Introduction to Machine Learning
Lecture 9

Mehryar Mohri
Courant Institute and Google Research
mohri@cims.nyu.edu

Kernel Methods

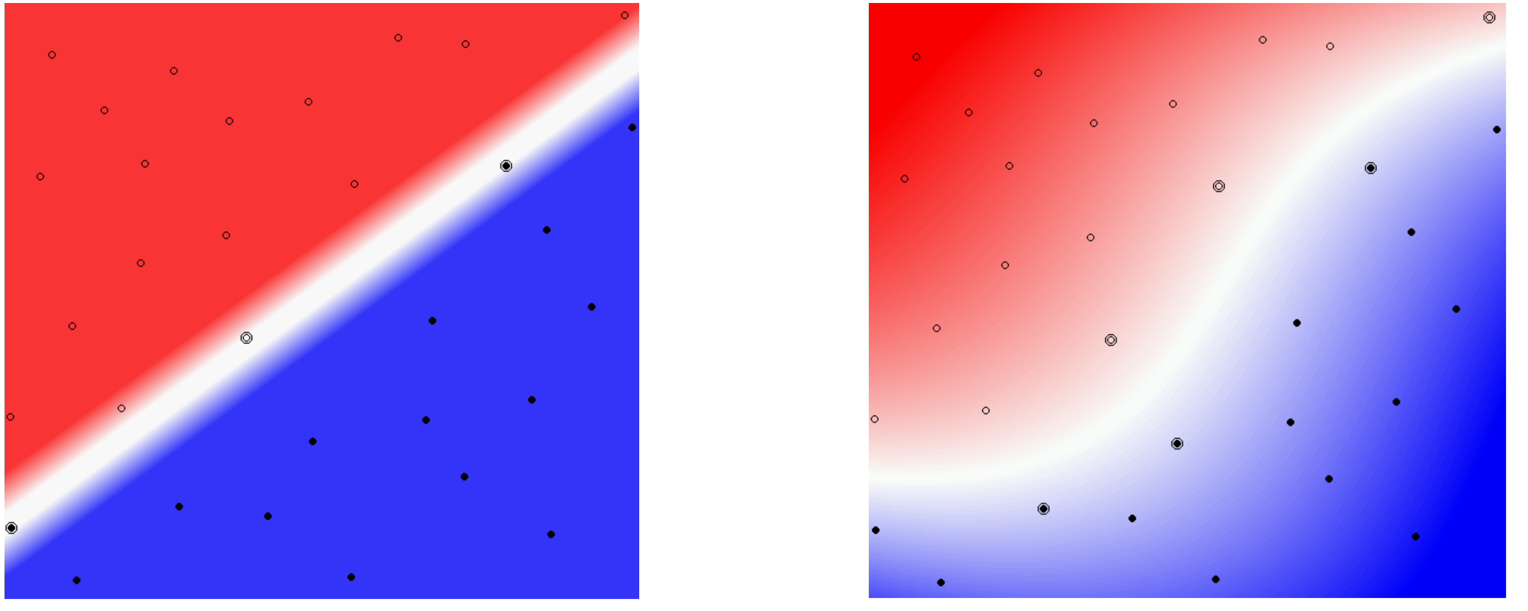
Motivation

- Non-linear decision boundary.
- Efficient computation of inner products in high dimension.
- Flexible selection of more complex features.

This Lecture

- Definitions
- SVMs with kernels
- Closure properties
- Sequence Kernels

Non-Linear Separation



- Linear separation impossible in most problems.
- Non-linear mapping from input space to high-dimensional feature space: $\Phi\colon X \to F$.
- Generalization ability: independent of $\dim(F)$, depends only on ρ and m.

Kernel Methods

- Idea:
  - Define $K \colon X \times X \to \mathbb{R}$, called kernel, such that:

$$\Phi(x) \cdot \Phi(y) = K(x, y).$$

  - K often interpreted as a similarity measure.
- Benefits:
  - Efficiency: K is often more efficient to compute than Φ and the dot product.
  - Flexibility: K can be chosen arbitrarily so long as the existence of Φ is guaranteed (symmetry and positive definiteness condition).

PDS Condition

- **Definition**: a kernel $K : X \times X \to \mathbb{R}$ is positive definite symmetric (PDS) if for any $\{x_1, \ldots, x_m\} \subseteq X$, the matrix $\mathbf{K} = [K(x_i, x_j)]_{ij} \in \mathbb{R}^{m \times m}$ is symmetric positive semi-definite (SPSD).
- $\mathbf{K}$ SPSD if symmetric and one of the 2 equiv. cond.'s:
  - its eigenvalues are non-negative.
  - for any $\mathbf{c} \in \mathbb{R}^{m \times 1}$, $\mathbf{c}^\top \mathbf{K} \mathbf{c} = \sum_{i,j=1}^{n} c_i c_j K(x_i, x_j) \geq 0.$
- **Terminology**: PDS for kernels, SPSD for kernel matrices (see (Berg et al., 1984)).

Example - Polynomial Kernels

- Definition:

$$\forall x, y \in \mathbb{R}^N,\ K(x, y) = (x \cdot y + c)^d, \quad c > 0.$$

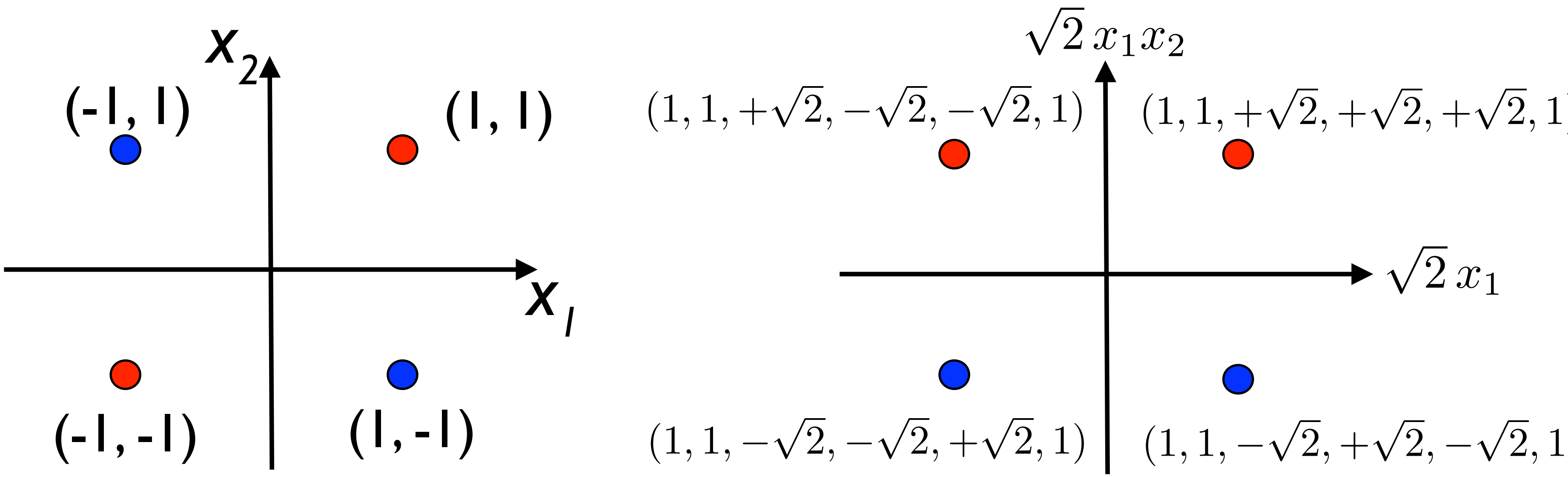
- Example: for $N=2$ and $d=2$,

$$K(x, y) = (x_1 y_1 + x_2 y_2 + c)^2$$

$$= \begin{bmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}\, x_1 x_2 \\ \sqrt{2c}\, x_1 \\ \sqrt{2c}\, x_2 \\ c \end{bmatrix} \cdot \begin{bmatrix} y_1^2 \\ y_2^2 \\ \sqrt{2}\, y_1 y_2 \\ \sqrt{2c}\, y_1 \\ \sqrt{2c}\, y_2 \\ c \end{bmatrix}.$$

XOR Problem

- Use second-degree polynomial kernel with $c = 1$:



Linearly non-separable

Linearly separable by $x_1 x_2 = 0.$

Other Standard PDS Kernels

- Gaussian kernels:

$$K(x, y) = \exp\left(-\frac{||x - y||^2}{2\sigma^2}\right), \;\; \sigma \neq 0.$$

- Sigmoid Kernels:

$$K(x, y) = \tanh(a(x \cdot y) + b), \;\; a, b \geq 0.$$

This Lecture

- Definitions
- SVMs with kernels
- Closure properties
- Sequence Kernels

Reproducing Kernel Hilbert Space

(Aronszajn, 1950)

- Theorem: Let $K \colon X \times X \to \mathbb{R}$ be a PDS kernel. Then, there exists a Hilbert space H and a mapping Φ from X to H such that

$$\forall x, y \in X, \;\; K(x, y) = \Phi(x) \cdot \Phi(y).$$

Furthermore, the following reproducing property holds:

$$\forall f \in H_0, \forall x \in X, \;\; f(x) = \langle f, \Phi(x) \rangle = \langle f, K(x, \cdot) \rangle.$$

- Notes:
  - H is called the reproducing kernel Hilbert space (RKHS) associated to K.
  - A Hilbert space such that there exists $\Phi: X \to H$ with $K(x, y) = \Phi(x) \cdot \Phi(y)$ for all $x, y \in X$ is also called a feature space associated to K. Φ is called a feature mapping.
  - Feature spaces associated to K are in general not unique.

Consequence: SVMs with PDS Kernels

(Boser, Guyon, and Vapnik, 1992)

- Constrained optimization:

$$\max_{\alpha} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

$\Phi(x_i) \cdot \Phi(x_j)$

$$\text{subject to: } 0 \leq \alpha_i \leq C \wedge \sum_{i=1}^{m} \alpha_i y_i = 0, i \in [1, m].$$

- Solution:

$$h(x) = \text{sgn}\Big(\sum_{i=1}^{m} \alpha_i y_i K(x_i, x) + b\Big),$$

$\Phi(x_i) \cdot \Phi(x)$

$\Phi(x_j) \cdot \Phi(x_i)$

with $b = y_i - \sum_{j=1}^{m} \alpha_j y_j K(x_j, x_i)$ for any x_i with $0 < \alpha_i < C$.

SVMs with PDS Kernels

- Constrained optimization:

Hadamard product

$$\max_{\boldsymbol{\alpha}} 2\, \mathbf{1}^\top \boldsymbol{\alpha} - (\boldsymbol{\alpha} \circ \mathbf{y})^\top \mathbf{K} (\boldsymbol{\alpha} \circ \mathbf{y})$$

$$\text{subject to: } \mathbf{0} \leq \boldsymbol{\alpha} \leq \mathbf{C} \wedge \boldsymbol{\alpha}^\top \mathbf{y} = 0.$$

- Solution:

$$h = \operatorname{sgn}\Big(\sum_{i=1}^{m} \alpha_i y_i K(x_i, \cdot) + b \Big),$$

with $b = y_i - (\boldsymbol{\alpha} \circ \mathbf{y})^\top \mathbf{K} \mathbf{e}_i$ for any x_i with $0 < \alpha_i < C$.

Generalization: Representer Theorem

(Kimeldorf and Wahba, 1971)

- Theorem: Let $K\colon X \times X \to \mathbb{R}$ be a PDS kernel and H its corresponding RKHS. Then, for any non-decreasing function $G\colon \mathbb{R} \to \mathbb{R}$ and any $L\colon \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$ the optimization problem

$$\operatorname*{argmin}_{h \in H} F(h) = \operatorname*{argmin}_{h \in H} G(\|h\|_H^2) + L\big(h(x_1), \ldots, h(x_m)\big)$$

admits a solution of the form $h^* = \sum_{i=1}^{m} \alpha_i K(x_i, \cdot)$.

If G is further assumed to be increasing, then any solution has this form.

- Proof: let $H_1 = \text{span}(\{K(x_i, \cdot) \colon i \in [1, m]\})$. Any $h \in H$ admits the decomposition $h = h_1 + h^\perp$ according to $H = H_1 \oplus H_1^\perp$.
  - Since G is non-decreasing,
  $$G(\|h_1\|^2) \leq G(\|h_1\|^2 + \|h^\perp\|^2) = G(\|h\|^2).$$
  - By the reproducing property, for all $i \in [1, m]$,
  $$h(x_i) = \langle h, K(x_i, \cdot) \rangle = \langle h_1, K(x_i, \cdot) \rangle = h_1(x_i).$$
  - Thus, $L\big(h(x_1), \ldots, h(x_m)\big) = L\big(h_1(x_1), \ldots, h_1(x_m)\big)$ and $F(h_1) \leq F(h)$.
  - If G is increasing, then $F(h_1) < F(h)$ and any solution of the optimization problem must be in H_1.

Kernel-Based Algorithms

- PDS kernels used to extend a variety of algorithms in classification and other areas:
  - regression.
  - ranking.
  - dimensionality reduction.
  - clustering.
- But, how do we define PDS kernels?

This Lecture

- Definitions
- SVMs with kernels
- Closure properties
- Sequence Kernels

Closure Properties of PDS Kernels

- Theorem: Positive definite symmetric (PDS) kernels are closed under:
  - sum,
  - product,
  - tensor product,
  - pointwise limit,
  - composition with a power series.

Closure Properties - Proof

- Proof: closure under sum:

$$\mathbf{c}^\top \mathbf{K} \mathbf{c} \geq 0 \wedge \mathbf{c}^\top \mathbf{K}' \mathbf{c} \geq 0 \Rightarrow \mathbf{c}^\top (\mathbf{K} + \mathbf{K}') \mathbf{c} \geq 0.$$

  - closure under product: $\mathbf{K} = \mathbf{M}\mathbf{M}^\top$,

$$\sum_{i,j=1}^{m} c_i c_j (\mathbf{K}_{ij} \mathbf{K}'_{ij}) = \sum_{i,j=1}^{m} c_i c_j \bigg(\Big[\sum_{k=1}^{m} \mathbf{M}_{ik} \mathbf{M}_{jk} \Big] \mathbf{K}'_{ij} \bigg)$$

$$= \sum_{k=1}^{m} \bigg[\sum_{i,j=1}^{m} c_i c_j \mathbf{M}_{ik} \mathbf{M}_{jk} \mathbf{K}'_{ij} \bigg] = \sum_{k=1}^{m} \mathbf{z}_k^\top \mathbf{K}' \mathbf{z}_k \geq 0,$$

with $\mathbf{z}_k = \begin{bmatrix} c_1 \mathbf{M}_{1k} \\ \cdots \\ c_m \mathbf{M}_{mk} \end{bmatrix}$.

- Closure under tensor product:
  - definition: for all $x_1, x_2, y_1, y_2 \in X$,

$$(K_1 \otimes K_2)(x_1, y_1, x_2, y_2) = K_1(x_1, x_2)K_2(y_1, y_2).$$

  - thus, PDS kernel as product of the kernels

$$(x_1, y_1, x_2, y_2) \rightarrow K_1(x_1, x_2) \quad (x_1, y_1, x_2, y_2) \rightarrow K_2(y_1, y_2).$$

- Closure under pointwise limit: if for all $x, y \in X$,

$$\lim_{n \to \infty} K_n(x, y) = K(x, y),$$

Then, $(\forall n, \mathbf{c}^\top \mathbf{K}_n \mathbf{c} \geq 0) \Rightarrow \lim_{n \to \infty} \mathbf{c}^\top \mathbf{K}_n \mathbf{c} = \mathbf{c}^\top \mathbf{K} \mathbf{c} \geq 0.$

- Closure under composition with power series:
  - assumptions: K PDS kernel with $|K(x,y)| < \rho$ for all $x, y \in X$ and $f(x) = \sum_{n=0}^{\infty} a_n x^n, a_n \geq 0$ power series with radius of convergence ρ.
  - $f \circ K$ is a PDS kernel since K^n is PDS by closure under product, $\sum_{n=0}^{N} a_n K^n$ is PDS by closure under sum, and closure under pointwise limit.
- Example: for any PDS kernel K, $\exp(K)$ is PDS.

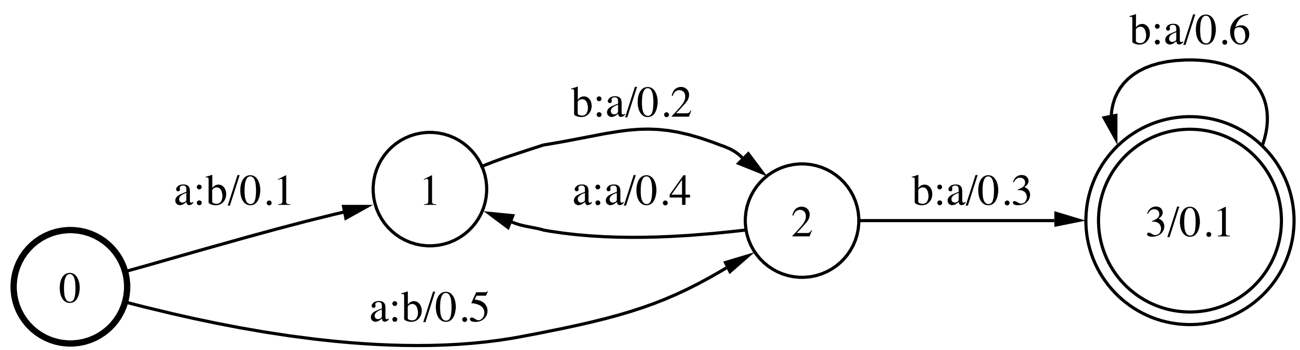
This Lecture

- Definitions
- SVMs with kernels
- Closure properties
- Sequence Kernels

Sequence Kernels

- Definition: Kernels defined over pairs of strings.
  - Motivation: computational biology, text and speech classification.
  - Idea: two sequences are related when they share some common substrings or subsequences.
  - Example: sum of the product of the counts of common substrings.

Weighted Transducers



$T(x, y) =$ Sum of the weights of all accepting paths with input x and output y.

$$T(abb, baa) = .1 \times .2 \times .3 \times .1 + .5 \times .3 \times .6 \times .1$$

Rational Kernels over Strings

(Cortes et al., 2004)

- Definition: a kernel $K\colon \Sigma^* \times \Sigma^* \to \mathbb{R}$ is rational if $K = T$ for some weighted transducer T.

- Definition: let $T_1\colon \Sigma^* \times \Delta^* \to \mathbb{R}$ and $T_2\colon \Delta^* \times \Omega^* \to \mathbb{R}$ be two weighted transducers. Then, the composition of T_1 and T_2 is defined for all $x \in \Sigma^*, y \in \Omega^*$ by

$$(T_1 \circ T_2)(x, y) = \sum_{z \in \Delta^*} T_1(x, z)\; T_2(z, y).$$

- Definition: the inverse of a transducer $T\colon \Sigma^* \times \Delta^* \to \mathbb{R}$ is the transducer $T^{-1}\colon \Delta^* \times \Sigma^* \to \mathbb{R}$ obtained from T by swapping input and output labels.

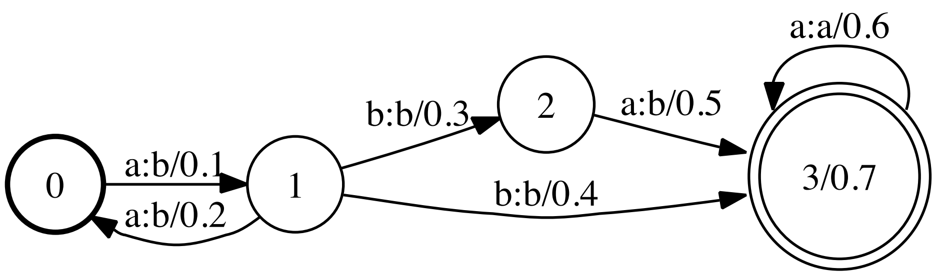
Composition

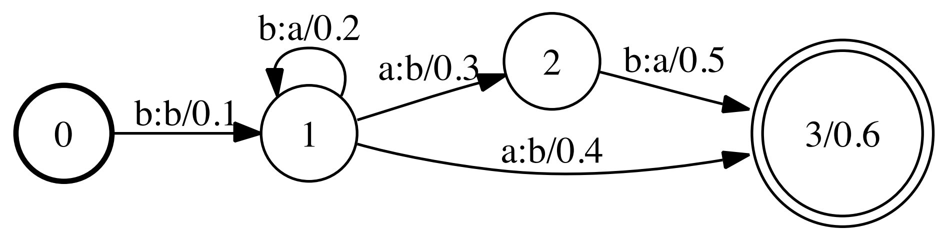
- Theorem: the composition of two weighted transducer is also a weighted transducer.
- Proof: constructive proof based on composition algorithm.
  - states identified with pairs.
  - ϵ-free case: transitions defined by

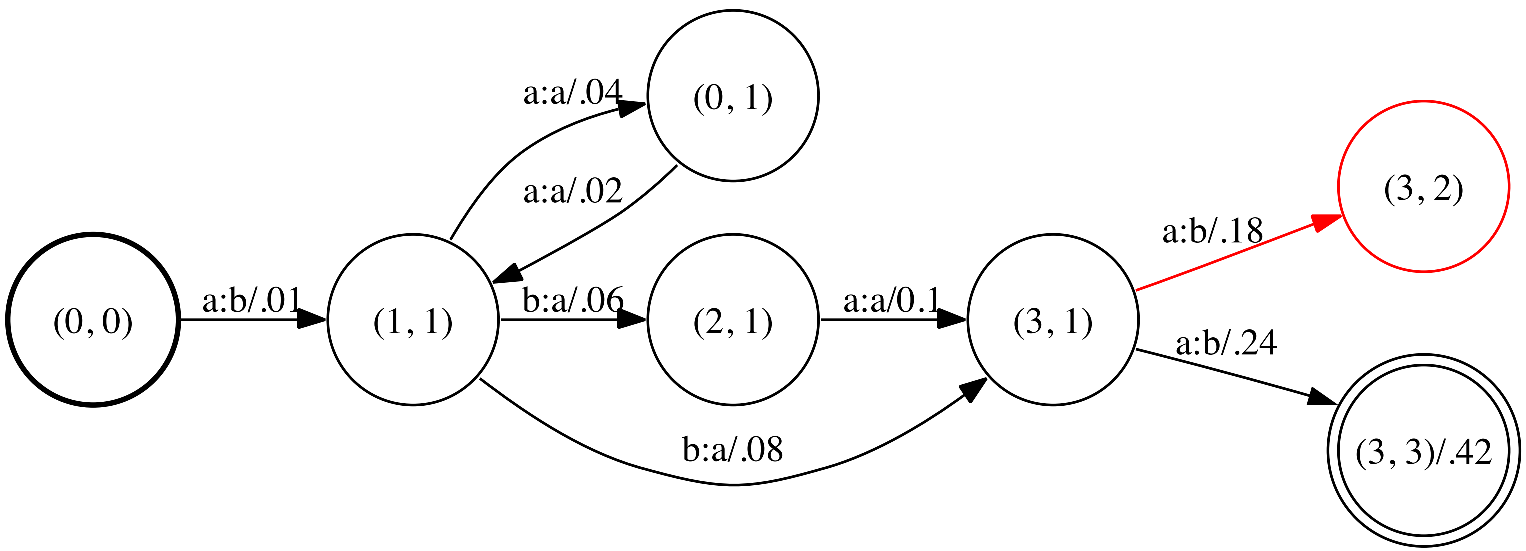
$$E = \biguplus_{\substack{(q_1, a, b, w_1, q_2) \in E_1 \\ (q'_1, b, c, w_2, q'_2) \in E_2}} \Big\{ \Big((q_1, q'_1), a, c, w_1 \times w_2, (q_2, q'_2) \Big) \Big\}.$$

  - general case: use of intermediate ϵ-filter.

Composition Algorithm
ε-Free Case





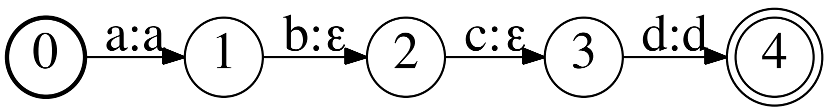


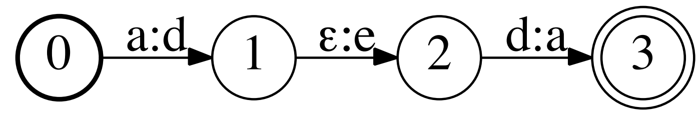
Complexity: $O(|T_1|\,|T_2|)$ in general, linear in some cases.

Redundant ε-Paths Problem

(MM et al. 1996)

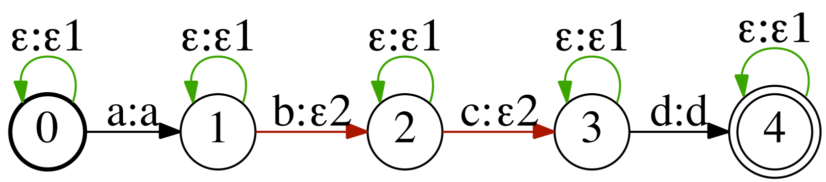
T_1

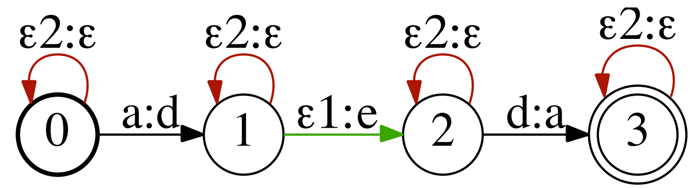




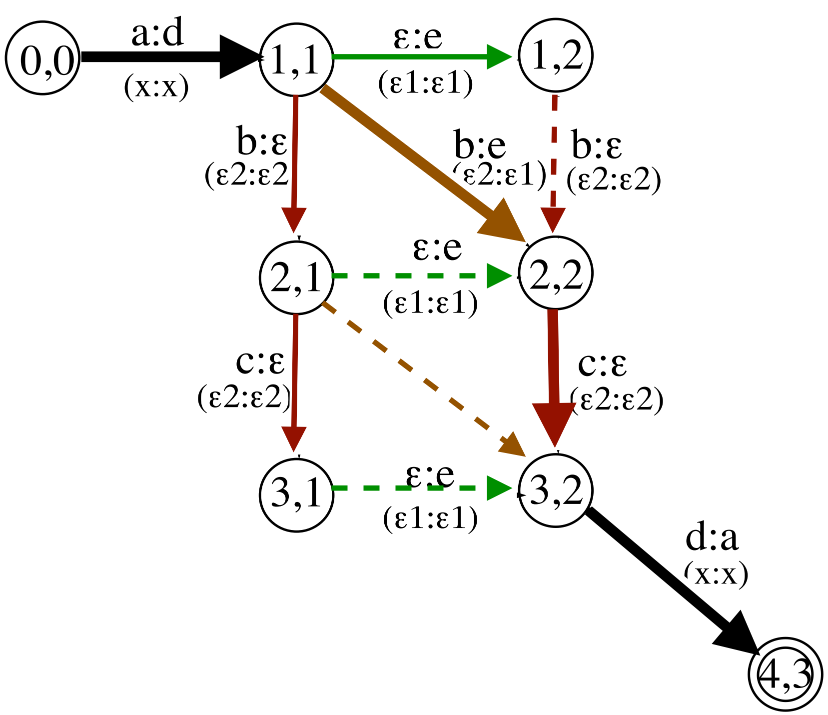
T_2

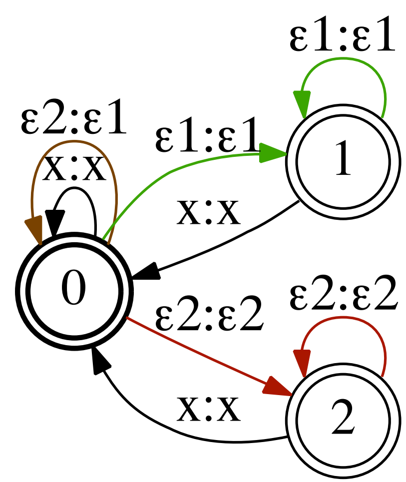
$\widetilde{T}_1$





$\widetilde{T}_2$





F

$$T = \widetilde{T}_1 \circ F \circ \widetilde{T}_2.$$

PDS Rational Kernels General Construction

- Theorem: for any weighted transducer $T: \Sigma^* \times \Sigma^* \to \mathbb{R}$, the function $K = T \circ T^{-1}$ is a PDS rational kernel.

- Proof: by definition, for all $x, y \in \Sigma^*$,

$$K(x, y) = \sum_{z \in \Delta^*} T(x, z)\, T(y, z).$$

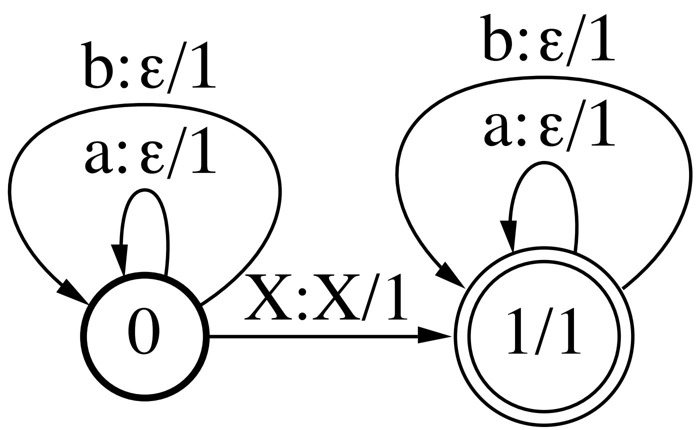
  - K is pointwise limit of $(K_n)_{n \geq 0}$ defined by

$$\forall x, y \in \Sigma^*, \;\; K_n(x, y) = \sum_{|z| \leq n} T(x, z)\, T(y, z).$$

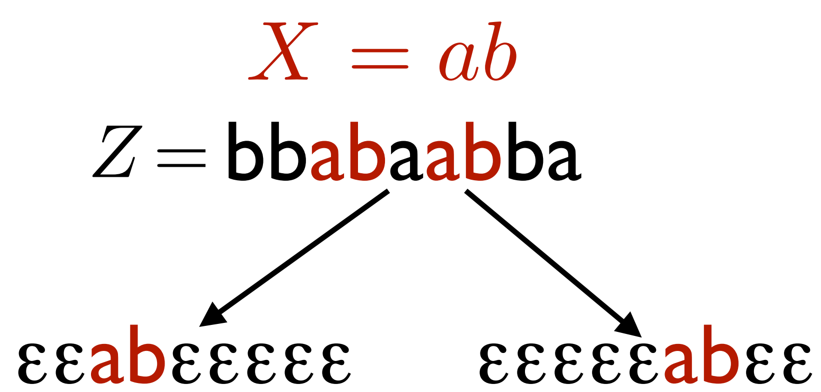
  - K_n is PDS since for any sample $(x_1, \ldots, x_m)$,

$$\mathbf{K}_n = \mathbf{A}\mathbf{A}^\top \text{ with } \mathbf{A} = (K_n(x_i, z_j))_{\substack{i \in [1, m] \\ j \in [1, N]}}.$$

Counting Transducers

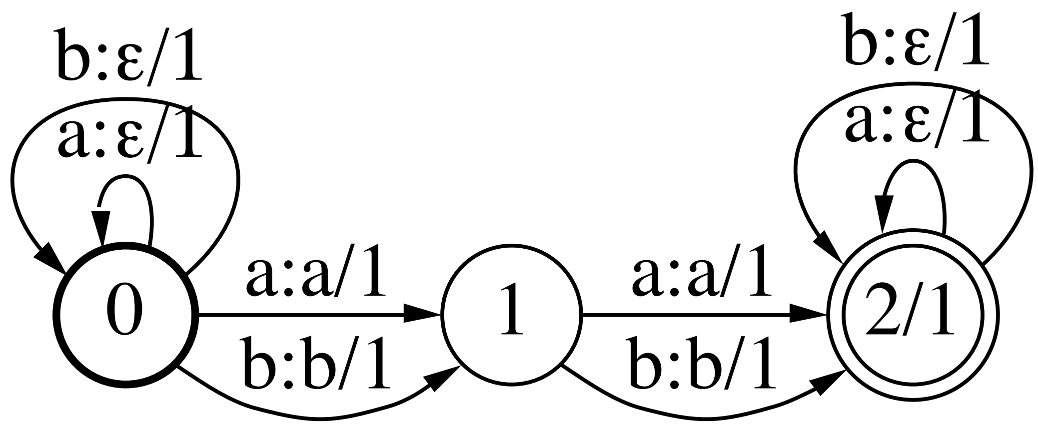


T_X



- X may be a string or an automaton representing a regular expression.
- Counts of Z in X: sum of the weights of accepting paths of $Z \circ T_X$.

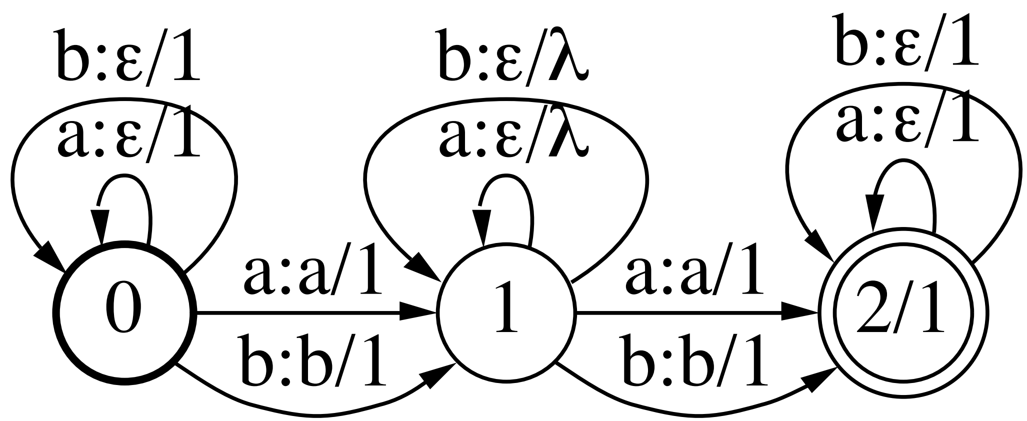
Transducer Counting Bigrams



T_{bigram}

Counts of Z given by $Z \circ T_{\text{bigram}} \circ ab$.

Transducer Counting Gappy Bigrams



$T_{\text{gappy bigram}}$

Counts of Z given by $Z \circ T_{\text{gappy bigram}} \circ ab$, gap penalty $\lambda \in (0, 1)$.

Kernels for Other Discrete Structures

- Similarly, PDS kernels can be defined on other discrete structures:
  - Images,
  - graphs,
  - parse trees,
  - automata,
  - weighted automata.

References

- N. Aronszajn, Theory of Reproducing Kernels, *Trans. Amer. Math. Soc.*, 68, 337-404, 1950.
- Peter Bartlett and John Shawe-Taylor. Generalization performance of support vector machines and other pattern classifiers. In *Advances in kernel methods: support vector learning*, pages 43–54. MIT Press, Cambridge, MA, USA, 1999.
- Christian Berg, Jens Peter Reus Christensen, and Paul Ressel. *Harmonic Analysis on Semigroups*. Springer-Verlag: Berlin-New York, 1984.
- Bernhard Boser, Isabelle M. Guyon, and Vladimir Vapnik. *A training algorithm for optimal margin classifiers*. In proceedings of COLT 1992, pages 144-152, Pittsburgh, PA, 1992.
- Corinna Cortes, Patrick Haffner, and Mehryar Mohri. Rational Kernels: Theory and Algorithms. *Journal of Machine Learning Research (JMLR)*, 5:1035-1062, 2004.
- Corinna Cortes and Vladimir Vapnik, Support-Vector Networks, *Machine Learning*, 20, 1995.
- Kimeldorf, G. and Wahba, G. *Some results on Tchebycheffian Spline Functions*, J. Mathematical Analysis and Applications, 33, 1 (1971) 82-95.

References

- James Mercer. Functions of Positive and Negative Type, and Their Connection with the Theory of Integral Equations. In *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, Vol. 83, No. 559, pp. 69-70, 1909.
- Mehryar Mohri, Fernando C. N. Pereira, and Michael Riley. *Weighted Automata in Text and Speech Processing*, In *Proceedings of the 12th biennial European Conference on Artificial Intelligence (ECAI-96), Workshop on Extended finite state models of language*. Budapest, Hungary, 1996.
- Fernando C. N. Pereira and Michael D. Riley. Speech Recognition by Composition of Weighted Finite Automata. In Finite-State Language Processing, pages 431-453. MIT Press, 1997.
- I. J. Schoenberg, Metric Spaces and Positive Definite Functions. *Transactions of the American Mathematical Society*, Vol. 44, No. 3, pp. 522-536, 1938.
- Vladimir N. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer, Basederlin, 1982.
- Vladimir N. Vapnik. *The Nature of Statistical Learning Theory.* Springer, 1995.
- Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, New York, 1998.

Appendix

Shortest-Distance Problem

- Definition: for any regulated weighted transducer T, define the shortest distance from state q to F as

$$d(q, F) = \bigoplus_{\pi \in P(q,F)} w[\pi].$$

- Problem: compute $d(q, F)$ for all states $q \in Q$.
- Algorithms:
  - Generalization of Floyd-Warshall.
  - Single-source shortest-distance algorithm.

All-Pairs Shortest-Distance Algorithm

(MM, 2002)

- Assumption: closed semiring (not necessarily idempotent).
- Idea: generalization of Floyd-Warshall algorithm.
- Properties:
  - Time complexity: $\Omega(|Q|^3(T_\oplus + T_\otimes + T_\star))$.
  - Space complexity: $\Omega(|Q|^2)$ with an in-place implementation.

Closed Semirings

(Lehmann, 1977)

- Definition: a semiring is closed if the closure is well defined for all elements and if associativity, commutativity, and distributivity apply to countable sums.

- Examples:
  - Tropical semiring.
  - Probability semiring when including infinity or when restricted to well-defined closures.

Pseudocode

Gen-All-Pairs(G)

for $i \leftarrow 1$ **to** $|Q|$ **do**
    for $j \leftarrow 1$ **to** $|Q|$ **do**
        $d[i,j] \leftarrow \bigoplus_{e \in E \cap P(i,j)} w[e]$
for $k \leftarrow 1$ **to** $|Q|$ **do**
    for $i \leftarrow 1$ **to** $|Q|, i \neq k$ **do**
        for $j \leftarrow 1$ **to** $|Q|, j \neq k$ **do**
            $d[i,j] \leftarrow d[i,j] \oplus (d[i,k] \otimes d[k,k]^* \otimes d[k,j])$
    for $i \leftarrow 1$ **to** $|Q|, i \neq k$ **do**
        $d[k,i] \leftarrow d[k,k]^* \otimes d[k,i]$
        $d[i,k] \leftarrow d[i,k] \otimes d[k,k]^*$
    $d[k,k] \leftarrow d[k,k]^*$

Single-Source Shortest-Distance Algorithm

(MM, 2002)

- **Assumption**: k-closed semiring.

$$\forall x \in \mathbb{K}, \quad \bigoplus_{i=0}^{k+1} x^i = \bigoplus_{i=0}^{k} x^i.$$

- **Idea**: generalization of relaxation, but must keep track of weight added to $d[q]$ since the last time q was enqueued.

- **Properties**:
  - works with any queue discipline and any k-closed semiring.
  - Classical algorithms are special instances.

Pseudocode

$\textsc{Generic-Single-Source-Shortest-Distance}\ (G, s)$

for $i \leftarrow 1$ **to** $|Q|$
  do $d[i] \leftarrow r[i] \leftarrow \overline{0}$
$d[s] \leftarrow r[s] \leftarrow \overline{1}$
$S \leftarrow \{s\}$
while $S \neq \emptyset$
  do $q \leftarrow head(S)$
    $\textsc{Dequeue}(S)$
    $r' \leftarrow r[q]$
    $r[q] \leftarrow \overline{0}$
    for each $e \in E[q]$
      do if $d[n[e]] \neq d[n[e]] \oplus (r' \otimes w[e])$
        then $d[n[e]] \leftarrow d[n[e]] \oplus (r' \otimes w[e])$
          $r[n[e]] \leftarrow r[n[e]] \oplus (r' \otimes w[e])$
          if $n[e] \notin S$
            then $\textsc{Enqueue}(S, n[e])$
$d[s] \leftarrow \overline{1}$

Notes

- Complexity:
  - depends on queue discipline used.

$$O(|Q| + (T_\oplus + T_\otimes + C(A))|E| \max_{q \in Q} N(q) + (C(I) + C(E)) \sum_{q \in Q} N(q))$$

  - coincides with that of Dijkstra and Bellman-Ford for shortest-first and FIFO orders.
  - linear for acyclic graphs using topological order.

$$O(|Q| + (T_\oplus + T_\otimes)|E|)$$

- Approximation: ϵ-k-closed semiring, e.g., for graphs in probability semiring.