データが少ない場合の対応

本章では、データが少ない場合の技術的対処について述べる。

1. データ量の重要性と必要量の見積もり

まずはデータ量の重要性を強調しておきたい。分析手法を高度化することに心血を注ぐよりも単純にデータ量を増やす方が費用対効果が高い場合が多い。現状のデータが予測用に蓄えられたものではなく、例えば経営状況の可視化のための直近3ヶ月分のデータしか保存されていないが、新たに需要予測を行いたいといった場合があるだろう。3ヶ月のデータを学習して次の月を予測すると、学習器にとって未知の季節を予測することになるため、精度を上げづらく、モデル選択も難しい。まずはデータの蓄積期間を3年間程度に延ばす施策が重要である。2年分あればゴールデンウィークのような年単位の周期性を捉えられる可能性があり、残り1年分はモデル選択や評価に使うことができる。また、人手でラベル付けするデータの場合、コストをかけてでもラベル付きデータを増やすことを検討することも重要である。小売需要予測のような一般性の高い問題の場合、類似と思われるデータを購入することも一案であろう。

必要なデータ量は、目的やデータの性質によって異なるものの、概ね次のように考えられる。線形回帰の場合、基本的には説明変数の数の数倍程度のデータ量が必要である。説明変数が10個の場合、意味のある予測を得るためには理想的なデータでサンプルサイズ20〜100点程度が最低ラインである。それ以下の場合、無理にパラメタ推定するのではなく、前時刻のデータや前周期、たとえば先週の同曜日の値をそのまま予測としたり、移動平均を予測としたり、精々それらの候補からデータに基づいてモデル選択のみ行うのがよいだろう。データのノイズ分散が大きい場合や説明変数が多い場合や、より高い精度が必要な場合、それらに対して線形に必要データ量が増える。また、時系列データでは近い時刻のデータの相関が強い場合も多く、10期（時刻）分程度は相関が強いとすると、10倍程度のサンプルが必要となる。

AICやLassoなどのスパース性を誘導する罰則項を用いて変数選択する場合は選択された後の変数の数が重要である。選択後の変数の数に対しては線形に多くのサンプルサイズが必要となるが、選択前の候補数に対しては対数オーダー程度で十分である。それも個々の変数ごとに変数選択する場合である。ARモデルの次数選択のような場合は実効的な変数の組合せの数は少ないので、候補の数はほとんど無視してよいだろう。

以上の議論は次の式におおよそまとめることができる。選択した次元数を k, 見かけの（組合せ選択前の）次元数を d, データの真のノイズ分散を σ^2, サンプルサイズを n とすると、k個の変数のみを用いた最善の（真のノイズを含まない決定論的な）予測モデル f^* と推定したモデル $\hat{f}$との予測二乗誤差の最悪値がおおよそ次のように抑えられる。

$$\|\hat{f} - f^*\|^2 \lesssim \frac{\sigma^2 k \log d}{n}$$

上式は線形回帰に対する結果[Liang+ 15]とスパース推定に関する結果[富岡 15]から主要な項を抽出したイメージである。ここで $\lesssim$ は、定数係数を除くという意味である。実用的にはその係数が気になるところであるが、実際にはあまり意味がないことが多い。というのも、真の確率分布を仮定しない理論保証は非常に保守的で、例えば実際には100点で達成できる精度に対して10,000点必要などと計算されるためである。

一方で、右辺の各変数を変えた場合の変化に関しては予測力が高いことが多く、既存の分析結果をもとに同様のデータに関して類推するには有用である。例えばサンプルサイズを10倍に増やすと誤差は1/10程度になると想定される。ただし実際に観測される誤差は最善のモデル f^* の誤差を含んだもの、つまりデータが無限にあった場合の予測誤差を加えたものであるが、これは最尤推定を行ったときの訓練誤差とテスト誤差の平均で見積もることができる。なぜならサンプルサイズに対して精度はおおよそ Fig. 1. のように推移するためである。以上により少しのデータがあれば達成したい精度に対する必要なサンプルサイズをおおまかに見積もることができる。

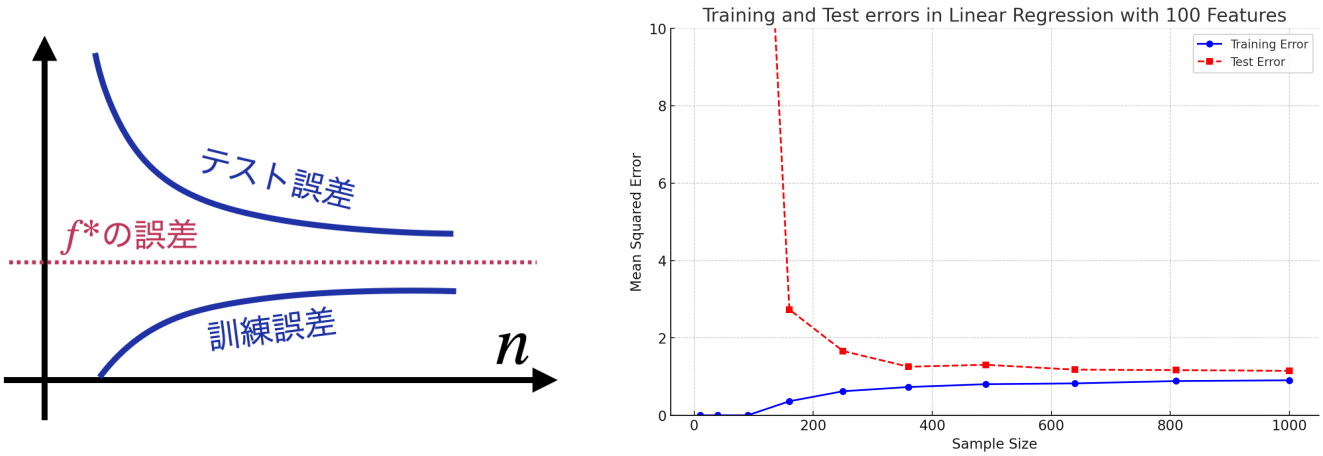

Fig. 1. 古典的な設定におけるデータ量とテスト誤差、訓練誤差、最適モデルの誤差の関係の模式図(左)と実際のシミュレーション例(右)

ただしあくまで大まかなイメージであり、当たりをつける程度にとどまる。時系列データでは非独立性や周期性の影響などにより上記理論通りにならないこともある。また、パラメタ数が比較的小さい $p < n$ の領域でのみ意味がある理論であり、ニューラルネットのようなパラメタ数の多いモデルの挙動は予想できない。モデルの実運用にあたっての性能見積もりは交差検証など実際のデータを踏まえて検証するのが妥当である。

データ量が非常に重要とはいえ、特に時系列データではデータが増やせない場合も多い。予測対象は人手によるラベルではなく自然に得られるデータであることが多い。また、マーケティング分野で既に3年分程度のデータがあるような場合、10年分に延ばしてもあまり有効ではないことが多い。消費者の志向性の変化や競争環境の変化、また東日本大震災やコロナ禍といった大きな社会変化の前後ではしばしば傾向が全く変化するためである。新商品の発売初期などさらにデータが少ない場合もある。ただしその場合は類似商品のデータや、商品マスタの活用など問題設定を変えることで擬似的にデータを増やせる場合もある。以降では、データ量が単純に増やせない場合の技術的対処法について述べる。

2. 問題設定の再検討

サンプルサイズに困ったとき、簡便さとインパクトの大きさの観点で最初に再検討すべきは問題設定、すなわち予測対象の選択と、目的変数および評価指標である。最終的にやりたいことに照らして必要最小限の問題設定を行うことが肝要である。問題によるところが大きいが、以下に幾つかの工夫の例を述べる。

類似データをまとめる/再活用する

予測対象が複数にわたる場合、それらの間でのモデルの共有化は小データにおいて主要なアプローチである。予測対象のカテゴリごとにまとめたり、新商品など特定の対象だけサンプルサイズが小さい場合は類似商品のモデルを流用したり、データを混ぜて再利用するといった対処が考えられる。どの予測対象をまとめて予測しても良いかがわからない場合、予測対象のIDや分類IDを説明変数に加えて、LightGBMなどの決定森系のモデルで推定すると、適宜モデルを切り替えてくれるかもしれない。さらに高度な手法は後の転移学習の節で議論する。

モデル化したい現象を分解する

予測対象を分解してそれぞれに学習し、学習後に組み合わせるのが強力な場合がある。たとえばコンビニの各商品の売上を予測したいとき、個々の商品の売上は桁が小さいため日々

のランダムな変動が激しく直接予測するのは難しい。そこで来店客数予測と客数あたりの売上割合予測に問題を分割すると、来店客数は桁が大きいためランダム性の影響は比較的小さく、予測しやすい。一方で来店客数あたりの売上割合は日々の変動が比較的小さいかもしれない。例えば土日と平日での売上数量の違いは主に来店客数の違いによるもので、客数あたりの売上割合は一定であれば、後者は土日か平日かを考慮しない単純なモデルで近似できる。さらに前項の方針を適用し、複数の類似店舗をまとめて1つのモデルで表現できれば、サンプルサイズを増やすことができる。

予測粒度を必要十分にする

予測の利用場面を考えて必要十分な予測問題とすることも非常に重要である。コンビニ需要予測の例で言えば、個々の商品の需要予測精度は必ずしも重要ではないかもしれない。顧客は梅おにぎりがなかったら何も買わずお茶だけ買って帰るのではなく、代わりに鮭おにぎりを買ってくれるかもしれない。したがって「おにぎり」という分類の粒度で在庫が維持できればある程度良しと考え、来店客数あたりのおにぎりの需要予測とする。実際には「梅おにぎり」の単位で発注する必要があるので、そのあとおにぎり売上数量あたりの梅おにぎり売上割合予測が必要となるが、これは一定と仮定することも考えられる。

損失関数を実務上の指標に近づける

予測が実測値に対して上振れる誤差と下振れる誤差は実務上対等ではないだろう。例えば需要予測が上振れると廃棄が発生するが、下振れると欠品時間が発生し、消費者の利便性に影響するためより重視されるかもしれない。そのような場合、分位点回帰などを使って上振れと下振れの程度を調整することで最終的な評価指標を上げられる可能性がある。

極端な値の予測は判別問題化する

数値の予測自体が重要ではない場合もしばしばある。例えば機器メンテナンスにおける劣化状態をモニタするセンサ値の推移を予測したいとき、劣化が速い箇所は稀であるため、劣化箇所のデータが少なく予測が難しい。一方このような目的変数の分布の歪度が高い（ほとんどの点で値が小さく、稀に非常に大きな値が出る）データは実際には値自体の予測ではなく、ある閾値より大きい値となるかどうかの判別だけが重要である場合も多い。劣化予測の例では、次の期間に劣化が危険水準に達するかどうかが分かれば補修の判断にとって十分である。その場合、初めから判別問題として考えた方が最終的な判断に繋がりやすいかもしれない。ただし、目的変数の数値情報を二値に削減することにより実質的な情報量が減少するという問題もある。そこで、ある程度の段階に分けて順序回帰としたり、閾値前後で0/1に区切るのではなくシグモイド関数などでソフトにラベル化するといったアプローチもある

[Tanimoto+ 22]。

3. 転移学習

前項で述べた予測対象の選定に関し、より柔軟にデータを再利用するアプローチが転移学習である。データのインスタンス自体を再利用するインスタンス転移学習が最も簡便であるが、前項と重なるため省略し、ここではパラメタ空間における転移について紹介する。

・パラメタ転移

最も標準的な転移学習アプローチの一つである。ソースタスクと呼ばれる、データが十分にある類似の問題で事前に学習した、基準となるパラメタ $\hat{\theta}_s$ を用いて、そこから著しく逸脱しない範囲でパラメタの学習を行う。典型的には以下のような損失関数を用いる。

$$\frac{1}{n}\sum_{i}^{n}(y_i - \theta^{\top}x_i)^2 + \alpha\|\theta\|^2 + \beta\|\theta - \hat{\theta}_s\|^2$$

第1項は予測誤差を表すMSE、第2項はL2正則化、第3項が基準パラメタからの逸脱を防止する項である。現状実際には、本手法だけを用いることはあまりなく、直接的に上記を実施する専用のOSSなどが整備されているわけではない。ニューラルネットの学習においてこのような基準パラメタとの類似度を一部考慮することがある程度である。ここでは転移学習の典型的な考え方として紹介した。

・特徴転移

特徴空間の距離構造（マハラノビス距離）を転移したり、特徴ごとに係数の転移/非転移を選択したりするアプローチである。中でも特徴拡張法 (Feature Augmentation Method) [Daumé III 07] は、 いらいらするほど簡単な方法とも呼ばれ、既存の線形モデルやカーネル法と簡単に組み合わせることができる。その方法は、ソースタスクの説明変数を $x_s \in \mathbb{R}^d$ を $(x_s, x_s, 0) \in \mathbb{R}^{3d}$ という特徴へ、目標タスクの説明変数 $x_t \in \mathbb{R}^d$ を $(x_t, 0, x_t) \in \mathbb{R}^{3d}$ へと3倍の長さに拡張し、全体のデータで同時に学習する。最初の d次元空間はソースと目標で共有された空間であり、係数に正則化をかけると、両ドメインで係数を共有できる変数に関してはこの部分が使われる。そうでない場合は2つ目の d 次元空間の中からソースタスクへ、3つ目のd次元の中から目標タスクへそれぞれ異なる係数が自然と用いられる。ソースタスクが2つある場合は4倍の空間とすればよい。

・ファインチューニング

特徴ベースに近いアプローチであり、深層学習の登場以降もっぱら用いられる転移学習手

法である。最近は転移学習というと断りなくファインチューニングのことを指すことも多い。深層学習の学習方法は現在基本的には損失関数の一次微分を用いる（確率的）勾配降下法であり、パラメタ空間を少しずつ勾配方向に移動することで学習が進む。そこで、ソースタスクで学習したパラメタ $\hat{\theta}_s$ を初期値として、目的のデータで少しの追加学習を行うことにより、実質的に前述のパラメタ転移のように $\hat{\theta}_s$ に近いパラメタとなる。実装としても初期値を設定するだけで使え、追加学習にかかる計算も少なく済むため非常に簡便である。

・表現転移

深層モデルの場合、表現層と呼ばれる入力に近い層のパラメタをソースタスクで学習して固定し、仮説層と呼ばれる出力に近い層、特に最終層のパラメタのみ学習する方法である。ファインチューニングでも、仮説層が比較的大きく変化することが多いが、より明確に、全く異なるタスクと考えられる場合に用いる。予測に必要な因子（表現）はソースタスクと共通であろうという仮定に基づいている。

・マルチタスク学習（深層学習以外）

パラメタ転移学習はソースタスクの学習後にそのパラメタを用いる、という一方通行のアプローチであったのに対し、マルチタスク学習は多数の対等なタスク群を同時に、相互に知識を転移しながら学習する方法である。パラメタ転移学習の損失関数においてソースと目標タスクのパラメタを同時に学習することで相互に類似させるのが基本的な方法である。

タスクが多数ある場合、一律にそれらを相互に近づけると、それらの間の異質性を無視することによるバイアスが大きくなる。そこで類似したタスクを自動的にクラスタリングしながら同時学習するのが Network Lasso [Hallac+ 15] である。 以下の損失関数を最小化する。

$$\frac{1}{nT}\sum_{t}^{T}\sum_{i}^{n}(y_{t,i}-\theta_t^{\top}x_{t,i})^2+\frac{\alpha}{T}\sum_{t}^{T}\|\theta_t\|_2^2+\frac{2\beta}{T(T-1)}\sum_{t<t'}w_{t,t'}\|\theta_t-\hat{\theta}_{t'}\|_2$$

第3項のタスク間のパラメータの差に対する正則化が、2乗ではなくL2距離の1乗となっていることがポイントである。これがもし2乗ならば、微分が1乗であるから、パラメタ同士が近くなるほど勾配が小さくなる。しかし1乗ノルムでは、パラメタ同士がぴったりと一致するまで勾配は一定であるため、一致しやすくなる。結果として、まとめても訓練精度に影響が少ないタスクが自然とクラスタリングされる。逆に、全く異なるタスクは遠く離れても勾配が一定のままであるため、距離の2乗に基づく正則化よりは係数を一致させる力が弱く、タスクの多様性を維持しやすい。

Fig. 2 は、5クラスタからなる25個のタスク、即ち回帰係数を生成し、元のクラスタが全く

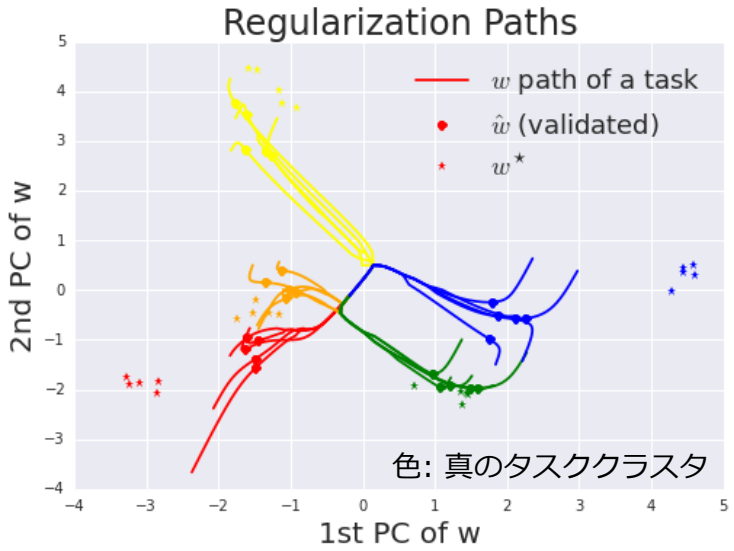

Fig. 2 正則化パラメタβを変化させたときの解の軌跡（$w_{t,t'}=1$）。w*は真の係数、$\hat{w}$はモデル選択された係数である。

未知の状態でNetwork Lassoを用いて全タスクを同時学習し、その正則化パラメタβを徐々に変化させたときの解の軌跡をプロットしたものである。色で示される真のタスククラスタごとにパラメタが集約される傾向が確認できる。

ただ、それでも全てのタスク間で正則化をかけると中心に集まる力が強いことから、ある程度の類似度を知識として与えるために重み $w_{t,t'}$ を用いることができる。例えば店舗間の距離が近い場合のみ$w_{t,t'}=1$とし、それ以外を0としたり、商品間の類似度として「食料品」という括りが同じなら $w_{t,t'}=0.3$ とし、「おにぎり」まで同じなら $w_{t,t'}=1.0$ とする、などが考えられる。

・マルチタスク学習（深層学習）

深層モデルにおけるマルチタスク学習は主に表現転移の形で行われる。即ち、仮説層、特に最終層のみを各タスク専用とし、それまでの層は共有としたモデルを各タスクに対して同時に訓練する。

・ドメイン適応

転移学習の中でも、学習すべきタスク、即ち $p(y|x)$ は単一で、入力変数分布 $p(x)$ が異なる設定をドメイン適応という。例えば暗視カメラによる赤外線画像を用いた画像認識のように、赤外線画像のデータが十分にないが、可視光のRGB画像は十分にある、といった場合が考えられる。画像は十分に情報量があり、どちらの画像を用いても人間には十分認識可能である

ことから $p(y|x)$ は同一と見なして問題ないだろう。マーケティングのような場合では、気温の影響など、観測可能な変化や店舗間の違いはドメイン適応が有効な場合も考えられる。一方で観測されていない変化、例えば消費者の意識変化や近隣店舗の値下げといった外部影響があると $p(y|x)$ 自体に変化があるかもしれない。しかし、 $p(y|x)$ が変化する問題は基本的に非常に学習が困難である。入力変数の分布が異なることがわかっている場合、 $p(x)$ のみが異なり $p(y|x)$ は一貫しているという仮説を試してみる価値はあるだろう。

・教師なしドメイン適応

ドメイン適応の中でも、目標ドメインの目的変数ラベルが全く得られない状況を考える。これに対する手法としては深層ニューラルネットを用いた敵対的ドメイン適応 [Tzeng+ 17] がよく知られている。入力から出力までの中間の特定層の出力を特徴表現と呼び、ここでドメインに依存しない表現を得るように表現抽出層を訓練する。具体的には、特徴表現からドメインを識別するニューラルネットを同時に訓練し、表現抽出層はこのドメイン識別精度を悪くするような表現抽出を行うように訓練する。これにより例えば、RGB画像の色相によらない、明度に基づいた特徴が抽出され、それは赤外線画像でも有効と期待される。

なお、同様の設定において半教師あり学習が適用可能と思われるかもしれない。半教師あり学習は、一部のデータのみラベルが付けられ、大半はxだけが観測されるという設定である。しかし、時系列データ解析の多くは原因から結果を予測する問題であり、半教師あり学習は因果の逆方向のモデルを学習する場合にのみ有効であることが知られている [Schölkopf+ 12]。画像認識のように、真の画像ラベル $y \rightarrow$ 画像 x という因果関係に対して、逆方向である$p(y|x)$を推論する場合にのみ有効であり、時系列に沿った予測のように $x \rightarrow y$ の因果関係に対して順方向に予測する場合には有効ではない。これは一言で言えば、 $p(x)$ が $p(y|x)$ に関する情報を持っていないため、入力側のデータだけいくら集めても $p(y|x)$ に関するヒントにはならないのである。

4. スパース性を活用した仮説空間の絞り込み

サンプルサイズが小さいときにまず考えるべきことは、予測対象の問題に関する人間の知識を活用して仮説空間、つまりモデルの候補集合を絞り込むことであろう。その内容は予測対象の問題ごとに異なるため、一般的に言えることは少ない。また、知識を柔軟にモデルの候補集合に反映させる方法としてはベイズ推論が挙げられるが、その方法はベイズ推論の章に譲る。本章では、考えられる複数の仮説空間を設計し、その中から有効な組み合わせを選択す

Fig. 3 スライディング窓は、一定期間のデータを1点とするデータを、1時刻ずつずらしてサンプリングする

るという汎用的なアプローチに着目し、その鍵となるスパース性を誘導する学習法について述べる。ただし、仮説空間の選択に費やせる程度にはそれなりに大きいサンプルが前提となる。

データ量の見積もりの項で述べたように、使用する説明変数の数、即ちゼロでないパラメタ数に比例して大きいサンプルが必要となる。一般に、より多くの説明変数を使えば近似誤差が減る一方で、サンプルサイズが限られるときの推定誤差が増えるため、有効と思われる説明変数であってもあえて無視する方がよい場合もある。

ARIMAモデル等の時系列モデルの多くは、そのパラメタサイズとともに ARIMA(p,d,q) などと表示され、そのハイパーパラメタ (p,d,q) の選択が問題となる。その選択には時系列CVや walk forward validation といった時系列向けのモデル評価手法を基本とすべきである。簡便な手法として、Auto-ARIMA は、情報量基準を用いて自動的にハイパーパラメタを決定する実装である。ただし、情報量基準は i.i.d. を仮定しているため、時系列データでは基本的に理論保証がないことに注意が必要である。特に、自己相関が強い場合や、x_t の時系列から $(x_1, x_2, x_3), (x_2, x_3, x_4), \ldots$ のようにスライディングウィンドウによるサンプリングを行う場合がよくあるが、これらは独立ではなく相関が高いと考えられる。つまりサンプルサイズを水増ししたような状況となり、パラメタの多い複雑なモデルが選択されやすいため注意が必要であ

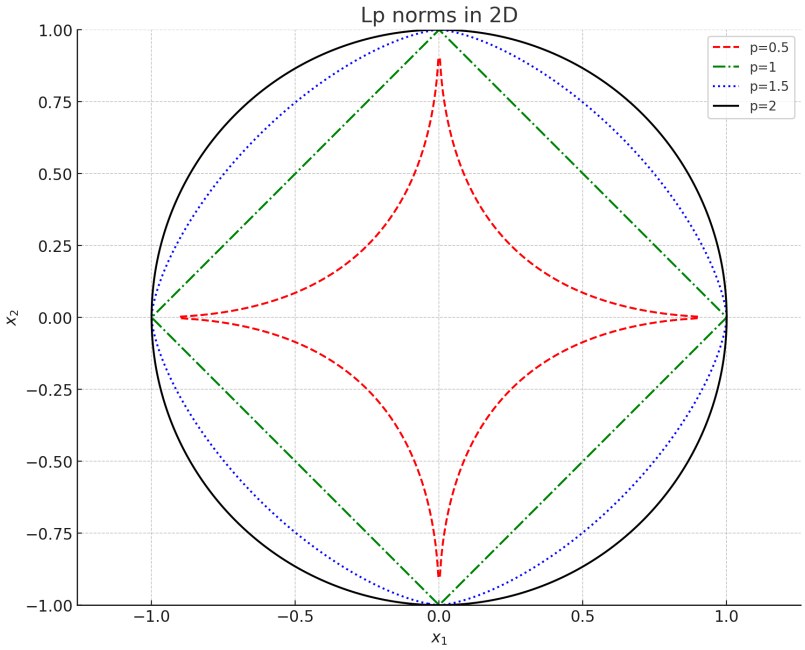

Fig. 4 L_pノルムの等高線。pが小さいほど、x_1, x_2の両方が非ゼロの領域の$\|x\|_p \leq 1$の範囲が狭く、したがって少なくとも片方を 0 に近づける効果が高い

る。

目的変数の時系列以外の説明変数や、任意に作った特徴量など、変数を使用する優先度がついていない場合、各変数ごとに取捨選択する必要がある。しかしその組合せすべてに対して情報量基準やモデル評価手法で評価するのは現実的ではない。そこで取捨選択を自動化するのがスパース性を誘導する正則化項である。代表的なものは Lasso や、L2正則化を加えた Elastic Net である。線形モデルを仮定するとその損失関数は以下のような形式である。

$$\frac{1}{n}\sum_{i}^{n}(y_i - \theta^\top x_i)^2 + \alpha\|\theta\|_2^2 + \beta\|\theta\|_p$$

第1項は予測誤差を表すMSE、第2項はL2正則化、第3項がスパース性を誘導するLp正則化 $(0 \leq p < 2)$ である。α, βは正則化の強さを表すハイパーパラメタであり、モデル選択が必要である。

Lpノルムは $0 < p$ のとき $\|x\|_p = \left(\sum_m x_m^p\right)^{\frac{1}{p}}$ で定義される量であり、$p = 0$ のとき非ゼロの次元の個数を表す。いくつかの p について$\|x\|_p = \|(x_1, x_2)\|_p = 1$ の等高線を Fig. 2 に示す。 p は小さいほどスパース性を強制する力が強く、非ゼロの場合の絶対値の大きさ自体には罰則

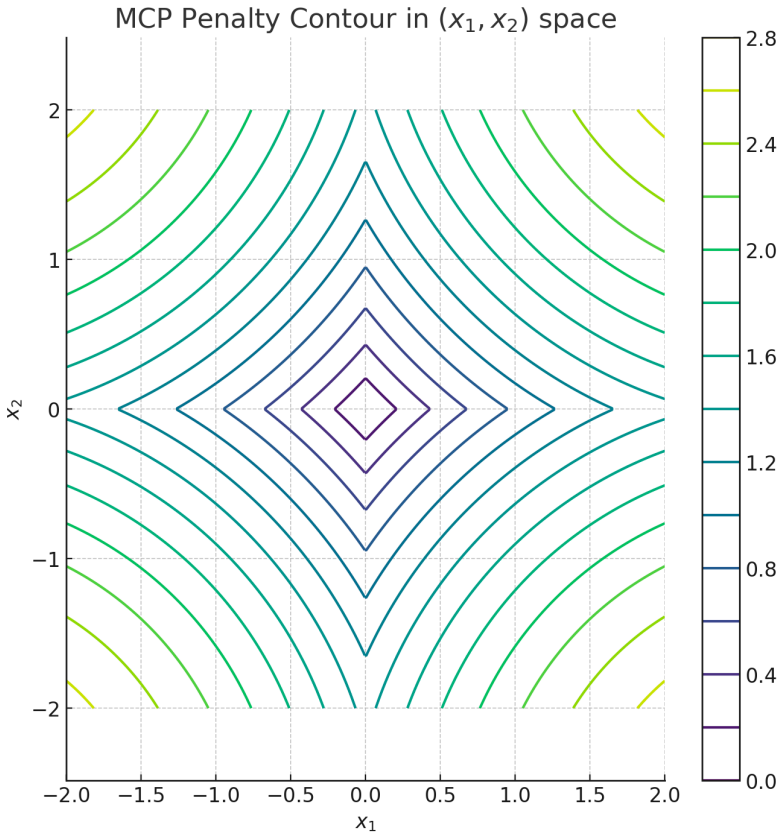

Fig.5 MCP罰則項の等高線

を課しにくくなる。AICやBICなどの情報量基準は$p = 0$に相当する。$0 \leq p < 1$ のとき損失関数は非凸であり、最適化計算が困難になる。$p = 1$、つまりLassoは凸であり計算が易しく、その中で最もスパース性を誘導する力が強いためしばしば用いられる。

さらに、p を小さくしつつ最適化計算も現実的であるような正則化として、SCAD (Smoothly Clipped Absolute Deviation) やMCP (Minimax Concave Penalty) といった、係数の絶対値が大きい範囲では $p = 0$ に近く、ゼロに近いとき $p = 1$ に近くなるような正則化も提案されている。MCPの等高線を Fig. 3 に示す。係数の絶対値が大きい、真に重要な変数に対しては勾配を小さくして推定の偏りを低減しつつ、小さい係数においては凸となり計算効率よく 0 に縮小させる。

多変量時系列間の相互作用をモデル化したVARモデルは、各変量間の時間差での影響を、行列のパラメタとして表現する。たとえば、道路の各箇所の渋滞状況は時間差で近隣の箇所へ伝播するため、箇所ごとにモデル化するよりも全箇所をベクトルとして扱い、各次元の間の相互作用を考慮に入れる利点があるだろう。ここで、交互作用を考えるとパラメタの数は増加するため、スパース性を活用するなど仮説空間の絞り込みが重要になる。

この場合、行列要素のスパース性に加えて、ランクという別のスパース性を考えることができる。d次元ベクトル $x \in \mathbb{R}^d$ の時系列に対する1次のVARモデル $x_{t+1} = A x_t$ のパラメタ $A \in \mathbb{R}^{d^2}$ に対して、以下の特異値分解が一つ定まる。

$$A = U \Sigma V^\top$$

ここで U, V はそれぞれユニタリ行列であり、回転を表し、Σ は対角行列であり、軸に沿った拡大縮小を表す。すなわち行列Aを掛けることは、x を回転させ、それぞれの軸に沿って拡大縮小し、再度回転させる操作として表現される。対角行列 Σ の対角成分を $\sigma = (\sigma_1, \sigma_2, \cdots)$ としたとき、その要素の大部分が 0 であれば、その軸に対応するU, V のパラメタは 0 が掛けられるため意味をなさなくなる。σの非ゼロの成分の数をランクといい、ランクが k のときパラメタ数は$2dk$程度となり、$2k < d$であれば元のパラメタ数d^2に対して少なくできる。学習はLassoと同様に、各特異値の絶対値の和を核型ノルムと呼び、これを正則化項とする。

$$\|A\|_* := \sum_j^d |\sigma_j|$$

核型ノルム正則化（低ランク性）の解釈は次のようなものである。観測される高次元量 x が本質的には低次元 (k次元) の空間での状態遷移に支配されていると仮定し、その状態空間へ射影してその空間で状態遷移を表す拡大縮小を適用したのち高次元に戻して予測値としている。このような解釈には線形代数に関する知識または慣れが必要であるが、もし直観的に納得が得られないとしても、多くの場合にパラメタ数を劇的に減少させることができ、計算的にも安定しているため、高次元の行列パラメタを扱う場合は基本的に試す価値があるように思われる。

Network lassoや核型ノルムのように、スパース性はどのような空間で非ゼロ要素数をカウントするかによって様々なバリエーションが存在する。いずれの方法も、仮説空間を何らかの捉え方で取捨選択することによって、選択後の実質的な仮説空間をシンプルに抑えつつ現象をうまく捉える戦略であると整理できる。逆にいえば、一旦広い仮説空間を考えつつ、その中にあると想定されるシンプルな構造の候補を、要素として書いたときの非ゼロ要素数として表現できれば、スパース性を活用できる。

5. 長期先予測とデータ拡張

真に予測したいのが次の時刻である場合は実務上多くないだろう。小売りにおける需要予

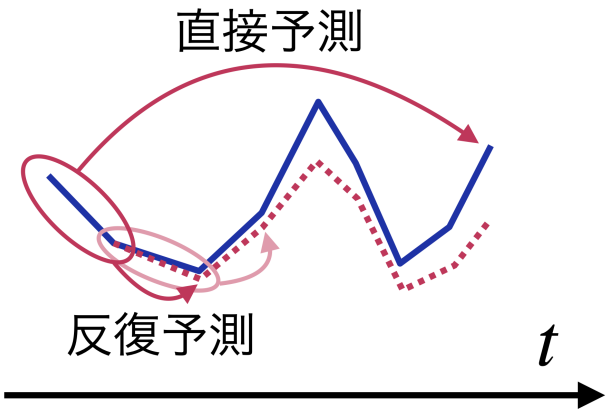

Fig. 6 直接多ステップ先予測と反復予測。
反復予測では、1期先予測モデルを繰り返し
適用して目的の期を予測する

測であれば、賞味期限等の関係で12時間ごとに入荷があり、その時間枠での需要を予測する必要がある一方、発注からの所要時間を考えると2、3日前に予測が必要となるかもしれない。そのような場合、1期先予測モデルを学習し、1期後の予測値を入力として2期先を予測、と繰り返す反復予測か、直接3日後の需要を予測するか、どちらが好ましいであろうか。

その選択方針は次のように考えることができる。反復予測アプローチは、データ量を確保しやすい一方、予測値を入力として予測するため誤差が累積しやすい。特にモデルクラスの誤指定による近似誤差が問題となる。製造業におけるセンサ系列など、比較的素性がよくわかっているシステムを対象とする場合は問題は少ない。一方、需要予測を含む消費者行動のモデリングなど、素性がよくわからないシステムを対象とする場合、モデルクラスに真のモデルが含まれることは仮定できない。そのような場合、近似誤差が累積する繰り返し予測ではなく予測先の期を直接予測する方がよい場合が多い。

直接予測では、予測する時刻と予測先時刻の中間的な時刻のデータ捨ててしまう分、学習に用いるデータ量が限られる。モデルクラスの誤指定による誤差の累積と、データ量の少なさによる推定誤差のトレードオフがある。最終的には検証データでの精度をもって方式を選択するのがよいだろう。

直接予測と反復予測の折衷案的なアプローチは様々提案されている。例えば、直接予測を基本としつつ、捨てていた中間データを再利用し、目的変数をデータ拡張する手法である[Hayashi+ 19]。Fig. 6のように、中間データ $\left(x_{t+1}, \cdots, x_{t+\tau-1}\right)$ やその一部を入力として目的変数

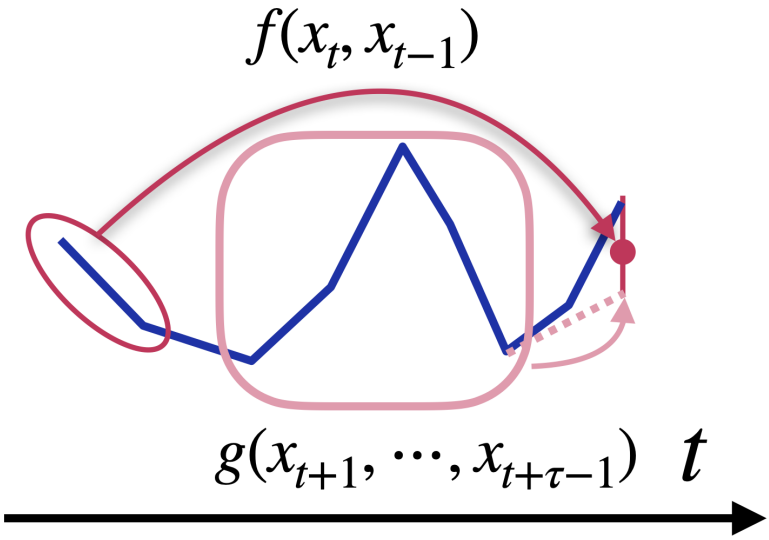

Fig. 7 中間データを一般化蒸留によって再利用するアプローチ

を予測する補助モデル g を学習し、その予測値と、元々の目的変数を一定割合 $\lambda \in [0,1]$ で混合した値を目的変数とする。

時系列に限らず、最終的なモデルと異なる入力変数を用いた目的変数の予測値を教師情報として用いるアプローチを一般化蒸留という。予測時に入手できないが訓練時のみ存在する、目的変数に対する予測力の高い変数を活用するために用いる。この手法は、一般化蒸留を時系列データに応用したものといえる。

特にgを反復予測モデルとしたとき、反復予測と直接予測の中間的な性質となる。すなわち、モデルの指定が正しい下でのモデル推定誤差は反復予測が最小、直接予測が最大であり、一般化蒸留を用いるとその中間となる [Karlsson+ 22]。

一方でモデル誤指定の際の近似誤差の影響についてはその逆となる。よって、データの少なさによる推定精度か、モデル誤指定による近似誤差のうちどちらがより問題であるかに応じて、補助モデルの予測値と元々の目的変数をどの程度重視するかという割合 λ をチューニングするのがよい、ということになる。ただし、この辺りのチューニングによる精度の違いは非常に微妙であり、直接予測や反復予測と比べて必ず改善するわけではない。どうしても中間が気になる場合に検討する程度と考えてよいだろう。

最後に、時系列予測のほかに、スマホのセンサ系列から行動を認識するといった、時系列分類問題もある。この場合、分類問題で標準的に用いられるデータ拡張戦略の多くが適用可

能である。たとえば、2点のデータを説明変数空間で、ラベルをワンホットエンコーディングした空間でそれぞれ内分点を取ることでデータを拡張するmixup戦略などがあるが、本章の範囲を超えるため、詳細は [Iwana+ 21] などを参照されたい。

まとめ

時系列分析においてデータ量の少ない場合は銀の弾丸のような汎用的で効果的な対処法は現状基本的にない。問題ごとのドメイン知識を活用するか、データを増やすか、類似のデータを活用するといった方針が基本になる。その上で、類似データを候補から選択する手段としてのマルチタスク学習（Network lasso）や、スパース性を活用する方法、繰り返し予測をデータ拡張として部分的に活用する時系列一般化蒸留などの手段を検討するのがよいだろう。

参考文献

Daumé III, Hal. "Frustratingly Easy Domain Adaptation." *ACL*. 2007.

Liang, Tengyuan, Alexander Rakhlin, and Karthik Sridharan. "Learning with square loss: Localization through offset rademacher complexity." *Conference on Learning Theory (COLT)*. PMLR, 2015.

Hallac, David, Jure Leskovec, and Stephen Boyd. "Network lasso: Clustering and optimization in large graphs." *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining (KDD)*. 2015.

Hayashi, Shogo, Akira Tanimoto, and Hisashi Kashima. "Long-term prediction of small time-series data using generalized distillation." *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019.

Iwana, Brian Kenji, and Seiichi Uchida. "An empirical survey of data augmentation for time series classification with neural networks." *Plos one* 16.7. 2021.

Karlsson, Rickard KA, et al. "Using time-series privileged information for provably efficient learning of prediction models." *International Conference on Artificial Intelligence and Statistics (AISTATS)*. PMLR, 2022.

Schölkopf, Bernhard, Dominik Janzing, Jonas Peters, Eleni Sgouritsa, Kun Zhang, and Joris Mooij. On causal and anticausal learning. In *29th International Conference on Machine Learning (ICML)*. 2012.

Tanimoto, Akira, et al. "Improving imbalanced classification using near-miss instances." *Expert Systems with Applications*. 2022.

冨岡亮太. スパース性に基づく機械学習. 講談社, 2015.

Tzeng, Eric, Judy Hoffman, Kate Saenko, and Trevor Darrell. "Adversarial discriminative domain adaptation." In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*. 2017.