proof: The theorem is true because the homomorphism ρ preserves the operations of the algebra

$$\rho(h) \;=\; \rho\Bigg(\sum_{k_1,\ldots k_r} h_{k_1,\ldots,k_r}\, {g_1}^{k_1} \ldots {g_r}^{k_r} \Bigg) \;=\; \sum_{k_1,\ldots k_r} h_{k_1,\ldots,k_r}\, \rho(g_1)^{k_1} \ldots \rho(g_r)^{k_r} \qquad \blacksquare$$

- ASP $\Rightarrow$ Vector space $\equiv$ Signals $+$ Algebra $\equiv$ Filters $+$ Homomorphism $\equiv$ Filter Instantiation.

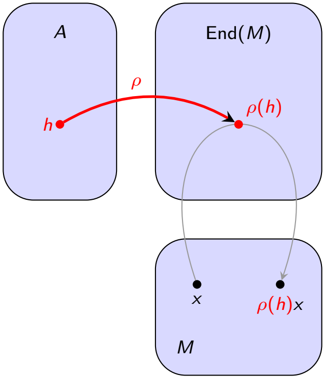
- In principle, the homomorphism $\rho : A \to \text{End}(M)$ has to be specified for all possible filters $h \in A$.

- In reality, it suffices to specify ρ for the generator set

$$g_i \quad \Rightarrow \quad \mathsf{S}_i = \rho(g_i)$$

- Other filters are polynomials on g_i $\Rightarrow$ $h = p_A(\mathcal{G})$

- Which instantiate to polynomials on S_i $\Rightarrow$ $\rho(h) = p_M(\mathcal{S})$



- ASP $\Rightarrow$ Vector space $\equiv$ Signals $+$ Algebra $\equiv$ Filters $+$ Homomorphism $\equiv$ Filter Instantiation.

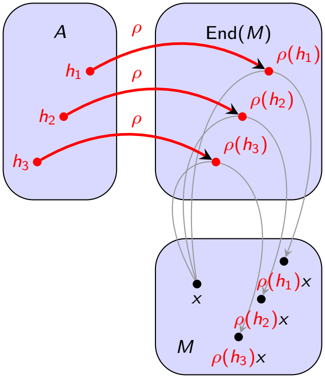
- In principle, the homomorphism $\rho : A \rightarrow \text{End}(M)$ has to be specified for all possible filters $h \in A$.

- In reality, it suffices to specify ρ for the generator set

$$g_i \quad \Rightarrow \quad \mathsf{S}_i = \rho(g_i)$$

- Other filters are polynomials on g_i $\Rightarrow$ $h = p_A(\mathcal{G})$

- Which instantiate to polynomials on S_i $\Rightarrow$ $\rho(h) = p_M(\mathcal{S})$



- ASP $\Rightarrow$ Vector space $\equiv$ Signals $+$ Algebra $\equiv$ Filters $+$ Homomorphism $\equiv$ Filter Instantiation.

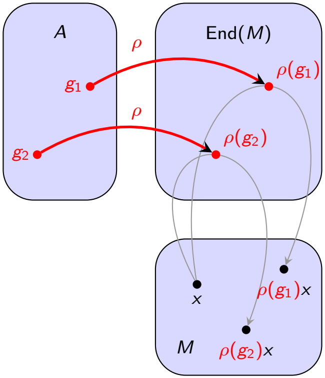
- In principle, the homomorphism $\rho : A \to \text{End}(M)$ has to be specified for all possible filters $h \in A$.

- In reality, it suffices to specify ρ for the generator set

$$g_i \quad \Rightarrow \quad \mathsf{S}_i = \rho(g_i)$$

- Other filters are polynomials on g_i $\Rightarrow$ $h = p_A(\mathcal{G})$

- Which instantiate to polynomials on S_i $\Rightarrow$ $\rho(h) = p_M(\mathcal{S})$



- GSP $\Rightarrow$ Signals in $\mathbb{R}^n$ + Algebra of Polynomials + Homomorphism ρ defined by the map

$$h = \sum_{k=0}^{K} h_k t^k \quad \rightarrow \quad \rho(h) = \sum_{k=0}^{K} h_k \mathbf{S}^k$$

- Equivalent to the (much) simpler specification of the homomorphism $\Rightarrow \rho(t) = \mathbf{S}$

$\Rightarrow$ This is possible because the algebra of polynomials is generated by $g = t$

Frequency Representation of an Algebraic Filter

Definition (Frequency Representation)

In an algebra A with generators $g_i \in \mathcal{G}$ we are given the filter h expressed as the polynomial

$$h \;=\; \sum_{k_1,\dots k_r} h_{k_1,\dots,k_r}\, {g_1}^{k_1} \dots {g_r}^{k_r} \;=\; p_A(\mathcal{G})$$

The frequency representation of h over the field F[1] is the polynomial function with variables $\lambda_i \in \mathcal{L}$

$$p_F(\mathcal{L}) \;=\; \sum_{k_1,\dots k_r} h_{k_1,\dots,k_r}\, {\lambda_1}^{k_1} \dots {\lambda_r}^{k_r}$$

- The two polynomials are different creatures $\Rightarrow$ The frequency representation is a simpler object

[1] The field is unspecified in the definition. But unless otherwise noted F refers to the field over which the vector space M is defined

Three Polynomials (or More)

- The central components of an ASP model are three different polynomials

$\Rightarrow$ The filter. The filter's instantiation on the space of endomorphisms The frequency response

- These three polynomials have the same coefficients. They are related. But similar though they look

$\Rightarrow$ They are different objects. They utilize different operations. They have different meanings.

P1: The Filter

- A polynomial on the elements g_i of the generator set $\mathcal{G}$ of the algebra A

$$p_A(\mathcal{G}) = \sum_{k_1,\ldots k_r} h_{k_1,\ldots,k_r}\, {g_1}^{k_1} \ldots {g_r}^{k_r}$$

- Sum, product, and scalar product are the operations of the algebra A
- The abstract definition of a filter. Untethered to a specific signal model

P2: The Instantiation of the Filter in the space of Endomorphisms $\text{End}(M)$

- A polynomial on the elements $\mathsf{S}_i = \rho(g_i)$ of the shift operator set $\mathcal{S}$

$$p_M(\mathcal{S}) \;=\; \sum_{k_1,\ldots k_r} h_{k_1,\ldots,k_r}\, \mathsf{S}_1^{\;k_1} \ldots \mathsf{S}_r^{\;k_r}$$

- Sum, product, and scalar product are the operations of the algebra of Endomorphisms of M
- The concrete effect that a filter has on a signal $\mathbf{x}$. Tethered to a specific signal model

- "Or more" $\Rightarrow$ The same abstract filter can be instantiated in multiple signal models

P3: The Frequency Response

- A polynomial function where the variables $\lambda_i \in \mathcal{L}$ take values on the field F

$$p_F(\mathcal{L}) \;=\; \sum_{k_1,\ldots k_r} h_{k_1,\ldots,k_r}\, {\lambda_1}^{k_1} \ldots {\lambda_r}^{k_r}$$

- Sum and product are the operations of field F. $\Rightarrow$ E.g, a polynomial with real variables
- Simpler representation of a filter. Untethered to a specific signal model (except for the field)

- The tool we use for analysis. $\Rightarrow$ To explain discriminability, stability and transferability

(P1) Abstract filter $\Rightarrow p_A(t) = \sum_{k=0}^{K} h_k t^k$. Abstract definition. Untethered to any specific graph

(P2) Filter instantiated on a graph $\Rightarrow p_M(\mathbf{S}) = \sum_{k=0}^{K} h_k \mathbf{S}^k$. Concrete instantiation. Tethered to $\mathbf{S}$

$\Rightarrow$ On another graph $\Rightarrow p_M(\hat{\mathbf{S}}) = \sum_{k=0}^{K} h_k \hat{\mathbf{S}}^k$. Concrete instantiation. Tethered to $\hat{\mathbf{S}}$

$\Rightarrow$ On a graphon $\Rightarrow p_M(T_W) = \sum_{k=0}^{K} h_k T_W^{(k)}$. Concrete instantiation. Tethered to graphon W

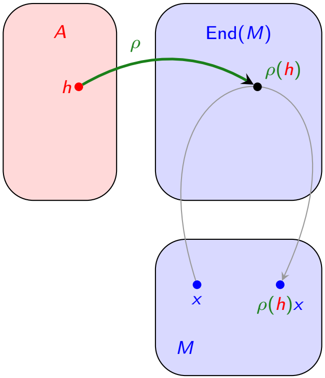
(P3) Frequency response $\Rightarrow p_F(\lambda) = \sum_{k=0}^{K} h_k \lambda^k$. Simple function of one variable. Same for all instances

Convolutional Information Processing

- Algebraic filters are a generic abstraction of the common features of convolutional signal processing

- Graph, time, and image convolutions can be expressed as particular cases of algebraic filters

- To specify an ASP model we need to specify
  - $\Rightarrow$ A vector space M where signals x live
  - $\Rightarrow$ An algebra A where convolutional filters h live
  - $\Rightarrow$ A homomorphism ρ mapping filters to $\text{End}(M)$
- The signal x is processed to the output $\Rightarrow y = \rho(h)x$



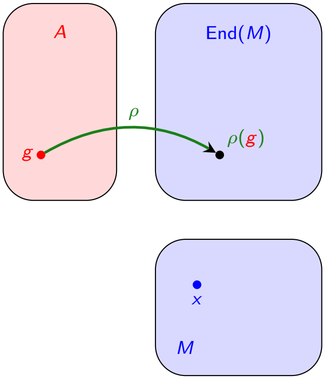
- To specify an ASP model we need to specify

  $\Rightarrow$ A vector space M where signals x live

  $\Rightarrow$ An algebra A where convolutional filters h live

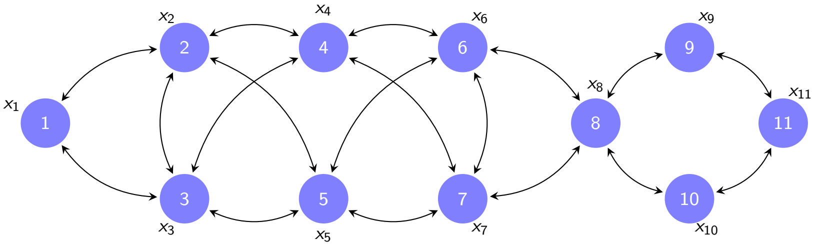
  $\Rightarrow$ The images $S = \rho(g)$ of the algebra generators g

- The signal x is processed to the output $\Rightarrow y = \rho(h)x$



Task

Process signals x that are supported on a graph with n nodes. A matrix representation of the graph is given in the matrix S.



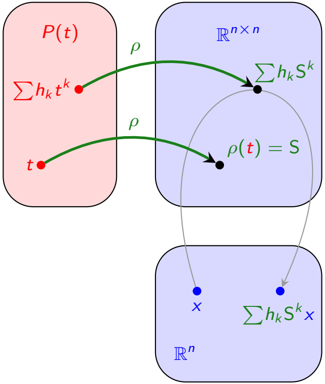
- GSP in the graph S is a particular case of ASP in which

$\Rightarrow M = \mathbb{R}^n \;\Rightarrow$ Vectors with n components

$\Rightarrow A = P(t) \;\Rightarrow$ The algebra of polynomials $h = \sum_k h_k t^k$

$\Rightarrow$ Shift operator $\rho(t) = \mathbf{S} \;\Rightarrow$ Resulting in filters

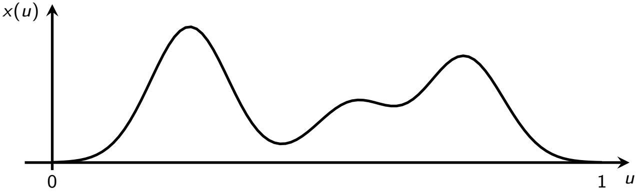
$$\rho(h) \;=\; \rho\left(\sum_k h_k t^k\right) \;=\; \sum_k h_k \mathbf{S}^k$$

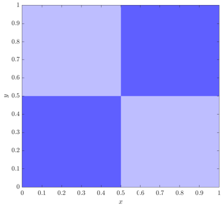


- Processing $\mathbf{x}$ with filter $\rho(h)$ yields output $\Rightarrow \mathbf{y} \;=\; \rho(h)\mathbf{x} \;=\; \rho\left(\sum_k h_k t^k\right)\mathbf{x} \;=\; \sum_k h_k \mathbf{S}^k \mathbf{x}$

Task

Process signals x supported on a graphon W. The signals have finite energy. I.e, $x \in L_2([0,1])$

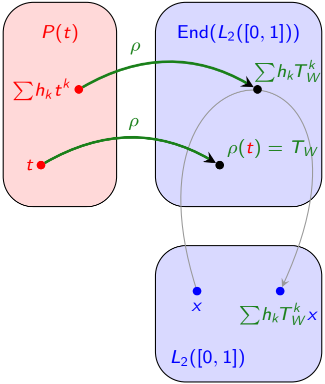




- WSP in the graphon W is a particular case of ASP

$\Rightarrow M = L_2([0,1])$ $\Rightarrow$ Finite-energy functions in $[0,1]$

$\Rightarrow A = P(t)$ $\Rightarrow$ The algebra of polynomials $h = \sum_k h_k t^k$



- WSP in the graphon W is a particular case of ASP

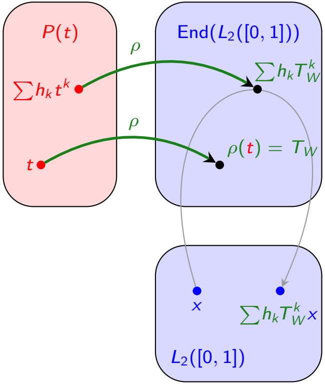
$\Rightarrow$ Shift operator is $\rho(t) = T_W$ defined as

$$(T_w \mathbf{x})(u) = \int_0^1 W(u,v)\mathbf{x}(v)dv$$

$\Rightarrow$ This mapping of the generator t yields filters

$$\rho(h) = \rho\left(\sum_k h_k t^k\right) = \sum_k h_k T_w^k$$

where T_w^k represents k applications of T_w



- Processing $\mathbf{x}$ with filter $\rho(h)$ yields output $\Rightarrow \mathbf{y} = \rho(h)\mathbf{x} = \rho\left(\sum_k h_k t^k\right)\mathbf{x} = \sum_k h_k T_W^k \mathbf{x}$

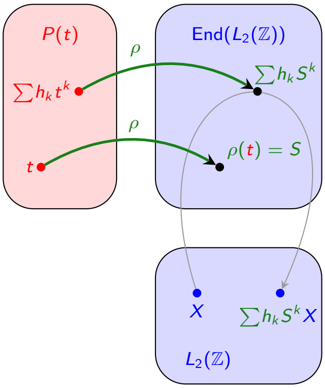
Task

Process sequences X with values $(X)_n = x_n$ for integer indexes $n \in \mathbb{Z}$. The sequences have finite energy. We say that $X \in L_2(\mathbb{Z})$

- DTSP is a particular case of ASP in which

$\Rightarrow M = L_2(\mathbb{Z}) \;\Rightarrow$ Finite-energy sequences in $\mathbb{Z}$

$\Rightarrow A = P(t) \;\Rightarrow$ The algebra of polynomials $h = \sum_k h_k t^k$



- DTSP is a particular case of ASP in which

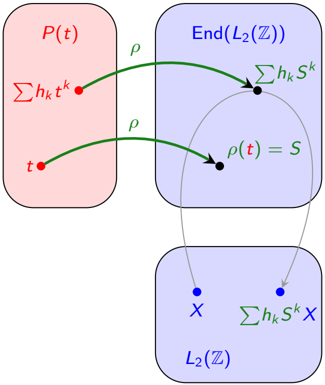
  $\Rightarrow$ Shift operator is a time shift $\rho(t) = S$ such that

$$(SX)_n = (X)_{n-1}$$

  $\Rightarrow$ This mapping of the generator t yields filters

$$\rho(h) \;=\; \rho\left(\sum_k h_k t^k\right) \;=\; \sum_k h_k S^k$$

  where S^k represents k applications of S



- Processing X with $\rho(h)$ yields $\Rightarrow \left(Y\right)_n \;=\; \left(\rho(h)X\right)_n \;=\; \left(\sum_k h_k S^k X\right)_n \;=\; \sum_k h_k \left(X\right)_{n-k}$

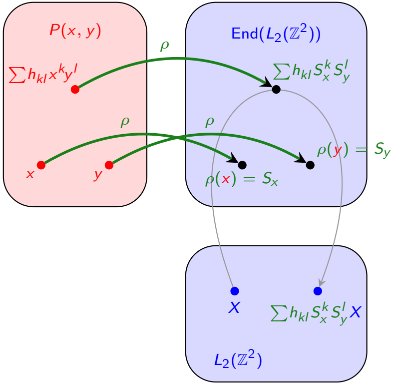
Task

Process images, defined as sequences X with values $(X)_{mn} = x_{mn}$ that depend on two integer indexes $m, n \in \mathbb{Z}$. The sequences have finite energy. We say that $X \in L_2(\mathbb{Z}^2)$

- IP is a particular case of ASP in which

$\Rightarrow$ $M = L_2(\mathbb{Z}^2)$ $\Rightarrow$ Finite-energy sequences in $\mathbb{Z}^2$

$\Rightarrow$ $A = P(x, y)$ $\Rightarrow$ Two-letter polynomials $h = \sum_k h_{kl} x^k y^l$



- IP is a particular case of ASP in which

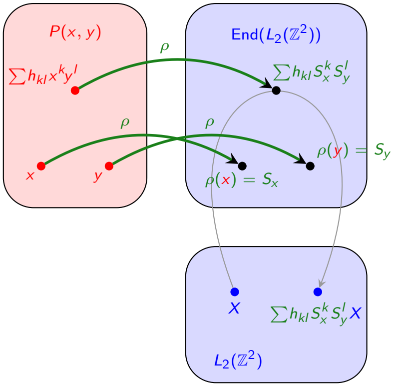
  $\Rightarrow$ Two shift operators $\rho(x) = S_x$ and $\rho(y) = S_y$

$$(S_x X)_{mn} = (X)_{(m-1)n} \quad (S_y X)_{mn} = (X)_{m(n-1)}$$

  $\Rightarrow$ This mapping of the generators x and y yields filters

$$\rho(h) \;=\; \rho\left(\sum_k h_k t^k\right) \;=\; \sum_k h_{kl} S_x^k S_y^l$$

S_x^k or S_x^k represent k or l applications of S_x orS_l



- Processing X yields $\Rightarrow \left(Y\right)_{mn} = \left(\rho(h)X\right)_{mn} = \left(\sum_{kl} h_{kl} S_x^k S_y^l X\right)_{mn} = \sum_k h_k \left(X\right)_{(m-k)(n-l)}$

ASP is a Generic Analysis Tool

- Algebraic SP encompasses Graph SP, graphon SP, Time SP, and Image SP as particular cases

$\Rightarrow$ Other particular cases exist. Notably, Group SP

- ASP provides a framework for fundamental analyses that hold for all forms of convolutional filters

Algebraic Neural Networks

- We introduce Algebraic Neural Networks (AlgNNs) to generalize neural convolutional networks

- Algebraic Neural Networks (AlgNNs) are stacked layered structures

$\Rightarrow$ Each layer consists of the algebraic signal model $(A_\ell, M_\ell, \rho_\ell)$ and σ_ℓ $\Rightarrow$ nonlinearity and pooling

- The output at layer ℓ in the AlgNN is given by

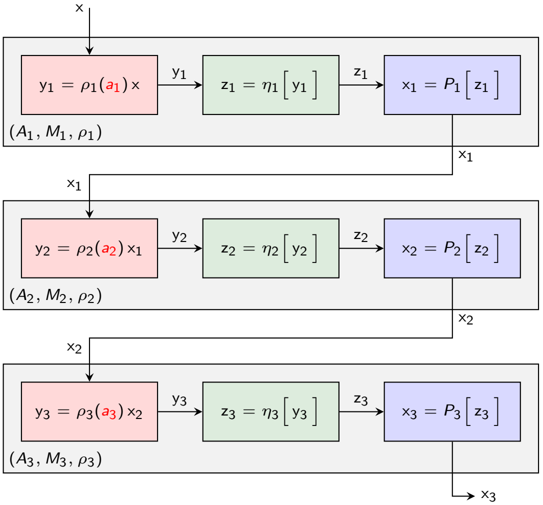
$$\mathrm{x}_\ell = \sigma_\ell\left(\rho_\ell(a_\ell)\mathrm{x}_{\ell-1}\right) \quad \text{with} \quad a_\ell \in A_\ell$$

- The operation is equivalent to

$$\mathrm{x}_\ell = \Phi(\mathrm{x}_{\ell-1}, \mathcal{P}_\ell, \mathcal{S}_\ell)$$

$\Rightarrow$ $\mathcal{P}_\ell \subset A_\ell$ represents the properties of the filters and $\mathcal{S}_\ell$ is the shifts associated to (M_ℓ, ρ_ℓ)

- Algebraic Neural Network $\{(A_\ell, M_\ell, \rho_\ell)\}_{\ell=1}^{3}$ architecture with three layers.



- The processing at each layer can be performed by a family of filters

$$x_\ell^f = \sigma_\ell \left(\sum_{g=1}^{F_\ell} \rho_\ell(a_\ell^{gf}) x_{\ell-1}^g \right)$$

$\Rightarrow$ where a_ℓ^{gf} is the filter processing gth feature $x_{\ell-1}^g$ and F_ℓ is the number of features

- Layers may use (different) specific algebraic signal models $(A_\ell, M_\ell, \rho_\ell)$

- Trainable parameters are the filters $\{a_\ell^{fg}\}_{\ell fg}$. Numerically, we train directly on $\rho_\ell(a_\ell^{fg})$.

Pooling

- The pooling increases the computational efficiency and improve the performance
- The operation is attributed to the composition operator $\sigma_\ell = P_\ell \circ \eta_\ell$

  $\Rightarrow$ η_ℓ is a pointwise nonlinearity and P_ℓ is a pooling operator

- The operator σ_ℓ is only assumed to be Lipschitz and to have zero as a fixed point $\sigma_\ell(0) = 0$
- The pooling operator P_ℓ projects elements from a given vector space M_ℓ to another $M_{\ell+1}$

- Traditional CNNs particularize the algebraic signal model in AlgNNs as the typical signal model.

- Make $M = \mathbb{C}^N$ and A the polynomial algebra in the variable t and module $t^N - 1$

  $\Rightarrow$ $\mathsf{S} = C$ is the cyclic shift operator satisfying $C^k = C^{k \mod N}$.

- The fth feature at layer ℓ is given by

$$\mathbf{x}_\ell^f = \sigma_\ell \left(\sum_{g=1}^{F_\ell} \sum_{i=0}^{K-1} h_i^{gf} C_\ell^i \mathbf{x}_{\ell-1}^g \right)$$

- In this case P_ℓ is a sampling operator while $\eta_\ell(u) = \max\{0, u\}$ is the ReLU.

- The GNNs particularize the algebraic signal model in AlgNNs as the graph signal model.
- Let $M = \mathbb{C}^N$ with components x_n of $\mathrm{x} \in M$ associated with graph nodes
- Let A be the polynomial algebra with elements $a = \sum_{k=0}^{K-1} h_k t^k$

  $\Rightarrow$ The homomorphism filter is given by $\rho(a) = \sum_{k=0}^{K-1} h_k \mathrm{S}^k$

  $\Rightarrow$ $\mathrm{S} \in \mathbb{C}^{N \times N}$ is the graph matrix representation, e.g., adjacency matrix, Laplacian matrix, etc.
- The fth feature at layer ℓ is given by

$$\mathrm{x}_\ell^f = \sigma_\ell \left(\sum_{g=1}^{F_\ell} \sum_{k=0}^{K-1} h_k^{gf} \mathrm{S}^k \mathrm{x}_{\ell-1}^g \right)$$

Perturbation Models

- We define perturbations in the context of algebraic signal processing

- To define perturbations (deformations) in ASP we recall the notion of generator of an algebra
- Generators: The set $\mathcal{G} \subseteq A$ generates A if all $a \in A$ are polynomial functions of elements of $\mathcal{G}$
- Shift Operators: The set $\mathcal{S}$ of homomorphism images $\mathbf{S} = \rho(g)$ is the set of shift operators
- Definitions of generators and shift operators allows writing filters as polynomials on shift operators

$$\rho(a) = p_M\big(\rho(\mathcal{G})\big) = p_M\big(\mathcal{S}\big) = p\big(\mathcal{S}\big)$$

- We define perturbations of Algebraic models as perturbations of shift operators $\Rightarrow \tilde{\mathsf{S}} = \mathsf{S} + \mathsf{T}(\mathsf{S})$

- The ASP model (A, M, ρ) is consequently perturbed to the triplet $(A, M, \tilde{\rho})$ such that

$$\tilde{\rho}(a) = p_M\big(\tilde{\rho}(g)\big) = p_M\big(\tilde{S}\big)$$

  That is, the polynomials that define filters are the same. But they use the perturbed shift operator

- Graphs $\Rightarrow$ Shift operator S represents a graph $\Rightarrow$ Perturbed operator $\tilde{\mathsf{S}}$ represents different graph

- Time $\Rightarrow$ S represents translation equivariance $\Rightarrow$ $\tilde{\mathsf{S}}$ represents quasi-translation equivariance

- Our definition limits the perturbation of the homomorphism to the perturbation of the shift operator

  $\Rightarrow$ Motivated by practice: seen in graph, discrete time, group and graphon signals

- The model perturbs the homomorphism ρ but not the algebra A $\Rightarrow$ A define the type of operations

- Notice that a perturbation $\tilde{x} = Tx$ acting on the signal x can be interpreted as a transformation of S

$$\mathbf{S}\tilde{x} = \mathbf{S}(Tx) = (\mathbf{S}T)x = \tilde{\mathbf{S}}x$$

Perturbation Model: Deformations

- We will consider a first order generic model of small perturbation for the shift operator(s) S

- It includes an absolute or additive perturbation T_0 and a relative or multiplicative perturbation $T_1 S$

$$T(S) = T_0 + T_1 S$$

- The operators T_0 and T_1 are compact normal with norms satisfying that $\|T_0\| \leq 1$ and $\|T_1\| \leq 1$

- $\|T_0\| \leq 1$, $\|T_1\| \leq 1$ not strong requirements as our interest is on perturbations with $\|T(S)\| \ll 1$

Example: perturbations of graph filters

- Apply the same filter h to the same signal x on different graphs shift operators S and $\tilde{S}$

$$H(S)\,x \;=\; \sum_{k=0}^{K-1} h_k S^k x \qquad\qquad H(\tilde{S})\,x \;=\; \sum_{k=0}^{K-1} h_k \tilde{S}^k x$$

- Filter $H(S)\,x \;\Rightarrow$ Coefficients h_k. Input signal x. Instantiated on shift S
- Filter $H(\tilde{S})\,x \;\Rightarrow$ Same Coefficients h_k. Same Input signal x. Instantiated on perturbed shift $\tilde{S}$

- Perturbed model $\tilde{S}$ is the matrix $\tilde{S} = T_0 + T_1 S \;\Rightarrow T_0$ additive and T_1 relative perturbations

Stability Theorems

- We define the notion of stability in the context of algebraic signal processing
- We discuss the stability properties of algebraic filters and algebraic neural networks

Stability: recalling basic notions and notation

- In the algebraic signal model (A, M, ρ) filters are defined by the operators $\rho(a) \in \text{End}(M)$, $a \in A$

- If A is generated by the set $\mathcal{G} \subset A$ $\Rightarrow$ $a \in A$ can be written as $a = p(g)$ with $g \in \mathcal{G}$, p: polynomial

- Any filter $\mathbf{H} \in \text{End}(M)$ is defined by operators $p(\mathbf{S})$ where $\mathbf{S} = \rho(g)$ $\Rightarrow$ Filters are functions of $\mathbf{S}$

- Filters are polynomial functions of the shift operators $\mathbf{S} \in \mathcal{S}$ $\Rightarrow$ We use denote filters as $p(\mathbf{S})$

Stable Operators:

We say operator $p(\mathrm{S})$ is stable if there exist constants C_0, $C_1 > 0$ such that

$$\left\|p(\mathrm{S})\mathrm{x} - p(\tilde{\mathrm{S}})\mathrm{x}\right\| \leq \left[C_0 \sup_{\mathrm{S}\in\mathcal{S}} \|\mathrm{T}(\mathrm{S})\| + C_1 \sup_{\mathrm{S}\in\mathcal{S}} \left\|D_{\mathrm{T}}(\mathrm{S})\right\| + \mathcal{O}\left(\|\mathrm{T}(\mathrm{S})\|^2\right)\right] \|\mathrm{x}\|$$

for all $\mathrm{x} \in \mathcal{M}$ and $D_{\mathrm{T}}(\mathrm{S})$ denoting the Fréchet derivative of T.

- $\left\|p(\mathrm{S})\mathrm{x} - p(\tilde{\mathrm{S}})\mathrm{x}\right\|$ is bounded by the size of the deformation. Measured by value and rate of change
- Stability is not a given $\Rightarrow$ Counter examples in GNN and processing of time signals.

- Filters are polynomials on shift operators $\Rightarrow$ Isomorphic to polynomials with complex variables

- Lipschitz Filter: Polynomial $p : \mathbb{C} \to \mathbb{C}$ is Lipschitz if $\|p(\lambda) - p(\mu)\| \leq L_0 \|\lambda - \mu\|$ for some L_0

- Integral Lipschitz: Polynomial $p : \mathbb{C} \to \mathbb{C}$ is Integral Lipschitz if $\left\| \lambda \frac{dp(\lambda)}{d\lambda} \right\| \leq L_1$ for some L_1

- Restricted attention to algebras with a single generator. Generalizations are cumbersome but ready

Theorem (Stability of Algebraic Filters)

A filter that is Lipschitz and Integral Lipschitz is stable, i.e.

$$\left\|p(\mathbf{S})\mathbf{x} - p(\tilde{\mathbf{S}})\mathbf{x}\right\| \leq \left[(1+\delta)\left(L_0 \sup_{\mathbf{S}} \|\mathbf{T}(\mathbf{S})\| + L_1 \sup_{\mathbf{S}} \|D_{\mathbf{T}}(\mathbf{S})\|\right) + \mathcal{O}\left(\|\mathbf{T}(\mathbf{S})\|^2\right)\right] \|\mathbf{x}\|$$

- Good news $\Rightarrow$ Algebraic filters can be made stable to perturbations
- Alas, we either have stability or discriminability. Integral Lipschitz Filter $\Rightarrow \left\|\lambda \frac{dp(\lambda)}{d\lambda}\right\| \leq L_1$
- Commutativity factor affects stability constant but does not generate instability

Theorem (Stability of Algebraic Neural Networks, Single Layer)

Let $\Phi_\ell(\mathbf{S}, \mathbf{x})$ and $\Phi_\ell(\tilde{\mathbf{S}}, \mathbf{x})$ be the operators associated with layer ℓ of an Algebraic NN. If the layer filters are Lipschitz and Integral Lipschitz,

$$\left\|\Phi_\ell(\mathbf{S},\mathbf{x}) - \Phi_\ell(\tilde{\mathbf{S}},\mathbf{x})\right\| \leq \left[(1+\delta)\left(L_0 \sup_{\mathbf{S}} \|\mathbf{T}(\mathbf{S})\| + L_1 \sup_{\mathbf{S}} \|D_{\mathbf{T}}(\mathbf{S})\|\right) + \mathcal{O}\left(\|\mathbf{T}(\mathbf{S})\|^2\right)\right] \|\mathbf{x}\|$$

- Good news $\Rightarrow$ Algebraic NNs can be made stable to perturbations. It's the same bound
- Individual layers lose discriminability. Integral Lipschitz Filter $\Rightarrow \left\|\lambda \dfrac{dp(\lambda)}{d\lambda}\right\| \leq L_1$
- Nonlinearity mixes frequency components $\Rightarrow$ Recover discriminability in subsequent layers

Theorem (Stability of Algebraic Neural Networks, Multi-Layer)

Let $\Phi(\mathbf{S}, \mathbf{x})$ and $\Phi(\tilde{\mathbf{S}}, \mathbf{x})$ be the operators associated with an Algebraic NN on L layers. If the layer filters are Lipschitz and Integral Lipschitz,

$$\left\|\Phi(\mathbf{S}, \mathbf{x}) - \Phi(\tilde{\mathbf{S}}, \mathbf{x})\right\| \leq L\left[(1+\delta)\left(L_0 \sup_{\mathbf{S}} \|\mathbf{T}(\mathbf{S})\| + L_1 \sup_{\mathbf{S}} \|D_{\mathbf{T}}(\mathbf{S})\|\right) + \mathcal{O}(\|\mathbf{T}(\mathbf{S})\|^2)\right] \|\mathbf{x}\|$$

- It is still the same bound $\Rightarrow$ simply scaled by the number of layers L

Theorem (Stability of Algebraic Neural Networks, Multiple Generators)

Let $\Phi(\mathbf{S}, \mathbf{x})$ and $\Phi(\tilde{\mathbf{S}}, \mathbf{x})$ be the operators associated with an Algebraic NN with M generators on L layers. If the layer filters are Lipschitz and Integral Lipschitz,

$$\left\| \Phi(\mathbf{S}, \mathbf{x}) - \Phi(\tilde{\mathbf{S}}, \mathbf{x}) \right\| \leq ML \left[(1+\delta) \left(L_0 \sup_{\mathbf{S}} \|\mathbf{T}(\mathbf{S})\| + L_1 \sup_{\mathbf{S}} \|D_{\mathbf{T}}(\mathbf{S})\| \right) + \mathcal{O}(\|\mathbf{T}(\mathbf{S})\|^2) \right] \|\mathbf{x}\|$$

- It is still the same bound $\Rightarrow$ simply scaled by the number of generators M and layers L

Spectral Representations

- In this lecture we discuss of notion of spectral decompositions in algebraic signal models

- Recalling algebraic signal model (A, M, ρ) $\Rightarrow$ A: algebra, M: vector space and ρ: homomorphism

- Where (M, ρ) is a representation of A $\Rightarrow$ Each representation of A defines an algebraic signal model

- Given (M_1, ρ_1) and (M_2, ρ_2) representations of A $\Rightarrow$ we can define a direct sum $(M_1 \oplus M_2, \rho)$

- Where the action of $\rho(a)$ on $M_1 \oplus M_2$ is given according to $\rho(a)(\mathrm{x}_1 \oplus \mathrm{x}_2) = (\rho_1(a)\mathrm{x}_1 \oplus \rho_2(a)\mathrm{x}_2)$

Definition (Subrepresentation)

Let (M, ρ) be a representation of A. Then, a representation (U, ρ) of A is a subrepresentation of (M, ρ) if $U \subseteq M$ and U is invariant under all operators $\rho(a)$ for all $a \in A$, i.e. $\rho(a)u \in U$ for all $u \in U$ and $a \in A$.

Definition (Irreducible Representations)

A representation (M, ρ) with $M \neq 0$ is irreducible or simple if the only subrepresentations of (M, ρ) are $(0, \rho)$ and (M, ρ).

Definition (Intertwining operators)

Let (M_1, ρ_1) and (M_2, ρ_2) be two representations of an algebra A. A homomorphism or interwining operator $\phi : M_1 \to M_2$ is a linear operator which commutes with the action of A, i.e.

$$\phi(\rho_1(a)v) = \rho_2(a)\phi(v),$$

And, ϕ is said to be an isomorphism of representations if it is an isomorphism of vectors spaces.

- If (M_1, ρ_1) and (M_2, ρ_2) are isomorphic representations we use the notation $(M_1, \rho_1) \cong (M_2, \rho_2)$

Definition (Fourier decomposition)

For (A, M, ρ) we say that there is a spectral or Fourier decomposition of (M, ρ) if

$$(M, \rho) \cong \bigoplus_{(U_i, \phi_i) \in \text{Irr}\{A\}} (U_i, \phi_i)^{\oplus m(U_i, M)}$$

where $m(U_i, M)$ indicates the number of irreducible subrepresentations of (M, ρ) that are isomorphic to (U_i, ϕ_i). Any signal $\mathrm{x} \in M$ can be therefore represented by the map Δ given by

$$\Delta : \quad M \to \bigoplus_i U_i^{\oplus m(U_i, M)}$$
$$\mathrm{x} \mapsto \hat{\mathrm{x}}$$

The projection of $\hat{\mathrm{x}}$ in each U_i are the Fourier components represented by $\hat{\mathrm{x}}(i)$.

Fourier Decompositions

- It is worth pointing out that the Fourier decomposition of any representation is defined by two sums

$$(M, \rho) \cong \bigoplus_{(U_i, \phi_i) \in \mathrm{Irr}\{A\}} (U_i, \phi_i)^{\oplus m(U_i, M)}$$

- Non isomorphic subrepresentations (external) and one (internal) for Isomorphic subrepresentations

- $\bigoplus$ on the frequencies of the representation while $\bigoplus$ on components associated to a given frequency

- Δ is an intertwining operator $\Rightarrow \Delta(\rho(a)\mathrm{x}) = \rho(a)\Delta(\mathrm{x}) \Rightarrow$ convolution $\rho(a)\mathrm{x} = \Delta^{-1}(\rho(a)\Delta(\mathrm{x}))$

- The projection of $\rho(a)\mathrm{x}$ on U_i is given by $\phi_i(a)\hat{\mathrm{x}}(i)$ where $\phi_i : A \to \mathrm{End}(U_i)$ is a homomorphism
- The collection of the projections of $\rho(a)\mathrm{x}$ on U_i for all i is the spectral representation of $\rho(a)\mathrm{x}$
- The product in $\phi_i(a)\hat{\mathrm{x}}(i)$ translates into different operations depending on the dimension of U_i
- If $\dim(U_i) = 1$, the operator $\phi_i(a)$ is a scalar while if $\dim(U_i) > 1$ and finite $\phi_i(a)$ is a matrix

Remarks

- The link between A and Fourier decomposition is exclusively given by $\phi_i(a)$ which is acting on $\hat{x}(i)$
- Not possible by selecting filters in A to modify the spaces U_i in the spectral decomposition of (M, ρ)
- Two sources of differences between two operators $\rho(a)$ and $\tilde{\rho}(a)$ associated to $(\mathcal{M}, \rho)$ and $(\mathcal{M}, \tilde{\rho})$
- One source of difference can be modified by selecting subsets of A and this is embedded in ϕ_i and $\tilde{\phi}_i$
- Second source of difference $\Rightarrow$ the difference between the spaces U_i and $\tilde{U}_i$ and cannot be modified

- In CNNs the filtering is defined by the polynomial algebra $A = \mathbb{C}[t]/(t^N - 1)$, therefore, we have

$$\rho(a)\mathrm{x} = \sum_{i=1}^{N} \phi_i \left(\sum_{k=0}^{K-1} h_k t^k \right) \hat{\mathrm{x}}(i)\mathbf{u}_i = \sum_{i=1}^{N} \sum_{k=0}^{K-1} h_k \left(e^{-\frac{2\pi ij}{N}} \right)^k \hat{\mathrm{x}}(i)\mathbf{u}_i,$$

- $\mathrm{u}_i(v) = \frac{1}{\sqrt{N}} e^{\frac{2\pi jvi}{N}}$ is the ith column vector of the DFT matrix and $\phi_i(t) = e^{-\frac{2\pi ji}{N}}$ its eigenvalue

- In GNNs the filtering is defined by the polynomial algebra $A = \mathbb{C}[t]$, therefore we have

$$\rho(a)\mathrm{x} = \sum_{i=1}^{N} \phi_i \left(\sum_{k=0}^{K-1} h_k t^k \right) \hat{\mathrm{x}}(i)\mathbf{u}_i = \sum_{i=1}^{N} \sum_{k=0}^{K-1} h_k \lambda_i{}^k \hat{\mathrm{x}}(i)\mathbf{u}_i$$

- u_i is the ith eigenvector of $\rho(t) = \mathrm{S}$, which is the matrix graph, and $\phi_i(t) = \lambda_i$ its ith eigenvalue