

## 1. RBF ネットワーク<sup>1</sup>

RBF ネットワークでは基底関数として一般にガウス関数が用いられる, その重ね合わせにより応答局面は

$$\hat{y}(x) = \sum_{j=1}^m w_j h_j(x)$$

で表され,  $m$  は中間層素子数,  $w_j$  は重みを表す。また  $h_j(x)$  は基底関数であり, 次式で与えられる。

$$h_j(x) = \exp\left(-\frac{(x-x_j)^T(x-x_j)}{r_j^2}\right)$$

上式において  $x_j$  と  $r_j$  はそれぞれ  $j$  番目の基底関数の中心と半径である。

学習用データ  $x_i$  と対になる教師データ  $y_i (i=1,2,\dots,p)$  とすると, RBF ネットワークにおける学習は次式を最小化する問題となる。

$$E = \sum_{i=1}^p (y_i - \hat{y}(x_i))^2 + \sum_{j=1}^m \lambda_j w_j^2 \rightarrow \min$$

この第2項の  $\lambda_j$  は, 一部の素子が過剰に反応するのを避けるための重みに対する抑制パラメータである。RBF ネットワークの学習とは,  $E$  を最小にする重みベクトル  $w$  をみつけることであり, 逆行列計算に帰着される。(詳細は省略)

## 2. 基底関数の半径

RBF ネットワークにおいて, パラメータ  $x_j$  および  $r_j$  の値を適切に定めることは重要である。ここでは, 各学習点  $x_i (i=1,2,\dots,p)$  の位置に基底関数を置くことにする。さらに  $r_j$  および  $\lambda_j$  に関しては, 通常, 交差検証法(cross validation)によって定めるが, 簡易法として以下の式<sup>2</sup>が提案されている。

$$r = \frac{d_{\max}}{\sqrt[n]{nm}} \quad (1)$$

ただし,  $n$  は設計変数の数,  $m$  はサンプル点数,  $d_{\max}$  は基底関数の中心間の最大距離

基底毎に異なる半径にする北山氏の提案したのが以下の式です。

$$r_j = \frac{d_{j,\max}}{\sqrt[n]{n}\sqrt{m-1}} \quad (2)$$

ここで,  $d_{j,\max}$  は,  $j$  番目のサンプル点からの最大距離

この基底関数ごとに半径を適用するのは, サンプル点の疎密が生じるような場合に有効であると北山氏が記述しています。

サンプル間の距離を使用しますので, 各設計変数は, 設計上下限值や最大値・最小値等でス

<sup>1</sup> RBF ネットワークによる逐次近似最適化 北山哲士 他 2010 日本機械学会

<sup>2</sup> 多目的最適化と工学設計 注 4.2.1 中山弘隆 他 現代図書

ケーリングをすることが重要です。

### 3. APC—Ⅲアルゴリズム

基底関数の中心と半径を同時に計算するアルゴリズムで、NTT データ数理システムで開発・販売している VMstudio の、Radial Basis Function Network 機能に採用されている方法<sup>3</sup>です。サンプル点を、指定半径内にクラスタリングする方法のようです。クラスタリング数は、指定半径の大きさにより決まり、各クラスター内の点の中心点とそのクラスターに属するサンプル点から、そのクラスターの半径を計算します。割と簡単な方法で、サンプル点数のワンパスでのクラスタリング方法です。詳しくは柱脚文献を参考して下さい。

指定半径  $R_0$  は、サンプル点間の最小距離の平均に  $\alpha$  倍（ユーザー指定、省略値 1.2）して求めます。

$$R_0 = \alpha \frac{1}{P} \sum_{i=1}^P \min_{i \neq j} (\|x_i - x_j\|)$$

ここで、 $P$  はサンプル点数

### 4. 例

R(S-PLUS)のライブラリーMASS 内の TOPO データセットの 1 列(x)と 2 列(y)を使用する。 サンプル数 52

図 4-1 は、サンプル点の散布図です。右図は、最小値と最大値で[0,1]にスケール化した図です。以降このスケール化したデータを使います。図 4-2 の左図は、基底関数の中心をサンプル点にした時の式(1)の半径を黒色の円で、式(2)の半径を橙色の円で表示しています。図 4-2 の右図は、APC—Ⅲアルゴリズムで計算した基底関数の中心を橙点で、各中心点での半径（分散から計算）の円を黒色で表示しました。

式(1)の半径は、0.1319119

式(2)の半径の平均値は、0.1038159

APC—Ⅲアルゴリズムでの平均値は、0.1662552  
です。

注) VMStudio での基底関数の中心と分散から RBF の係数を計算する方法は、1 章 RBF ネットワークで記述した方法と異なりますので、図 4-2 の右図の半径とは、異なるかもしれません。

---

<sup>3</sup> Hwang Young-Sup and Bang Sung-Yang. An Efficient Method to Construct a Radial Basis Function Neural Network Classifier. Neural Networks, 10, 8, pp. 1495--1503 (1997)

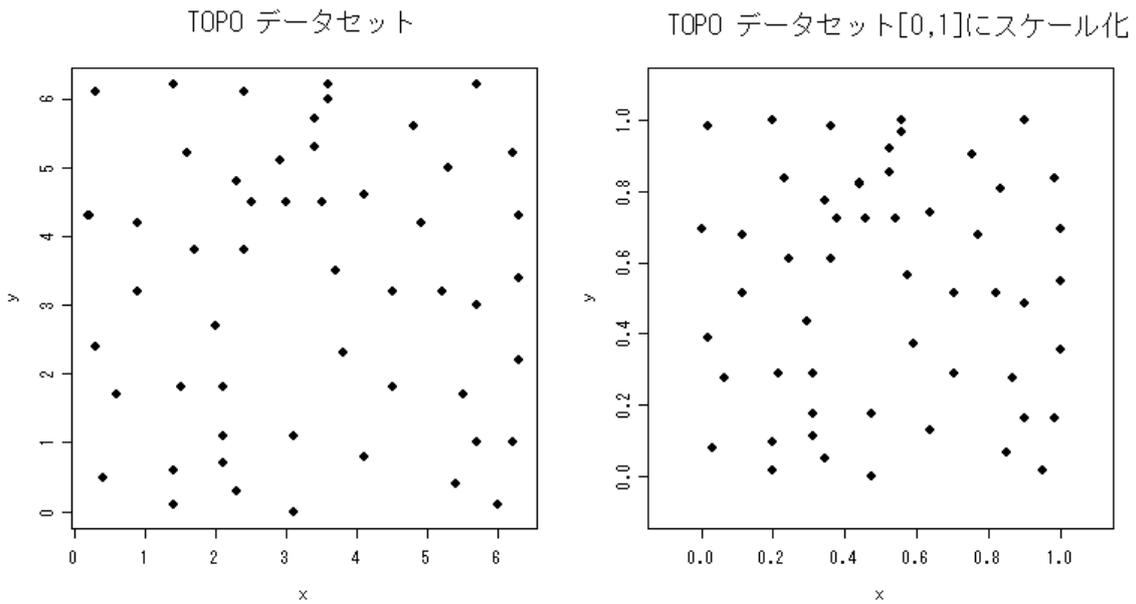


図 4-1 TOPO データセットの散布図

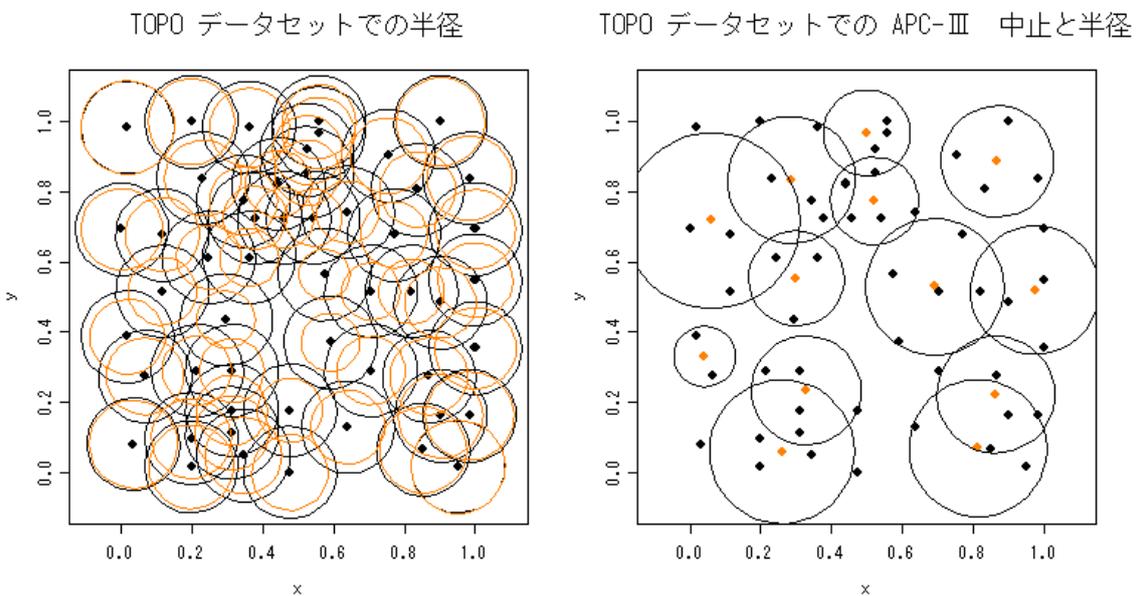


図 4-2 TOPO データセットでの基底関数の中心と半径

### 5. サポートベクターマシン(SVM)のガウシアンカーネル

SVM ガウシアンカーネルは、一般には

$$K(x, x') = \exp\left(-\frac{\|x - x'\|^2}{r^2}\right)$$

と表示され、一見 RBF ネットワークと同じに見えるが、基底を置く場所がサポートベクトルの点として自動的に出てくるところが大きく異なりますが、パラメータ  $r$  の指定方法が各種パッケージにより異なります。ここでは、このパラメータの指定方法の違いおよびどのよ

うな値を設定したら良いか省略時の値を含め RBF の半径と関連づけて検討します。

### 5. 1 各パッケージ (ライブラリー) でのパラメータ

#### (1) LIBSVM(A Library for Support Vector Machines)<sup>4</sup>

svm-train.exe プログラムの `-g` オプションで  $\exp(-\gamma \cdot |u-v|^2)$  の  $\gamma$  を指定

#### (2) R kernlab ライブラリー

ksvm 関数の kpar 引数で

$k(x, x_0) = \exp(-\sigma \cdot \|x - x_0\|^2)$  の  $\sigma$  を指定

#### (3) R e1071 ライブラリー

svm 関数の引数 gamma で

$\exp(-\gamma \cdot |u-v|^2)$  の gamma を指定

#### (4) S-PLUS svm ライブラリー

svm 関数の引数 gamma で

$\exp(-\gamma \cdot |u-v|^2)$  の gamma を指定

#### (5) VMStudio

$\exp\left(-\frac{|x-x'|^2}{\sigma^2}\right)$  の  $\sigma^2$  を指定

LIBSVM の `-g` オプションの  $\gamma$ , R(kernlab)の  $\sigma$ , R(e1071)の  $\gamma$ , S-PLUS(SVM)の  $\gamma$  は, すべて同じである。

### 5. 2 各パッケージ (ライブラリー) のパラメータの省略値

#### (1) LIBSVM(A Library for Support Vector Machines)

省略値:  $1/n$  ただし,  $n$  は設計変数の数

#### (2) R kernlab ライブラリー

省略値: sigest 関数で推定された値 詳しくは後述

`sigma=mean(sigest(x,scaled=FALSE)[c(1,3)]`

#### (3) R e1071 ライブラリー

省略値:  $1/n$  ただし,  $n$  は設計変数の数

#### (4) S-PLUS svm ライブラリー

省略値:  $1/n$  ただし,  $n$  は設計変数の数

#### (5) VMStudio

省略値: 0.1

R kernlab 以外はサンプル点の設計変数のスケール方法に関係なく固定値である。次に各パッケージ (ライブラリー) でのスケールの方法に関して述べる。

### 5. 3 各パッケージ (ライブラリー) のスケーリング

#### (1) LIBSVM(A Library for Support Vector Machines)

学習(svm-train.exe)前に, svm-scale.exe でスケーリングを行う。

---

<sup>4</sup> Chih-Chung Chang and Chih-Jen Lin

<http://www.csie.ntu.edu.tw/~cjlin/libsvm/index.html>

libsvm-3.1.zip

svm-scale -l lower -u upper ...

[lower, upper]にスケールする。-l の省略値は -1, -u の省略値は+1

スケールリングのオプションを指定しない時は、各設計変数は、データの最小値・最大値で[-1,1]にスケールされます。

(2) R(kernlab)ライブラリー, R(e1071)ライブラリー, S-PLUS(svm)ライブラリー

(k)svm 関数の scaled 引数でスケールリング処理の有無を指定

TRUE 指定でスケールリングする, FALSE 指定でスケールリングなし。scaled 指定がない場合は、スケールリングする。スケールリングは、R(S-PLUS)の scale 関数を使用しています。設計変数毎に平均値と標準偏差で標準化します。(スケールリング後 平均値が0, 標準偏差が1となる) スケールリング後の範囲は理論上 $[-\infty, +\infty]$ です。

(3) VMStudio

内部で自動スケール 最小値0, 最大値1 のようです。(要確認)

上記3種類で、スケールリング方法が違います。またパラメータの省略値はスケールリングをした場合を想定した固定値ですが、R(kernlab)だけが特殊です。

#### 5. 4 R(kernlab)でのパラメータの省略値

sigest 関数を使用して,

```
kpar <- list(sigma=mean(sigest(x,scaled=FALSE)[c(1,3)])
```

で設定しています。

以下は、sigest 関数のヘルプからです。

sigest

sigest estimates the range of values for the sigma parameter which would return good results when used with a Support Vector Machine (ksvm). The estimation is based upon the 0.1 and 0.9 quantile of  $|x - x'|^2$ . Basically any value in between those two bounds will produce good results.

Value

Returns a vector of length 3 defining the range (0.1 quantile, median and 0.9 quantile) of the sigma hyperparameter.

下記の srange の値が sigest 関数の戻り値です。

```
srange <- 1/quantile(dist[dist!=0],probs=c(0.9,0.5,0.1))
```

ここで、dist は、サンプル点からランダムに重複ありで  $m(=\text{frac} \cdot n)$ 個のサンプリングを2組作成し、それぞれ対応するデータ間の距離の二乗(各変数の差の二乗和)のベクトルです。frac はサンプリング比率(省略値 0.5), n はサンプル数です。

m 個のデータ間距離の二乗ベクトルの 10%分位点と 90%分位点のそれぞれの逆数の平均値を sigma としています。

この dist は、ガウシアンカーネルの一般式の  $r^2$  に相当します。

注) 中央値や平均値を使用せずに、10%分位点と 90%分位点を採用しています。

また、dist の平均値の逆数の平均値でないのが特徴です。

採用した sigma に対応する  $r^2$  は、dist の 10%分位点と 90%分位点の算術平均ではありません。調和平均です。

図5-1は、4章 例 で使用した TOPO データセットで、sigest 関数により 1000 回計算した sigma 値の分布図です。左図は、R(S-PLUS)で採用されているスケールリングでの結果で、右図は[0,1]でのスケールリング結果です。表5-1は、1000 データの要約(summary)である。R(e1071)ライブラリーS-PLUS(svm)の省略値  $0.5(1/n)$ に比べて、大きめの値が多いようです。[0,1]のスケールリングでは、全体にかなり大きな値なので、sigest 関数による省略値の使用は、scaled=TRUE 指定で、平均値・標準偏差による標準化指定と併用するのが

良いでしょう。但し交差検証法(cross validation)の時の  $\sigma(\gamma)$  の範囲の参考にはなると思います。

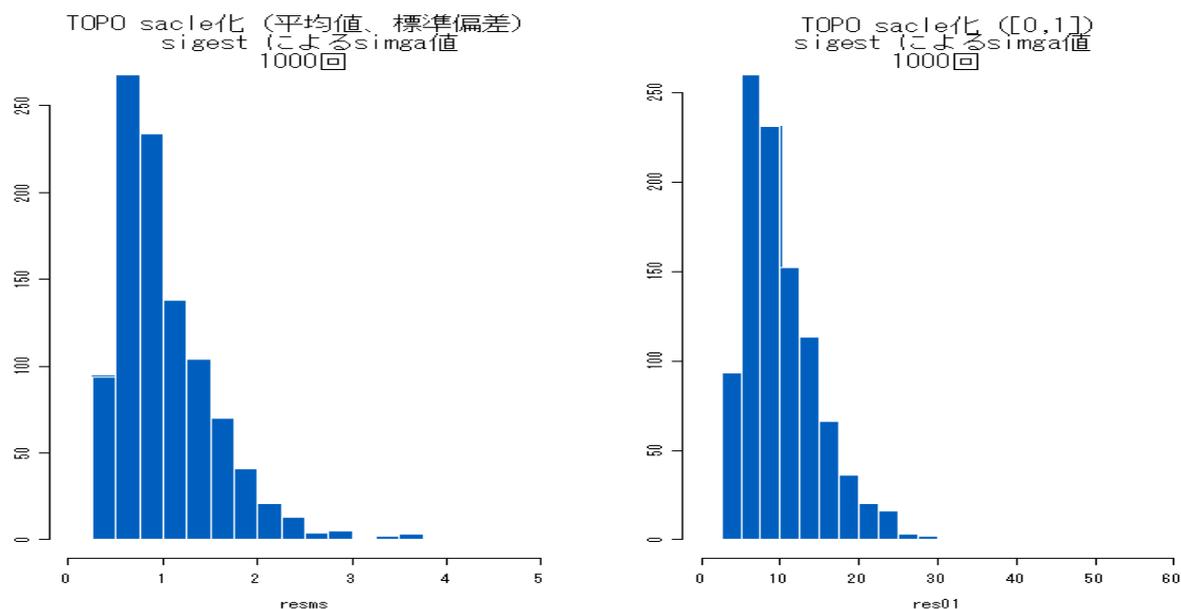


図 5-1 TOPO データセットでの sigest 関数による  $\sigma$  値の分布

表 5-1 TOPO データセットでの sigest 関数による  $\sigma$  値の要約(summary)

|        | Min.  | 1st Qu. | Median | Mean   | 3rd Qu. | Max    |
|--------|-------|---------|--------|--------|---------|--------|
| 標準化    | 0.273 | 0.655   | 0.906  | 1.032  | 1.281   | 4.861  |
| [0,1]化 | 2.320 | 6.512   | 9.040  | 10.241 | 12.667  | 57.281 |

以下は  $r = 1/\sqrt{\sigma}$  変換による 1000 個の結果です。

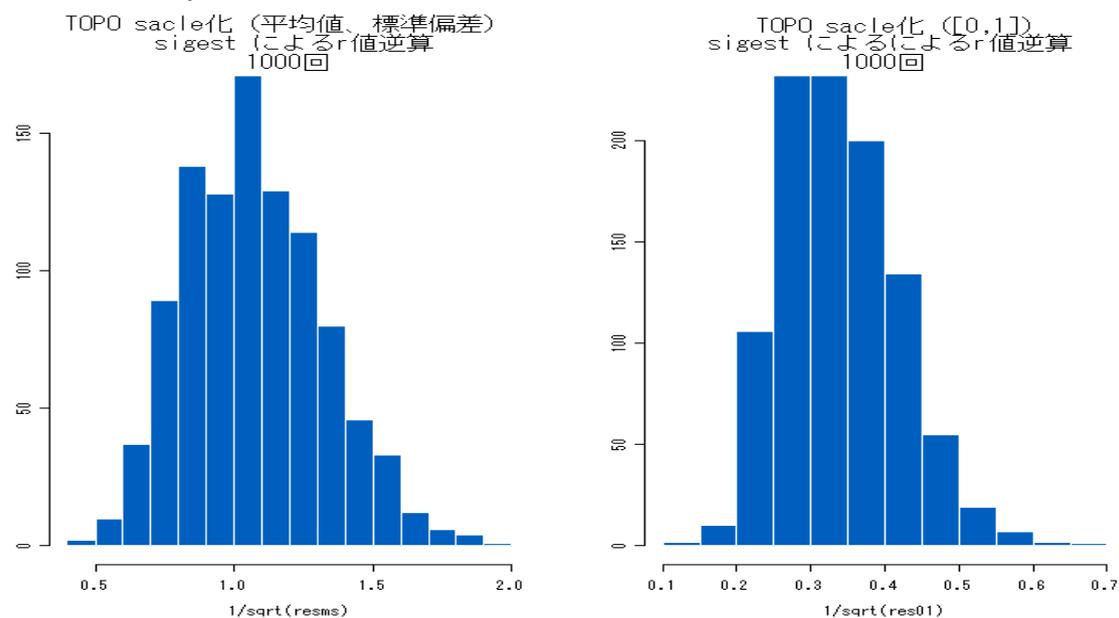


図 5-2 TOPO データセットでの sigest 関数による  $r$  値の分布

表5-2 TOPO データセットでの sigest 関数による r 値の要約(summary)

|        | Min.  | 1st Qu. | Median | Mean  | 3rd Qu. | Max   |
|--------|-------|---------|--------|-------|---------|-------|
| 標準化    | 0.454 | 0.883   | 1.050  | 1.072 | 1.236   | 1.914 |
| [0,1]化 | 0.132 | 0.281   | 0.333  | 0.339 | 0.392   | 0.656 |

4章 例 での[0,1]でのサンプリングで検討した半径 r と比較すると、少し大きめの値になっているようです。

sigest 関数では、 $m(=frac*n)$ 個のサンプリングで計算していますので、関数を使用する毎に結果が異なりますが、全データを使用すると距離計算に時間がかかりますので、サンプル数が大きい時には有効です。ここで利用した例では、全データ間の距離が割と楽に計算出来ます。表5-3は、全データを使用しての距離(r)の二乗(dist)の分布の10%分位点と90%分位点、中央値及び10%分位点と90%分位点から計算した sigma と半径(r)の値です。

表5-1の sigma の Mean, 表5-2の r の Mean と大差がないです。表5-1, 2の値は、全体データからのサンプリングでもありますので、当然かもしれません。

表5-3 TOPO データセットでの全データでの sigma 値と r 値

|        | 90%   | 50%   | 10%   | sigma  | r     |
|--------|-------|-------|-------|--------|-------|
| 標準化    | 8.716 | 3.269 | 0.451 | 1.166  | 0.926 |
| [0,1]化 | 0.853 | 0.324 | 0.045 | 11.697 | 0.292 |

以上