

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4635088号
(P4635088)

(45) 発行日 平成23年2月16日(2011.2.16)

(24) 登録日 平成22年11月26日(2010.11.26)

(51) Int.Cl.

G06F 17/17 (2006.01)

F 1

G06F 17/17

請求項の数 10 (全 22 頁)

(21) 出願番号	特願2008-533701 (P2008-533701)
(86) (22) 出願日	平成18年9月28日 (2006.9.28)
(65) 公表番号	特表2009-510633 (P2009-510633A)
(43) 公表日	平成21年3月12日 (2009.3.12)
(86) 國際出願番号	PCT/US2006/038199
(87) 國際公開番号	W02007/038765
(87) 國際公開日	平成19年4月5日 (2007.4.5)
審査請求日	平成21年9月24日 (2009.9.24)
(31) 優先権主張番号	60/721,753
(32) 優先日	平成17年9月28日 (2005.9.28)
(33) 優先権主張国	米国(US)
(31) 優先権主張番号	11/535,921
(32) 優先日	平成18年9月27日 (2006.9.27)
(33) 優先権主張国	米国(US)

早期審査対象出願

最終頁に続く

(54) 【発明の名称】 LOO誤差を用いた分類または回帰

(57) 【特許請求の範囲】

【請求項 1】

データのトレーニングセットを用いて、与えられた新しい点に対応する値を予測するための関数を構築するマシン学習を行なうための正則化最小二乗分類または回帰の処理を行なう R L S 分類器 / 回帰論理回路と、メモリと、を少なくとも備えた正則化最小二乗分類または回帰システムにおいて、前記正則化最小二乗分類または回帰の処理を行なうための方法であって、

前記 R L S 分類器 / 回帰論理回路は、少なくとも、入力回路と、カーネル行列生成回路と、固有値分解 / SVD 回路と、係数演算回路と、LOO誤差演算回路と、正則化パラメータ選択回路とで構成され、

前記方法は前記 R L S 分類器 / 回帰論理回路によって行なわれ、

前記方法は、

前記入力回路が、前記データのトレーニングセットを受信するステップと、

前記カーネル行列生成回路が、 $K_{ij} = K(\bar{x}_i, \bar{x}_j)$ を満たし、 $K(\bar{x}_i, \bar{x}_j) = \exp\left(\frac{\|\bar{x}_i - \bar{x}_j\|^2}{2\sigma^2}\right)$ の形

式を有する $n \times n$ 行列 K であり、この場合の x は、前記トレーニングセットに含まれるデータ点のベクトルであり、 σ はユーザ選択の帯域幅パラメータであるような、カーネル行列 K を前記トレーニングセットに基づいて生成するステップと、

前記固有値分解 / SVD 回路が、前記カーネル行列 K の固有値分解を演算するステップと、

10

前記係数演算回路が、複数の正則化パラメータを受信し、

前記カーネル行列 K の前記固有値分解 $K = Q Q^t$ を用いて、前記正則化パラメータの各々について、 n は前記トレーニングセットのデータ点の数を表し、 I は標準的な単位行列を表し、 y は新しい点に対応する値を表すような線形システム $(K + nI)c = y$ に従って係数 c を演算するステップと、

前記L00誤差演算回路が、前記正則化パラメータの各々についてのL00誤差を、 n は前記トレーニングセットの d 次元におけるデータ点の数であるような $O(n^3 + n^2d)$ 時間と $O(n^2)$ 空間において、正規化カーネル行列 G と前記求めた係数 c とを用いて $c/\text{diag}(G^{-1})$ に従って演算するステップと、

前記正則化パラメータ選択回路が、前記L00誤差の最小値を有する前記正則化パラメータを選択するステップと、

20

前記RLS 分類器 / 回帰回路が、前記選択した正則化パラメータと前記トレーニングセットに基づいて、 X は $n \times d$ のデータ行列を表すような $w = X^t c$ で定義される超平面関数 w を決定し、

前記決定した超平面関数 w を前記メモリに記憶するステップと

を含むことを特徴とする正則化最小二乗分類または回帰の処理を行うための方法。

【請求項 2】

前記超平面関数 w を前記正則化最小二乗分類で用い、

前記正則化最小二乗分類は、係数 c または超平面関数 w の少なくとも 1 つに基づいて、入力されたデータ点のセットに対応するクラス y を予測することをさらに含むことを特徴とする請求項 1 に記載の正則化最小二乗分類または回帰に関する方法。

30

【請求項 3】

前記カーネル行列を前記メモリに記憶することをさらに含むことを特徴とする請求項 1 に記載の正則化最小二乗分類または回帰の処理を行うための方法。

【請求項 4】

データのトレーニングセットを用いて、与えられた新しい点に対応する値を予測するための関数を構築するマシン学習を行なうための正則化最小二乗分類または回帰の処理を行なう RLS 分類器 / 回帰論理回路と、メモリと、を少なくとも備えた正則化最小二乗分類または回帰システムであって、

前記 RLS 分類器 / 回帰論理回路は、少なくとも、入力回路と、カーネル行列生成回路と、固有値分解 / SVD 回路と、係数演算回路と、L00誤差演算回路と、正則化パラメータ選択回路とで構成され、

40

前記入力回路は、前記データのトレーニングセットを受信し、

前記カーネル行列生成回路は、 $K_{ij} = K(\bar{x}_i, \bar{x}_j)$ を満たし、 $K(\bar{x}_i, \bar{x}_j) = \exp\left(\frac{\|\bar{x}_i - \bar{x}_j\|^2}{2\sigma^2}\right)$ の形

式を有する $n \times n$ 行列 K であり、この場合の x は、前記トレーニングセットに含まれるデータ点のベクトルであり、 σ はユーザ選択の帯域幅パラメータであるような、カーネル行列 K を前記トレーニングセットに基づいて生成し、

前記固有値分解 / SVD 回路は、前記カーネル行列 K の固有値分解を演算し、

前記係数演算回路は、複数の正則化パラメータ λ を受信し、前記カーネル行列 K の前記固有値分解 $K = Q \Lambda Q^T$ を用いて前記正則化パラメータ λ の各々について、 n は前記トレーニングセットのデータ点の数を表し、 I は標準的な単位行列を表し、 y は新しい点に対応する値を表すような線形システム $(K + nI)c = y$ を解くことで係数 c を演算し、

前記LOO誤差演算回路は、前記正則化パラメータ λ の各々についてのLOO誤差を、 n は前記トレーニングセットの d 次元におけるデータ点の数を表すような $O(n^3 + n^2d)$ 時間と $O(n^2)$ 空間において、 G は正規化カーネル行列であるような $c/\text{diag}(G^{-1})$ と前記求めた係数 c とに基づいて演算し、

前記正則化パラメータ選択回路は、前記LOO誤差の最小値を有する前記正則化パラメータ λ を選択し、

前記RLS 分類器 / 回帰回路は、前記選択した正則化パラメータ λ と前記トレーニングセットに基づいて、 X は $n \times d$ のデータ行列を表すような $w = X^T c$ で定義される超平面関数 w を決定し、

前記決定した超平面関数 w を前記メモリに記憶する

ことを特徴とする正則化最小二乗分類または回帰システム。

【請求項 5】

前記超平面関数 w を前記正則化最小二乗分類で用い、

前記正則化最小二乗分類は、係数 c または超平面関数 w の少なくとも 1 つに基づいて、入力されたデータ点のセットに対応するクラス y を予測する予測モデルをさらに備えることを特徴とする請求項 4 に記載の正則化最小二乗分類または回帰システム。

【請求項 6】

前記カーネル行列を記憶するメモリをさらに備えることを特徴とする請求項 4 に記載の正則化最小二乗分類または回帰システム。

【請求項 7】

前記入力されたデータ点および前記データのトレーニングセットは、顔認識データ、対象物追跡データ、室内ナビゲーションデータ、医療画像解析データおよび音声認識データからなる群のいずれか 1 つを表すことを特徴とする請求項 2 に記載の正則化最小二乗分類または回帰に関する方法。

【請求項 8】

前記超平面関数 w を前記正則化最小二乗分類で用い、さらに、係数 c および前記超平面関数 w の少なくとも 1 つに基づいて、入力されたデータ点のセットに対応する値のセットを予測することを特徴とする請求項 1 に記載の正則化最小二乗分類または回帰に関する方法。

【請求項 9】

前記入力されたデータ点および前記データのトレーニングセットは、顔認識データ、対象物追跡データ、室内ナビゲーションデータ、医療画像解析データおよび音声認識データからなる群のいずれか 1 つを表すことを特徴とする請求項 5 に記載の正則化最小二乗分類または回帰システム。

【請求項 10】

前記超平面関数 w を正則化最小二乗分類で用い、さらに、係数 c および前記超平面関数 w の少なくとも 1 つに基づいて、入力されたデータ点のセットに対応する値のセットを予測

10

20

30

40

50

することを特徴とする請求項 4 に記載の正則化最小二乗分類または回帰システム。

【発明の詳細な説明】

【技術分野】

【0001】

本願は、米国特許仮出願第 60 / 721753 号（出願日：2005 年 9 月 28 日）発明の名称：「Making Regularized Least Squares Practical」および米国特許出願第 11 / 535921 号 発明の名称：「Regularized Least Squares Classification/Regression」の優先権を主張するものであり、これら出願のすべての記載をここに引用するものである。

【0002】

本発明は、マシン学習（Machine Learning）に関し、具体的には、正則化最小二乗法の分類 / 回帰に関する。

【背景技術】

【0003】

ここに d 次元の n 個のデータ点からなる $S = (\bar{x}_1, y_1), (\bar{x}_2, y_2), \dots, (\bar{x}_n, y_n)$ のデータセットがあるとすると、機能的学習タスクでは、新しい点 \bar{x} が与えられると、これに対応する y 値を予測する関数 $f(\bar{x})$ を生成する。このような問題を解く一般的なフレームワークとしては、以下のようない最小化問題を解くための Tikhonov 正則化（Evgeniou ら、2000 年）がある。

【0004】

【数 1】

$$\min_{f \in H} \frac{1}{n} \sum_{i=1}^n V(f(\bar{x}_i), y_i) + \lambda \|f\|_K^2 \quad (\text{式 1})$$

【0005】

ここで、 H は、対応するカーネル関数 K を有する再生核ヒルベルト空間（Reproducing Kernel Hubert Space）または RKHS（Aronszajn、1950 年）であり、 $\| \cdot \|_K$ は、RKHS における二乗ノルムである。 λ は、トレーニングセットの適合精度と、（RKHS ノルムの） f の円滑さとの間のトレードオフを制御するための正則化定数であり、 $V(f(\bar{x}), y)$ は、 \bar{x} が判明し、 $f(\bar{x})$ が予測されており、これに対応する実値が y であるときのプライスピード（Price Paid）を表す損失関数である。異なる V を選択することで、異なる学習法が得られる（Evgeniou ら、2000 年）。特に、ヒンジ損失（Hinge Loss）の $V(f(\bar{x}), y) = \max(1 - y f(\bar{x}), 0)$ には SVC という、よく知られたサポートベクターマシン（Support Vector Machine）が含まれ（Vapnik、1998 年）、一方、二乗損失の $V(f(\bar{x}), y) = (f(\bar{x}) - y)^2$ には、単純な正則化最小二乗分類器が含まれる（Wahba、1990 年）。

【0006】

二乗損失を含む幅広い損失関数では、いわゆる Representer 定理は、Tikhonov の最小化問題の解では下の式 2 を用いることが証明されている（Scholkopf ら、2001 年；Wahba、1990 年）。

【0007】

10

20

30

40

【数2】

$$f_s(\bar{x}) = \sum_{i=1}^n c_i K(\bar{x}, \bar{x}_i) \quad (\text{式2})$$

【0008】

Representer定理は、RKHSに関数を求める無限次元問題を係数 c_i を求めるn次元問題となる。正則化最小二乗（以下、「RLS」という）の場合、損失関数は微分可能であり、単純な代数により、線形システム $(K + \lambda I)c = y$ を解くことで係数 c_i を求めることができる。ここで、一般に用いられている省略表記に従えば、Kは $K_y = K(\bar{x}_i, \bar{x}_j)$ を満たす $n \times n$ の行列である。

10

【0009】

RLSアルゴリズムには、いくつかのすばらしい特性がある。概念がシンプルであり、数行のMATLABコードを用いて効率的な実施が可能である。回帰も分類もまったく同じフレームワークに置くことができる（つまり、同じコードで回帰と分類の問題を解くことができる）。直感的には、二乗損失関数は、回帰の問題には好適である一方で、ヒンジ損失と比較すると、分類の問題には適した選択とは言い難い。

具体的には、ある点 $(\bar{x}, y=1)$ があるとすると、二乗損失は $f(\bar{x})=10$ と $f(\bar{x})=-9$ とに等しくペナルティを科すが、SVMのヒンジ損失は後者のみにペナルティを科す。しかしながら、広範囲な各種問題では、正則化最小二乗分類器（以下、「RLSC」ともいう）だけでなく、SVM（サポートベクタマシン）も同時に実行する（Rifkin, 2002年； RifkinとKlautau, 2004年）。

20

【発明の開示】

【発明が解決しようとする課題】

【0010】

その一方で、大規模なデータベースや非線形カーネルでは、RLSは、より一般的なSVMと比較すると、重大な問題を抱える。詳細には、RLSを解く直接的な手法では、カーネル行列全体を扱うため、 $O(n^3)$ 時間と、（さらに悪いことに） $O(n^2)$ 空間が必要になる。このような問題は、行列・ベクトル積のみが必要となる共役勾配などの反復法を用いることで緩和できる。しかし、dが大きい場合、反復ごとにカーネル行列を再計算するには、 $O(n^2d)$ 仕事が必要となるため、dが大きい場合には禁止される恐れがある。一方、SVMは演算上極めて魅力的であるのは、正しく分類された点の平坦な損失関数は、正しく分類された点がすべてゼロの係数 c_i を有する疎の解を導出するためである。トレーニング動作中常に正しく分類される1対のトレーニング点同士のカーネル積を最新のSVMアルゴリズムで演算することはない。とりわけ、SVMアルゴリズムはカーネル行列の小さい分数のみを生成するため、SVMの方がRLSよりもはるかに非線形問題に適することになる。

30

【0011】

より詳細には、線形カーネル $K(\bar{x}_i, \bar{x}_j) = \bar{x}_i \cdot \bar{x}_j$ を用いるときには、SVMとRLSCとの相対的な効果は反転する。係数 c に係わる RLSC 問題は、Sherman-Morrison-Woodbury 公式を直接用いるか、共役勾配法を反復的に用いるかの、 $O(nd^2)$ 時間と $O(nd)$ メモリの問題となる。

40

SVMのほうが幾分速いが大差はない。というのは、最新のSVMアルゴリズムは座標上昇アルゴリズム（Coordinate Ascent Algorithms）であって、このアルゴリズムは、実施可能領域の境界部に働き、残りの係数を固定したままでいくつかの係数を最適化するため、カーネル行列Kの明示的エントリ（Explicit Entries）が必要となるからだ。行列・ベクトル積で表現可能であり、これにより線形カーネルがもたらす強みのすべてを利用できる内

50

点SVMにおいていくつかの試みを行った。しかしながら、このようなアプローチは、最新の線形SVMと同じ性能を達成するまでには至っておらず、正則化の線形分類器をトレーニングする方法としては、依然としてRLS分類が最速の方法である。

【0012】

したがって、正則化最小二乗をより実践的に用いるような技術が求められている。

【課題を解決するための手段】

【0013】

本発明の一実施形態では、コンピュータに実装する正則化最小二乗（RLS）分類／回帰に関する方法を提供する。本方法は、データのトレーニングセットを受信し、このトレーニングセットを用いて行列分解（例えば、固有値分解やSVD）を演算することを含む。本方法は、複数の正則化パラメータを受信し、前記行列分解を用いて正則化パラメータの各々について係数を演算することをさらに含む。本方法は、前記に続いて、前記正則化パラメータの各々についてL00（leave-one-out）誤差も演算することを含む。本方法は、前記選択した正則化パラメータに対応する係数cまたは超平面関数wの少なくとも1つに基づいて未来のデータ点を予測することをさらに含む。特定の場合には、トレーニングセットを用いた行列分解の演算には、トレーニングセットを用いて特異値分解（以下、「SVD」ともいう）を演算することが含まれ、この場合のSVDは行列分解である。別の特定の場合には、トレーニングセットを用いた行列分解の演算には、入力したトレーニングセットを用いてカーネル行列を生成し、このカーネル行列の固有値分解を演算することが含まれ、この場合の固有値分解は行列分解である。このような場合の一例では、前記カーネル行列を明示的に表現し、本方法は前記カーネル行列を記憶することを含む。このような場合の別の例では、前記カーネル行列を明示的に表現し、本方法は、 $O(n^3+n^2d)$ 時間と $O(n^2)$ 空間でのすべての正則化パラメータについて前記L00誤差を演算することをさらに含み、この場合のnは前記トレーニングセットのd次元におけるデータ点の数である。10

また、このような場合のさらに別の例では、前記カーネル行列は $K_{ij} = K(\bar{x}_i, \bar{x}_j)$

$$K(\bar{x}_i, \bar{x}_j) = \exp\left(-\frac{\|\bar{x}_i - \bar{x}_j\|^2}{2\sigma^2}\right)$$
30
を満たし、この形式を有する $n \times n$ 行列 K であり、この場合の x は、前記トレーニングセットに含まれるデータ点のベクトルであり、σ はユーザ選択の帯域幅パラメータである。あるいは、前記カーネル行列は、例えば、（特に、行列が大きすぎて一度に記憶できない場合に）行列・ベクトル積を用いて表現することができる。このような場合の一例では、IFGT（改良版高速ガウス変換）を用いて前記行列・ベクトル積を概算する。

【0014】

本発明の別の実施形態では、1つ以上のプロセッサによって実行されるときに、このプロセッサに正則化最小二乗（RLS）分類／回帰の処理を実行させるような指令が符号化された（例えば、1つ以上のコンパクトディスク、フレキシブルディスク、サーバ、メモリスティック、ハードドライブなどの）マシン読み取り可能な媒体を提供する。この処理は、例えば、前記の方法の類似例や変形例であってもよい。40

【0015】

本発明の別の実施形態では、正則化最小二乗（RLS）分類／回帰システムを提供する。システムの機能性（例えば、前記の方法やその変形例など）は、多数の手段、例えば、ソフトウェア（1つ以上のコンピュータ読み取り可能記憶媒体に符号化された実行可能な指令など）、ハードウェア（ゲートレベル論理や1つ以上のASICなど）、ファームウェア（I/O能力を有する1つ以上のマイクロコントローラおよびここに記載する機能を実行するための搭載ルーチン）、あるいはこれらを組み合わせものに実装可能である。特定の場50

合では、本システムを、実行可能なRLS分類モジュール、あるいはこれらのモジュールを記憶するモジュールセットを備えた、デスクトップやラップトップコンピュータなどの演算環境に実装する。

【0016】

ここに記載する特徴や利点は必ずしも包括的なものではなく、特に、さらに多くの追加的な特徴や利点があることは、本願の図面や記載に照らすならば当業者には明らかである。さらに、本明細書中に用いられる表現は、主に、読みやすさや例示の目的から選んだものであって、これによって本発明の主題の範囲が限定されるものではない。

【発明を実施するための最良の形態】

【0017】

正則化最小二乗法をより実践的に用いる技術を開示し、より具体的には、従来技術に係わる演算上の制約を低減する技術を開示する。

【0018】

<概要>

本発明の一実施形態では、正則化最小二乗分類器（RLSC）を提供する。周知の通り、分類器は（対象物の特性や状況の特徴に関する値など）入力値を受けると、これら入力値に関するラベルを個別に出力するようにプログラムされている、あるいはそのように構成されている。顔認識、対象物追跡、室内ナビゲーション、医療画像解析、音声認識などは分類法が適用された応用例である。このような分類器を、固定型分類器あるいは学習用分類器として実装可能である。さらに、ここに記載する技術は、本開示に照らせば明らかなように、RLS回帰の問題にも適用可能である。

【0019】

前記に説明した通り、RLS の損失関数は微分可能であり、係数 c は線形システムの $(K + \lambda n I)c = y$ を解くことで求めることができる。なお、 K は $K_{ij} = K(\bar{x}_i, \bar{x}_j)$ を満たす $n \times n$ の行列である。

$$K(\bar{x}_i, \bar{x}_j) = \exp\left(-\frac{\|\bar{x}_i - \bar{x}_j\|^2}{2\sigma^2}\right)$$

本発明のある実施形態によれば、

形式のガウスカーネル（Gaussian Kernel）を検討する。なお、 σ はユーザ選択の帯域幅パラメータである。

本開示に照らせば明らかなように、この他の実装も可能である。RLS アルゴリズムにはいくつかのすぐれた特性があり、その中には、（MATLAB やこの他の好適な演算環境／ソフトウェア パッケージなどを用いることで）符号化が簡便であったり、回帰と分類の問題へ適用できる点も含まれる。さらに、RLS は L00 誤差 $f_{S^i}(\bar{x}_i)$ について陽関数表示の閉論理式（closed-form formula）を用いる。

なお、 $S^i = S - \bar{x}_i$ は、 i 番目のデータ点を取り除いたトレーニングセットを意味する。前記のような RLS を解く反復法では、陽関数表示で逆正則化カーネル行列の対角を利用して、L00 値を算出する直接式の利点を活かすことができない。

SVMの解法の開発者によっては、L00の算出や（Rifkin、2002年）、境界付け（Joachims、1990年）を可能にした者もいるが、正確に演算を行おうとすれば演算の費用がかさみ、境界付けには重大な誤差が付隨的に発生する恐れもある。ここで述べるように、中規模から小規模のデータセットについては、交差検定（cross-validation）の「黄金基準（gold standard）」としてL00を効果的に用いることができる。この基準はほとんど偏りがなく、10-fold cross-validationなどの手順に関連するばらつきは極めて高く、データセットの分割を変えることで実質的に異なる結果がもたらされる。大規模なデータセツ

10

20

30

40

50

トについては、L00の重要度は低くなるが、ここで述べるように、同じく利用には適している。

【0020】

従って、本発明の一実施形態では、RLSに関連するいくつかのタスクについて、極めて安価な演算費用で正則化パラメータを広範囲な値全体について変化させる方法を示すとともに、分類器を構築し、同時にトレーニングセットを用いてこれらの分類器でL00値を演算することで、RLSをより実践的なものにする。正則化の手法では、の選択は重要であり、選択が悪いと分類器の精度が損なわれることになる。RLS（およびSVM）を頻繁に「モデル選択」のフレームワークに用いるため、膨大な数の値を検証する能力は、すなわち演算上の強みを意味する。RLSを特徴選択の問題（例えば、顔・対象認識など）の構成要素として用いるときや、カーネルのハイパーパラメータの選択が必要なときに、この違いが特に重要になる。ここに記載する方法論の1つは、 $(O(n^2))$ カーネル行列を記憶・操作可能な中規模の問題について、1つの値について「直接的に」解く場合と比較すると小さい定数係数（おおよそ3）でスローダウンはあるものの、膨大な数の値について「解」cとL00値の両方を求めるところから始まる。次に、本方法論では、多数の基準データセットにおいてとを広範囲に変化させる大規模な実験を行って、大きい値を用いることで正確な分類がしばしば得られることが実験観察から明らかである。加えて、を大きくして大規模な問題を解く能力を利用する方法と、多数の演算例についても説明する。10 20

【0021】

<システム構造>

【0022】

図1は、本発明のある実施形態に係わる構成の演算環境10を示すブロック図である。

【0023】

この事例から分かるように、演算環境10には、バス119を介してメモリ107に動作可能に接続されるプロセッサ101と、RLS分類器／回帰モジュール105（および、オペレーティングシステムおよびアプリケーションなどのこの他の実行可能な符号）を記憶する記憶装置103と、キーボード113と、グラフィックアダプター109と、ポイントティング装置117と、ネットワークアダプター115とを備える。ディスプレイ111は、グラフィックアダプター109に動作可能に接続される。30

【0024】

プロセッサ101はいずれのプロセッサであってもよく、各種のオペレーティングシステム（UNIX（登録商標）など）や演算環境10のアプリケーション／ドライバ（MATLAB（登録商標）など）を実行可能なプロセッサのセットであってもよい。多数の最適なプロセッサ（例えば、Intel Pentium（登録商標）、MAC（登録商標）のG4またはG5のプロセッサ、AMD（商標）のKプロセッサ、いずれのコプロセッサ、オンボードキャッシュなどのこの他の代表的な演算最適化）を使用可能であることは、本開示に照らせば明らかである。メモリ107は、例えば、ファームウェアROM、RAMおよびまたはフラッシュメモリであってもよく、（例えば、512Mバイト以上の）プロセッサ101が用いる指令やデータを記憶する。一実施形態では、記憶装置103は（例えば、10ギガバイト以上の）ハードディスクドライブであるが、データを持続的に記憶できるものであれば、メモリステイックや固体メモリ装置などのいずれの装置であってもよい。記憶装置103は、一般に行われるよう、実行可能な符号およびまたはデータを含む複数のファイルを記憶可能である。演算環境10は実行可能なファイルをメモリ107にロードして、プロセッサ101でこれを実行する。図示する実施形態では、RLS分類器モジュール105は記憶装置103に実行可能な符号として記憶され、メモリ107にロードされて、1つ以上の処理を行うためにプロセッサ101で実行される。40

【0025】

記憶装置103に記憶されるファイルは、例えば、MATLAB形式（Mファイルとも呼ばれ

50

る)であってもよい。記憶装置103には、他のファイル形式も記憶できるため、例えば、EXEファイル、DLLファイル、INIファイル、あるいは演算環境を適切に動作させるために必要なファイル種別であればいずれであってもよい。このようなファイルは、Microsoft社のオペレーティングシステムを組み込んだ演算環境において代表的なものである。この他のファイル形式としては、Apple社のMacintosh(登録商標)およびUNIXベースのコンピュータで利用されるものであることは、本開示に照らせば明らかであろう。

【0026】

ポインティング装置117は、マウス、トラックボール、あるいはこの他のユーザによる入力装置などがあり、一般に行われているように、キーボード113と組み合わせて用いることでユーザが演算環境10とのやり取り(入力データおよびプロンプトへの応答など)が可能となる。グラフィックアダプター109は、画像やこの他の情報をディスプレイ111に表示する。ネットワークアダプター115は、所望であれば、一般に行われるよう(例えは、従来の有線・無線技術などを用いた)インターネットやLANあるいはこれらを組み合わせたような外部ネットワークを介して通信可能に演算環境10に接続する。

【0027】

演算環境10は、コンピュータプログラムモジュールを実行して、ここに記載するようなRLS分類器モジュール105の機能を含むRLS機能をもたらす。RLS分類器モジュール105の構造と機能については、図2から図5を参照しつつ詳細に説明する。本開示に照らせば明らかなように、演算環境10は、汎用のデスクトップ型またはノート型のコンピュータなどの処理システム、あるいは、ここに記載するRLS分類(またはRLS回帰)を実行するように構成された専用の処理システムを介して実現してもよい。場合によっては、演算環境10は、1.7GHzのPentium-Mプロセッサおよび2ギガバイトのRAMを搭載したIBM社製T42ラップトップによって実現される。この他の集中的な演算処理は、標準的なユーザのデスクトップ環境でも実行可能であるが、ここに記載するRLS処理を実行するために処理能力を反転させる必要性に応じて限定することもできる。

【0028】

<分類器／回帰モジュール>

図2は、本発明のある実施形態に係わるRLSに分類器／回帰モジュール105内のモジュールを図示するブロック図である。

【0029】

図示するように、モジュール105には、入力モジュール210と、カーネル行列生成器203と、固有値分解/SVDモジュール205と、係数(c)演算モジュール207と、L00(leave-one-out)誤差演算モジュール209と、正則化パラメータ(λ)選択モジュール211と、予測モジュール213と出力モジュール215とを備える。一実施形態では、これらモジュールの各々は、所定の機能性を提供するための実行可能なソフトウェア(例えは、C、C++、この他の最適な指令セットなど)によって実現される。ただし、このようなモジュールは、ハードウェア(例えは、ゲート論理など)、ファームウェア(各モジュールの機能性を実行するルーチンを搭載したマイクロコントローラ)あるいはこれらの組み合わせによっても実施可能である。本開示に照らせば、ここに例示されるモジュールは本発明の一実施形態を示したものであることは理解できるであろう。したがって、この他の実施形態には、追加的な、あるいは異なるモジュールや機能性を含んでいてよい。同様に、他の実施形態では、ここに記載するいくつかのモジュールを欠いたり、モジュール間で前記RLSの機能性を異なる形態によって分散させることによってもよい。

【0030】

10

20

30

40

入力モジュール201は、トレーニング用データセットを受信するようにプログラミングされているか、あるいはそのように設計されている。このデータを用いて、所定の用途で用いられるR L S分類器／回帰モジュール105のトレーニングを行う。このようなデータセットはd次元におけるn個のデータ点 $S = (\bar{x}_1, y_1), (\bar{x}_2, y_2), \dots, (\bar{x}_n, y_n)$ として表現することができる。前記に示した通り、帰納的な学習タスクとは、新しい点 x が与えられると、これに対応する y 値を予測するための関数 $f(\bar{x})$ を構築することである。さらに、この入力モジュール201は、対距離 (Pairwise Distances) のカーネル行列を生成する準備として、入力されたデータセットについてフォーマットと構築化を(必要ならば)適宜行う。

10

カーネル行列生成器203は、カーネル行列 (K) が $O(n^2)$ のサイズであるような対距離のカーネル行列 (K) を生成するようにプログラミングされている、もしくはそのように設計されている。なお、以下に順次説明するのだが、本発明の一実施形態によれば、カーネル行列生成器203は、特異値分解の処理を用いることができる線形カーネルの場合には適用しない。

【0031】

本発明のある実施形態に係わるRLSCアルゴリズムでは、以下の式3を用いればLOO (leave-one-out) 誤差値が陽関数表現で与えられる。

【0032】

【数3】

20

$$\begin{aligned} LOOerror &= \frac{G^{-1}Y}{\text{diag}(G^{-1})} \\ &= \frac{c}{\text{diag}(G^{-1})} \end{aligned} \quad (\text{式3})$$

【0033】

上記のように、トレーニング重み c (係数 c ともいい、一般に、 c は関数展開上のトレーニング点を表す) と正則化カーネル行列 G の逆の対角を与えることで LOO 値を算出することができる。固有値分解 $K = Q\Lambda Q^T$ を与えると、 $c(\lambda)$ は $O(n^2)$ にて算出可能となる。また、LOO 誤差も $O(n)$ 時間にて算出可能となる。 $G(\lambda)^{-1}$ の成分が 1つであれば $O(n)$ 時間にて演算可能であるが、これについてはモジュール207を参照しつつ後述する。このように、 $\text{diag}(G^{-1})$ を算出することが可能であり、LOO 誤差を $O(n^2)$ 時間にて算出する。

30

【0034】

係数 c 演算モジュール207は、正則化パラメータ () の集合を受信し、固有値分解 / SVDモジュール205が算出した行列分解を用いて、各 について係数 c を算出するようプログラムされている、もしくはそのように設計されている。

40

より詳細には、与えられた λ について、係数 c 演算モジュール207は、 K に λI を加えることすべての固有値が λ だけシフトするが、対応する固有値： $(K + \lambda I) = V(\Lambda + \lambda I)V^T$ は変化しないことを利用して、対応する係数 c および $\text{diag}(G^{-1})$ の算出を実行する。特定の一実施形態では、係数 c 算出モジュール207は、以下の式4を用いて各 λ について c の算出を実行する。

【0035】

【数4】

$$\begin{aligned}
 c &= V(\Lambda + \lambda I)^{-1} V^T y \\
 G_y^{-1} &= (Q(\Lambda + \lambda I)^{-1} Q^T)_y \\
 &= \sum_{k=1}^n \frac{Q_{yk} Q_{yk}}{\Lambda_{kk} + \lambda}
 \end{aligned} \tag{式4}$$

【0036】

したがって、係数 c について 1 つの RLS 問題を解くまでに $O(n^3)$ の時間が必要になる。K の固有値分解を算出すれば、基本的に追加費用を必要とせずに、 λ のファイングリッド (quite fine grid) 上で $c(\lambda)$ と $\|LOO\ error(\lambda)\|$ を算出することができる。このような手法の利用には $O(n^3)$ 時間と $O(n^2)$ メモリが必要になる。

10

【0037】

固有値分解 / SVD モジュール 205 は、カーネル行列 K の固有値分解を算出するようにプログラムされている、もしくはそのように設計されている。かつたまは、(線形カーネルの場合についても) 固有値分解 / SVD モジュール 205 は、データ行列 (トレーニングセットのデータ) の特異値分解 (SVD) を算出することができる。

20

【0038】

線形カーネル $k(X_i, X_j) = X_i^T \cdot X_j$ の特殊な場合では、 $d < n$ の \mathbb{R}^d にいるとする。もし $d > n$ ならば、カーネルが線形であることを無視して、前記の固有値分解を用いる。 $d < n$ ならば、行列 X はトレーニングセット全体を含み、1 列に 1 データ点とし、X の i 番目の列を X_i^T するように定義する。なお、上記は数学的な表記による決定であって、データをディスク上やメモリ上に表記する方法を決定するものではない。この方法では、X には n 個の列と d 個のコラムがある。線形カーネルを有する場合は、学習中の関数も線形である：

【0039】

30

【数5】

$$\begin{aligned}
 f(x) &= \sum_i c_i k(X_i, x) \\
 &= c^T X x \\
 &= w^T x
 \end{aligned} \tag{式5}$$

【0040】

ここで、超平面 w は $X^T c$ として定義する。新規のデータ点を分類するときには、n 個のカーネル積 (通常 $O(nd)$ 時間かかる) の加重和を算出せずに、 w を用いて $O(d)$ 時間に分類することができる。 $n \times d$ データの行列 X は、行列 X のランクが列数と同じ (full column rank) であり、経済的な規模の特異値分解 $X = USV^T$ を有するものとする。

40

なお、この場合、 $U^T U = VV^T = V^T V = I_d$ である。なお、 $UU^T \neq I_n$ であり、S は特異値の準正定対角行列 (positive semi-definite diagonal matrix) である。カーネル行列 K は、 $K = XX^T = (USV^T)(VSU^T) = US^2V^T$ によって与えられる。膨張係数 c 、超平面関数 w および LOO 誤差値に関するそれぞれの表現を導くことができる。本発明の一実施形態によれば、非線形 LOO の関連式を直接用いることで、1 セットの公式を求めるのだが、ここでは SVD を用いることとすると、

50

【0041】

【数6】

$$\begin{aligned}
 c &= (K + \lambda I)^{-1} Y \\
 &= (US^2U' + \lambda I)^{-1} Y \\
 &= U(S^2 + \lambda I)^{-1} U' Y + \lambda^{-1} U_{\perp} U_{\perp}' Y \\
 w &= X' c \\
 &= VSU'(U(S^2 + \lambda I)^{-1} U' Y + \lambda^{-1} U_{\perp} U_{\perp}' Y) \\
 &= VS(S^2 + \lambda I)^{-1} U' Y \\
 diag(G^{-1}) &= diag(U(S^2 + \lambda I)^{-1} U' + \lambda^{-1} U_{\perp} U_{\perp}') \\
 &= diag(U(S^2 + \lambda I)^{-1} U' + \lambda^{-1} (I - UU')) \quad (式6)
 \end{aligned}$$

10

【0042】

SVDの算出に、 $O(nd^2)$ 時間と $O(nd)$ メモリが必要になる。

【0043】

U を算出すると、 λ の所定値について、 c と $diag(G^{-1})$ を $O(nd)$ 時間で求めることができる。 $n >> d$ のとき、 $O(nd^3)$ 時間ではなく $O(nd^2)$ 時間で良好な λ を求められる点で、大幅な節約と言える。高次元の大規模データセットでは、メモリ上でのデータセットの調整が困難な場合がある。さらに、基本的にSVDはout-of-coreで算出したたり、コンピュータ群により平行的に演算することは可能であるが、このような作業ツールは容易に手に入るものではない。

20

従って、本発明のある実施形態によれば、線形RLSの代替的な線形代数を用いる。具体的には、線形カーネルを用いて超平面を求めるときに、RLS問題を c ではなく w で直接表

$$\min_{w \in R^d} L(w) = \frac{1}{2} \|Y - Xw\|_2^2 + \frac{\lambda}{2} \|w\|_2^2.$$

30

w で微分した $\frac{\partial L}{\partial w} = X'Xw - X'Y + \lambda w$ をゼロに設定すると下の式7のようになる。

【0044】

【数7】

$$\begin{aligned}
 w &= (X'X + \lambda I)^{-1} X'Y \\
 &= V(S^2 + \lambda I)^{-1} V'X'Y \\
 &= V(S^2 + \lambda I)^{-1} V'(VSU')Y \\
 &= VS(S^2 + \lambda I)^{-1} V'U'Y \quad (式7)
 \end{aligned}$$

40

【0045】

V と S^2 を $d \times d$ $X'X$ の固有値分解から求めることができる。さらに、 $U = XVS^{-1}$ であるため、 U を含む表現であれば XVS^{-1} として記述することができる。この式の利点は、 $X'X$ や $X'y$ を生成する並列プログラムやout-of-coreプログラムはプログラミングの作業がそれほど困難でない点である。この手法で最も時間がかかるのは、共分散行列 $X'X$ を生成する段階であり、 $O(nd^2)$ 時間がかかり、さらに後の固有値分解に $O(d^3)$ 時間がかかる。この手法では、 $X'X$ と $X'y$ を生成するための環境整備の労力は必要だが、 $d \times d$ 行列を処理する能力さえ与えれば演算により実施可能である。実際には、 $n >> d$ であり、ここに記載するような共分散法を用いる場合の線形関数の処理では $w(\cdot)$ のみを求めて、 c と $diag(G^{-1})$ の演算に

50

必要な $U=XVS^{-1}$ を求めるよりもテストセット (holdout set) で動作確認を行うほうが好ましい。SVDおよび共分散に基づく手法には、同じような（漸近的な）演算上の複雑さがある。しかし、SVD手法は数値的に安定していると言える。

【0046】

選択した所定の λ について、カーネル行列 K の固有値分解（あるいはデータ行列の SVD）をモジュール 205 で算出すると、モジュール 207 は、前記の通りに c と $\text{diag}(G^{-1})$ の生成を実行する。L00 (leave-one-out) 誤差算出モジュール 209 は、正則化パラメータ

(λ) の集合を受信し、各 λ について L00 誤差を算出する（例えば L00 誤差 = $\frac{c}{\text{diag}(G^{-1})}$ ）
10 ようにプログラムされている、もしくはそのように設計されている。正則化パラメータ (λ) 選択モジュール 211 は、（例えば、値を比較するなどして）最小の L00 誤差を有する λ を選択するようにプログラムされている、もしくはそのように設計されている。

【0047】

予測モジュール 213 は、最も優れた λ に対応する RLS 関数に基づいて（例えば、膨張係数 c または超平面関数 w に基づいて）データ点をさらに予測する。出力モジュール 215 は、入力されたデータセットに基づいて RLS の予測を出力するようにプログラムされている、もしくはそのように設計されている。さらに、出力モジュール 215 は、出力された予測データのフォーマット化と構築を（必要ならば）適宜行って、このデータをディスプレイ、あるいは報告できるように準備をする。
20

【0048】

本実施形態により実現される時間短縮について、最良の事例を示す。より詳細には、表 1 を参照しつつ、4192 点ある 14 次元の Galaxy Dim データセットを検討する。データセット全体と、このデータセットの半分のランダムなサブセットについて、従来の直接法と本発明のある実施形態に係わる固有値分解法とをそれぞれに用いて L00 誤差値を算出し、そのタイミングを比較した。表 1 に報告したタイミング結果は概算値（例えば、 $\pm 5\%$ ）であるため、近似するタイミングの順番は変わる場合もある。報告した時間はすべて、5 回のテストの平均値であり；分散値は平均値と比べて小さかった。時間はすべて経過時間で表した；ディスクの交換は行なわず、システム時間は合計時間では問題にならない程度であった。
30

【0049】

【表 1】

	データ点数	
	2048	4092
直接解	12.4	106.0
固有値分解	28.1	268.6
時間 / λ	0.4	1.7
25 λ の合計時間 (直接法)	310.0	2650.0
25 λ の合計時間 (固有値分解法)	38.1	311.1
25 λ の比	8.1	8.5
$\infty \lambda$ の比	31.0	62.5

【0050】

表 1：直接線形代数を用いて Galaxy Dim データセットの L00 値を算出するのにかかった時間（秒）

【0051】

上記から分かるように、この規模の問題（例えば、2048 点から 4092 点まで）で
50 3 以上の λ を検討するときには、本発明のある実施形態に係わる固有値分解法による L0

$O(n^3)$ ではなく $O(n^2)$ であるため、問題の規模が大きくなるほど、(十分な数の λ がある場合には) 固有値分解の利点も相対的に大きくなる。この程度の n の範囲では、直接法と固有値分解法はいずれも $O(n^3)$ であるが、問題の規模が 2 倍になれば、約 8 倍 (2048 点では 8.1 倍、4092 点では 8.5 倍) の時間がかかることになる。異なる 25 個のについて LOO 値を算出するのにかかった合計時間は、2048 点で 38.1 秒であり、データセット全体の 4192 点では 311.1 秒であった。ここで説明するアルゴリズムは $O(n^2)$ 空間を用いるため、中規模のデータセットに好適である。大規模なデータセットに好適な手法については後述する。

【0052】

10

< および の変化 >

効果を例示する目的から、RLS 分類器 / 回帰モジュール 105 の実施形態を本明細書に記載するように実現し、これを用いて大規模な実験を行った。その中で、所定のデータセットについて、値の範囲全体で帯域幅パラメータ λ を変化させて、多数の正則化パラメータの各 値について LOO 値を算出する。

図 3 a、図 3 b、図 4 a、図 4 b は、4 つのデータセットについての実験結果をそれぞれ示したものである。なお、この他のデータセットも利用可能であり、UCI machine learning repository (UCI = University of California at Irvine ; 例えば <http://www.ics.uci.edu/~mlearn/MLRepository.html> を参照のこと) なども類似する結果を有するため利用可能である。図 3 a、図 3 b、図 4 a、図 4 b のラベル付けされた縦線の m はカーネル行列の最小成分を表す。

20

【0053】

図 3 a、図 3 b、図 4 a、図 4 b からは多数の興味深い観察事項が読み取れる。例えば、 λ の選択が適切であれば、最高レベルの精度に等しいか、それに近似するレベルの精度が得られるような λ の値の範囲が幅広いことが分かる。さらに、非常に大きい λ で正確な分類 / 回帰を得ることができる。特に、カーネル行列のすべての成分が限りなく 1 に近似する(例えば、0.9999 以上)ように λ を選択することができる。同時に、大きい λ を用いて好適に分類を行うためには、 λ をできる限り小さくする必要がある。ここで、数式の限界を理解することが重要になる。

【0054】

30

実験的な手法を用いて、LOO 算出の精度をチェックするために異なる 2 つのテストをおこなって、RLS 分類器 / 回帰モジュール 105 の性能テストを行うことができる。1 つ目のテストはスポットチェックであり、このテストでは、データの小サブセットについて各点 x をトレーニングセットから明示的に取り除き、データの残りの部分について分類器を構築して、点 x について分類器のテストを行う。2 つ目のテストはデータを若干擾動させるためのものであり、このテストでは、各数値に $(1+\epsilon)$ 形式の独立ランダム変量を乗じることで、すべての値(トレーニングデータ、 y 値、 λ および σ) を擾動させる。なお、 ϵ は、平均が 0、偏差が 10^{-13} のガウス確率変数(Gaussian random variable)である。この 2 つのテストから、図 3 a、図 3 b、図 4 a、図 4 b に反映された精度は許容可能である。しかしながら、 λ が小さくなるほど、この精度は低下する。一般に、 $\lambda < 1e-11$ での LOO 値は、所定の用途によっては許容可能なレベルに達しないこともある。

40

【0055】

通常、RLSC(および SVM) は、テストのデータ点を中規模数の近似点によって分類するような重み付け局所法として働くものだ、という直感的事実に基づけば、このように大きい λ で好適な分類が行える能力が得られることは予想できない。大きい λ を用いる手法では、最も遠い点は、最も近いものとほぼ同じインパクトを有する。以下では、本発明の各種実施形態に係わる、大きい λ を利用した大規模な正則化最小二乗分類器(および回帰)の問題を迅速かつ正確に解く方法について説明する。

50

【0056】

<大規模問題と大きい >

大規模な問題でカーネル行列 K を表現することは、メモリの制約上、明らかに実用的ではない。そこで、1つの手法として、ベクトルにカーネル行列 K を乗じる能力だけを必要とする共役勾配を用いる。 $(K + \lambda I)c = y$ の方程式で複数の λ を解くために、CG アルゴリズムを直接用いることはできない。むしろ、行列 K に乘じるベクトル x は λ に依存する。しかしながら、本発明の特定の実施形態によれば、CG を多重化することができる。詳細には、因数分解 $K = QTQ'$ を考慮する。この場合の Q は対角であり、 T は対称三重対角 (symmetric tridiagonal) (この因数分解は現代の固有値分解の第1ステップ) である。Lanczos 原理 (Lanczos process) で Q および T を反復的に構築する。

具体的には、もし

$$Q = (q_1 \dots q_r)$$

および

$$T = \begin{pmatrix} \alpha_1 & \beta_1 & 0 & \cdots & 0 & 0 \\ \beta_1 & \alpha_2 & \beta_2 & \cdots & 0 & 0 \\ 0 & \beta_2 & \alpha_3 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & \alpha_{r-1} & \beta_{r-1} \\ 0 & 0 & 0 & \cdots & \beta_{r-1} & \alpha_r \end{pmatrix}$$

10

20

であれば、下の式を参照することで Q と T を（正確な計算によって）算出できる。

【0057】

【数8】

$$q_k = \frac{x_{k-1}}{\beta_{k-1}} \quad (\text{式8})$$

$$\alpha_k = q'_k K_{qk} \quad (\text{式9})$$

30

$$x_k = K_{qk} - \alpha_k q_k - \beta_{k-1} q_{k-1} \quad (\text{式10})$$

$$\beta_k = \|x_k\|_2 \quad (\text{式11})$$

ここで、 $\beta_0 = \|x_0\|_2$, q_0 であり、 x_0 は任意の non-zero ベクトルである。

40

【0058】

処理が r ステップ分だけ実行されるとき、得られる Q は、連続するベクトル $x_0, Kx_0, K^2x_0, \dots, K^{r-1}x_0$ の Gram-Schmidt の直交化法 (Gram-Schmidt orthonormalization) によって与えられる直交基底、つまり、Krylov の部分空間 (Krylov subspace) である。

【0059】

$x_0 = y$ ならば、 $x = QT^1Q'y$ は、共役勾配法で求めた $Kx = y$ についての r 番目の反復解である。さらに、 $K \rightarrow K + \lambda I$ の対角シフトにおいては、式 8 から式 11 で算出した数量は $Q, T \rightarrow Q, T + \lambda I$ になる、よって、 $x = Q(T + \lambda I)^{-1}Q'y$ は $(K + \lambda I)x = y$ についての r 番目の反復解である。したがって、多数の λ についての解を求めるために必要なものは、1 つの正確な Lanczos 原理の三重対角化だけであり、これらの解は迅速に求めることができる。三重対角行列の線形システムで乗じたり、この線形システムを解くときに必要なものは線形時間だけであるからだ。

【0060】

10

また、類似する手法で $L00$ 値を算出することも可能である。より詳細には、 K を部分的に三重対角化して、 $\text{diag}(K(K+\lambda I)^{-1})$ を $\text{diag}(QTQ'(QTQ'+\lambda I)^{-1})$ として概算化することができる：

【0061】

【数 9】

$$\begin{aligned} (QTQ' + \lambda I)^{-1} &= Q(T + \lambda I)^{-1}Q' \\ &\quad + \frac{1}{\lambda}(I - QQ') \\ QTQ'(QTQ' + \lambda I)^{-1} &= QT(T + \lambda I)^{-1}Q' \end{aligned} \quad (式 12)$$

20

【0062】

このように、 $\text{diag}(QTQ'(QTQ'+\lambda I)^{-1})$ の単一要素を算出するには、 Q の 1 列に $T(T + \lambda I)^{-1}$ を乗じ、要素ごとに $O(r)$ 時間で Q の同じ列から得られた結果に点を付すことができる。R 回反復した所定の Q と T における $\text{diag}(K(K+\lambda I)^{-1})$ の概算を求めるのに必要な合計時間は、 $O(nr)$ である。実際には、 T の固有値分解が完全に形成されるのであれば、時間は若干遅くなってしまっても、より単純な手法を用いることもできる。この場合の費用は $O(nr + r^3)$ であり、 r が \sqrt{n} よりもはるかに小さいため、違いはほとんど問題にはならない。

30

【0063】

丸め誤差は、 q_i における直交性の損失を引き起こす傾向がある。したがって、式 1 から式 4 を直接的に用いると不安定を招く恐れがある。そこで、ARPACK (Lehoucq および Sorensen, 1996 年; Lehoucq ら, 1997 年) ライブラリによって、数的に安定したバージョンである、Lanczos 原理 (Lanczos process: アーノルディ原理 (Arnoldi process) を安定的に実装したものであり、アーノルディ原理の特殊事例である) が提供されており；ここに記載する実験にこのライブラリを用いる。

【0064】

ここに記載する数学は に関係なく適用する。しかし、できる限り大きい を用いることが有用である理由は少なくとも 2 つあり、いずれの理由も K の構造に関係する。ガウスカーネル (Gaussian kernel) と、一般の位置 (general position) における点を用いて、正確な計算においては、 K はいずれの についてもフルランク (full rank) となる。しかし、 が大きくなるほど K の崩壊スペクトルが速くなって、下位ランクの行列を有する K の概算が容易になり、 $O(n^2)$ 未満時間で正確な行列・ベクトル積 Kx を生成することが容易になる。非線形の RLS 分類を迅速に行うための前記作業の大半は、このような概算法に着目したものであるが、大きい を用いた概算法により取り扱いが容易になることには結びついてはいなかった。大きな を用いることで、同時に多数の利点が得られる。例えば、正則化最小二乗分類に必要な行列・ベクトル積が少なくてすむのは、 K の固有構造の崩壊が迅速であり、行列・ベクトル積 Kx の概算が容易であるからだ。

40

50

【0065】

<改良型高速ガウス変換 (Improved Fast Gauss Transform (IFGT)) >

本発明の一実施形態によれば、改良型の高速ガウス変換 (IFGT) を用いて行列・ベクトル積Kxを生成することができる。また、ニストロム法 (Nystrom method) など、この他の手法も用いててもよい。IFGTにより中心のセットを取り出し、これら中心付近の截端（丸め）エルミート展開 (truncated Hermite expansions) を合計したものを用いて行列・ベクトル積Kxを算出する。本明細書の目的から、IFGTをブラックボックスとして取り扱う；つまり、行列A、ベクトルx、パラメータpが与えられるならば、IFGTは、Axに対する「第p次数」の概算値を算出する。第p次数を1つ処理するために、IFGTは、 $O(nd^p)$ 時間と $O(nd^p)$ 空間を要し；CGの反復が一定数であるとすれば、得られるアルゴリズムは、時間と空間のいずれも $O(nd^p)$ になる。 $d^p \ll n$ あるときには、このやり方は非常に効果的である。10

【0066】

Yangらは、が小さい固定値の場合には、IFGTは最終的にはゆっくりと多項式次数pへ収束することを証明する実験を行っている。さらにYangは、小さいp ($p = 1, 2$)についてのIFGTの動作時間が非常に魅力的であることも証明している。しかし、Yangは、やを変化させることは考慮していない。むしろ、Yangは、各データセット (.5d) に1つの値を用いているが、の指定はしていない。Yangのデータセットと同じものをここで検討するのだが、最適なを求めて適切なを選択するため、より良好で均一な動作性のを達成することができる。

ここで、作用素 (operator) A のノルムは $\max_{\|v\|=1} \|Av\|$ で定義されることを思い出して、作用素 ($K - IFGT(K)$) と K のそれぞれの相対ノルムを考慮すると；作用素ノルムの演算では、例えば、Arnoldi 原理を20回反復して、作用素の最大固有値を高い精度で求めるごとで、作用素ノルムを算出する。IFGTは非対称であるため、Lanczos 原理ではなく Arnoldi 原理を用いる：一般に、 $y' \cdot IFGT(x) \neq IFGT(y)' \cdot x$ である。図5では、この相対ノルムを広範囲なシグマに対してプロットしたものを示しており、Galaxy Dimデータセットの p = 1, 2, 3 についてプロットした。なお、(本発明の一実施形態に係わる) IGBT を備えた RLS 分類器の性能では、次数が増して σ が増えるほど性能が向上し、浮動小数点の精度が限界に達すると、最終的に曲線が平坦な状態になる。2030

【0067】

<実験>

ここで再びGalaxy Dimデータセットを考慮する。特に、Galaxy Dimはこの規模 (4192点) の事例に適しており；このサブセットは、漸近解析を良好に理解するには十分な大きさの規模であり、カーネル行列K全体を必要とする方法と必要としない方法を比較するには十分な小ささの規模である。表1を再び参照して、2048点のデータのサブセットでは、異なる25個のについてL00値を算出するのにかかった合計時間は38.1秒であり、4192点のデータセット全体では311.1秒であった。ここに記載する方法を用いて、第3次数のIFGTと = 100を用いて実施される行列・ベクトルの積算を行うと、同じ演算を2048点について行った場合の時間は19.3秒であり、4192点全体では38.3秒であった。本発明の一実施形態によれば、>= 1e-11について結果はすべて必然的に同じであり、フルの固有値分解とIFGTベース法を用いて、2点以下のデータ点を個別に分類した。必然的にタイミングは線形であり、データセットの規模ではIFGTは線形であるが、データセットの規模が大きくなれば、収束までの演算の反復回数が若干増える。低次数のIFGTでは、IFGTの1反復にかかる時間は実質的に短くなるものの、精度が落ちる可能性もある。40

【0068】

ここで、カーネル行列全体を直接処理することは一般的に実用的ではないが、「マッシ

50

ュルーム」という 8 1 2 4 点、2 2 次元のデータセットについて検討する。ここに記載の IFGT 法による RLS を用いて、 $\lambda = 100$ 、2 6 0 秒で、 ϵ の範囲（例えば、 10^{-16} から 10^{+16} まで）において、このデータセットの L00 値を算出する。 $1e - 11$ となるように最適な λ を演算し、L00 値の演算精度が 8 8 . 8 % になるようにした。4 0 6 2 点のデータのサブセットには 1 3 7 秒の作業が必要であった；ここで再び、IFGT アルゴリズムの RLS は必然的に線形時間であることを確認されたし。

【0069】

<方法論>

図 6 は、本発明のある実施形態に係わる RLS 分類 / 回帰の方法を示すものである。この方法は、例えば、図 1 および図 2 を参照しつつ説明した RLS 分類 / 回帰モジュール 105 を用いて実行される。

10

【0070】

この方法はトレーニングセットのデータを受信することを含む（S501）。線形カーネルの場合にはステップ S503 を飛ばして、ステップ S505 に進む。あるいは、入力したトレーニングセットを用いて対距離でのカーネル行列を生成する（S503）。なお、本方法では、ここに記載する RLS 分類器 / 回帰モジュールのトレーニングが完了すると、実際のデータセットを受信することをさらに含む。

【0071】

次に、前記の通りのカーネル行列の固有値分解を演算する、あるいは（線形カーネルの場合には）データ行列の SVD を演算する（S505）。本発明の実施形態によっては、固有値分解と SVD の両方を演算するように構成する。本方法は、正則化パラメータ（ λ ）の集合を受信することをさらに含む（S507）。この λ の集合の範囲と分解能は、手動（ユーザ入力など）により、あるいは（ユーザ入力で指定した目標の範囲と分解に基づいて、あらかじめ記憶されたりハードコード化されたものから読み出したり、ダイナミックに演算するなどして）自動的に求めることができ、分類器の所望の精度とロバスト性などの要因に依存する。実施形態によっては、 λ の集合は 10^{-16} から 10^{+16} までの範囲であり、互いに均等な空間を置いた 200 から 400 までの中間的な λ の分解能を含む。このような広範囲性は、（例えば、小さすぎる範囲から大きすぎる範囲で）収束が可能となり、このような収束を達成するために過度の演算を必要とせずに十分な分解能が得られる。

20

【0072】

引き続いて、本方法は、（固有値分解または SVD のいずれを用いるとしても）ステップ S505 で算出した行列の分解を用いて λ ごとに係数（c）を算出する（S509）。実施形

30

態によっては、前記の $c = V(\Lambda + \lambda I)^{-1}V'y$ を用いて前記の通りに係数 c を算出する。あるい

は、（線形カーネルの場合には） $c = U(S^2 + \lambda I)^{-1}U'Y + \lambda^{-1}U_\perp U_\perp'Y$ を用いて算出することもできる。なお、前記に説明した通り、 $\text{diag}(G^{-1})$ も算出することもできる。係数 c を算出すると、次に、各 λ について L00 誤差を算出する（ステップ S511）。

40

【0073】

次に、L00 誤差が最小値の λ を選択する（S513）。前記に説明した通り、適切なを選択することは、分類器の精度を確保するために重要である。さらに、最適な λ に関する RLS 関数（例えば、最適な λ の係数 c または超平面関数 w）に基づいて将来のデータ点を予測する（S515）。分類の実用化（つまり、顔や対象、音声など、予測するもの）あるいは回帰の実用化（つまり、寿命やサラリーなど、予測する数量の大きさ）については、本開示に照らすならば、変更可能であることは明らかである。

【0074】

本発明の 2 つの実施形態に係わる L00・RLS の高速処理の 2 つの手法は、 λ の範囲全体に用いることが可能であることを再び確認されたし：つまり、中規模データセットでは $O(n^3 +$

50

n^2d 時間、小さい $O(n^2d)$ メモリの手法と、大きい α を用いる大規模データセットでは $O(nd^p)$ 時間、 $O(nd^p)$ メモリ（この場合の p は用いる IFGT の次数）の手法である。

【0075】

本発明の実施形態は、例示的な説明の目的から前記の通りの説明を行った。これは包括的なものではなく、本発明を開示された形態に厳密に限定することを意図したものではない。したがって、本開示に照らすならば各種の修正や変形が可能である。例えば、係数 c のみを正確に求め、 $\text{diag}(G^{-1})$ を求める必要がない場合には、ここに記載する Lanczos 原理での許容範囲を大きくしたり、あるいはもっと小さい α を用いることも可能である。しかしながら、大きい α の方が、 K のスペクトル崩壊が速く、IFGT も正確であるために好ましい；しかし、大きい α はすなわち小さい α を意味するため、本明細書の Lanczos 原理では許容範囲は小さくなる。従って、条件の選択を適切かつ慎重に行う必要がある。さらに、（ここに記載する）ガウスカーネルを有する RLS 分類器の考え方は、おおまかな局所学習の手法としては、小さいあるいは中程度の α に依存し；大きい α については、最も遠い点は最も近い点とほぼ同じインパクトを有する。さらに、RLS 分類器の漸近解析を検討する場合は、 α および d ならば、RLS 分類器は、 α / d のレートに依存するような次数を有する特定の多項式分類の手法に収束する。同じく、ここに記載する α を用いた手法は、RLS 分類の問題と RLS 回帰の問題の両方に等しく適用することができる。本発明の範囲は、本明細書に記載された詳細によってではなく、むしろ付属の請求項によって限定されるべきである。

【図面の簡単な説明】

【0076】

【図 1】本発明のある実施形態に係わる構成の演算環境を示すブロック図である。

【図 2】本発明のある実施形態に係わる構成の正則化二乗 (RLS) 分類器 / 回帰モジュール 105 を示すブロック図である。

【図 3】 a および d は、本発明のある実施形態に係わる、 α と d の範囲での RLS 分類器 / 回帰についての精度を示したものである。

【図 4】 a および d は、本発明のある実施形態に係わる、 α と d の範囲での RLS 分類器 / 回帰についての精度を示したものである。

【図 5】本発明の実施形態に係わる、事例データセットの $|K - FGT(K)| / ||K||$ を示したものであり、この場合の P は多項式展開の次数を表す。

【図 6】本発明のある実施形態に係わる RLS 分類 / 回帰の方法を示したものである。

【符号の説明】

【0077】

- 10 演算環境
- 101 プロセッサ
- 103 記憶装置
- 105 RLS 分類器 / 回帰モジュール
- 107 メモリ
- 109 グラフィックアダプタ
- 111 ディスプレイ
- 113 キーボード
- 115 ネットワークアダプタ
- 117 ポイントティング装置
- 119 バス
- 201 入力モジュール
- 203 カーネル行列生成器
- 205 固有値分解 / SVD モジュール
- 207 係数 (c) 演算モジュール
- 209 LOO 誤差演算モジュール
- 211 正則化パラメータ (α) 選択モジュール

10

20

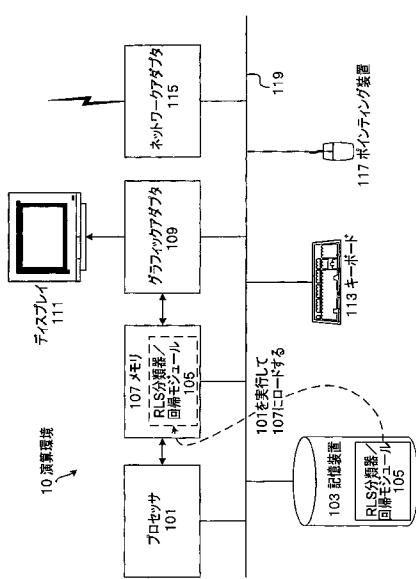
30

40

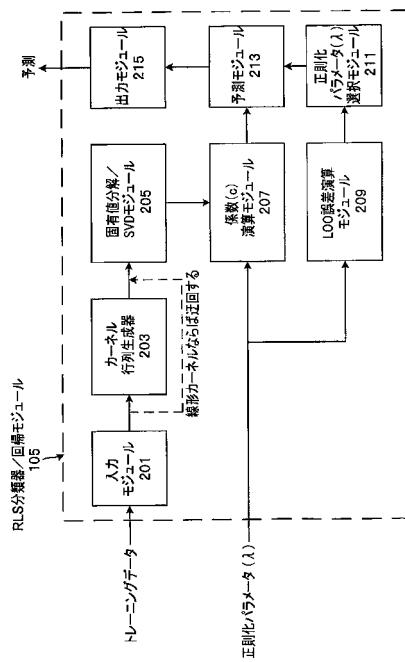
50

2 1 3 予測モジュール
2 1 5 出力モジュール

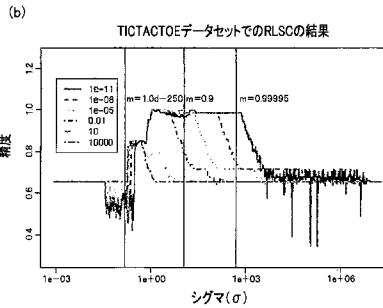
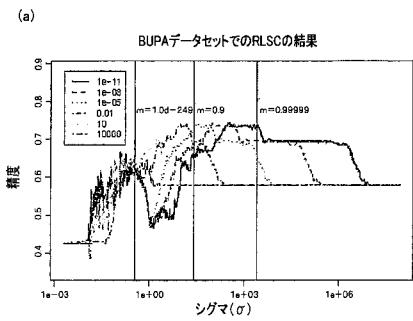
【図 1】



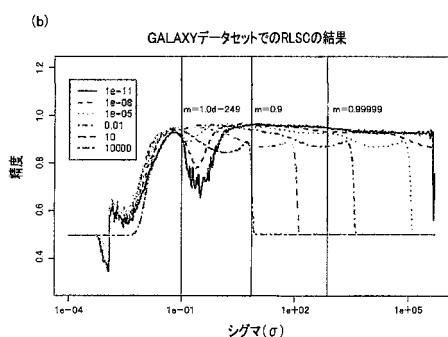
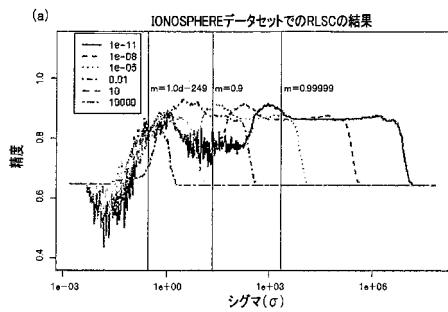
【図 2】



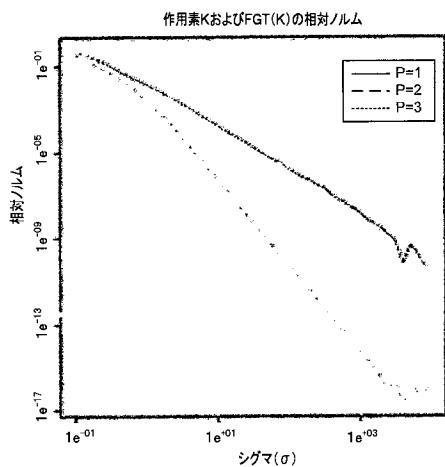
【図3】



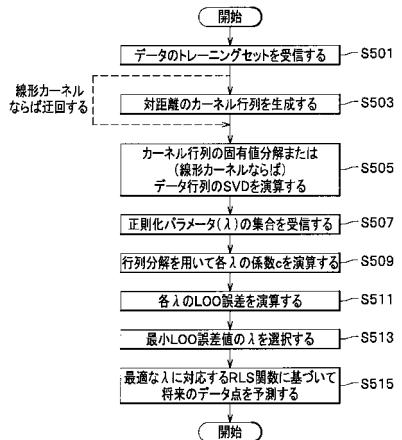
【図4】



【図5】



【図6】



フロントページの続き

(72)発明者 リパート、ロス

アメリカ合衆国、オハイオ州 93067-9705、レイモンド、ステート ルート 739
21001、ホンダ アール セブン ディ アメリカズ、リーガル デパートメント気付

審査官 北元 健太

(56)参考文献 特開2000-234951(JP,A)

特開2003-210460(JP,A)

国際公開第2004/046877(WO,A2)

R. M. Rifkin, *Everything Old Is New Again: A Fresh Look at Historical Approaches*, PhD thesis at MIT, 2002年 9月, pp.25-181

河田聰、南茂夫, 科学計測のための画像データ処理, CQ出版株式会社, 1997年 8月20日, 第3版

(58)調査した分野(Int.Cl., DB名)

G06F 17/10 - 17/18

G06N 3/00 - 3/12