Vectorial Data
- Examples for learning are often encoded as real-valued vectors:
  - **Text mining:** E.g., “All human beings are born free and equal in dignity and rights” becomes
    \[
    (\downarrow, \downarrow, \downarrow, \ldots)
    \]
- **Robot/process control:**
  \[\text{state} = (x, \dot{x}, y, \dot{y}, \rho, \dot{\rho})\]
- Learning methods based on calculus, linear algebra, multi-variate statistics, optimization theory

Non-Vectorial/Structured Data
- Sets, sequences, strings, trees, graphs, logical formulas
- Difficult to apply methods for vectors
- New problems: variability, mapping, matching, complexity
- **Trees:** DOM tree kernels for classification of XML docs
  \[
  \text{Tree node} \text{ are considered a set of subtrees “adjusted to the left”}
  \]
  
  **DOM-tree kernel:** common subtrees “adjusted to the left”
  **Set tree kernel:** successors of a tree node are considered a set
  **String tree kernel:** common subtrees with gaps → string kernel applied to list of successors
  **Fuzzy tree kernel:** modification of set tree kernel

Graph Similarities as Kernels
- **Zelinka-Similarity:** = number of nodes \(n\) in the greatest common subgraph
- Computation of \(n\): Hopfield network
- T-Similarity not positive-semidefinite
- **Methods:**
  - PPC: Pairwise proximity classifier
  - Convexification
  - naive

Rel. Reinforcement Learning
- **Probabilistic planning problems in unknown environments**
- **Model learning = learning probabilistic functions mapping graphs to graphs.**

Graph: e.g., classification of images, classification of chemical compounds
- Kernels for general graphs
- (sub-) graph isomorphisms, complexity
- Kernel vs. similarity

Genres

Graph Similarities as Kernels
- **Graph Similarity ↔ Graph Match ↔ Graph Kernel?**
  \[
  (\sigma, 2, 4, 6, 8, 10)
  \]
  
  **PPC:** 1.0 1.0 0.96 0.94 0.77
  **naive:** 1.0 1.0 0.95 0.96 0.80

(From Pasula, Zettlemoyer, and Kaelbling 2004)

Genre Classification
- To which extend can the genre of a document be determined from its structure?

DOM Tree Kernel
- Document structure described by DOM tree
- A tree kernel \(k(T, T')\) determines similarity of \(T_1\) and \(T_2\) based on common subtrees
- Until now only kernels for ordered parse trees (Collins & Duffy, 2002)
- In contrast to parse trees, text elements might be missing/inserted/permitted

Genre Classification
- To which extend can the genre of a document be determined from its structure?

Graph Similarities as Kernels
- **Graph Similarity ↔ Graph Match ↔ Graph Kernel?**
  \[
  (\sigma, 2, 4, 6, 8, 10)
  \]
  
  **PPC:** 1.0 1.0 0.96 0.94 0.77
  **naive:** 1.0 1.0 0.95 0.96 0.80

(From Pasula, Zettlemoyer, and Kaelbling 2004)