Computer Vision, Speech Communication & Signal Processing Group,
National Technical University of Athens, Greece (NTUA)
Robotic Perception and Interaction Unit,
Athena Research and Innovation Center (Athena RIC)

Dynamical Systems on Weighted Lattices: Nonlinear Processing and Optimization

Petros Maragos

IEEE SPS DL Talk, Feb. 2018, University of Southern California, Los Angeles

Outline of this Talk

1. **Motivations & Synopsis of Max-Plus Algebra and Lattices:**
   - **Max-plus Arithmetic and Matrix algebra**
   - **Lattices and Monotone Signal/Vector Operators**
2. **Theoretical Extensions and Unification: Weighted Lattices and Max-* Algebra**
3. **Dynamical Systems on Weighted Lattices:**
   - **Applications in DSP, Vision, Multimedia**
   - **Control, Stability**
   - **Max-product systems and Generalized Viterbi algorithm**
   - **Optimization, Sparsity**

Operation	Meaning
$\bigvee$	Maximum/Supremum: applies for scalars, vectors and matrices
$\bigwedge$	Minimum/Infimum: applies for scalars, vectors and matrices
$\boxed{\star}$ ($\boxed{\star}'$)	General max-$\star$ (min-$\star'$) matrix multiplication
$\boxplus$ ($\boxplus'$)	Max-sum (min-sum) matrix multiplication
$\boxtimes$ ($\boxtimes'$)	Max-product (min-product) matrix multiplication
$\circledast$ ($\circledast'$)	General max-$\star$ (min-$\star'$) signal convolution
$\oplus$ ($\oplus'$)	Max-sum (min-sum) signal convolution
$\otimes$ ($\otimes'$)	Max-product (min-product) signal convolution

max-sum and min-sum

matrix multiplications

$$\boldsymbol{C} = \boldsymbol{A} \boxplus \boldsymbol{B} = [c_{ij}] \quad , \quad c_{ij} = \bigvee_{k=1}^{n} a_{ik} + b_{kj}$$

$$\boldsymbol{C} = \boldsymbol{A} \boxplus' \boldsymbol{B} = [c_{ij}] \quad , \quad c_{ij} = \bigwedge_{k=1}^{n} a_{ik} + b_{kj}$$

signal convolutions

$$(f \oplus h)(t) = \bigvee_{k=-\infty}^{+\infty} f(t-k) + h(k)$$

$$(f \oplus' h)(t) = \bigwedge_{k=-\infty}^{+\infty} f(t-k) + h(k)$$

Linear vs. Max-Plus Algebra: Scalar Operations

$+ \longrightarrow \max$

$\times \longrightarrow +$

Max-plus has properties similar to linear algebra:

- Commutativity: $a \vee b = b \vee a$
- Associativity: $a \vee (b \vee c) = (a \vee b) \vee c$
- Distributivity: $a + (b \vee c) = (a + b) \vee (a + c)$
- Idempotency: $3 \vee 3 = 3$
- Inverse?: $3 \vee x = 6 \Rightarrow x = 6$

  $3 \vee x = 3 \Rightarrow x = ?$

Linear versus Max-Plus Systems

- State space representation: linear vs. max-plus

$$x(k) = Ax(k-1) + Bu(k) \qquad x(k) = A \boxplus x(k-1) \vee B \boxplus u(k)$$

$$y(k) = Cx(k) + Du(k) \qquad y(k) = C \boxplus x(k) \vee D \boxplus u(k)$$

- **Matrix products**
  - Linear: $$[AB]_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$
  - Max-plus: $$[A \boxplus B]_{ij} = \bigvee_{k=1}^{n} a_{ik} + b_{kj}$$
- **Example**

$$\begin{bmatrix} 4 & -1 \\ 2 & -\infty \end{bmatrix} \boxplus \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}, \left\{ \begin{array}{r} \max(x+4, y-1) = 3 \\ x+2 = 1 \end{array} \right\} \Longrightarrow \begin{array}{c} x = -1 \\ y \leq 4 \end{array}$$

- What can we model with max-plus systems?

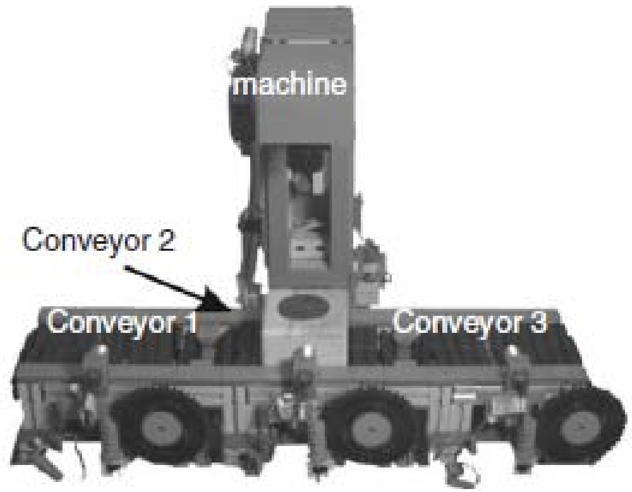
Areas using Max/Min(+) Algebra

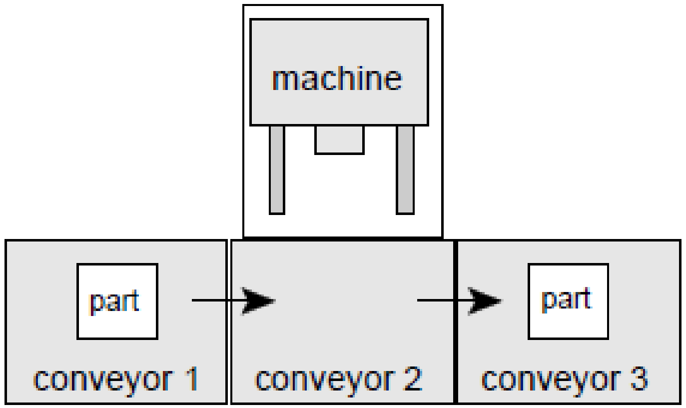
- **Scheduling & Operations Research**: **Minimax Algebra** [Cuninghame-Green 1979]: mainly Max-Plus.
- **Image & Vision, Nonlinear DSP:** Image Algebra [Ritter et al, 1980s-90s], Math. Morphology [Serra 88; Heijmans & Ronse 1992-94]. Morphological Filters, Median & Rank [Maragos & Schafer 1987]. Nonlinear S-S PDEs [Brockett & Maragos 1994]. Distance Transforms (Borgefors 1984; Felzenszwalb etal 2004]
- **Control**: Discrete-Event Dynamical Systems [Cohen etal. 1985; Kamen 1993; Cassandras etal. 2013; Heidergot etal. 2006]. Dioid algebra [Cohen etal. 1989, Baccelli etal. 1992-2001, Gaubert & Max-plus Group 1997; Lahaye & Hardouin etal. 2004, Gondran & Minoux 2008], Max-Linear Systems [Butkovic 2010, van den Boom & de Shutter 2012]
- **Speech & Language** Processing: Finite-State Automata: Tropical Semiring [Mohri, Pereira et al, 1990s; Hori & Nakamura 2013]
- **Graphical Models, Machine Learning**: Max-Sum and Max-Product algorithms in Belief Propagation [Pearl 1988; Bishop 2006].
- **Mathematics, Optimization**: Convex analysis & Optimization [Bellman & Karush 1960’s; Rockafellar 1970; Lucet 2010]. Idempotent Mathematics [Maslov 1987; Litvinov et al.]. Tropical Geometry [Gaubert & Katz 2011; Maclagan & Sturmfels 2015].

Earlier Special Cases, Applications and Motivations

Automated Manufacturing as Max-plus System

Discrete event systems (*)





$x_i(k)$**:** time product k enters conveyor i
$u(k)$: time we put product k in conveyor 1
t_i: conveyor i waiting time
Only one product in a conveyor during each cycle

$$x_1(k) = \max\left(x_1(k-1) + t_1, u(k)\right)$$
$$x_2(k) = \max\left(x_1(k) + t_1, x_2(k-1) + t_2\right)$$
$$x_3(k) = \max\left(x_2(k) + t_2, x_3(k-1) + t_3\right)$$

$$A = \begin{bmatrix} t_1 & -\infty & -\infty \\ 2t_1 & t_2 & -\infty \\ 2t_1 + t_2 & 2t_2 & t_3 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ t_1 \\ t_1 + t_2 \end{bmatrix}$$

$$\boxed{x(k) = A \boxplus x(k-1) \vee B \boxplus u(k)}$$

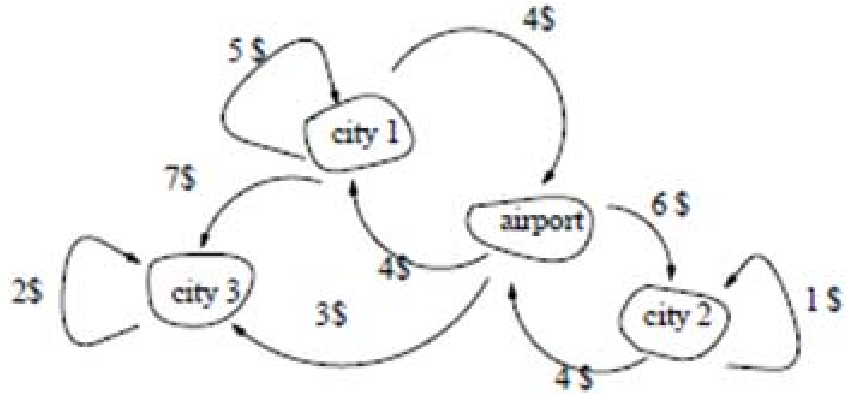
(*) Example from: [G. Schullerus, V. Krebs, B. De Schutter & T. van den Boom, "Input signal design for identification of max-plus-linear systems", Automatica 2006.]

Longest/Shortest Paths as Max/Min-plus Systems

Dynamic Programming

- **Taxi drivers (*)**

$$\boldsymbol{x}(k+1) = \boldsymbol{A}^T \boxplus \boldsymbol{x}(k)$$



$$A^T = \begin{bmatrix} 5 & 4 & -\infty & 7 \\ 4 & -\infty & 6 & 3 \\ -\infty & 4 & 1 & -\infty \\ -\infty & -\infty & -\infty & 2 \end{bmatrix}$$

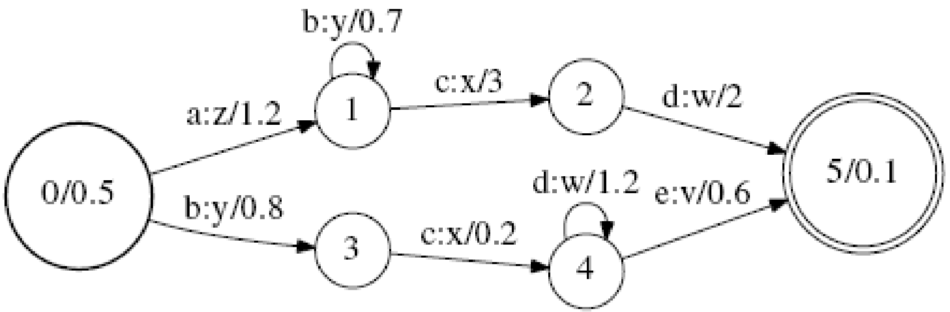
$$money_i(k) = \max_j \left(money_j(k-1) + a_{ji} \right)$$

x_1, x_2, x_3, x_4 correspond to city 1, airport, city 2 and city 3

(*) Example from:
[S. Gaubert and Max-Plus group, "Methods and applications of (max,+) linear algebra“, STACS 1997.]

WFSTs for Speech Recognition: Tropical (Min-Plus) Algebra

Weighted Finite State Transducer (WFST)



[Mohri, Pereira & Ripley, CSL 2002]

[Hori and Nakamura, 2013]

HMM Transducer

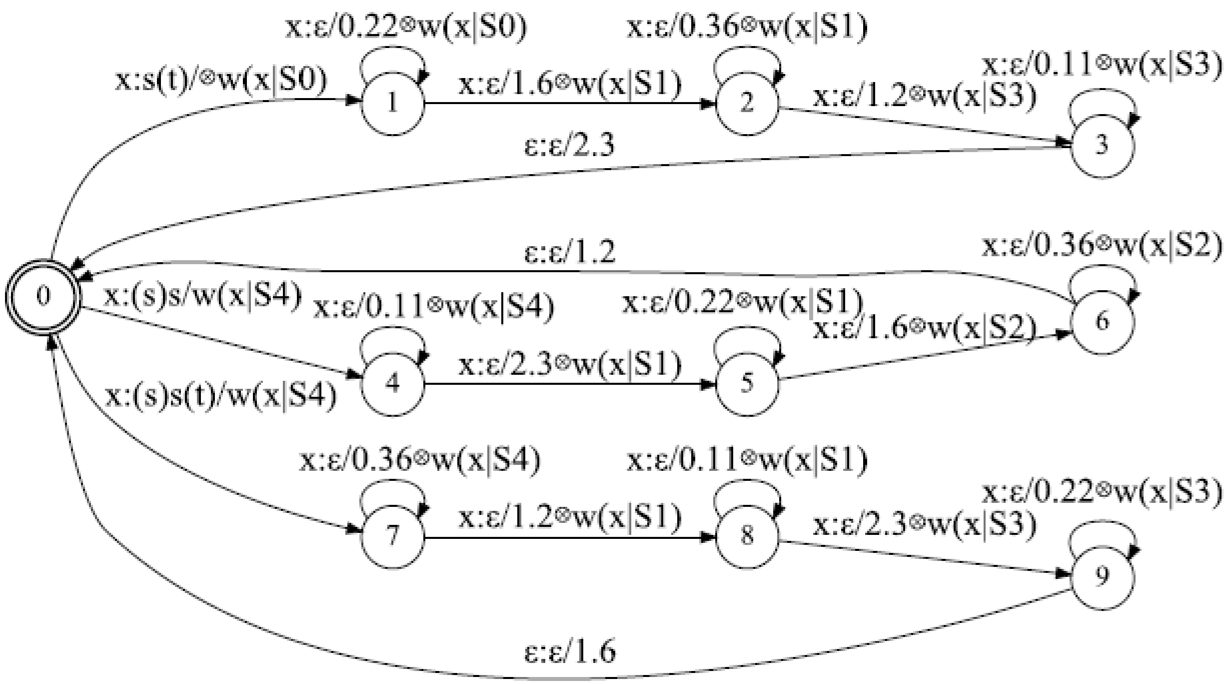


Image Analysis: Euclidean Morphological Operators

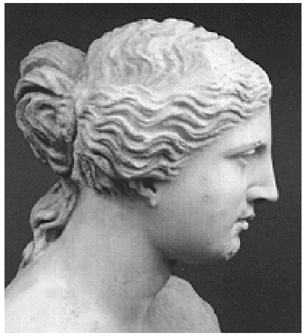
Dilation: $(f \oplus g)(x) = \vee_y f(y) + g(x-y)$

Erosion: $(f \ominus g)(x) = \wedge_y f(y) - g(y-x)$

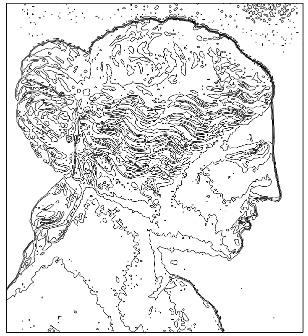
Opening: $f \circ g = (f \oplus g) \ominus g$

Closing: $f \bullet g = (f \oplus g) \ominus g$

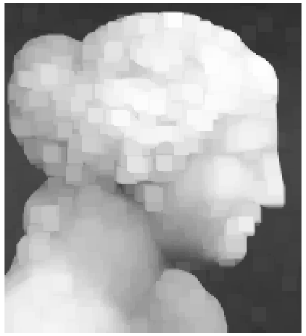
IMAGE



IM. LEVEL CURVES



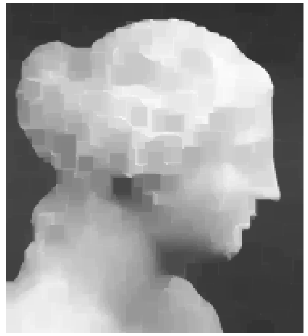
DILATION 9x9



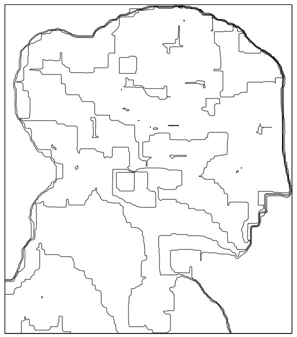
EROSION 9x9

OPENING 9x9

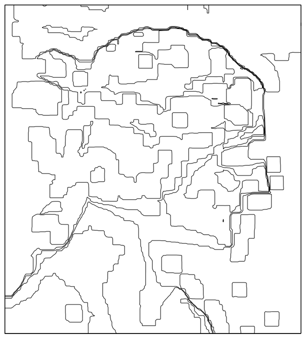
CLOSING 9x9



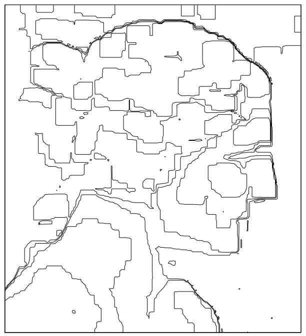
CLOS. LEVEL CURVES



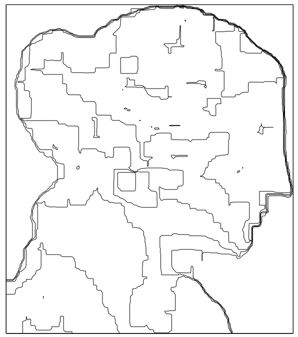
EROS. LEVEL CURVES



OPEN. LEVEL CURVES



CLOS. LEVEL CURVES



Areas using Supremal / Infimal Convolutions

- Image processing and Computer vision (M. Morphology)
- Nonlinear filtering (Order statistics)
- Convex analysis and Optimization
- Discrete Event Dynamical Systems
- Distance computation

$$D_F(p) = \min_{q \in \mathcal{G}} \{ d(p,q) + F(q) \}$$

Sup/Inf-plus Representation of TI Increasing Operators

Theorem (Maragos, IEEE T-PAMI 1989):

Every operator ψ on the set F of extended-real-valued functions that is *translation-invariant (TI) and monotone increasing* can be represented as a supremum (infimum) of inf-plus (sup-plus) convolutions of the input by functions in its *kernel* K. If F consists of u.s.c. functions and the operator is also upper-semicontinuous, then ψ has a *basis* and the representation becomes *minimal*.

$$\psi(f) = \mathrm{SUP}_{g \in K} f \ominus g = \mathrm{INF}_{h \in K^*} f \oplus h$$

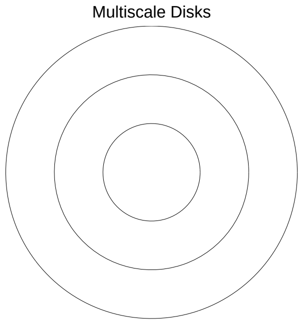
Applications:

- Composite Morphological operators
- Median, Rank, Stack filters
- Linear filters
- Image Denoising
- Curve Evolution, Curvature Motion

PDE for 2D Multiscale Flat Dilations

- initial image, multiscale flat convex structur. elems. (disks)

$f(x, y)$



tB

- multiscale dilations by disks : $\delta(x, y, t) = (f \oplus tB)(x, y)$

$t = 3$

$t = 6$

$t = 9$

- PDE:

$$\frac{\partial \delta}{\partial t} = \|\nabla \delta\| = \sqrt{\left(\frac{\partial \delta}{\partial x}\right)^2 + \left(\frac{\partial \delta}{\partial y}\right)^2}$$

[Brockett & Maragos 1992]

[Alvarez et al. 1993]

(Flat) Lattices and Monotone Operators

Axioms of Lattices $(\mathcal{L}, \vee, \wedge)$

Properties of Lattice Operations

Sup-Semilattice	Inf-Semilattice	Description
L0. $X \vee Y \in \mathcal{L}$	L0′. $X \wedge Y \in \mathcal{L}$	Closure
L1. $X \vee X = X$	L1′. $X \wedge X = X$	Idempotence
L2. $X \vee Y = Y \vee X$	L2′. $X \wedge Y = Y \wedge X$	Commutativity
L3. $X \vee (Y \vee Z) = (X \vee Y) \vee Z$	L3′. $X \wedge (Y \wedge Z) = (X \wedge Y) \wedge Z$	Associativity
L4. $X \vee (X \wedge Y) = X$	L4′. $X \wedge (X \vee Y) = X$	Absorption
L5. $X \leq Y \Longleftrightarrow Y = X \vee Y$	L5′. $X \geq Y \Longleftrightarrow Y = X \wedge Y$	Consistency
L6. $O \vee X = X$	L6.′ $I \wedge X = X$	Identity
L7. $I \vee X = I$	L7.′ $O \wedge X = O$	Absorbing Null

Lattices of Shapes (Sets) and real Signals/Vectors (Functions)

Set Lattices:

Collection: $P(E) = \{\text{subsets } X \subseteq E\}$

partial order: $X \subseteq Y$ [set inclusion]

supremum: $\bigcup_i X_i$ [set union]

infimum: $\bigcap_i X_i$ [set intersection]

negation: X^c [set complement: $E \setminus X$]

Function Lattices (for Signals or Vectors):

Collection: $\{\text{functions } f : E \to \text{complete lattice } K \subseteq \overline{\mathbb{R}}\}$

For Signals $E = \mathbb{Z}^m, \ \mathbb{R}^m$. For Vectors $E = \{1,\ldots,n\}$

partial order: $f \le g$ $[f(x) \le g(x) \ \ \forall x]$

supremum: $\vee_i f_i$ $[(\vee_i f_i)(x) = \vee_i f_i(x)]$

infimum: $\wedge_i f_i$ $[(\wedge_i f_i)(x) = \wedge_i f_i(x)]$

negation: $-f$ $[(-f)(x) = -f(x) \text{ or } c - f(x)]$

Lattice Operator Properties

$$\begin{aligned}
\text{identity}: &\quad \mathsf{id}(X) = X \quad \forall X \\
\text{extensive}: &\quad \mathsf{id} \leq \Psi \\
\text{antiextensive}: &\quad \Psi \leq \mathsf{id} \\
\text{idempotent}: &\quad \Psi\Psi = \Psi \\
\text{involution}: &\quad \Psi\Psi = \mathsf{id}
\end{aligned}$$

$$\begin{aligned}
\text{increasing}: &\quad X \leq Y \Longrightarrow \Psi(X) \leq \Psi(Y) \\
\text{decreasing}: &\quad X \leq Y \Longrightarrow \Psi(X) \geq \Psi(Y)
\end{aligned}$$

Operators on Complete Lattices

($\leq$ = partial ordering, V = supremum, Λ = infimum)

- ψ is **increasing** iff $f \leq g \Rightarrow \psi(f) \leq \psi(g)$.
- δ is **dilation** iff $\delta(\vee_i f_i) = \vee_i \delta(f_i)$.
- ε is **erosion** iff $\varepsilon(\wedge_i f_i) = \wedge_i \varepsilon(f_i)$.
- α is **opening** iff increasing and antiextensive $(\alpha(f) \leq f)$,

  and idempotent $(\alpha = \alpha^2)$.
- β is **closing** iff increasing and extensive $(\beta(f) \geq f)$,

  and idempotent $(\beta = \beta^2)$.
- (ε, δ) is **adjunction** iff $\delta(g) \leq f \Leftrightarrow g \leq \varepsilon(f)$. **Residuation pair**

  Then: ε is erosion, δ is dilation, $\delta\varepsilon$ is opening (projection), $\varepsilon\delta$ is closing (projection).

[Serra (1988), Heijmans (1994)]

Max-* Algebra
and
Weighted Latticess

Max-* Scalar Arithmetic: Complete Lattice-Ordered Double Monoid (CLODUM)

‘addition’: $x \vee y = \max(x, y)$

‘dual addition’: $x \wedge y = \min(x, y)$

‘multiplication’ $\star$ distributes over ‘addition’: $a \star (x \vee y) = a \star x \vee a \star y$

‘d.multiplication’ $\star'$ distributes over ‘d.addition’: $a \star' (x \wedge y) = a \star' x \wedge a \star' y$

$(\mathcal{K}, \vee, \wedge, \star, \star')$ is called a **clodum** if:
(C1) $(\mathcal{K}, \vee, \wedge)$ is a complete distributive lattice.
(C2) $(\mathcal{K}, \star)$ is a monoid whose operation $\star$ is a dilation.
(C3) $(\mathcal{K}, \star')$ is a monoid whose operation $\star'$ is an erosion.

Three Clodum Examples:

- *Max-plus* : $(\overline{\mathbb{R}}, \vee, \wedge, +, +')$
- *Max-times* : $([0, +\infty], \vee, \wedge, \times, \times')$
- *Max-min* : $([0, 1], \vee, \wedge, \min, \max)$

Scalar Conjugation in a CLODUM

A clodum $\mathcal{K}$ is called *self-conjugate* if it has a lattice negation $a \mapsto a^*$ that maps each element a to its *conjugate* a^* s.t.

$$\left(\bigvee_i a_i\right)^* = \bigwedge_i {a_i}^* \quad , \quad \left(\bigwedge_i b_i\right)^* = \bigvee_i {b_i}^*$$

$$(a \star b)^* = a^* \star' b^*$$

Examples:

max-plus clodum: $a^* = -a$

max-times clodum: $a^* = 1/a$

max-min clodum: $a^* = 1 - a$

General (Max-*) Matrix and Signal Algebra

- scalars from a clodum $(\mathcal{K}, \vee, \wedge, \star, \star')$
- vector/matrix and signal 'addition' = pointwise max

$$
\begin{aligned}
\boldsymbol{x} \vee \boldsymbol{y} &= [x_1 \vee y_1, \ldots, x_n \vee y_n]^T \\
\boldsymbol{A} \vee \boldsymbol{B} &= [a_{ij} \vee b_{ij}] \\
(F \vee G)(t) &= F(t) \vee G(t)
\end{aligned}
$$

- vector/matrix and signal 'multiplication' by scalar

$$
\begin{aligned}
c \star \boldsymbol{x} &= [c \star x_1, \ldots, c \star x_n]^T \\
c \star \boldsymbol{A} &= [c \star a_{ij}] \\
(c \star F)(t) &= c \star F(t)
\end{aligned}
$$

- matrix (max-$\star$) 'multiplication'

$$
\{\boldsymbol{A} \boxast \boldsymbol{B}\}_{ij} = \bigvee_{k=1}^{n} a_{ik} \star b_{kj}
$$

- signal (sup-$\star$) convolution

$$
(F \circledast H)(t) = \bigvee_{k} F(k) \star H(t-k)
$$

Unified Framework for Nonlinear (max-*) Systems
Dynamical Systems on Weighted Lattices

$$
\begin{aligned}
\mathbf{x}(t+1) &= \mathbf{A}(t) \boxast \mathbf{x}(t) \vee \mathbf{B}(t) \boxast \mathbf{u}(t) \\
\mathbf{y}(t) &= \mathbf{C}(t) \boxast \mathbf{x}(t) \vee \mathbf{D}(x) \boxast \mathbf{u}(t)
\end{aligned}
$$

- * = + → max-plus (max-sum) systems
- * = x → max-times (max-product) systems
- * = fuzzy intersection norm → fuzzy/probabilistic dynamical systems
- * = Boolean product → Boolean dynamical systems

Axioms of Weighted Lattices

Sup-Semilattice	Inf-Semilattice	Description
L1. $X \vee Y \in \mathcal{W}$	L1′. $X \wedge Y \in \mathcal{W}$	Closure of $\vee, \wedge$
L2. $X \vee X = X$	L2′. $X \wedge X = X$	Idempotence of $\vee, \wedge$
L3. $X \vee Y = Y \vee X$	L3′. $X \wedge Y = Y \wedge X$	Commutativity of $\vee, \wedge$
L4. $X \vee (Y \vee Z) = (X \vee Y) \vee Z$	L4′. $X \wedge (Y \wedge Z) = (X \wedge Y) \wedge Z$	Associativity of $\vee, \wedge$
L5. $X \vee (X \wedge Y) = X$	L5′. $X \wedge (X \vee Y) = X$	Absorption between $\vee, \wedge$
L6. $X \leq Y \iff Y = X \vee Y$	L6′. $X \leq' Y \iff Y = X \wedge Y$	Consistency of $\vee, \wedge$ with partial order $\leq$
L7. $O \vee X = X$	L7′. $I \wedge X = X$	Identities of $\vee, \wedge$
L8. $I \vee X = I$	L8′. $O \wedge X = O$	Absorbing Nulls of $\vee, \wedge$
L9. $X \vee (Y \wedge Z) = (X \vee Y) \wedge (X \vee Z)$	L9′. $X \wedge (Y \vee Z) = (X \wedge Y) \vee (X \wedge Z)$	Distributivity of $\vee, \wedge$
WL10. $a \star X \in \mathcal{W}$	WL10′. $a \star' X \in \mathcal{W}$	Closure of $\star, \star'$
WL11. $a \star (b \star X) = (a \star b) \star X$	WL11′. $a \star' (b \star' X) = (a \star' b) \star' X$	Associativity of $\star, \star'$
WL12. $a \star (X \vee Y) = a \star X \vee a \star Y$	WL12′. $a \star' (X \wedge Y) = a \star' X \wedge a \star' Y$	Distributive scalar-vector mult over vector sup/inf
WL13. $(a \vee b) \star X = a \star X \vee b \star X$	WL13′. $(a \wedge b) \star' X = a \star' X \wedge b \star' X$	Distributive scalar-vector mult over scalar sup/inf
WL14. $e \star X = X$	WL14′. $e' \star' X = X$	Scalar Identities
WL15. $\perp \star X = O$	WL15′. $\top \star' X = I$	Scalar Nulls
WL16. $a \star O = O$	WL16′. $a \star' I = I$	Vector Nulls

[Maragos, Math. Control, Signals and Systems, 2017]

Dynamical Systems on Weighted Lattices

Refs:

- P. Maragos, "Dynamical Systems on Weighted Lattices: General Theory", *Math. Control, Signals and Systems*, 2017.

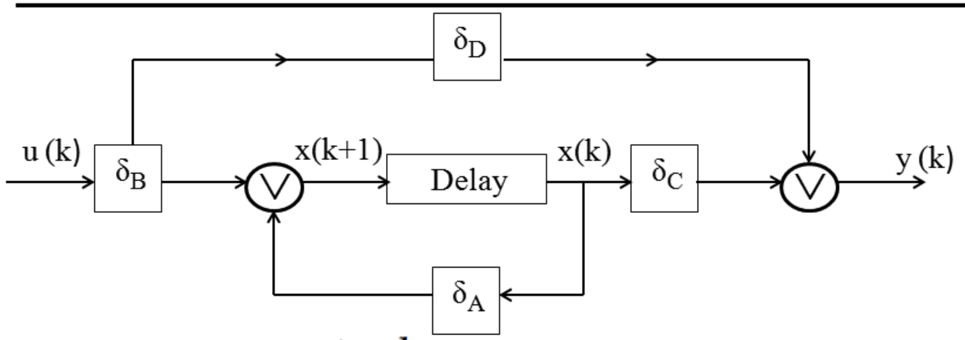
Dynamical Max-* Systems on Weighted Lattices

$$\begin{aligned} \mathbf{x}(t+1) &= \mathbf{A}(t) \boxed{\star} \mathbf{x}(t) \vee \mathbf{B}(t) \boxed{\star} \mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}(t) \boxed{\star} \mathbf{x}(t) \vee \mathbf{D}(x) \boxed{\star} \mathbf{u}(t) \end{aligned}$$

- * = + → max-plus (max-sum) systems
- * = x → max-times (max-product) systems
- * = fuzzy intersection norm → fuzzy dynamical systems
- * = Boolean product → Boolean dynamical systems

- **Problems to solve**:
  - State and output responses
  - Transform domain
  - Solve max-* equations: $\mathbf{A} \boxed{\star} \mathbf{x} = \mathbf{b}$
  - Stability
  - Controllability
  - Feedback
  - System Identification
  - State Estimation from Observations

Discrete-time Nonlinear Control Systems



- **max-sum control:**

$$\mathbf{x}(k+1) = \mathbf{A} \boxplus \mathbf{x}(k) \vee \mathbf{B} \boxplus \mathbf{u}(k)$$
$$\mathbf{y}(k) = \mathbf{C} \boxplus \mathbf{x}(k) \vee \mathbf{D} \boxplus \mathbf{u}(k)$$

max-plus

discrete event dynamic systems (DEDS),

minimax algebra, morphological vision systems

- **max-Tnorm control:**

$$\mathbf{x}(k+1) = \mathbf{A} \,\square_T\, \mathbf{x}(k) \vee \mathbf{B} \,\square_T\, \mathbf{u}(k)$$
$$\mathbf{y}(k) = \mathbf{C} \,\square_T\, \mathbf{x}(k) \vee \mathbf{D} \,\square_T\, \mathbf{u}(k)$$

fuzzy systems, neural nets, probabilistic automata

- **max-product control:**

$$\mathbf{x}(k+1) = \mathbf{A} \boxtimes \mathbf{x}(k) \vee \mathbf{B} \boxtimes \mathbf{u}(k)$$
$$\mathbf{y}(k) = \mathbf{C} \boxtimes \mathbf{x}(k) \vee \mathbf{D} \boxtimes \mathbf{u}(k)$$

max-product

systems with nonnegative input/output/states

Max-Plus Operators

- **Vector Operators** $(\mathbf{A} = \text{matrix } [a_{ij}])$

dilation

$$\delta(\mathbf{x}) = \mathbf{A} \boxplus \mathbf{x} = [\bigvee_{j=1}^{n} x_j + a_{ij}]$$

adjoint erosion $[(\varepsilon, \delta) = \text{adjunction}]$

$$\varepsilon(\mathbf{x}) = -\mathbf{A}^T \boxplus' \mathbf{x} = [\bigwedge_{j=1}^{n} x_j - a_{ji}]$$

dual erosion: $\varepsilon'(\mathbf{x}) = \mathbf{A} \boxplus' \mathbf{x} = [\bigwedge_{j=1}^{n} x_j +' a_{ij}]$

Adjunctions of Vector operators
max-plus and min-diff
matrix-vector multiplication

- **Signal Operators** $(h = \text{impulse response})$

dilation = sup-sum convolution

$$\Delta(f)(k) = (f \oplus h)(k) = \bigvee_{i \in \mathbb{Z}} f(i) + h(k-i)$$

adjoint erosion $[(E, \Delta) = \text{adjunction}]$

$$E(f)(k) = (f \ominus h)(k) = \bigwedge_{i \in \mathbb{Z}} f(i) - h(i-k)$$

dual erosion: $E'(f)(k) = \bigwedge_{i \in \mathbb{Z}} f(i) +' h(k-i)$

Adjunctions of Signal operators
max-plus convolution
and
min-diff correlation

Solving State-Space Equations

- state space eqns:

$$\mathbf{x}(k+1) = \mathbf{A} \boxtimes \mathbf{x}(k) \vee \mathbf{B} \boxtimes \mathbf{u}(k)$$
$$\mathbf{y}(k) = \mathbf{C} \boxtimes \mathbf{x}(k) \vee \mathbf{D} \boxtimes \mathbf{u}(k)$$

- state response:

$$\mathbf{x}(k) = \mathbf{A}^{(k)} \boxtimes \mathbf{x}(0) \vee \bigvee_{i=0}^{k-1} A^{(k-1-i)} \boxtimes B \boxtimes \mathbf{u}(i)$$
$$= \delta_A^k[\mathbf{x}(0)] \vee \bigvee_{i=0}^{k-1} \delta_A^{k-1-i}\delta_B[\mathbf{u}(i)]$$

- output response:

$$\mathbf{y}(k) = \underbrace{\delta_C\delta_A^k[\mathbf{x}(0)]}_{\text{zero-input resp.}} \vee \underbrace{\bigvee_{i=0}^{k-1} \delta_C\delta_A^{k-1-i}\delta_B[\mathbf{u}(i)] \vee \delta_D[\mathbf{u}(k)]}_{\mathbf{y}_{zs}(k) \triangleq \text{'zero'-state resp.}}$$

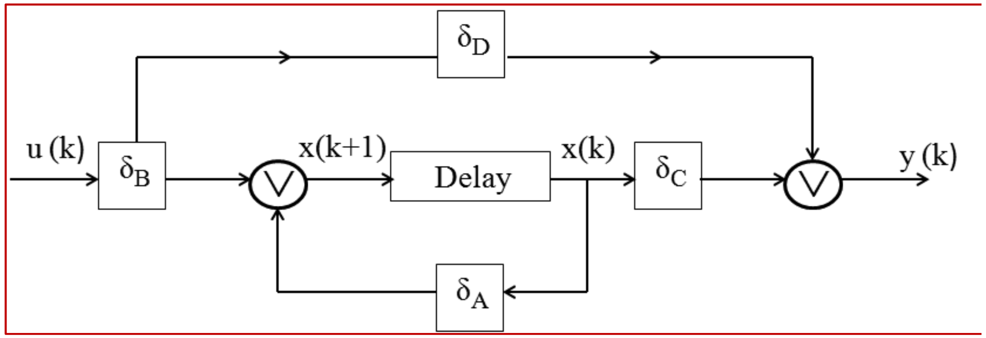
- single-input single-output systems:

$u(k) \mapsto y_{zs}(k)$ is a $\mathbb{T}$-dilation Δ:

$$y_{zs} = \Delta(u) = \bigvee_i u(i) \star h(k-i)$$

impulse response:

$$h(k) = \begin{cases} V_{\text{inf}}, & k < 0 \\ D, & k = 0 \\ C \boxtimes A^{(k-1)} \boxtimes B, & k > 0 \end{cases}$$



Max-plus Eigenvalues and Matrix Graph Cycles

Consider $n \times n$ matrix $\boldsymbol{A} = [a_{ij}]$ with elements from $(\overline{\mathbb{R}}, \vee, \wedge, +)$ and represent $\boldsymbol{A}$ by a directed weighted graph.

The **max-+ eigenproblem** for the matrix $\boldsymbol{A}$ consists of finding its *eigenvalues* λ and *eigenvectors* $\boldsymbol{v} \neq -\infty$ s.t.

$$\boldsymbol{A} \boxplus \boldsymbol{v} = \lambda + \boldsymbol{v}$$

For any graph cycle $\boldsymbol{\sigma}$, its cycle mean is weight$(\boldsymbol{\sigma})$/length$(\boldsymbol{\sigma})$. The *maximum cycle mean* is the **principal eigenvalue** of $\boldsymbol{A}$:

$$\lambda(\boldsymbol{A}) = \bigvee_{\text{all cycles } \boldsymbol{\sigma} \text{ of } \boldsymbol{A}} w(\boldsymbol{\sigma})/\ell(\boldsymbol{\sigma})$$

It is the largest eigenvalue of $\boldsymbol{A}$ and the only eigenvalue whose corresponding eigenvectors may be finite.

Generalizes to max-$\star$ algebra over any radicable clodum.

Max-plus Eigenvalues, Heaviest Paths on Graph, Stability

The **metric matrix** generated by $\boldsymbol{A}$ is the series

$$\boldsymbol{\Gamma}(\boldsymbol{A}) = \bigvee_{k=1}^{\infty} \boldsymbol{A}^{(k)}$$

If it converges, its elements equal the weights of heaviest paths of any length, and its columns can provide eigenvectors.

Theorem (heaviest paths):
(a) The infinite series converges in finite time to a matrix $\boldsymbol{\Gamma}(\boldsymbol{A}) = [\gamma_{ij}]$ and all $\gamma_{ij} < +\infty$ if and only if $\lambda(\boldsymbol{A}) \leq 0$:

$$\boldsymbol{A}^{(t)} \leq \boldsymbol{\Gamma}(\boldsymbol{A}) = \boldsymbol{A} \vee \boldsymbol{A}^{(2)} \vee \cdots \vee \boldsymbol{A}^{(n)} \quad \forall t \geq 1$$

(b) If the graph of $\boldsymbol{A}$ is strongly connected, then all $\gamma_{ij} > -\infty$.

Theorem (stability):
A max-$\star$ is BIBO absolutely stable iff $\lambda(\boldsymbol{A}) = 0$.

Both theorems generalize to max-$\star$ algebra.

Special Cases, Applications

Nonlinear Recursive Filtering

Max-plus Difference Eqn

$$y(t) = \left(\bigvee_{i=1}^{n} a_i + y(t-i) \right) \vee \left(\bigvee_{j=0}^{m} b_j + u(t-j) \right)$$

Dynamical System in State-Space

$$\begin{aligned} \boldsymbol{x}(t) &= \boldsymbol{A}(t) \boxplus \boldsymbol{x}(t-1) \ \vee \ \boldsymbol{B}(t) \boxplus \boldsymbol{u}(t) \\ \boldsymbol{y}(t) &= \boldsymbol{C}(t) \boxplus \boldsymbol{x}(t) \ \vee \ \boldsymbol{D}(t) \boxplus \boldsymbol{u}(t) \end{aligned}$$

Max-plus principal Eigenvalue =

$$\bigvee_{k=1}^{n} a_k / k$$

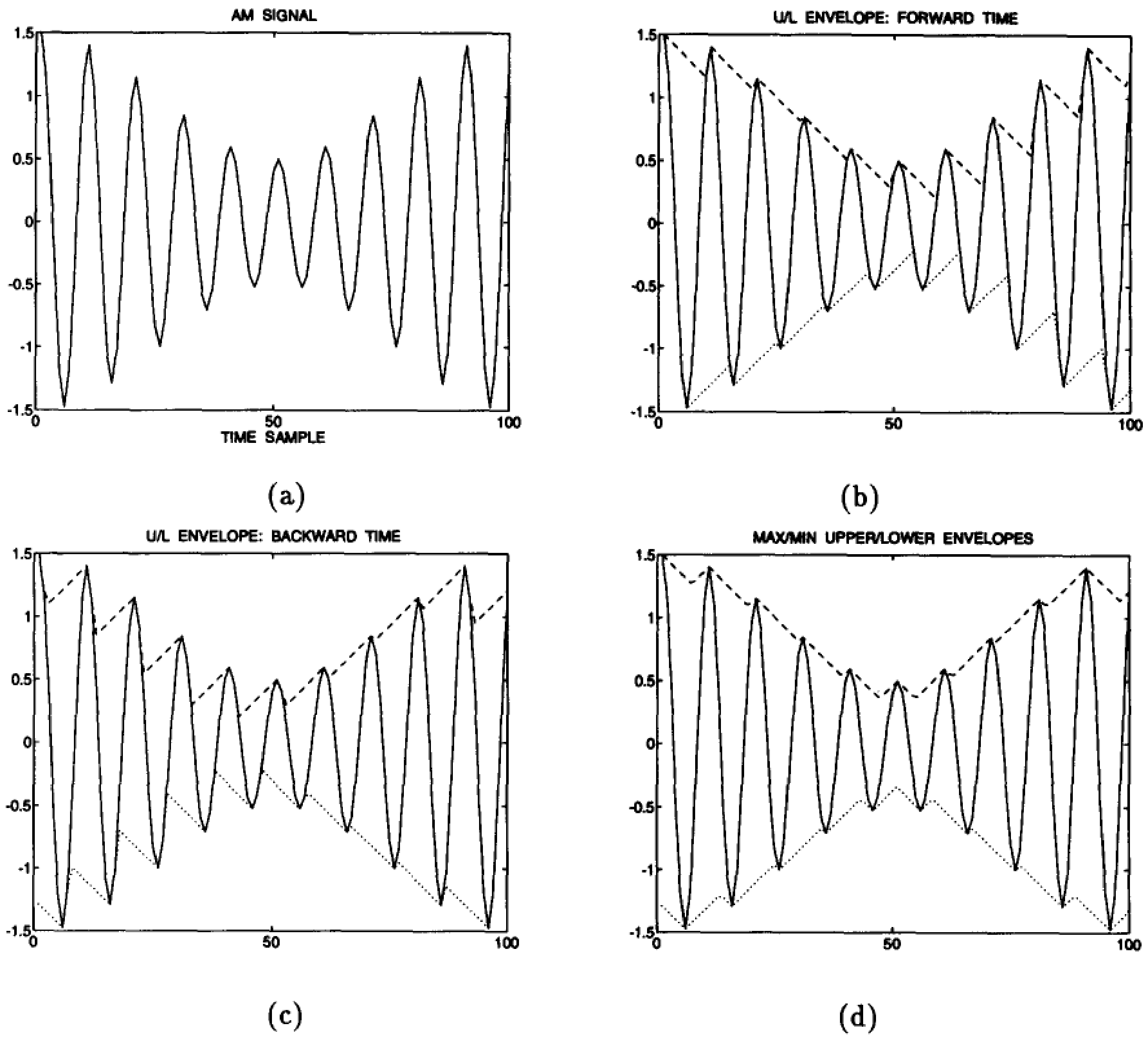
System's Matrices

$$\boldsymbol{A} = \begin{bmatrix} -\infty & 0 & -\infty & \dots & -\infty \\ -\infty & -\infty & 0 & \dots & -\infty \\ \vdots & \vdots & & & \vdots \\ -\infty & -\infty & -\infty & \dots & 0 \\ a_n & a_{n-1} & a_{n-2} & \dots & a_1 \end{bmatrix}, \ B = [b_0]$$

$$\boldsymbol{C} = [a_n, \dots, a_1], \quad D = [b_0]$$

Envelope Detection w. Max-Plus Difference Eqns

$$y[n] = \max(y[n \pm 1] \pm \alpha_0, x[n])$$



Min-Plus Difference Eqns and Distance Computation

$$y(t) = \left(\bigwedge_{i=1}^{n} a_i + y(t-i) \right) \wedge \left(\bigwedge_{j=0}^{m} b_j + u(t-j) \right)$$

$$y_1(t) \;=\; \min[y_1(t-1)+1, u_0(t)]$$

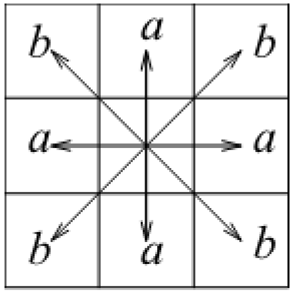
$$y_2(t) = \min[y_2(t+1)+1, y_1(t)]$$

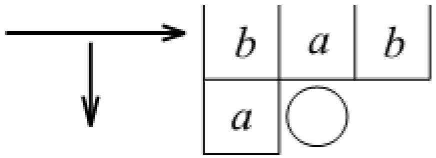
u_0	∞	0	∞	∞	∞	0	∞	∞	∞	∞	∞	0
y_1	∞	0	1	2	3	0	1	2	3	4	5	0
y_2	1	0	1	2	1	0	1	2	3	2	1	0

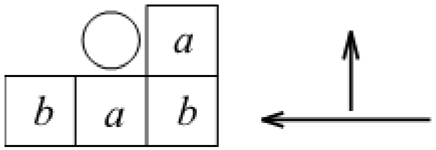
Distance Transforms via Min-plus Difference Eqns

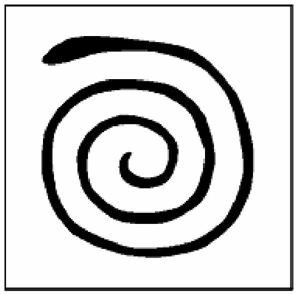
Two – Pass Algorithm

$$
\begin{aligned}
u_1[i,j] &= \min(u_1[i,j-1]+a, u_1[i-1,j]+a, \\
&\quad u_1[i-1,j-1]+b, u_1[i-1,j+1]+b, u_0[i,j]) \\
u_2[i,j] &= \min(u_2[i,j+1]+a, u_2[i+1,j]+a, \\
&\quad u_2[i+1,j+1]+b, u_2[i+1,j-1]+b, u_1[i,j])
\end{aligned}
$$

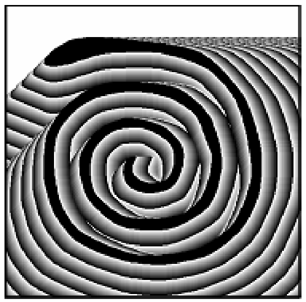








Initial Image



First Pass

Second Pass

Sequential Distance Computation with Obstacles

$$y_i(t) = [\bigwedge_{k=1}^{N+1} w_k + y_i(t-N+k-2)] \wedge u_{i-1}(t)$$

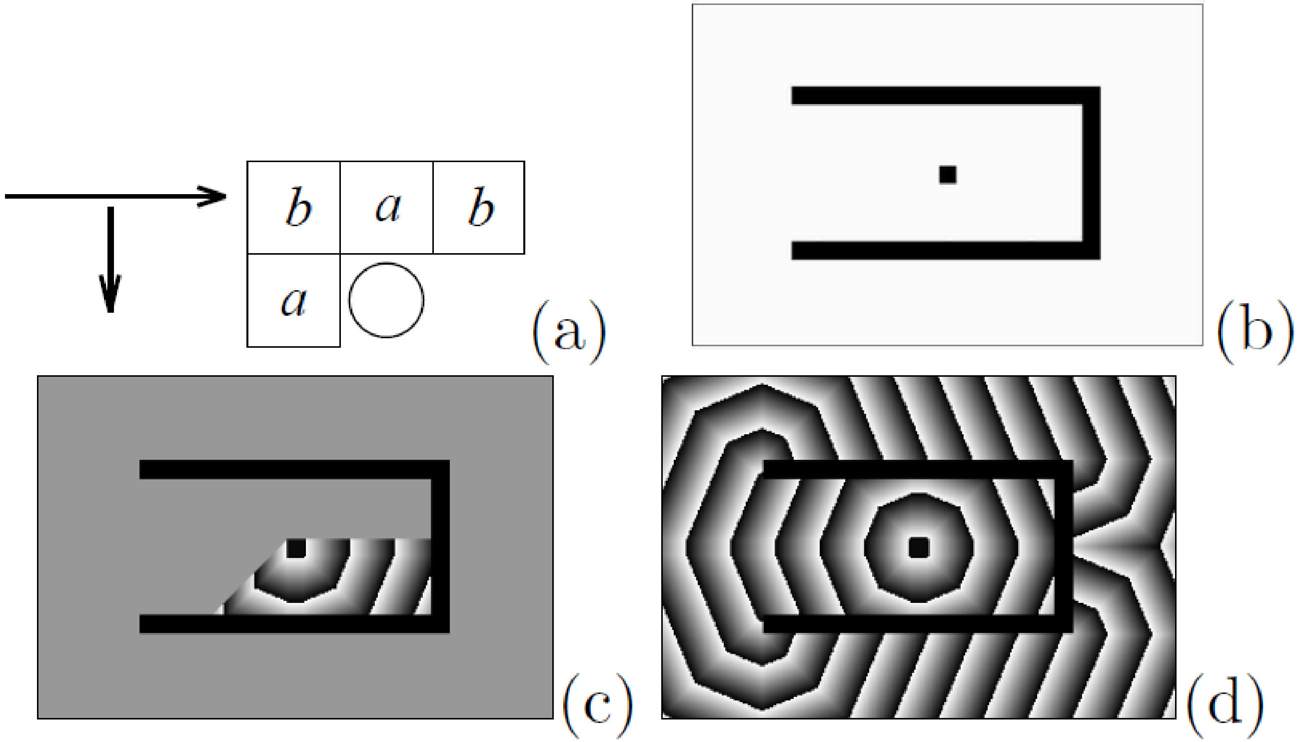
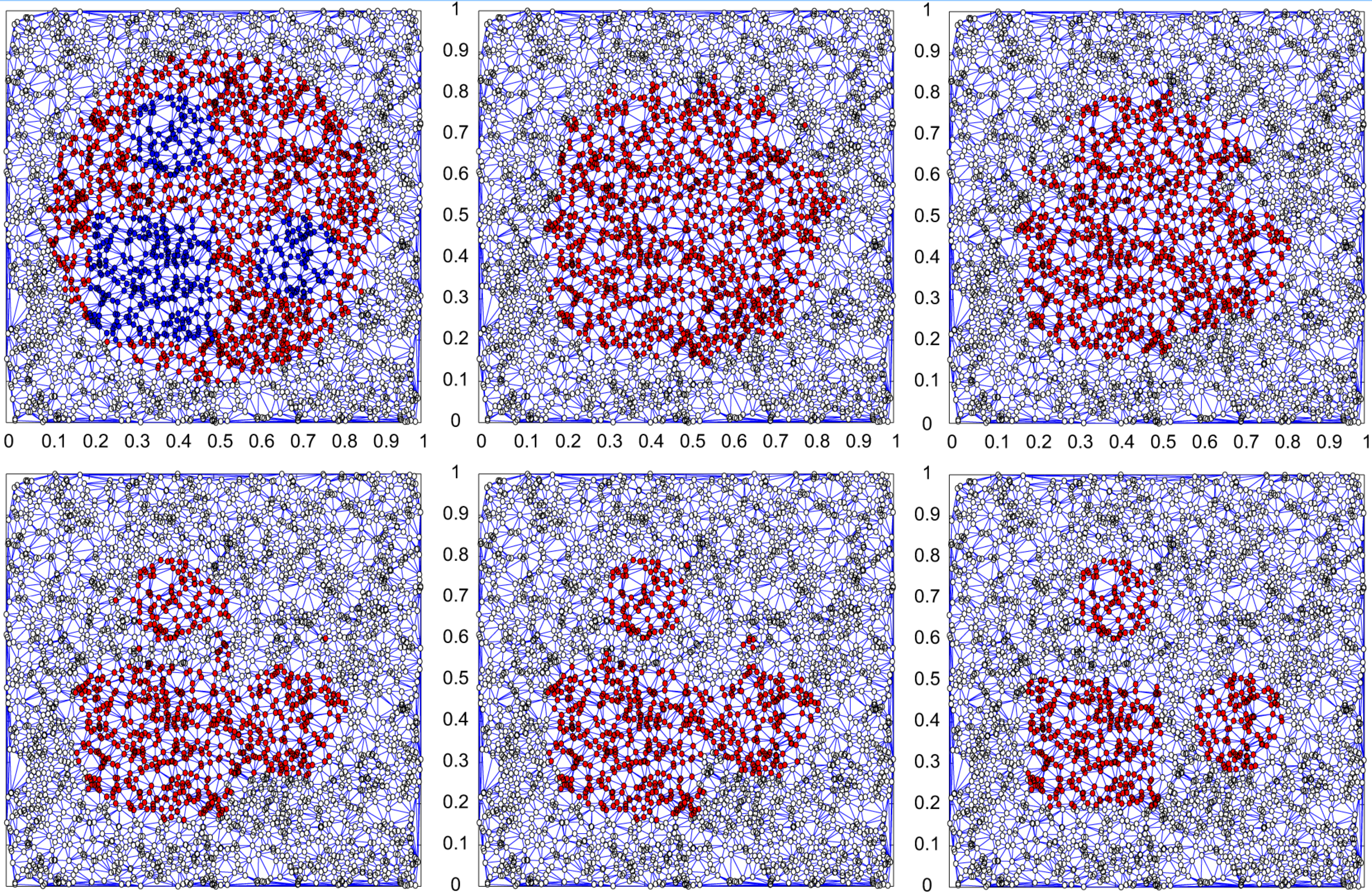


Figure 3: (a) Coefficient submask for forward pass of sequential distance transform. (b) Source set S and the obstacle wall set W. (c) First (forward) pass of constrained distance transform with steps $(a, b) = (24, 34)/25$. (d) Fourth (backward) pass and final result, shown with gray values modulo a constant.

Cluster Detection on Graphs w. Active Contours



Drakopoulos & Maragos (IEEE STSP 2012)

Linear and Nonlinear Spaces

Linear spaces (Vector Spaces):

Signal Superposition (+): $f(t) + g(t)$

Scaling (x): $c \cdot f(t)$

$$\sum_i c_i f_i(t)]$$

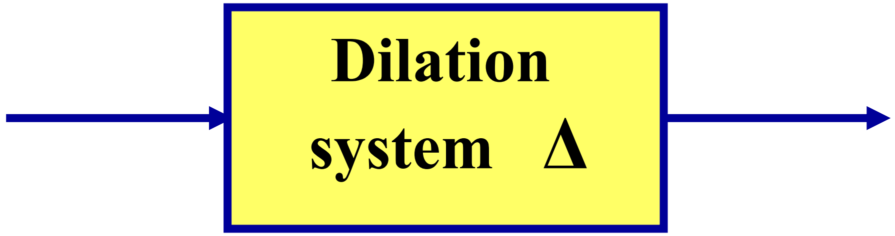
$$\sum_i c_i \Gamma[f_i(t)]$$

Nonlinear spaces (Weighted Lattices):

Signal Superposition : **max:** $f(t) \vee g(t)$ **min:** $f(t) \wedge g(t)$

Scaling (+): $c + f(t)$

$$\bigvee_i c_i + f_i(t)$$



$$\bigvee_i c_i + \Delta[f_i(t)]$$

Max-Product Systems

On linear spaces, a linear system Γ obeys *linear* superposition:

$$\Gamma(\sum_i a_i F_i) = \sum_i a_i \Gamma(F_i)$$

On a CWL, a *dilation and vertical-scaling invariant (DVI)* system δ obeys a max-product superposition:

$$\delta(\bigvee_i c_i F_i) = \bigvee_i c_i \delta(F_i),$$

A vector operator δ is DVI iff it is a matrix-vector max-product:

$$\delta(\boldsymbol{x}) = \boldsymbol{M} \boxtimes \boldsymbol{x}$$

$$\boldsymbol{P} = \boldsymbol{Q} \boxtimes \boldsymbol{R}, \quad p_{ij} = \bigvee_k q_{ik} \times r_{kj}$$

A signal operator Δ is DVI and time-invariant iff it is a max-product convolution:

$$\Delta(f)(t) = (f \otimes h)(t) = \bigvee_k h(k) f(t-k)$$

Max-Product Dynamical Systems

State-Space equations:

$$
\begin{aligned}
\boldsymbol{x}(t) &= \boldsymbol{A} \boxtimes \boldsymbol{x}(t-1) \ \vee \ \boldsymbol{B} \boxtimes \boldsymbol{u}(t) \\
\boldsymbol{y}(t) &= \boldsymbol{C} \boxtimes \boldsymbol{x}(t) \ \vee \ \boldsymbol{D} \boxtimes \boldsymbol{u}(t)
\end{aligned}
$$

State response:

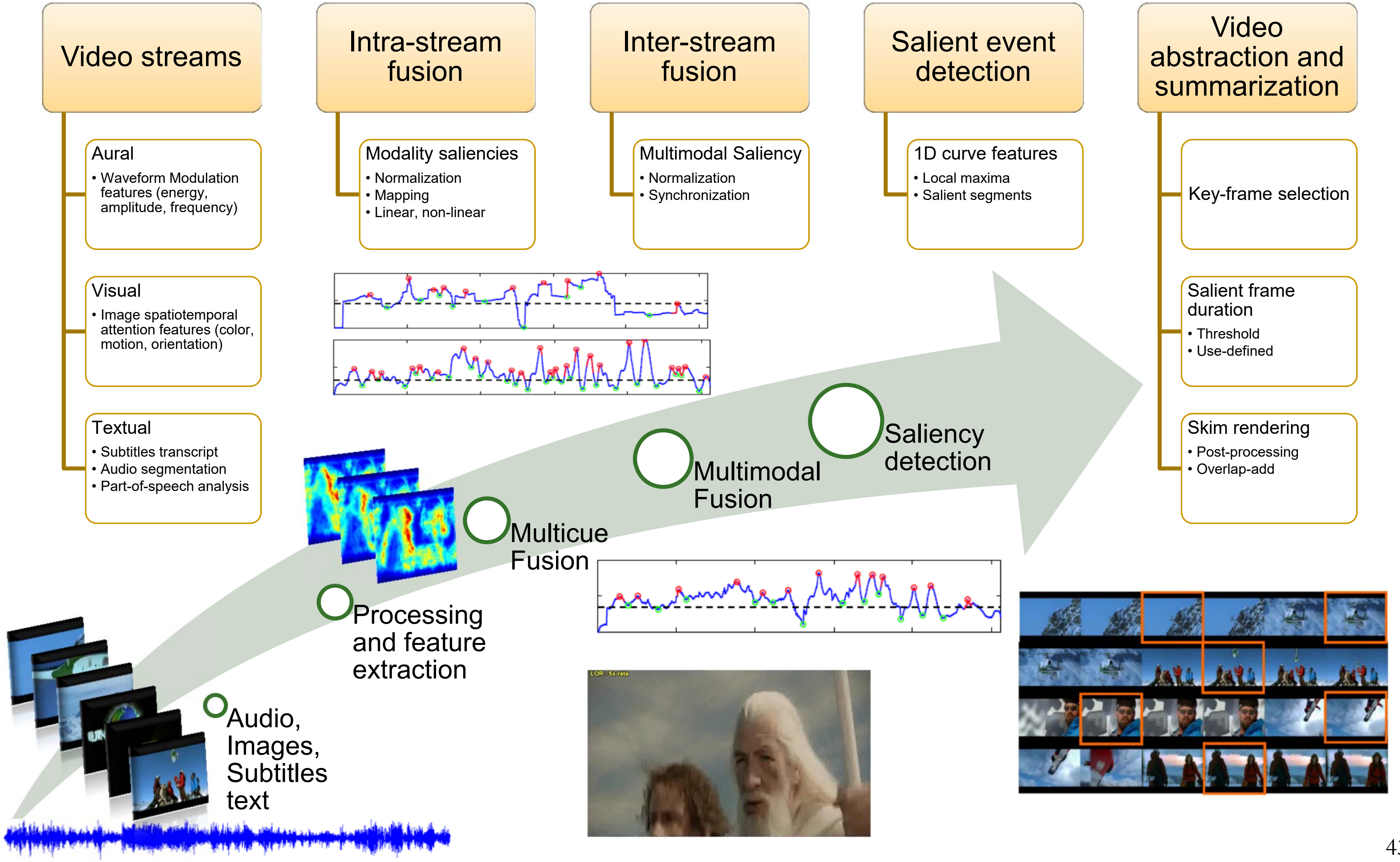
$$
\boldsymbol{x}(t) \quad = \quad \boldsymbol{A}^{(t)} \boxtimes \boldsymbol{x}(0) \vee \left(\bigvee_{i=1}^{t} \boldsymbol{A}^{(t-i)} \boxtimes \boldsymbol{B} \boxtimes \boldsymbol{u}(i) \right)
$$

Zero-state Output response (Max-Product Convolution):

$$
y(t) = (h \otimes u)(t) = \bigvee_{k} u(k)h(t-k)
$$

Audio-Visual-Text Saliency and Video Summarization

[Evangelopoulos, Zlatintsi, Potamianos, Maragos, Rapantzikos, Skoumas, Avrithis, IEEE Tr.-MM 2013.]



Saliency State Evolution with Max-Product (I)

State equations: state i=1 (Audio), i=2 (Visual), i=3 (A-V), i=4 (none)

$$x_i(t) = \left(\bigvee_{j=1}^{4} a_{ij} x_j(t-1) \right) \star p_i(t) \vee \left(\bigvee_{j=1}^{4} b_{ij} u_j(t) \right)$$

$x_i(t) = \Pr(\text{state } i \text{ being salient at time } t)$, $\quad u_i(t) = \text{control inputs}$

$p_i(t) = \text{likelihood of observed low-level feature vector } \boldsymbol{o}_t \text{ at state } i$

$a_{ij} = \text{state transitions probabilities}$, $\quad b_{ij} = \text{input weights}$

Special Case I: $*$ = product & $u(t) = 0$ → Viterbi algorithm in HMMs:

State: $x_i(t) =$ max Pr[observations until t and state(t)=i]

Output: $y(t) = \bigvee_i x_i(t)$ = Viterbi score at time t

Saliency State Evolution with Max-Product (II)

State equations: state i=1 (Audio), i=2 (Visual), i=3 (A-V), i=4 (none)

$$x_i(t) = \left(\bigvee_{j=1}^{4} a_{ij} x_j(t-1) \right) \star p_i(t) \vee \left(\bigvee_{j=1}^{4} b_{ij} u_j(t) \right)$$

$x_i(t) = \Pr(\text{state } i \text{ being salient at time } t)$, $\quad u_i(t) = $ control inputs

$p_i(t) = $ likelihood of observed low-level feature vector $\boldsymbol{o}_t$ at state i

$a_{ij} = $ state transitions probabilities, $\quad b_{ij} = $ input weights

Special Case II: $*$ = min, max & $u(t) = 0$ → HMM variants

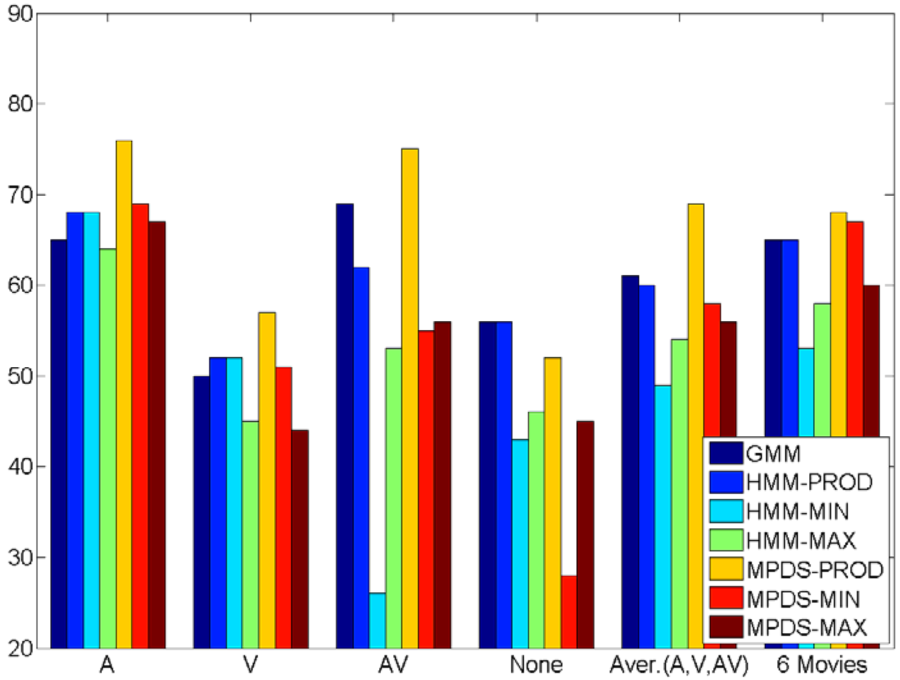
Special Case III: $*$ = prod, min, max & **Nonzero Inputs** $\boldsymbol{u}(t)$

High-level information (e.g. semantics) → $\boldsymbol{u}(t)$

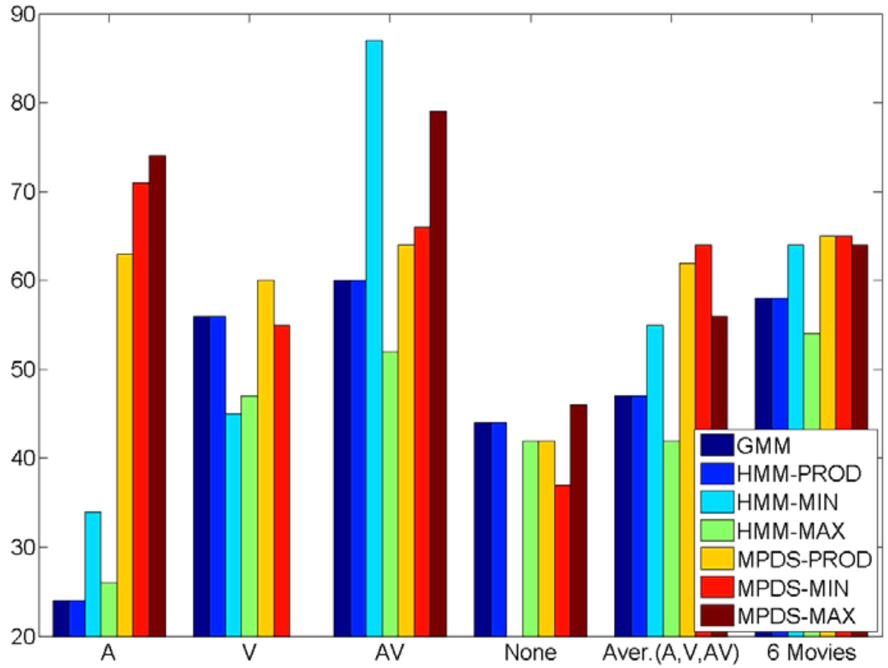
Experiments: state 1: Voice Activity Detector, state 2: Face Detector.

Evaluation Results on Movie Videos (F-Scores)

GMM Likelihoods



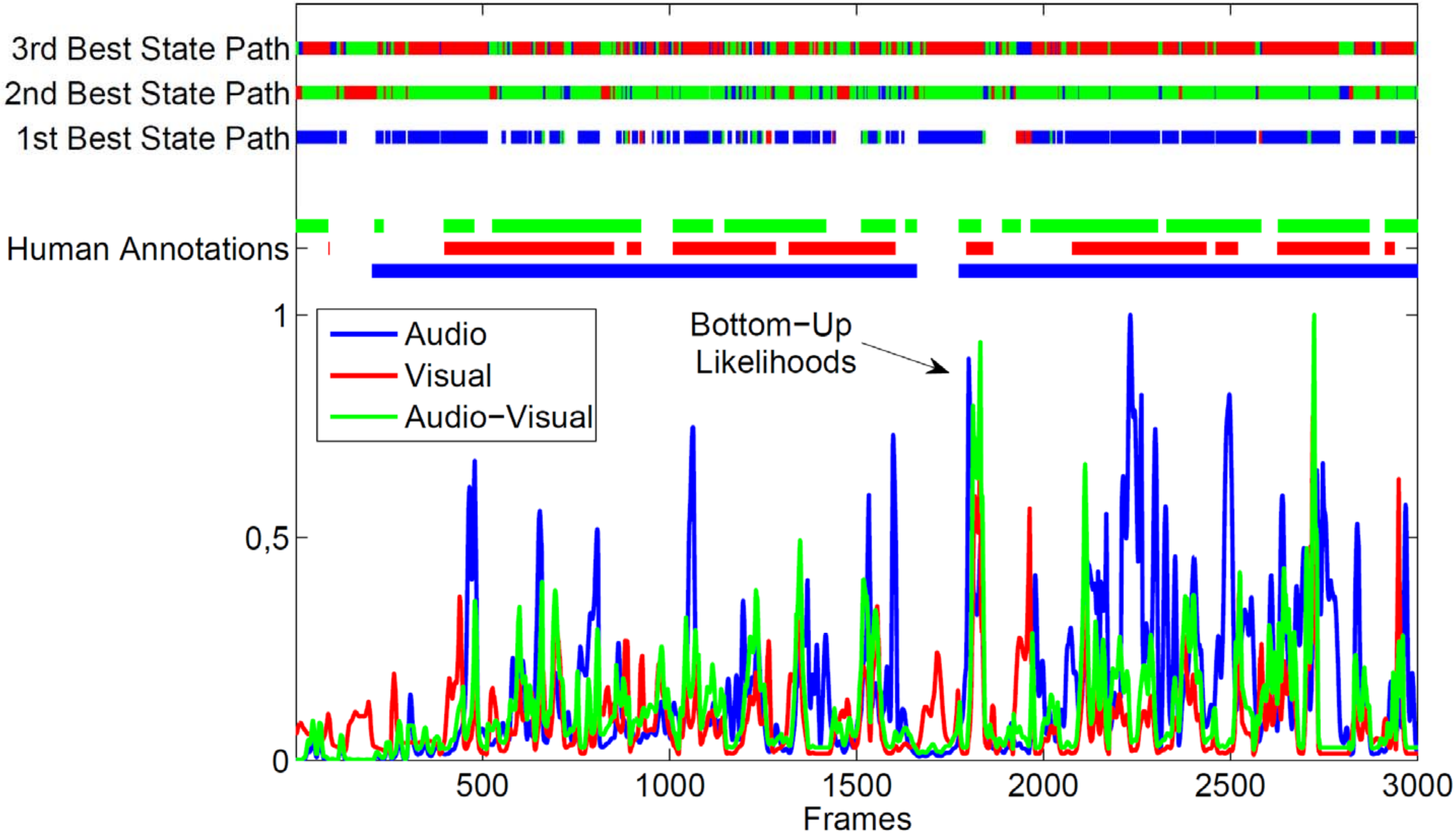
Bottom-Up Likelihoods



$$F_{score}^{-1} = P_{recision}^{-1} + R_{ecall}^{-1}$$

MPDS: Max-Product Dynamical System

Evolution of Bottom-Up Likelihoods



MPDS with product operation

[Maragos & Koutras, ICASSP 2015]

Max-plus equation

$$A \boxplus x = b$$

- Occurs in most applications of max-plus algebra.

- To solve it we need:
  - The min operator: $\wedge$
  - The adjoint (conjugate transpose) of a matrix: $A^* = (-A)^T$
  - The dual of matrix max-plus `multiplication:

$$[A \boxplus' B]_{ij} = \bigwedge_{k=1}^{n} a_{ik} + b_{kj}$$

- Existence of solution:
  - Principal solution $\bar{x} = A^* \boxplus' b$
  - Solution exists if and only if $\bar{x}$ is a solution, and then it is the greatest solution. Otherwise, it is always the greatest sub-solution.

Solve Max-* Equations via L1,Linf Minimization

$$\text{Solve} \quad \boldsymbol{A} \boxast \boldsymbol{x} = \boldsymbol{b}$$

Approximate solution via optimization:

$$\text{Minimize} \quad \|\boldsymbol{A} \boxast \boldsymbol{x} - \boldsymbol{b}\|_{1,\infty}$$

$$\text{s.t.} \quad \boldsymbol{A} \boxast \boldsymbol{x} \leq \boldsymbol{b}$$

Greatest solution:

$$\hat{\boldsymbol{x}} = \boldsymbol{A}^* \boxast' \boldsymbol{b}$$

$\boldsymbol{A}^*$ is conjugate transpose of $\boldsymbol{A}$

SOLVING EQUATIONS $\delta_A(\mathbf{x}) = \mathbf{A} \boxed{\star} \mathbf{x} = \mathbf{b}$

max/min-plus: $\delta_{\mathbf{A}}(\mathbf{x}) = \mathbf{A} \oplus \mathbf{x}$, $\varepsilon_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}^{*} \oplus' \mathbf{x}$

- find unique erosion ε_A s.t. $(\varepsilon_A, \delta_A)$ is adjunction
- erosion $\varepsilon_A(\mathbf{b})$ provides largest $\mathbf{x}$ s.t. $\delta_A(\mathbf{x}) \leq \mathbf{b}$:

$$\varepsilon_A(\mathbf{b}) = \bigvee \{\mathbf{x} : \delta_A(\mathbf{x}) \leq \mathbf{b}\}$$

- opening guarantees $\delta_A(\varepsilon_A(\mathbf{b})) \leq \mathbf{b}$:
- the solution $\varepsilon_A(\mathbf{b})$ is optimal w.r.t. minimizing the ℓ_∞ norm and being $\leq b$:

$$\varepsilon_A(\mathbf{b}) = \text{argmin}_{\mathbf{x}}[\bigvee_{i=1}^{n}(b_i - \{\mathbf{A} \boxed{\star} \mathbf{x}\}_i)]$$

$$\text{s.t.} \quad \mathbf{A} \boxed{\star} \mathbf{x} \leq \mathbf{b}$$

Infeasible Equations - Comparison

Linear

- Approximation

$$x_{ls} = A^+ b$$

- Maps b to range space

$$b \to AA^+ b$$

- Least Squares

$$\min_y ||y - b||_2$$
$$\text{s.t.} \quad y = Ax$$

Max-plus

- Approximation

$$\bar{x} = A^* \boxplus' b$$

- Maps b to range space

$$b \to A \boxplus A^* \boxplus' b$$

- Minimum $L_{1,\infty}$ norm

$$\min_y ||y - b||_{1,\infty}$$
$$\text{s.t.} \quad y = A \boxplus x, y \leq b$$

- controllability

$$\mathbf{x}(n) = \mathbf{A}^{(n)}\mathbf{x}(0) \vee \underbrace{[\mathbf{A}^{(n-1)} \boxast \mathbf{B}, \cdots, \mathbf{B}]}_{\mathcal{C}} \boxast \begin{bmatrix} u(0) \\ \vdots \\ u(n-1) \end{bmatrix}$$

- observability

$$\begin{bmatrix} y(0) \\ \vdots \\ y(n-1) \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{C} \\ \vdots \\ \mathbf{C} \boxast \mathbf{A}^{(n-1)} \end{bmatrix}}_{\mathcal{O}} \boxast \mathbf{x}(0)$$

$$\vee \; [h(n-1), \cdots, h(0)] \boxast \begin{bmatrix} u(0) \\ \vdots \\ u(n-1) \end{bmatrix}$$

Sparsity in Max-Plus Algebra and Systems

Max-Plus Algebra

- Let $\mathbb{R}_{\max} = \mathbb{R} \cup \{-\infty\}$
- Max operator: $\max\{x, y\} = x \vee y$
- Addition: $x + y$
- Null element: $-\infty$

- Max-plus equation:

$$\boldsymbol{A} \boxplus \boldsymbol{x} = \boldsymbol{b}$$

Or

$$\bigvee_{j=1}^{n} (A_{ij} + x_j) = b_j, \ \text{for } i = 1, \ldots, m$$

Sparsest Solution to Max-Plus Equation

- A sparse vector $\boldsymbol{x} \in \mathbb{R}^n_{\max}$ has many $-\infty$ elements.
- Let $\operatorname{supp}(\boldsymbol{x})$ be the support (the set of finite indices)
- We solve the following problems:

Exact solution

$$\min_{\boldsymbol{x} \in \mathbb{R}^n_{\max}} \quad |\operatorname{supp}(\boldsymbol{x})|$$
$$\text{subject to} \quad \boldsymbol{A} \boxplus \boldsymbol{x} = \boldsymbol{b}$$

Approximate solution

$$\min_{x \in \mathbb{R}^n_{\max}} \quad |\operatorname{supp}(\boldsymbol{x})|$$
$$\text{subject to} \quad \|\boldsymbol{b} - \boldsymbol{A} \boxplus \boldsymbol{x}\|_1 \leq \epsilon$$
$$\boldsymbol{A} \boxplus \boldsymbol{x} \leq \boldsymbol{b}$$

- NP-complete problems. Use greedy algorithms.
- Submodularity tools to provide suboptimality bounds.

Applications

- Resource optimization in Discrete Event Systems

$$\begin{aligned} \min_{x \in \mathbb{R}^n_{\max}} \quad & |\mathrm{supp}(\boldsymbol{x})| \\ \text{subject to} \quad & \|\boldsymbol{b} - \boldsymbol{A} \boxplus \boldsymbol{x}\|_1 \leq \epsilon \\ & \boldsymbol{A} \boxplus \boldsymbol{x} \leq \boldsymbol{b} \end{aligned}$$

$\boldsymbol{x}$: times machines are used
$\boldsymbol{A} \boxplus \boldsymbol{x}$: times of production
$\boldsymbol{b}$: deadlines

Minimize the number of machines used, keep storage time small, meet the deadlines

- Sparsity-aware system identification
  - Max-Plus System:

$$\boldsymbol{y}_i = \boldsymbol{G} \boxplus \boldsymbol{u}_i \qquad (\boldsymbol{y}_i, \boldsymbol{u}_i) \text{ input-output pairs}$$

Identify G from input-output pairs, without knowing the sparsity pattern of G
.

  - Sufficient conditions for exact recovery.

System Identification-Example

- Max-plus system:

$$G_{true} = \begin{bmatrix} 2 & 3 & -\infty \\ 1 & 1 & -\infty \\ -\infty & 2 & 6 \end{bmatrix}$$

- Input-output pairs:

$$(\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3, \boldsymbol{u}_4) = \begin{bmatrix} 0 & 10 & 0 & 2 \\ 10 & 0 & 0 & 0 \\ 5 & 5 & 10 & 2 \end{bmatrix} \qquad (\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}_3, \boldsymbol{y}_4) = \begin{bmatrix} 13 & 12 & 3 & 4 \\ 11 & 11 & 1 & 3 \\ 12 & 11 & 16 & 8 \end{bmatrix}$$

- Our method:

$$\boldsymbol{G}_{sparse} = \boldsymbol{G}_{true}$$

- Previous methods:

$$\bar{G} = \begin{bmatrix} 2 & 3 & -7 \\ 1 & 1 & -9 \\ 1 & 2 & 6 \end{bmatrix} \neq \boldsymbol{G}_{\boldsymbol{true}}$$

[A. Tsiamis and P. Maragos, “Sparsity in Max-Plus Algebra and Systems”, 2018, arXiv:1801.09850]

Dynamical Weighted Lattices: Applications

$$\mathbf{x}(t+1) = \mathbf{A}(t) \boxed{\star}\, \mathbf{x}(t) \vee \mathbf{B}(t) \boxed{\star}\, \mathbf{u}(t)$$
$$\mathbf{y}(t) = \mathbf{C}(t) \boxed{\star}\, \mathbf{x}(t) \vee \mathbf{D}(x) \boxed{\star}\, \mathbf{u}(t)$$

Minimax Algebra **(* = +, x, …, binary operation distributes over max/min)**

$$\mathbf{C} = \mathbf{A} \vee \mathbf{B} \quad , \quad c_{ij} = a_{ij} \vee b_{ij}$$

$$\mathbf{C} = \mathbf{A} \boxed{\star}\, \mathbf{B} \quad , \quad c_{ij} = \bigvee_k a_{ik} \star b_{kj}$$

Applications

- Automated Manufacturing (production lines)
- Scheduling (ex. Transportation)
- Operations Research
- Nonlinear Control: Discrete Event Dynamical Systems
- Weighted Finite State Transducers for Speech Recognition
- Shortest Paths on Graphs, Dynamic Programming
- Graphical Models (ex. Viterbi, Max-product algorithm)

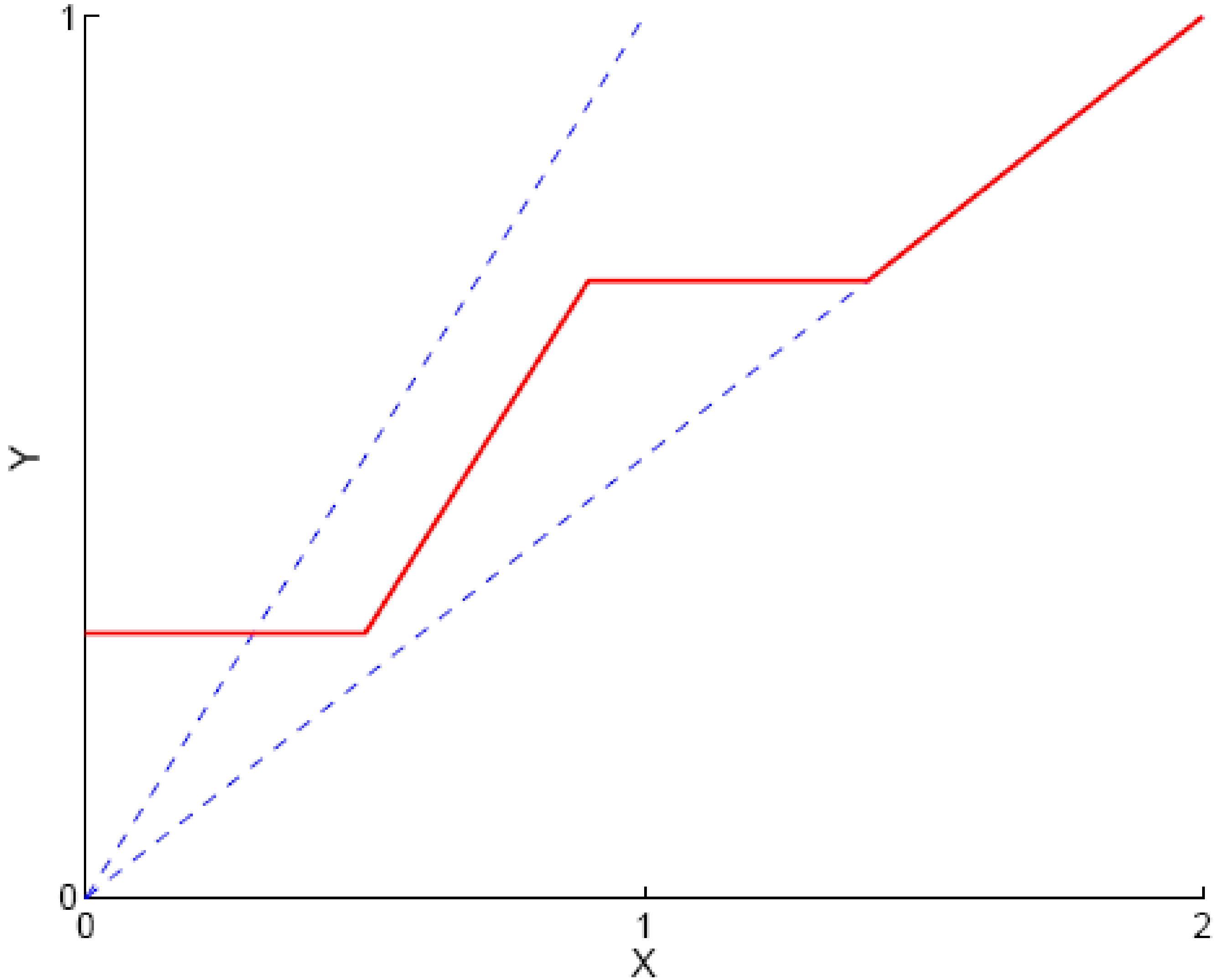
Dynamical Systems on Weighted Lattices: Theory

$$\begin{aligned} \mathbf{x}(t+1) &= \mathbf{A}(t) \boxast \mathbf{x}(t) \ \vee \ \mathbf{B}(t) \boxast \mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}(t) \boxast \mathbf{x}(t) \ \vee \ \mathbf{D}(x) \boxast \mathbf{u}(t) \end{aligned}$$

- * = + → max-plus (max-sum) systems
- * = x → max-times (max-product) systems
- * = fuzzy intersection norm → fuzzy dynamical systems
- * = Boolean product → Boolean dynamical systems

- **Problems solved or to solve**:
  - State and output responses
  - Transform domain
  - Solve max-* equations: $\mathbf{A} \boxast \mathbf{x} = \mathbf{b}$
  - Stability
  - Control, Feedback
  - System Identification
  - State Estimation from Observations
  - Connections with Logic/Language/Probability (Boolean, Fuzzy)

Polygonal Spaces



$$y = \min[\max(x - 0.2, 0.3), \max(x/2, 0.7)]$$

Contributions

- Complete Weighted Lattices: Nonlinear Spaces
- Theory for Data Analysis and Dyn. Systems on CWLs
- Applications to Multimodal SP, Control & Optimization
- Additional results:
  - Stochastic Stability in Max-product & Max-plus systems with Markovian Jumps: [Kordonis, Maragos & Papavassilopoulos, *Automatica*, to appear]. Also arXiv:1711.03018.
  - Max-plus Perceptrons: Geometry & Training algorithms: [Charisopoulos & Maragos, *ISMM* 2017]

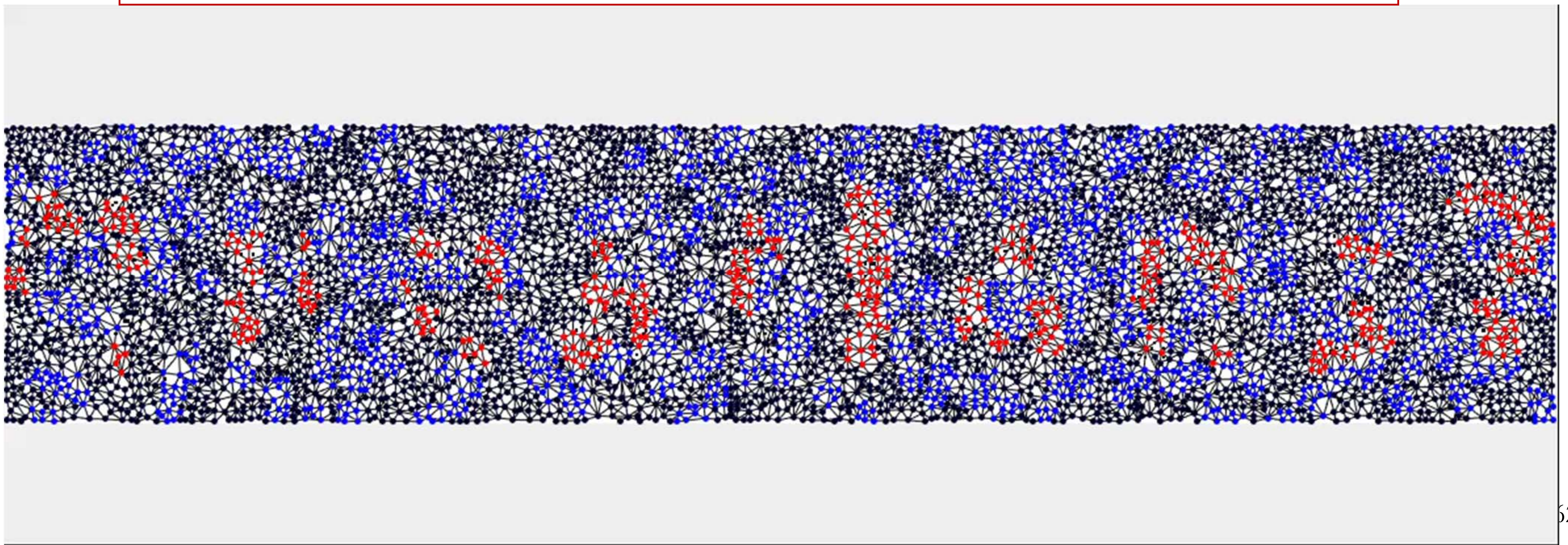
- On-going work:
  - Parameter estimation from observation data
  - Learning problems
  - Control/Feedback in perception/cognition
  - Geometry problems

Collaborators and References

Kordonis, Yannis
Koutras, Petros
Tsiamis, Anastasios

For more information, demos, and current results:

http://cvsp.cs.ntua.gr and http://robotics.ntua.gr



APPENDIX

Transform Domain for Systems on CWL

$$\mathbf{x}(t+1) = \mathbf{A}(t) \boxed{\star} \mathbf{x}(t) \vee \mathbf{B}(t) \boxed{\star} \mathbf{u}(t)$$

$$\mathbf{y}(t) = \mathbf{C}(t) \boxed{\star} \mathbf{x}(t) \vee \mathbf{D}(x) \boxed{\star} \mathbf{u}(t)$$

- * = + → max-plus systems:
  Slope Transforms (Legendre-Fenchel Conjugate)
  max-plus convolution ←→ Sum of slope transforms

[Maragos, IEEE Tr. SigPro, 1995,
Heijmans & Maragos, SigPro 1997]

	GMM Likelihoods							Bottom-Up (BU) Likelihoods						
	GMM	HMM Variant			MPDS			BU	HMM Variant			MPDS		
State		Prod.	Min	Max	Prod.	Min	Max		Prod.	Min	Max	Prod.	Min	Max
A	65	68	68	64	76	69	67	24	24	34	26	63	71	74
V	50	52	52	45	57	51	44	56	56	45	47	60	55	14
AV	69	62	26	53	75	55	56	60	60	87	52	64	66	79
None	56	56	43	46	52	28	45	44	44	11	42	42	37	46
Aver.	60	59	47	52	**65**	51	53	46	46	44	42	**57**	57	53

Table 1: F-scores $\left(F_{score}^{-1} = P_{recision}^{-1} + R_{ecall}^{-1}\right)$ for the HMM Variant and the Max-Product Dynamic System (MPDS) using either the GMM estimated or the bottom-up likelihoods. For the operation $\star$ we have employed three different versions: product, minimum and maximum.

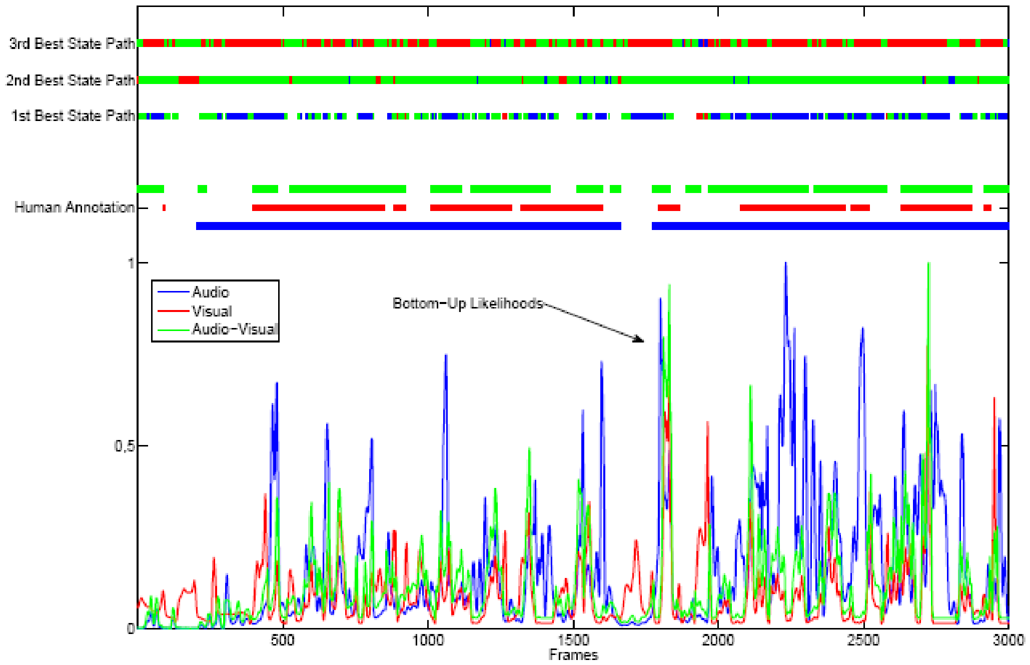
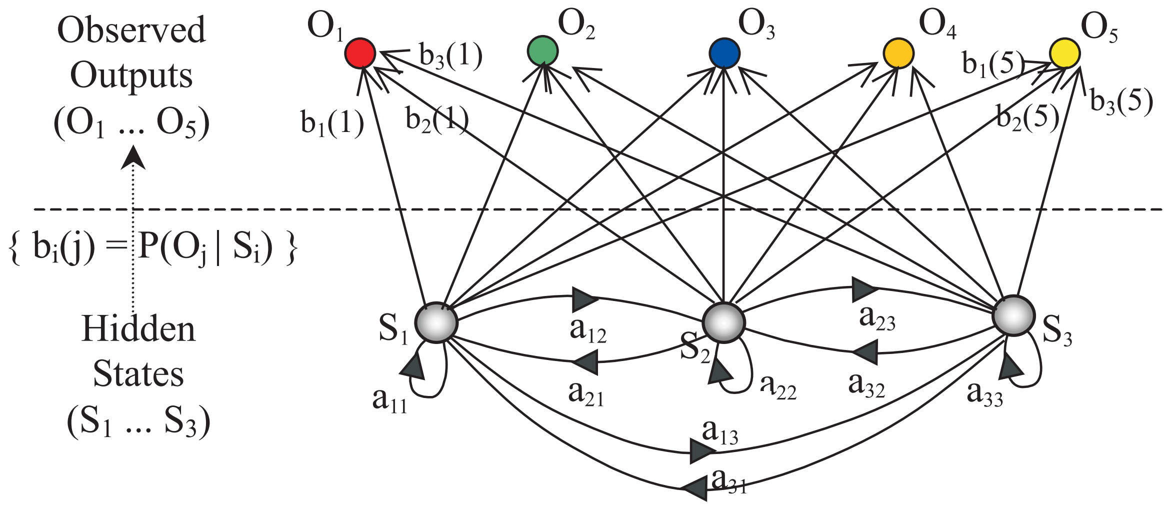


Fig. 1: Evolution of audio (blue), visual (red) and audio-visual(green) bottom-up likelihoods. We also see the human annotations and the 3-Best state paths using the Max-Product Dynamic System (MPDS) with product

[Maragos & Koutras, ICASSP 2015]

HMM (Hidden Markov Models)



- t = 1, 2, 3, …: **Discrete Time**
- $\boldsymbol{O} = (O_1, O_2, \ldots, O_T)$ **: Observation Sequence**
- $\boldsymbol{T}$ **= Length of Observation Sequence**
- $\boldsymbol{N}$ **= Number of States**
- $\boldsymbol{M}$ **= # of Observation Symbols / Mixtures**
- **States** $\{S_1, S_2, \ldots, S_N\}$

HMM: $\lambda = (A, B, \pi)$

- $\boldsymbol{A} = [a_{ij}], \quad a_{ij} = \Pr\left(S_j \text{ at } t{+}1 \mid S_i \text{ at } t\right)$

  State Transition Probability Matrix

- $\boldsymbol{B} = \{b_j(k)\}, \quad b_j(k) = \Pr\left(v_k \text{ at } t \mid S_j \text{ at } t\right)$

  Observations Probability Distributions

- $\boldsymbol{\pi} = \{\pi_i\}, \quad \pi_i = \Pr\left(q_i \text{ at } t{=}1\right)$

  Initial State Probability

Problems to Be Solved in HMM

- **<u>Problem 1</u>: Classification – Scoring (*Forward-Backward Algorithm*)**

  Given an observed sequence $O = (O_1, O_2, \ldots, O_T)$ **and a model** ***λ=(π, A, B),***
  compute likelihood $\Pr(O \mid \lambda)$

- **<u>Problem 2</u>: State Estimation (*Viterbi Algorithm*)**

  Given an observed sequence $O = (O_1, O_2, \ldots, O_T)$ **estimate an optimum state sequence** $Q^* = (q_1, q_2, \ldots, q_T)$ **and compute max score** $\Pr(O, Q^* \mid \lambda)$

- **<u>Problem 3</u>: Training (*EM Algorithm*)**

  Given an observed sequence $O = (O_1, O_2, \ldots, O_T)$ **adjust model parameters** ***λ=(π, A, B)*** **to maximize likelihood** $\Pr(O \mid \lambda)$

Transform Domain for Systems on CWL

$$
\begin{aligned}
\mathbf{x}(t+1) &= \mathbf{A}(t) \boxast \mathbf{x}(t) \;\vee\; \mathbf{B}(t) \boxast \mathbf{u}(t) \\
\mathbf{y}(t) &= \mathbf{C}(t) \boxast \mathbf{x}(t) \;\vee\; \mathbf{D}(x) \boxast \mathbf{u}(t)
\end{aligned}
$$

- * = + → max-plus systems:
  Slope Transforms (Legendre-Fenchel Conjugate)
  max-plus convolution ←→ Sum of slope transforms
- * = x → max-times systems
  Exponential Slope Transform
  max-times convolution ←→ Product of exp slope transforms

Max-* Eigenvalues and Matrix Graph Cycles

Consider a $n \times n$ matrix $\boldsymbol{A} = [a_{ij}]$ with elements from a radicable clodum $\mathcal{K}$, and represent $\boldsymbol{A}$ by a directed weighted graph.
$\perp, \top$ are least and greatest elements, e is mult-identity of $\mathcal{K}$
example: if $\mathcal{K} = (\overline{\mathbb{R}}, \vee, \wedge, +)$, then $\perp = -\infty$, $\top = +\infty$, $e = 0$.

The **max-$\star$ eigenproblem** for the matrix $\boldsymbol{A}$ consists of finding its *eigenvalues* λ and *eigenvectors* $\boldsymbol{v} \neq \perp$ s.t.

$$\boldsymbol{A} \boxtimes \boldsymbol{v} = \lambda \star \boldsymbol{v}$$

For any cycle $\boldsymbol{\sigma}$ on the graph, its cycle mean is $w(\boldsymbol{\sigma})^{\star(1/\ell(\boldsymbol{\sigma}))}$.
The *maximum cycle mean* is the **principal eigenvalue** of $\boldsymbol{A}$:

$$\lambda(\boldsymbol{A}) = \bigvee_{\text{all cycles } \boldsymbol{\sigma} \text{ of } \boldsymbol{A}} w(\boldsymbol{\sigma})^{\star(1/\ell(\boldsymbol{\sigma}))}$$

It is the largest eigenvalue of $\boldsymbol{A}$ and the only eigenvalue whose corresponding eigenvectors may be finite.

Max-* Eigenvalue, Heaviest Paths on Graph, Stability

The **metric matrix** generated by $\boldsymbol{A}$ is the series

$$\boldsymbol{\Gamma}(\boldsymbol{A}) = \bigvee_{k=1}^{\infty} \boldsymbol{A}^{(k)}$$

If it converges, its elements equal the weights of heaviest paths of any length, and its columns can provide eigenvectors.

Theorem (heaviest paths):
(a) The infinite series converges in finite time to a matrix $\boldsymbol{\Gamma}(\boldsymbol{A}) = [\gamma_{ij}]$ and all $\gamma_{ij} < \top$ if and only if $\lambda(\boldsymbol{A}) \leq e$:

$$\boldsymbol{A}^{(t)} \leq \boldsymbol{\Gamma}(\boldsymbol{A}) = \boldsymbol{A} \vee \boldsymbol{A}^{(2)} \vee \cdots \vee \boldsymbol{A}^{(n)} \quad \forall t \geq 1$$

(b) If the graph of $\boldsymbol{A}$ is strongly connected, then all $\gamma_{ij} > \perp$.

Theorem (stability):
A max-$\star$ is BIBO absolutely stable iff $\lambda(\boldsymbol{A}) = e$.

Solve Max-* Equations via L1,Linf Minimization

$$\mathbf{A} \boxast \mathbf{x} = \mathbf{b}$$

$$\text{Minimize} \quad ||\boldsymbol{A} \boxast \boldsymbol{x} - \boldsymbol{b}||$$

$$\text{subject to} \quad \boldsymbol{A} \boxast \boldsymbol{x} \le \boldsymbol{b}$$

greatest solution

$$\hat{\boldsymbol{x}} = \boldsymbol{A}^* \boxast' \boldsymbol{b}, \quad \boldsymbol{A}^* = \overline{\boldsymbol{A}}^T$$