A Multivariate Sequence Kernel

Trajectoires '11 Lecture

Sorbonne 1, Paris

October 14, 2011

Cees H. Elzinga

PARIS/SILC Research Group Vrije Universiteit Amsterdam

vrije Universiteit amsterdam

Purpose

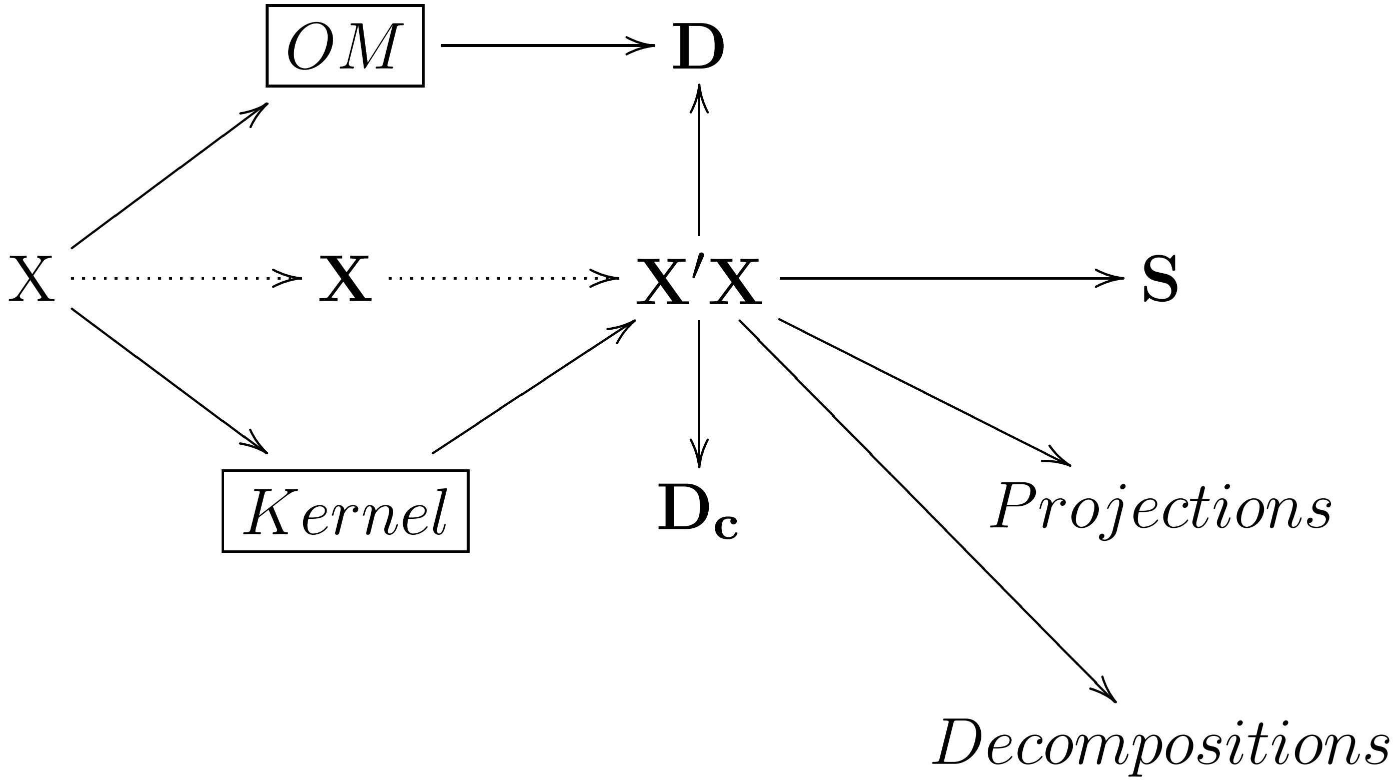
- Create vector-representation of
  - multivariate sequences
  - with numeric and symbolic variables,
  - that allows for
    - distances in Euclidean space,
    - sequences of unequal length
    - well defined similarity
    - distances to centroid: characteristic sequence, K-means
    - R_V-correlation
    - Fisher-discriminant
    - Decompositions: testing factor-structure hypotheses
- and use kernel function for feasible calculations

Unfortunately, there is **NO** software yet!

No examples

vrije Universiteit amsterdam

OM vs Kernel



In collaboration with

- Dr. Hui Wang
  - Reader in Computer Science
    Computer Science Research Institute
    University of Ulster
- Dr. Zhiwei Lin
  - Researcher
    SAP UK

vrije Universiteit amsterdam

Structure

- Data I
- The Challenge
- Preliminaries
  - Feature Vectors
  - Kernel
  - Data II
- Neighborhoods
- Hypertuples
- Hyperspace as Featurespace
- Kernel in Hyperspace

vrije Universiteit amsterdam

Structure

- Data I
  - Example
  - Structure
- The Challenge
- Preliminaries
  - Feature Vectors
  - Kernel
  - Data II
- Neighborhoods
- Hypertuples
- Hyperspace as Featurespace
- Kernel in Hyperspace

vrije Universiteit amsterdam

Multivariate Sequences

- simultaneous “timeseries” from the same objects
  - numerical
    - ECG: 12 signals
    - stock commodity prices
  - symbolical
    - life course facets (job, family, residence)
      - Gary Pollock 2007
      - Gauthier et al 2010
  - “mixed”: symbolical & numerical

vrije Universiteit amsterdam

MV Sequence: Example

- Consist of tuples (x_1, x_2, x_3, x_4)
  - $x_1 \leftarrow$ labor market status
    symbolic: $\{E, U, N, R, \ldots\}$
  - $x_2 \leftarrow$ monthly income
    numerical: $[0, 10000]$
  - $x_3 \leftarrow$ household type
    symbolic: $\{P, S, C, M, SC, \ldots\}$
  - $x_4 \leftarrow$ hours spend in housekeeping
    numerical: $[0, 400]$

vrije Universiteit amsterdam

MD Sequence: Structure

- Tuples of v variables vary over time
  - $$x = \begin{pmatrix} (x_{11}, \ldots x_{1v}) \\ \vdots \\ (x_{t1}, \ldots, x_{tv}) \\ \vdots \\ (x_{n1}, \ldots x_{nv}) \end{pmatrix}$$
- n-long sequence consists of n v-tuples
- data consist of N sequences of n_i v-tuples
- $v = 1$: sequence is ordinary time-series or string

vrije Universiteit amsterdam

Structure

- Data I
- Challenge & Strategy
- Preliminaries
  - Feature Vectors
  - Kernel
  - Data II
- Neighborhoods
- Hypertuples
- Hyperspace as Featurespace
- Kernel in Hyperspace

vrije Universiteit amsterdam

The Challenge

- Quantify the sequences
- such that linear (statistical) models apply
  - to classify the sequences
  - to use as dependent variable

vrije Universiteit amsterdam

The Strategy

- Quantify the sequences
- such that linear statistical models apply
  - to classify the sequences
  - to use as (in-)dependent variable
- Map the sequences onto vectors in $\mathbb{R}^n$
- and use the vectors to partition and model

HOW to CONSTRUCT the VECTORS??

Structure

- Data I
- Challenge & Strategy
  - Feature Vectors
    - Principle
    - Example 1: Beetles in Beetle-Space
    - Example 2: Careers in Career-Space
  - Kernel
  - Data II
- Neighborhoods
- Hypertuples
- Hyperspace as Featurespace
- Kernel in Hyperspace

vrije Universiteit amsterdam

Fig Beetle

ersiteit amsterdam

Cottonwood Stag Beetle

vrije Universiteit amsterdam

Feature Vectors: Principles

- select d features or properties $\{p_1, \ldots, p_d\}$
- map each object x to a d-vector $\mathbf{x}$
  - $\mathrm{x} \mapsto \mathbf{x} = (x_1, \ldots, x_d)$
- determine the value of the $\mathbf{x}$-coordinates x_i
  - $x_i = \begin{cases} f(p_i) & \text{if object } \mathrm{x} \text{ has property } p_i \\ 0 & \text{otherwise} \end{cases}$
  - simple: $f(p_i) = 1,$ all i (feature “on”)

4 Beetles in Beetle Space $\{0,1\}^8$

Features	$\mathbf{a}$	$\mathbf{b}$	$\mathbf{c}$	$\mathbf{d}$
crawls	1	0	1	1
flies	0	1	0	1
big eyes	1	1	0	0
long antennas	1	1	1	0
stripes	1	0	0	1
dots	1	0	0	0
eats marshmellows	0	0	0	0
intimidating	0	0	1	1

- inner product $\mathbf{a}'\mathbf{b} = \sum_i a_i b_i = 2$ counts common features
- inner product $\mathbf{a}'\mathbf{a} = \sum_i a_i^2 = 5$ counts features
- discerns $2^8 = 256$ distinct beetles

vrije Universiteit amsterdam

Beetle Feature Vectors

- beetle feature space-matrix $\mathbf{X} = (\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d})$
- Gram-matrix $\mathbf{X}'\mathbf{X} = \begin{pmatrix} 5 & 2 & 2 & 2 \\ 2 & 3 & 1 & 1 \\ 2 & 1 & 3 & 2 \\ 2 & 1 & 2 & 4 \end{pmatrix}$, inner products
- beetle vectors have
  - length: $\|\mathbf{a}\| = \sqrt{\mathbf{a}'\mathbf{a}} = \sqrt{\sum_i a_i^2} = \sqrt{5} = 2.24$ (st. dev.)
  - distance: $d(a, b) = \mathbf{a}'\mathbf{a} + \mathbf{b}'\mathbf{b} - 2\mathbf{a}'\mathbf{b} = 4$
  - angle: $\angle(a, b) = \frac{\mathbf{a}'\mathbf{b}}{\|\mathbf{a}\| \cdot \|\mathbf{b}\|} = \frac{2}{\sqrt{5 \cdot 3}} = 0.52$ (correlation)

vrije Universiteit amsterdam

Careers in Career-Space

- Alphabet $\mathcal{A} = \{a, b, c\}$ (labor market states)
- all strings $\mathcal{A}^*$: set of all possible careers
  - career $\mathrm{x} = abbcaaccbbaaaab\ldots$
  - careers are concatenations of symbols from $\mathcal{A}$
- career features: all sub-careers
  - a, ac, $abacb$, $\ldots$
- map careers onto career-feature vectors

vrije Universiteit amsterdam

2 Careers in Career-Space

careers: $\mathrm{x} = abac \mapsto \mathbf{x}$, $\mathrm{y} = bacb \mapsto \mathbf{y}$

subcareers	$\mathbf{x}$	$\mathbf{y}$
a	1	1
$\vdots$	$\vdots$	$\vdots$
aa	1	0
ab	1	1
$\vdots$	$\vdots$	$\vdots$
aba	1	0
$\vdots$	$\vdots$	$\vdots$
acb	0	1
$\vdots$	$\vdots$	$\vdots$

Feature Vectors: Problems

- feature selection: relevance?
  - no beetles eat marshmellows (irrelevant)
  - some beetles have horns (not selected)
  - all beetles have 6 legs (non-discriminating)
- feature selection: how many are necessary/acceptable?
  - $\{0,1\}^d$-vectors generate at most 2^d classes
  - dimensionality of subsequence-space is colossal: countably infinite
    - numerical problems: how to evaluate Gram-matrix - use a KERNEL

Structure

- Data I
- Challenge & Strategy
- Preliminaries
  - Feature Vectors
  - Kernel
  - Data II
- Neighborhoods
- Hypertuples
- Hyperspace as Featurespace
- Kernel in Hyperspace

vrije Universiteit amsterdam

Kernel Function: Fast Trick

- careers x and y of length m and n
- with feature vectors $\mathbf{x}$ and $\mathbf{y}$
- naively calculating $\mathbf{x}'\mathbf{y}$ takes 2^{m+n} operations: not feasible
- kernel function[a] takes $m \cdot n$ operations: feasible
- kernel function: evaluates $\kappa(\mathrm{x}, \mathrm{y}) = \mathbf{x}'\mathbf{y}$ *without* constructing $\mathbf{x}$ and $\mathbf{y}$ explicitly
- problem: no method or recipe for kernel-design

vrije Universiteit amsterdam

[a] e.g. Elzinga, Rahmann & Wang, TCS 2008

Structure

- Data I
- Challenge & Strategy
  - Feature Vectors
  - Kernel
  - Data II
- Neighborhoods
- Hypertuples
- Hyperspace as Featurespace
- Kernel in Hyperspace

vrije Universiteit amsterdam

MV Sequence: Structure

- Tuples of v variables vary over time
  - $$x = \begin{pmatrix} (x_{11}, \ldots x_{1v}) \\ \vdots \\ (x_{t1}, \ldots, x_{tv}) \\ \vdots \\ (x_{n1}, \ldots x_{nv}) \end{pmatrix}$$
- n-long sequence consists of n v-tuples
- data consist of N sequences of n_i v-tuples
- $v = 1$: sequence is ordinary time-series or string

vrije Universiteit amsterdam

Domains and Data-Space

- Domains of variables are finite
  - Symbolic variable x_j: $x_j \in D_j = \{P, S, \ldots\}$
    Domain size: $|D_j|$
  - Numeric variable x_j: $x_j \in D_j = [min, Max]$
    Domain size: $|D_j| = Max - min$
- Data Space: $\Omega = \bigtimes_{j=1}^{v} D_j, \quad |\Omega| = \prod_j |D_j|$
  - Ω consists of all possible v-tuples
  - MV-sequences arise by catenating tuples from Ω
  - Ω is our “alphabet”

vrije Universiteit amsterdam

Structure

- Data I
- Challenge & Strategy
  - Feature Vectors
  - Kernel
  - Data II
- Neighborhoods
- Hypertuples
- Hyperspace as Featurespace
- Kernel in Hyperspace

Neigborhoods

vrije Universiteit amsterdam

General Idea: A Neighborhood Space

- Objects live in neighborhoods
- Objects share neighborhoods
  - the more neighborhoods shared, the more alike, the less distant
- Map objects to neighborhood space using neighborhood vectors
  - neighborhoods as features
- Count (common) neighborhoods through products of neighborhood vectors

Neighborhood

- you live in a neighborhood
- you share properties, features with your neighbors
  - income
  - education
  - ....
- Neighborhood: a set of objects sharing a property P
  - $N(P) = \{x : x \vdash P\}$
- **Your** neighborhood: a neighborhood where **you** live
  - $N(P, x) = N(P) \Leftrightarrow x \in N(P)$
  - common neighborhood: $N(P, x, y) = N(P) \Leftrightarrow x, y \in N(P)$
- What about neighborhoods of measurements??

vrije Universiteit amsterdam

Neighborhood of Symbolic Measurements

- Let $D_i = \{a, b, c\}$; a symbolic domain.
- Neighborhood: a set $N(P) = \{x : x \vdash P\}$
- Interpret P as: "*is a subset of a domain* D"
  - D generates $2^{|D|}$ neighborhoods
- Neighborhoods of a: $\{a\}$, $\{a, b\}$, $\{a, c\}$, $\{a, b, c\}$
- Number of v-neighborhoods of a: $\phi_v(a) = 2^{|D_i|-1}$
- Common v-neighborhoods of a, b: $\phi_v(a, b) = 2^{|D_i|-2}$

Neighborhood of Numeric Measurements

- Let $D_i = [m, M]$, an **ordered** set
- Interpret P as: "*is an **ordered** subset of a domain* D"
  - D generates $\binom{M-m+1}{2}$ neighborhoods
- Neighborhoods of $x \in D_i$: $\binom{M-x+1}{1} \cdot \binom{x-m+1}{1}$
- Number of v-neighborhoods of x:
  $\phi_v(x) = (M - x + 1)(x - m + 1)$
- Common v-neighborhoods of x, y:
  $\phi_v(x, y) = (M - \max(x, y) + 1)(\min(x, y) - m + 1)$

vrije Universiteit amsterdam

Neighborhoods of v-tuples: Hypertuples

- a v-tuple of measurements: $\mathrm{x}_i = (\mathrm{x}_{i1}, \ldots, \mathrm{x}_{iv})$
- a v-tuple of neighborhoods: $h_i = (d_1, \ldots, d_v)$
- if x_{ij} “covered” by d_j for $j = 1, \ldots, v$:
  - h_i is a t-neighborhood of x_i
  - h_i is called a “hypertuple”: consists of **sets**
  - many distinct h_i “cover” x_i: $\phi_t(\mathrm{x}_i)$
  - how many??

vrije Universiteit amsterdam

Counting hypertuples

- hypertuples covering a tuple x_i

$$\phi_t(\mathrm{x}_i) = \prod_{k=1}^{v} \phi_v(\mathrm{x}_{ij})$$

- hypertuples common to $\mathrm{x}_i, \mathbf{y}_j$:

$$\phi_t(\mathrm{x}_i, \mathrm{y}_j) = \prod_{k=1}^{v} \phi_v(\mathrm{x}_{ik}, \mathrm{y}_{jk})$$

vrije Universiteit amsterdam

Hypersequences in hyperspace

- an n-sequence of v-tuples of measurements:
  $x = x_1 \dots x_n$
- an m-sequence of v-tuples of t-neighborhoods:
  $h = h_1 \dots h_m$, $m \leq n$
- if each x_i is covered by some h_i
  - h is an md-neighborhood of x
  - h is called "hypersequence"
  - many distinct h "cover" the same x: $\phi_{md}(x)$
  - many distinct h cover both x and y: $\phi_{md}(x, y)$

vrije Universiteit amsterdam

Hypersequences: an example

i	x	h
1	$x_1 = \langle 2, 6, b, p \rangle$	$h_1 = \langle \{[1,3]\}, \{[5,8]\}, \{a, b\}, \{p, r\} \rangle$
2	$x_2 = \langle 3, 6, b, r \rangle$	$h_2 = \langle \{[2,4]\}, \{[6,7]\}, \{a, b, c\}, \{q, r\} \rangle$
3	$x_3 = \langle 4, 8, a, q \rangle$	$h_3 = \langle \{[3,4]\}, \{[8,8]\}, \{a\}, \{p, q, r\} \rangle$

- h_i are arbitrary within constraint that the x_i must be properly covered

vrije Universiteit amsterdam

Structure

- Data I
- Challenge & Strategy
  - Feature Vectors
  - Kernel
  - Data II
- Neighborhoods
- Hypertuples
- Hyperspace as Featurespace
- Kernel in Hyperspace

Dataspace & Hyperspace

- Ω: the set of all possible v-tuples
  - the "multidimensional alphabet"
- Ω^*: the set of all sequences of v-tuples - Dataspace
- $\mathcal{H}$: the set of all distinct hypertuples: t-neighborhoods
  - the "hyperalphabet" consists of tuples of sets
- $\mathcal{H}^*$: the set of all sequences of hypertuples - Hyperspace
- $\Omega^* \subset \mathcal{H}^*$, $\mathcal{H}^*$ is finite since the domains are finite

vrije Universiteit amsterdam

Vectors in hyperspace

- Order the hypersequences in $\mathcal{H}^*$
  - i.e. assign a unique integer $r(h)$ to each $h \in \mathcal{H}^*$

- construct a vector $\mathbf{x} = (x_1, x_2, \ldots)$ for each $\mathrm{x} \in \Omega^*$:

$$x_{r(h)} = \begin{cases} 1 & h \in \mathcal{H}^* \text{ covers } \mathrm{x} \in \Omega^* \\ 0 & \text{otherwise} \end{cases}$$

- $\mathbf{x}'\mathbf{x}$ counts the number of covers of x
- $\mathbf{x}'\mathbf{y}$ counts the number of common covers of x and y
- constructing $\mathbf{x}$ and $\mathbf{y}$ and directly evaluating $\mathbf{x}'\mathbf{y}$ is not feasible

vrije Universiteit amsterdam

Structure

- Data I
- Challenge & Strategy
  - Feature Vectors
  - Kernel
  - Data II
- Neighborhoods
- Hypertuples
- Hyperspace as Featurespace
- Kernel in Hyperspace

vrije Universiteit amsterdam

An Efficient Kernel

- x and y are MD-sequences of lengths k and n
- x^m denotes the first m v-tuples of x

$$\phi(\mathrm{x}^k, \mathrm{y}^n) = \phi(\mathrm{x}^k, \mathrm{y}^{n-1}) + \phi(\mathrm{x}^{k-1}, \mathrm{y}^n)$$

$$- \phi(\mathrm{x}^{k-1}, \mathrm{y}^{n-1}) \cdot (2 - \phi_t(\mathrm{x}_k, \mathrm{y}_n))$$

- initialize $\phi(\mathrm{x}^0, \mathrm{y}^j) = 1 = \phi(\mathrm{x}^j, \mathrm{y}^0)$
- $\phi(\mathrm{x}^k, \mathrm{y}^n) = \mathbf{x}'\mathbf{y}$ takes time proportional to $k \cdot n$

vrije Universiteit amsterdam

An Efficient Kernel?

- $\phi_t(\mathrm{x}_k, \mathrm{y}_n) = \prod_{i=1}^{v} \phi_v(\mathrm{x}_{ki}, \mathrm{y}_{ni})$
  - for longer sequences,
  - for large domains,
  - for many variables,
  - this generates very big numbers: special big-number arithmetic required
- $\mathbf{x}'\mathbf{x}$ and $\mathbf{y}'\mathbf{y}$ very big compared to $\mathbf{x}'\mathbf{y}$
  - the columns of the Gram-matrix “almost orthogonal”
  - big distances, small angles

vrije Universiteit amsterdam

Questions?

vrije Universiteit amsterdam