

Derived linear equation in \mathbb{R}^n

$$K\alpha = \lambda n\alpha \quad (9.7)$$

for $\alpha \in \mathbb{R}^n$ (eigenvectors),

$$K \in \mathbb{R}^{n \times n}, K_{ij} = \mathbb{K}(x^{(i)}, x^{(j)}) \text{ (“kernel Gram matrix”)}$$

Intuition: Look at linear kernel.

$$\mathbb{K}(x, y) = x^T y, \quad (9.8)$$

$$\text{Feature map } x \mapsto \underbrace{\Phi(x) = x}_{\text{identity}} \quad (9.9)$$

Sample covariance matrix

$$\hat{\Sigma} = \frac{1}{n} \sum_i x^{(i)} x^{(i)T} = \frac{1}{n} \overbrace{X^T X}^{d \times d}, \quad (9.10)$$

$$\text{where } X = \begin{pmatrix} - & x^{(1)} & - \\ & \vdots & \\ - & x^{(n)} & - \end{pmatrix}$$

and $x^{(i)} \in \mathbb{R}^d, i = 1, \dots, n$.

The kernel Gram matrix is

$$K = \overbrace{X X^T}^{n \times n}. \quad (9.11)$$

The covariance matrix is $d \times d$ in this case, but could be infinite dimensional in another kernel feature space. The kernel Gram matrix will always be $n \times n$, where n is the number of data points.

9.2 Graph kernels

9.2.1 All-subsets kernel

Given an index set $\mathcal{I} = \{1, 2, \dots, d\}$ and vector $x = (x_1, x_2, \dots, x_d)$, define for any $A \subseteq \mathcal{I}$:

$$\phi_A(x) = \prod_{i \in A} x_i. \quad (9.12)$$

The feature map is $\Phi : x \rightarrow \mathbb{R}^{2^d}$,

$$\Phi(x) = (\phi_\emptyset(x), \phi_{\{1\}}(x), \dots) \quad (9.13)$$

$$= (\phi_A(x))_{A \subseteq \mathcal{I}} \in \mathbb{R}^{2^d}. \quad (9.14)$$

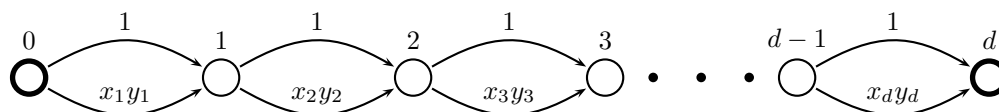
How do we compute that?

$$\mathbb{K}(x, y) = \langle \Phi(x), \Phi(y) \rangle_{\mathbb{R}^{2^d}} \quad (9.15)$$

$$= \sum_{A \subset \mathcal{I}} \phi_A(x) \phi_A(y) \quad (9.16)$$

$$= \sum_{A \subset \mathcal{I}} \prod_{i \in A} x_i y_i. \quad (9.17)$$

This expression can be interpreted as the *weighted* sum of all paths $0 \rightarrow d$ in the directed graph below. (The weight for each path is the product of all edges in the path.)



If T_k is the weighted sum of all paths $0 \rightarrow k$, then

$$T_d = (1 + x_d y_d) T_{d-1} \quad (9.18)$$

By induction,

$$\mathbb{K}(x, y) = \prod_{j=1}^d (1 + x_j y_j) \quad (9.19)$$

Side note: Polynomial kernel is related but different.

$$\mathbb{K}(x, y) = \left(1 + \sum_{j=1}^d x_j y_j\right)^k \quad (9.20)$$

Polynomial kernel has higher order powers, e.g. $x_i^3 y_i^3$, etc.

The all-subsets kernel is a special case of a multiplicative kernel. Given $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_d$ and kernels $\mathbb{K}_j : \mathcal{X}_j \times \mathcal{X}_j \rightarrow \mathbb{R}$, you can define a new kernel $\mathbb{K} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ via:

$$\mathbb{K}(x, y) = \prod_{j=1}^d \mathbb{K}_j(x_j, y_j) \quad (9.21)$$

9.2.2 ANOVA kernel

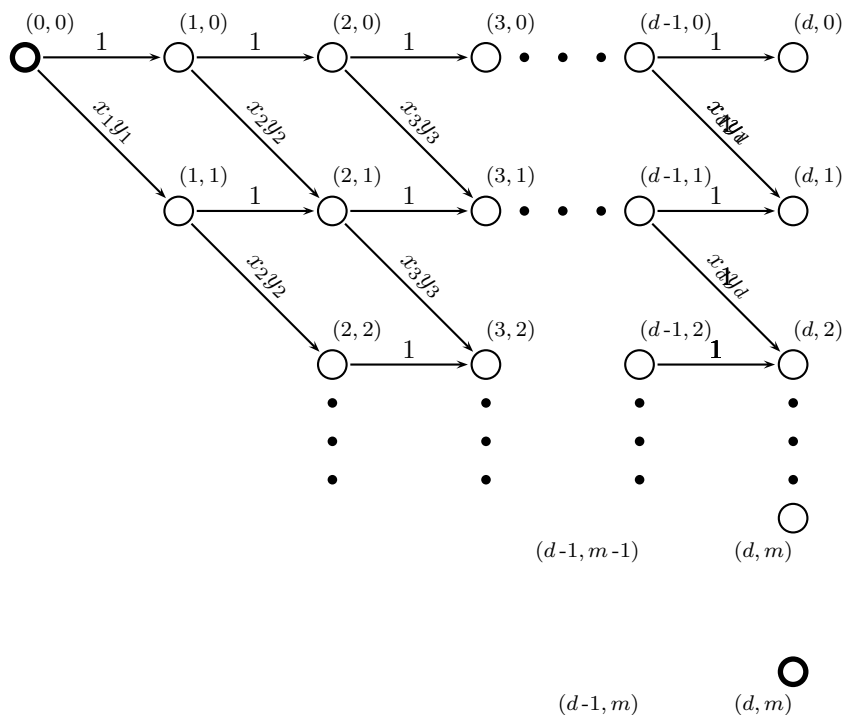
The all-subsets kernel can be generalized to other kinds of graph kernels. For an ANOVA kernel, restrict all-subsets kernel to subsets of cardinality exactly m , where $m \leq d$.

$$\text{Feature map } \Phi_m(x) = (\phi_A(x))_{|A|=m} \in \mathbb{R}^{\binom{d}{m}} \tag{9.22}$$

$$\mathbb{K}_{AN(m)}(x, y) = \sum_{|A|=m} \prod_{i \in A} x_i y_i \tag{9.23}$$

$$= \sum_{1 \leq i_1 < i_2 < \dots < i_m \leq d} \left[\prod_{l=1}^m x_{i_l} y_{i_l} \right] \tag{9.24}$$

This last expression can be interpreted as the weighted sum of all paths $(0, 0) \rightarrow (d, m)$ in the graph below. (Again the weight for each path is the product of all edges in the path.)



If T_s^r is the weighted sum of all paths $(0, 0) \rightarrow (r, s)$, then:

$$T_0^0 = 1 \tag{9.25}$$

$$T_{r+1}^r = 0 \tag{9.26}$$

$$T_{-1}^r = 0 \tag{9.27}$$

$$T_s^r = x_r y_r T_{s-1}^{r-1} + T_s^{r-1} \tag{9.28}$$

$\mathbb{K}_{AN(m)}(x, y) = T_m^d$ can be solved with dynamic programming in $O(dm)$ time. For more on the topic, see Scholkopf & Smola.

9.2.3 String kernels

(Haussler 1999) Frequently we want to compare strings (i.e. finite sequences of characters from some alphabet) for, say, document classification. The naive “bag-of-words” way is simply to count the number of occurrences of given substrings (e.g. words within a document). More sophisticated would be to compare all strings by all the ordered substrings they contain! Define:

$$\Sigma \triangleq \text{finite alphabet, e.g. } \{a \dots z\}. \quad (9.29)$$

$$\Sigma^n \triangleq \text{set of all strings of length } n. \quad (9.30)$$

$$\Sigma^* \triangleq \bigcup_{n=1}^{\infty} \Sigma^n = \text{set of all finite strings.} \quad (9.31)$$

$$\text{Given } s \in \Sigma^*, |s| \triangleq \text{length of } s = (s(1), \dots, s(|s|)) \quad (9.32)$$

$$\text{Given index set } \mathcal{I} \triangleq 1 \leq i_1 < i_2 < \dots < i_k \leq |s|, \quad (9.33)$$

$$S(\mathcal{I}) \triangleq (s(i))_{i \in \mathcal{I}} \quad (9.34)$$

$$l(S(\mathcal{I})) \triangleq \text{length of subsequence} = i_k - i_1 + 1 \quad (9.35)$$

(E.g. if $s = \text{chat}$, length of subsequence cat is 4.) Feature space is indexed by all strings of length n . Given $u \in \Sigma^n$,

$$(\Phi(s))_u = \sum_{\mathcal{I} | S(\mathcal{I})=u} \lambda^{l(\mathcal{I})}, \text{ where } \lambda \in (0, 1] \quad (9.36)$$

(E.g. if $\lambda = 0.5$, cat is stronger than $c \dots a \dots t$.)

Example: Let $n = 3$. In the case of $u = \text{asd}$,

$$\Phi(\text{Nasdaq})_{\text{asd}} = \lambda^3; \quad (9.37)$$

$$\Phi(\text{lasso dimension})_{\text{asd}} = 2\lambda^5, \text{ if not counting space.} \quad (9.38)$$

So for $\Phi : \Sigma^* \rightarrow \mathbb{R}^{|\Sigma^n|}$, we must compute, for strings s and t :

$$\mathbb{K}_n(s, t) = \sum_{u \in \Sigma^n} \Phi(s)_u \Phi(t)_u \quad (9.39)$$

END OF LECTURE