

# グラフスペクトルカーネルにおける算術オーバーフローの回避計算

Calculation of the Graph Spectra Kernel without Occurring Arithmetic Overflow

\*1中尾寛郎

Hiroo Nakao

\*2猪口明博

Akihiro Inokuchi

\*2鷺尾隆

Takashi Washio

\*1岡田孝

Takashi Okada

\*1関西学院大学

Kwansei Gakuin University

\*2大阪大学 産業科学研究所

The Institute of Scientific and Industrial Research, Osaka University

Kernel methods such as the SVM are becoming increasingly popular due to their high performance in graph classification. We proposed a kernel function, called the SPEC kernel, using graph spectra and the Interlace Theorem. Although the SPEC kernel was theoretically defined to classify graphs with large sets of vertices, arithmetic overflow occurs when calculating the SPEC kernel for the graphs in its implementation. In this paper, we propose a method for calculating the SPEC kernel without occurring arithmetic overflow. The fundamental performance of the method is evaluated using artificial datasets.

## 1. はじめに

近年、化学やコンピュータネットワークなどの分野で、グラフで表現されたデータを直接扱うことが急速に増加しつつある。このような状況の中で、与えられたグラフから特徴的な性質や共通部分構造などを発見するグラフマイニングが盛んに研究されており、クラスラベルをもつグラフを分類するグラフ分類問題は、グラフマイニングの分野において主要な問題となっている。グラフマイニングにおいて扱われるグラフの例は、化学分野における分子構造や、人を頂点、人間関係を辺で表したときの人間関係ネットワークなどである。グラフ分類の目的は、クラスラベルが既知のグラフからクラスラベルを予測するための規則を学習し、クラスラベルが未知のグラフのクラスラベルを予測することである。

分類問題を解く一つの解法として、優れた学習モデルである SVM (Support Vector Machine) がよく用いられる。SVM は、特徴空間に与えられたデータの特徴の分布を表す訓練データを 2 クラスに分類するための線形分離超平面を学習する手法である。SVM では、特徴空間でのデータ間の内積が計算できれば、最適な線形分離超平面を構成することができる。つまり、SVM を用いてグラフを分類するためには、2 グラフの内積値、すなわち類似度が必要となる。この類似度を計算するための関数をカーネル関数と呼ぶ。

類似度を求める代表的なカーネル関数として、Random Walk カーネルと近傍ハッシュカーネルがある [Kashima 2003, Hido 2009]。これらを用いた SVM は、数個の頂点からなる比較的小規模な部分グラフでクラスラベルが決まるデータには有効であるが、それ以上の頂点からなる部分グラフでクラスラベルが決まるデータには効果的でない。一方、後者のデータに対応するカーネル関数として、SPEC カーネル (graph SPECtra kernel) [Vihn 2010] が提案されている。SPEC カーネルは、グラフスペクトルと Interlace 定理 [Hwang 2004] を用いて定義されるカーネル関数である。SVM に用いられるカーネル関数は、原則として半正定である必要がある [Taylor 2004]、SPEC カーネルはその性質を満たす。SPEC カーネルは、理論上、頂点数が大きなグラフを扱うために提案されたカーネル関数であるが、頂点数が大きなグラフを扱う場合、実装上、計算過程に

において算術オーバーフローを起こすという課題があり、実世界で使用することが困難である。

本稿では、SPEC カーネルの算術オーバーフローを回避する SPEC カーネルの計算方法を提案する。提案手法では、SPEC カーネルの行列を、カーネル行列の対角要素で除算することで、算術オーバーフローを回避する。カーネル行列の対角要素で除算しても、カーネル関数は半正定であるという性質が失われないので、提案手法により計算される SPEC カーネルを、SVM のカーネル関数として用いることができる。

## 2. SPEC カーネルとその課題

グラフ分類問題とは、グラフ  $g_i$  とクラスラベル  $y_i \in \{+1, -1\}$  のペアの集合である訓練データ  $\{(g_i, y_i)\} (i = 1, \dots, n)$  から、グラフのクラスラベルを予測するための関数  $f$  を学習し、クラスラベルが未知のテストデータに含まれる各グラフのクラスを正しく予測することであり、グラフがある部分グラフを含むか含まないかで、そのグラフのラベルが  $+1$  か  $-1$  となる。

グラフ分類問題において、2 つのグラフ  $g_i$  と  $g_j$  の類似度を計算するためには、あるグラフが別のグラフの部分グラフと同型であるかどうかを判定する部分グラフ同型判定問題を解く必要がある。しかし、部分グラフ同型判定問題を解くための計算複雑さは、NP-完全 [Garey 1979] であることが知られており、頂点数の多いグラフを扱う場合、非常に大きな計算時間が必要となる。これに対して、カーネル関数の中には、部分グラフ同型問題を解くことなく、その類似度を計算できる手法がある。

一般に、実世界において、大きな部分グラフは小さな部分グラフに比べ、より多くの有用な情報を持つ。しかし、Random Walk カーネルや近傍ハッシュカーネルといった既存のグラフカーネルは、大きな共通部分グラフを含むかどうかによってクラスラベルが決定するグラフ分類問題について、良い結果を残さないことが文献 [Vihn 2010] により報告されている。その主な原因は、それぞれの頂点を特徴付ける際に、これらのグラフカーネルが、ごく少数の近接する頂点しか使用しない点にある。従って、既存のグラフカーネルが、大きな共通部分グラフを含むかどうかによってクラスラベルが決定するグラフに対応するには限界がある。この問題を軽減するために、SPEC カーネルという手法が提案された。SPEC カーネルはグラフスペクトルと Interlace 定理を用いて定義されるカーネル関数であ

り、次のように導かれる。初めに、

$$\gamma(g_i) = \{\gamma_1(g_i), \dots, \gamma_{|g_i|}(g_i)\}, \gamma(g_j) = \{\gamma_1(g_j), \dots, \gamma_{|g_j|}(g_j)\} \quad (1)$$

で与えられる2つのグラフ  $g_i$  と  $g_j$  を考える。ただし、 $\gamma_1(g_i) \leq \gamma_2(g_i) \leq \dots \leq \gamma_{|g_i|}(g_i)$ 、 $\gamma_1(g_j) \leq \gamma_2(g_j) \leq \dots \leq \gamma_{|g_j|}(g_j)$  であり、 $|g_i|$  はグラフ  $g_i$  の頂点数である。もし頂点数  $m$  の共通誘導部分グラフ  $g_s$  が  $g_i$  と  $g_j$  に含まれるならば、Interlace 定理により  $\gamma_h(g_s)$  の範囲は、

$$[\gamma_h(g_i), \gamma_{h+|g_i|-m}(g_i)], [\gamma_h(g_j), \gamma_{h+|g_j|-m}(g_j)] \quad (2)$$

の共通部分となる。ただし、 $1 \leq h \leq m \leq \min(|g_i|, |g_j|)$  とする。また、式 (2) は全ての  $h$  に対して互いに共通部分を持たなければならない。このことは、次の条件式が全ての  $h$  について共に成り立つときにのみ満たされる。

$$\gamma_h(g_i) \leq \gamma_{h+|g_j|-m}(g_j), \gamma_h(g_j) \leq \gamma_{h+|g_i|-m}(g_i) \quad (3)$$

更に、式 (3) より次式を導くことが出来る。

$$(\gamma_{h+|g_j|-m}(g_j) - \gamma_h(g_i)) (\gamma_{h+|g_i|-m}(g_i) - \gamma_h(g_j)) \geq 0 \quad (4)$$

また、式 (4) の指数をとって、次の不等式が得られる。

$$u(\cdot) = \exp(\gamma_{h+|g_j|-m}(g_j)\gamma_{h+|g_i|-m}(g_i) + \gamma_h(g_i)\gamma_h(g_j)) \times \exp(-\gamma_{h+|g_j|-m}(g_j)\gamma_h(g_j)) \exp(-\gamma_{h+|g_i|-m}(g_i)\gamma_h(g_i)) \geq 1 \quad (5)$$

$u(\cdot)$  は、下記のように置換することによって、 $u(\cdot) = \exp(\lambda'_{mh} \theta'_{mh}) \times (\phi_{\lambda_{mh}} \phi_{\theta_{mh}})$  と書き換えることが出来る。

$$\lambda'_{mh} = \begin{bmatrix} \gamma_h(g_i) \\ \gamma_{h+|g_i|-m}(g_i) \end{bmatrix}, \theta'_{mh} = \begin{bmatrix} \gamma_h(g_j) \\ \gamma_{h+|g_j|-m}(g_j) \end{bmatrix}, \\ \phi_{\lambda_{mh}} = \exp(-\gamma_{h+|g_i|-m}(g_i)\gamma_h(g_i)) \\ \phi_{\theta_{mh}} = \exp(-\gamma_{h+|g_j|-m}(g_j)\gamma_h(g_j))$$

文献 [Vihn 2010] に示される通り、 $u(\cdot)$  は半正定カーネルであるため、 $u(\cdot)$  をカーネル関数として以下のように書き換えることが出来る。

$$k'_{mh}(\lambda_{mh}, \theta_{mh}) = \exp(\lambda'_{mh} \theta'_{mh}) \times (\phi_{\lambda_{mh}} \phi_{\theta_{mh}}), \quad (6)$$

ここで、 $\lambda_{mh}$  と  $\theta_{mh}$  は次式で与えられる。

$$\lambda_{mh} = \begin{bmatrix} \gamma_h(g_i) \\ \gamma_{h+|g_i|-m}(g_i) \\ \gamma_h(g_i)\gamma_{h+|g_i|-m}(g_i) \end{bmatrix}, \theta_{mh} = \begin{bmatrix} \gamma_h(g_j) \\ \gamma_{h+|g_j|-m}(g_j) \\ \gamma_h(g_j)\gamma_{h+|g_j|-m}(g_j) \end{bmatrix}$$

$g_i$  と  $g_j$  が、頂点数  $m$  の共通誘導部分グラフ  $g_s$  を含む時に、式 (6) は大きな値を返す。さらに、 $g_i$  と  $g_j$  は  $m$  個以下の頂点で構成される  $g_s$  の共通誘導部分グラフを含むため、 $h = 1, \dots, m$  において、式 (6) は大きな値を返す。それゆえ、 $h = 1, \dots, m$

において、式 (6) の総乗を取ることで、次式の半正定カーネル関数を得る。

$$k_m(\lambda_m, \theta_m) = \prod_{h=1}^m k'_{mh}(\lambda_{mh}, \theta_{mh}), \quad (7)$$

ここで、 $\lambda_{mh}$  と  $\theta_{mh}$  は次式で与えられる。

$$\lambda_m = [\lambda_{m1}^T, \lambda_{m2}^T, \dots, \lambda_{mm}^T]^T, \theta_m = [\theta_{m1}^T, \theta_{m2}^T, \dots, \theta_{mm}^T]^T$$

式 (2) の各  $h$  の範囲が、少なくとも1つの  $m$  ( $m = 1, \dots, |g_s|, \dots, \min(|g_i|, |g_j|)$ ) について共通部分を持つ、すなわち  $g_i$  と  $g_j$  がある共通部分グラフ  $g_s$  を含む時に、 $m = |g_s|$  において式 (7) は大きな値を返す。さらに、 $g_i$  と  $g_j$  は  $|g_s|$  個以下の頂点で構成される  $g_s$  の共通誘導部分グラフを含むため、 $m = 1, \dots, |g_s|$  において、式 (7) は大きな値を返す。それゆえ、 $m = 1, \dots, \min(|g_i|, |g_j|)$  に対して、式 (7) の総和を取ることで、より大きな共通誘導部分グラフが存在すればより大きな値となる次式の半正定カーネル関数を得る。

$$s(g_i, g_j) = \sum_{m=1}^{\min(|g_i|, |g_j|)} k_m(\lambda_m, \theta_m), \quad (8)$$

ここで、 $\lambda$  と  $\theta$  は次式で与えられる。

$$\lambda = [\lambda_1^T, \lambda_2^T, \dots, \lambda_m^T]^T, \theta = [\theta_1^T, \theta_2^T, \dots, \theta_m^T]^T$$

また、式 (2) の各  $h$  の範囲が、全ての  $m$  ( $m = 1, \dots, H$ ) について共通部分を持つ、すなわち  $g_i$  と  $g_j$  が、頂点数  $H$  のある共通部分グラフ  $g_s$  を含む時に、 $m = H$  において式 (7) は大きな値を返す。さらに、 $g_i$  と  $g_j$  は  $H$  個以下の頂点で構成される  $g_s$  の共通誘導部分グラフを持つため、 $m = 1, \dots, H$  において、式 (7) は大きな値を返す。それゆえ、 $m = 1, \dots, H$  に対して、式 (7) の総乗を取ることで、より大きな共通誘導部分グラフが存在すればより大きな値となる次式の半正定カーネル関数を得る。

$$d(g_i, g_j) = \prod_{m=1}^H k_m(\lambda_m, \theta_m), \quad (9)$$

式 (8) と式 (9) の2つのカーネル関数を SPEC カーネルと呼び、これらを用いる SVM は比較的大きな頂点数を持つ部分グラフでクラスラベルが決まるデータには効果的である。しかし、式 (9) で示した SPEC カーネルは、 $m = 1$  から  $H$  にわたって  $k_m(\lambda_m, \theta_m)$  を総乗するため、より大きな頂点数を持つグラフを扱う場合に算術オーバーフローが起き、類似度を正確に計算することが出来ない問題がある。算術オーバーフローとは、コンピュータによる計算において、演算結果がレジスタの表せる範囲や記憶装置上の格納域に記録できる範囲を超えてしまう現象である。例えば、プログラム言語の Java を用いて実装することを考えた場合、Double 型の変数の取りうる最大値は、 $(2 - 2^{-52}) \cdot 2^{1023} = O(10^{308})$  であり、それ以上の値を扱うことはできない。そこで本稿では、SPEC カーネルの算術オーバーフローを回避するために、新たなカーネル関数 NSPEC を提案する。

### 3. 提案手法

図 1 で示す半正定値カーネルの性質を利用し、NSPEC を提案する。図 1 において、 $k_1$  と  $k_2$  をそれぞれ任意の半正定カー

$$\begin{aligned}
 k(x, y) &= \exp(k_1(x, y)), \\
 k(x, y) &= \mathbf{x}^T A \mathbf{y}, \\
 k(x, y) &= k_1(\mathbf{x}_a, \mathbf{y}_a) \times k_2(\mathbf{x}_b, \mathbf{y}_b), \\
 k(x, y) &= f(x) \times k_1(x, y),
 \end{aligned}$$

図 1: 半正定値カーネルの性質

- 0)  $\mathbf{d}_{new}(\gamma(g_i), \gamma(g_j))\{$
- 1)  $K = 0$
- 2) for  $m \in [1, \dots, H]$
- 3) for  $h \in [1, \dots, m]$
- 4)  $K = K - \frac{1}{2}(\lambda'_{mh} - \theta'_{mh})^2$
- 5) end for
- 6) end for
- 7)  $K = \exp(K)$
- 8) return  $K$
- 9) }

図 2:  $d_{new}(g_i, g_j)$  の疑似コード

ネル関数,  $x$ と $y$ をそれぞれ任意のベクトル,  $A$ を $|x| \times |y|$ の対称半正定値行列,  $|x|$ をベクトル $x$ の次元数,  $f$ を任意の関数とする. 更に,  $x = (x_a^T, x_b^T)^T$ は,  $x_a$ と $x_b$ を連結して得られるベクトルとする.  $k(x, y)$ が図1で示すいずれかの性質を満たすとき,  $k(x, y)$ は半正定値カーネルである. ここで, 式(9)に示されたSPECカーネルをそのカーネル行列 $K$ の対角要素 $\sqrt{d(g_i, g_i)}$ と $\sqrt{d(g_j, g_j)}$ で除算し, 次のように定義する.

$$\begin{aligned}
 d_{new}(g_i, g_j) &= \frac{d(g_i, g_j)}{\sqrt{d(g_i, g_i)}\sqrt{d(g_j, g_j)}} \quad (10) \\
 &= \exp\left(-\frac{1}{2} \sum_{m=1}^H \sum_{h=1}^m (\lambda_m - \theta_m)^2\right) \quad (11)
 \end{aligned}$$

式(11)の指数部は必ず0以下となるため,  $g_i = g_j$ の時に1となり,  $g_i \neq g_j$ に対して0より大きく1未満の値をとる. 従って, NSPECは算術オーバーフローを起こさない.

また, 式(9)に示されるSPECカーネルは半正定値カーネルであるため, 式(10)においてカーネル関数とスカラー $\frac{1}{\sqrt{d(g_i, g_i)}}$ の積で, 半正定値カーネルの性質を満たし,  $\frac{d(g_i, g_j)}{\sqrt{d(g_i, g_i)}}$ は半正定値カーネルである. 次に,  $\frac{d(g_i, g_j)}{\sqrt{d(g_i, g_i)}}$ とスカラー $\frac{1}{\sqrt{d(g_j, g_j)}}$ の積であるため, 半正定値カーネルの性質を満たし, 本稿で提案するNSPECは半正定値カーネルである. 従って, 半正定値性を満たすカーネル関数を必要とするSVMに, 本稿で提案するNSPECを用いることができる. 本稿で提案したNSPECの疑似コードを図2に示す.

表 1: データ生成プログラムのパラメータとその規定値

グラフの平均頂点数	埋込みグラフの頂点数の割合	グラフの辺存在確率	グラフのラベル数
$ g  = 100$	$p_v = 0.7$	$p = 5\%$	$ L  = 3$

表 2: グラフの平均頂点数  $|g|$  を変化させたときの正答率

$ g $	OPTSPEC1		OPTSPEC2		NSPEC	
	Training	Test	Training	Test	Training	Test
36	99%	74%	92%	88%	94%	87%
60	*	*	90%	64%	96%	91%
100	*	*	86%	74%	92%	89%
180	*	*	85%	72%	93%	85%
300	*	*	93%	90%	90%	75%

表 3: 埋込みグラフの頂点数の割合  $p_v$  を変化させたときの正答率

$p_v$	OPTSPEC1		OPTSPEC2		NSPEC	
	Training	Test	Training	Test	Training	Test
0.1	*	*	68%	60%	84%	48%
0.3	*	*	67%	57%	86%	66%
0.5	*	*	83%	72%	87%	66%
0.7	*	*	86%	74%	92%	89%
0.9	*	*	98%	96%	89%	83%

#### 4. 評価実験

本稿では, 式(11)に示したNSPECを用いたSVMの性能評価を行った. ここでは, 式(11), 式(9), 式(8)を用いるSVMを, それぞれNSPEC, OPTSPEC1, OPTSPEC2と記す. 上記3つのグラフカーネルの値よりカーネル行列を生成し, LIBSVM\*1というSVMパッケージを用いて10分割交差検証を行った. それぞれのグラフカーネルを用いるSVMの正答率は, 次に示す人工データによって評価を行った.

本稿の実験の人工データは, 表1に記載する4つのパラメータを用いて生成した. 各データセットは, 平均 $|g|$ 個の頂点数を持つ50個のグラフで構成される. 各グラフの2つの頂点の間の辺は, 辺存在確率 $p$ によって接続されるかどうかが決める. また,  $|L|$ 個のラベルの内の1つを各頂点と各辺に割り当てる. また, 平均 $p_v \times |g|$ 個の頂点数を持つ3つのグラフ $g_{s1}, g_{s2}, g_{s3}$ を, 共通誘導部分グラフとして埋め込む. はじめに,  $g_{s1}$ を50個のグラフの内の25個のグラフにランダムに埋め込む. 続いて,  $g_{s2}$ と $g_{s3}$ も同様に, 50個のグラフの内の25個のグラフにランダムに埋め込む. そして, 最後に埋め込んだ $g_{s3}$ を含むグラフのクラスラベルを1とし, それ以外のグラフのクラスラベルを-1とする. また, 予備実験として, 式(11)により与えられるNSPECの $H$ の値を変化させ実験を行なったところ,  $H = |g| \times 0.1$ で最良の正答率が得られたため,  $H = |g| \times 0.1$ として実験を行った. 実験結果を表2から表5に示す. なお, この実験結果は, 同じパラメータで値の異なる人工データを5種類生成し, 各々のデータに対する正答率の平均をとったものである. また, "\*"は, 計算途中で算術オーバーフローを起こし, 結果が得られなかったことを示している.

表 4: グラフのラベル数  $|L|$  を変化させたときの正答率

$ L $	OPTSPEC1		OPTSPEC2		NSPEC	
	Training	Test	Training	Test	Training	Test
1	*	*	83%	55%	89%	63%
2	*	*	83%	74%	92%	79%
3	*	*	84%	70%	92%	89%
4	*	*	86%	74%	96%	88%
5	*	*	98%	98%	99%	96%

表 5: グラフの辺存在確率  $p$  を変化させたときの正答率

$p$	OPTSPEC1		OPTSPEC2		NSPEC	
	Training	Test	Training	Test	Training	Test
2.5	*	*	93%	80%	94%	87%
5	*	*	84%	70%	85%	68%
10	*	*	92%	92%	92%	89%
20	*	*	94%	80%	88%	79%

\*1 <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

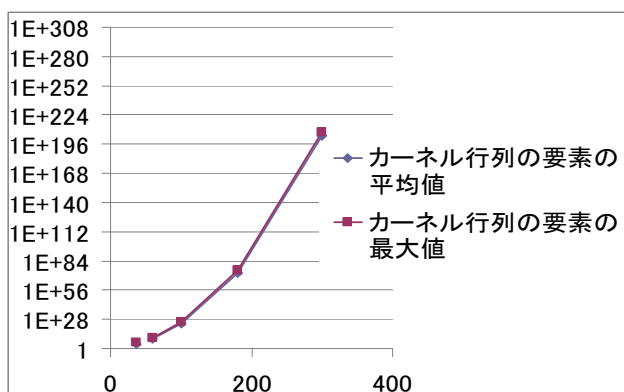


図 3: カーネル行列の要素の最大値と平均値

表 2 から表 5 は, 各表に示されるパラメータ以外を既定値のまま, 表に示されるパラメータを変化させて実験データセットを生成した場合の結果である. 表 2 から表 5 で示した実験結果より, OPTSPEC1 は,  $|g| = 36$  以外で算術オーバーフローを起こし, 結果が得られなかった. 一方, 本稿の提案手法は, 算術オーバーフローを起こさず, なおかつ OPTSPEC2 と同等の正答率を持つことが示された.

図 3 は, 式 (8) を用いたカーネル関数によるカーネル行列の要素の最大値と平均値を表したグラフである. この図の横軸はグラフのサイズ  $|g|$  であり, 縦軸はカーネル関数の値である. 表 2 で示したとおり, 本実験で行なった  $|g| = 300$  までは, OPTSPEC2 は算術オーバーフローを起こさなかった. しかし,  $|g| = 500$  程度の大きさのグラフでは, カーネル行列の要素の最大値と平均値は, Double 型の変数の取りうる最大値  $(2 - 2^{-52}) \cdot 2^{1023} = O(10^{308})$  を超え, 算術オーバーフローを起こす. 一方, NSPEC の OPTSPEC2 に対する優位性は, OPTSPEC2 が図 3 で示したように, 約 400 頂点のサイズのグラフにしか適用できないのに対し, NSPEC はサイズの大きなグラフにも適用できる点である.

## 5. まとめと今後の課題

本稿では, SPEC カーネルの算術オーバーフローを回避する SPEC カーネルの計算方法を提案した. 提案手法では, SPEC カーネルの行列をカーネル行列の対角要素で除算することで, 算術オーバーフローを回避する計算方法を提案した. また, その手法を実装し, 様々な人工データに適用することで本稿の NSPEC の有効性を評価・検証した.

本稿の評価実験は, 人工データについて, 既存の 2 種のグラフカーネルと NSPEC を用いた SVM の正答率比較を行った. NSPEC を用いた SVM の正答率は, 実験で用いた人工のデータセットに対し, SPEC カーネルを用いた SVM と同等の正答率が得られ, また, NSPEC は算術オーバーフローを起こさないことを確認した. 今後の課題は, 実世界で収集されたデータセットに対し, NSPEC を用いた SVM の正答率が, SPEC カーネルを用いた SVM と同等の正答率が得られることを検証することである.

## 参考文献

- [Vihn 2010] N. Vinh, et al. Graph Classification Based on Optimizing Graph Spectra. *Proc. of Int'l Conf. on Discovery Science*, pp.205-220 (2010)
- [Hwang 2004] S. Hwang. Cauchy's Interlace Theorem for Eigenvalues of Hermitian Matrices. *American Mathematical Monthly*, Vol.111, pp.157-159 (2004)
- [Kashima 2003] H. Kashima, et al. Marginalized Kernels Between Labeled Graphs. *Proc. of Int'l Conf. on Machine Learning*, pp.321-328 (2003)
- [Scholkopf 2002] B. Scholkopf, et al. Learning with Kernels. *MIT Press* (2002)
- [Hido 2009] S. Hido, et al. A Linear-Time Graph Kernel. *Proc. of Int'l Conf. on Data Mining*, pp.179-188 (2009)
- [Garey 1979] M. Garey, et al. Computers and Intractability. A Guide to the Theory of NP-Completeness, W. H. Freeman (1979)
- [Taylor 2004] J. Taylor, et al. Kernel Methods for Pattern Analysis. *Cambridge University Press* (2004)
- [元田 2006] 元田ら. データマイニングの基礎 (IT Text), 共立出版, pp.96-109 (2006)