

サンプルエントロピー測度に基づくニューラルネットワークのオンライン学習アルゴリズムと金融時系列予測への応用

Online Learning Algorithm of Nerual Network based on Sample Entropy Measure and Its Application to Financial Time Series Forecasting

80716542 杉崎弘一 (Koichi Sugisaki) Supervisor 大森浩充 (Hiromitsu Ohmori)

1 緒論

世の中には様々な複雑なシステムが存在する。特に金融市場は高い変動性、非線形性、不規則性が内在する非常に複雑なダイナミクスを持つため、予測は大変困難である。それゆえに金融時系列の予測は学術的にも、実用的にも大きな注目を浴びている。多変量モデルであるニューラルネットワーク (NN) は、予測対象の基礎となるダイナミクスに対し特定の仮定をおく必要のないデータ駆動型の手法であり、入出力関係に対する万能近似能力をもつため、金融時系列予測に幅広く用いられてきた。しかしながら、一般的な NN の誤差逆伝播法 (BP) は学習レートの設定が難しく、ネットワークが常に入出力関係を正しく学習するとは限らない。特に予測対象が時変の場合、その汎化性能は著しく低下する。そこで本研究では、これらの問題を改善するために、クロスサンプルエントロピー (CSampEn) を用いてネットワークの入出力の関連性のある程度事前情報として与え、入力に付加し、その関連性をリアルタイムで補正して学習を行うオンライン学習アルゴリズムを提案する。またその有効性を実際の金融時系列予測に適用することで検証した。

2 CSampEn の定義

CSampEn は時系列の不規則性を定量化する SampEn を時系列同士の関連性を定量化できるように拡張したものであり、時系列のパターンの同調性を評価することによって二つの異なるが互いに関連性のある時系列を比較するのに用いられる [1]。

N 個のサンプルから成る時系列, $\{u(g) : 1 \leq g \leq N\}$ と $\{v(g) : 1 \leq g \leq N\}$ において, $N - m$ 個の基準ベクトル $\mathbf{x}_m(i)$, $\{i | 1 \leq i \leq N - m\}$ と $\mathbf{y}_m(j)$, $\{j | 1 \leq j \leq N - m\}$ を生成する。ここで, $\mathbf{x}_m(i), \mathbf{y}_m(j)$ は, それぞれ $\mathbf{x}_m(i) = \{u(i), \dots, u(i + m - 1)\}$, $\mathbf{y}_m(j) = \{v(j), \dots, v(j + m - 1)\}$ を満たす m 個の成分を持つベクトルである。 m は正の整数, r はベクトル同士の類似性を決定する許容範囲である。このとき, CSampEn は次式で定義される。

$$CSampEn(m, r, N)(v \| u) = -\ln \frac{\sum_{i=1}^L \sum_{j=1}^L s_{i,j}^{m+1}}{\sum_{i=1}^L \sum_{j=1}^L s_{i,j}^m} \quad (1)$$

ただし,

$$s_{i,j}^m = \begin{cases} 1 & d[\mathbf{x}_m(i), \mathbf{y}_m(j)] \leq r \\ 0 & d[\mathbf{x}_m(i), \mathbf{y}_m(j)] > r \end{cases}$$

$$d[\mathbf{x}_m(i), \mathbf{y}_m(j)] = \max_{[k \in 0, m-1]} |u(i+k) - v(j+k)|$$

ここで, $u \equiv v, i \neq j$ の時は通常の SampEn の定義となる。一般に比較する二つの時系列のスケールは異なるため, CSampEn を使用するときには時系列を $u^*(i) = \frac{(u(i) - \bar{u})}{\sigma_u}$, $v^*(i) = \frac{(v(i) - \bar{v})}{\sigma_v}$ と平均 0, 分散 1 に正規化して用いる [1]。

3 変動重み付き CSampEn

CSampEn をリアルタイムで推定する場合, 時間シフトする窓を設けて新しくデータが入ってきたら, 過去のデータを 1 つ捨てるということを随時行ってゆく。 $\rho_i(t) = \rho(t)^{L-i}$ として指数重み付けすることで, システムの状態変化に敏感に反応できるアルゴリズムを提案する。CSampEn は次式のように逐次的に求めることができる。

$$fCSampEn(t+1) = fCSampEn(t) + \ln \left\{ \frac{\frac{fB^m(t+1)(v \| u)}{fB^m(t)(v \| u)}}{\frac{fB^{m+1}(t+1)(v \| u)}{fB^{m+1}(t)(v \| u)}} \right\} \quad (2)$$

ただし

$$fB^m(t+1)(v \| u) = fB^m(v \| u) + \left(\frac{\sum_{k=1}^L \rho_k(t+1) s_{k,L}^m(t+1)}{H(t+1)} + \frac{\sum_{k=1}^{L-1} \rho_k(t+1) s_{L,k}^m(t+1)}{H(t+1)} \right) - \left(\frac{L\rho_1(t) \sum_{k=1}^L s_{k,1}^m(t)}{H(t)} + \frac{(L-1)\rho_1(t) \sum_{k=2}^L s_{1,k}^m(t)}{H(t)} \right) + \sum_{i,j=2}^L \left(\frac{\rho_{\min(i,j)}(t+1)}{H(t+1)} - \frac{\rho_{\min(i,j)}(t)}{H(t)} \right) s_{i,j}^m(t)$$

$$H(t) = \sum_{k=1}^L (2k-1) \rho_{L-k}(t)$$

重みは次の評価関数を最小化するように決定する。

$$J(t) = \sum_{k=1}^{N-n+1} \lambda^{N-n+1-k} (fCSampEn(\rho) - CSampEn(k))^2 \quad (3) + \frac{\alpha}{L} \sum_{k=1}^L (\rho^{L-k}(t) - \rho^{L-k}(t-1))^2$$

ここで, $CSampEn(m, r, n)(v \| u), n(n \ll N)$ は, 現在の状態の不規則さの推定値である。 λ, α はそれぞれ固定された忘却係数, 正則化パラメータである。式 (3) を最小にする重み $\rho(t)$ は Gauss - Newton 法により逐次的に求める。

$$\rho(t) = \rho(t-1) - \mu(t)^{-1} \frac{dJ(\rho(t-1))}{d\rho} \quad (4)$$

$$\mu(t) = \sum_{k=1}^{N-n+1} \lambda^{N-n+1-k} \left(\frac{dfSampEn(\rho(t-1))}{d\rho} \right)^2 + \frac{2\alpha}{L} \sum_{k=1}^L (L-k)^2 \rho^{2(L-k)}(t-1)$$

4 最適学習レートをを用いたオンライン学習アルゴリズム

一般に BPNN において, 適切な固定学習レートを決定することは非常に困難である。学習レートの値が大きすぎると収束は速いが, 不安定で全く学習していない可能性がある。反対に学習レートが小さすぎると学習の安定性は保証されるが, 学習に膨大な時間がかかり, 収束速度が遅い。またリアルタイムでの予測を行いたい場合, 従来のバッチ型アルゴリズムよりも新しいデータが利用可能になったとき, そのデータを逐次学習しつつ, 予測を行えるオンライン学習アルゴリズムの方が望ましい。本節で紹介するオンライン学習アルゴリズムは, Yu, Kai らによって提案された最適学習レートをを用いる [2]。時刻 t を新たな訓練ペアが提示された瞬間としたとき, 入力層に p ノード, 中間層に q ノード, 出力層に k ノードを持つ三層 BPNN の出力は下記のように表せる。

$$\hat{Y}(t) = F_2 \left[V^T (t-1) F_1 (W(t-1) X(t)) \right] \quad (5)$$

ここで、 $X = (x_0, x_1, \dots, x_p)^T \in R^{(p+1) \times 1}$ は、BPNN の入力ベクトルであり、 $\hat{Y} = (\hat{y}_0, \hat{y}_1, \dots, \hat{y}_k)^T \in R^{k \times 1}$ は BPNN の出力ベクトルを表す。また

$$W = \begin{bmatrix} w_{10} & \dots & w_{1q} \\ \dots & \dots & \dots \\ w_{q0} & \dots & w_{qp} \end{bmatrix} \in R^{q \times (p+1)}, V = \begin{bmatrix} v_{10} & \dots & v_{k0} \\ \dots & \dots & \dots \\ v_{1q} & \dots & v_{kq} \end{bmatrix} \in R^{(q+1) \times k}$$

$F_1(W(t-1)X(t)) = (F_1(\text{net}_0(t)) \dots F_1(\text{net}_q(t)))^T \in R^{(q+1) \times 1}$, $\text{net}_i(t) = \sum_{j=0}^q w_{ij}(t-1)x_j(t)$, $i = 0, 1, \dots, q$ は、 i 番目の中間ノードの出力である。また $F_1(\bullet)$, $F_2(\bullet)$ は、活性化関数であり、 F_1 は \tanh 関数、 F_2 は線形伝達関数を用いる。評価関数は次のように与える。

$$E(t) = \frac{1}{2} \sum_{j=1}^t \sum_{i=1}^k e_{i,j}^2(t) + \lambda E(t-1) \quad (6)$$

$$e_j(t) = [e_{1,j}(t), e_{2,j}(t), \dots, e_{k,j}(t)]^T \in R^{k \times 1}, j = 1, \dots, t$$

ここで、 λ , $0 < \lambda \leq 1$ はモメンタムを表し、 t は訓練ペアの数を表す。また $e_j(t)$ は時刻 t で考慮される重みを用いて導出された j 番目の訓練ペアにおける誤差を表す。重みは式 (6) に対し、勾配法を適用することで、更新していく。また最適学習レートは下記の式で与えられる。

$$\eta(t) = \frac{[\xi(t)e(t) - \lambda\zeta(t-1)]^T e(t)}{[\xi(t)e(t) - \lambda\zeta(t-1)]^T [\xi(t)e(t) - \lambda\zeta(t-1)]} \quad (7)$$

ただし、

$$\xi(t) = F_2' \left[(F_1^T F_1) I_{k_2} + (X^T X) \left(\bar{V}^T F_1' F_1 \bar{V} \right) \right] F_2'$$

$$\zeta(t-1) = F_2' \left[\nabla_{\bar{V}}^T E(t-1) F_1 + \bar{V}^T F_1' \nabla_W E(t-1) X \right] F_2'$$

$$F_2' = \begin{bmatrix} F_{2(1)}' & 0 & 0 \\ \dots & \dots & \dots \\ 0 & 0 & F_{2(t)}' \end{bmatrix}, F_1^T F_1 = \begin{bmatrix} F_{1(1)}^T F_{1(1)} \dots F_{1(1)}^T F_{1(1)} \\ \dots & \dots & \dots \\ F_{1(t)}^T F_{1(1)} \dots F_{1(t)}^T F_{1(1)} \end{bmatrix}$$

$$X^T X = \begin{bmatrix} x_1^T x_1 & \dots & x_1^T x_t \\ \dots & \dots & \dots \\ x_t^T x_1 & \dots & x_t^T x_t \end{bmatrix}, F_1' F_1' = \begin{bmatrix} F_{1(1)}' F_{1(1)}' \dots F_{1(1)}' F_{1(1)}' \\ \dots & \dots & \dots \\ F_{1(t)}' F_{1(1)}' \dots F_{1(t)}' F_{1(1)}' \end{bmatrix}$$

ここで、

$$F_1' = \text{diag} [f_{1(1)}' \dots f_{1(q)}'] \in R^{q \times q}$$

$$F_2' = \text{diag} [f_{2(1)}' \dots f_{2(k)}'] \in R^{k \times k}$$

$$f_{1(i)}' = f_{1'}(\text{net}_i) = \frac{\partial f_1(\text{net}_i)}{\partial \text{net}_i}, i = 1, 2, \dots, q$$

$$f_{2(i)}' = f_{2'}[v_i^T F_1(WX)] = \frac{\partial f_2[v_i^T F_1(WX)]}{\partial [v_i^T F_1(WX)]}, i = 1, 2, \dots, k$$

また \bar{V} はバイアス成分を除いた純粋な重み行列を表す。モメンタムは下記の式にしたがって適応的に変化させる。

$$\lambda(t) = s(t-1)/s(t)$$

$$s(t) = [(\tau-1)/\tau]s(t-1) + (e(t)^T e(t)/\tau) \quad (8)$$

ここで、 τ はモメンタムの変化分を決める定数である。

5 SampEn に基づくオンライン学習法

NN を実際に予測器として用いるとき、訓練データにおいて、その入出力関係を正しく学習する必要がある。提案法では、まず入力変数と出力との関連性を CSampEn を用いてリアルタイムで推定し、得られた値に基づいて入力変数の値を修正する。図 1 は提案するオンライン学習法概念図を表す。ここで、 x は予測対象の時系列データを表し、 y は予測対象と関連性のある外部要因を表す。入力データ x, y は入出力との CSampEn(t)(m, r, N) の値に基づいて補正された後、前章で言及した学習アルゴリズムを用いて訓練される。 x, y をまとめて \mathbf{x} で表すと SampEn Filter では式 (9) のように入力データが補正される。

$$\mathbf{x}_{re,j}(t-i) = \frac{fCSampEn(\mathbf{x}_j(t) \|\mathbf{x}_j(t-i))^{-1}}{\sum_{j=1}^{num} \sum_{k=0}^{p-1} fCSampEn(\mathbf{x}_j(t) \|\mathbf{x}_j(t-k))^{-1}} \mathbf{x}_j(t-i) \quad (9)$$

ただし、 $\mathbf{x}_j(t-i) = [x_{t,j}(t-i), x_{t-1,j}(t-i), \dots, x_{1,j}(t-i)]^T$ は j 番目の入力変数のラグ i だけ遅れた時系列データを表す。

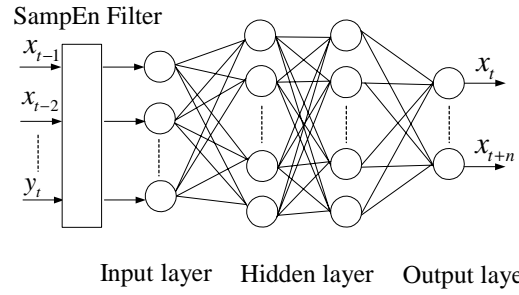


図 1: SampEn フィルタを導入した BPNN 構造

6 金融時系列予測への応用

提案手法を日経平均株価の終値のワンステップ (日次) 予測に適用することでその有効性を検討した。2005 年 5 月 ~ 2007 年 4 月初めまでのデータを訓練データとし、2007 年 4 月初め ~ 2008 年 4 月末までのデータをテストデータとした。入力には日経平均株価、日経平均先物、NASDAQ 総合指数、FTSE100 種株価指数、S & P500 指数先物、円ドルレート、長期金利回り、金先物価格の 8 変数の過去 5 日分のデータを平均 0 分散 1 に正規化して用いている。表 1 は 20 回の試行の平均値と標準偏差を表示している。また図 2 は、ベストパフォーマンスをグラフ化したものである。

表 1: 予測パフォーマンスの比較

入力データ	中間層	RMSE	方向正答率
日経のみ (1 変数 5 入力)	5	242.2 ± 2.649	49.11 ± 2.462
	10	239.4 ± 1.891	52.88 ± 1.262
	15	239.9 ± 2.852	51.06 ± 0.4335
多入力 (8 変数 40 入力) SampEn Filter なし	40	271.5 ± 22.45	58.41 ± 1.489
	80	287.1 ± 47.25	59.76 ± 2.856
多入力 (8 変数 40 入力) SampEn Filter あり	120	265.4 ± 37.16	60.46 ± 1.802
	40	211.4 ± 4.462	67.27 ± 1.231
	80	211.1 ± 4.611	66.72 ± 1.165
	120	218.6 ± 3.272	66.40 ± 0.4846

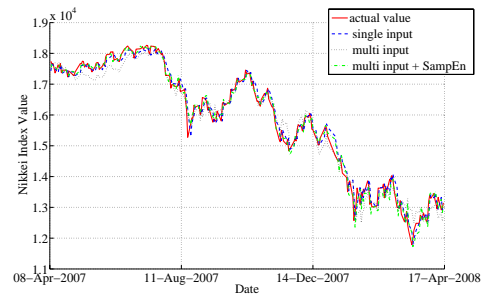


図 2: ベストパフォーマンスの比較

7 結論

提案手法により、入出力関係を適切に学習することが可能になり、予測精度の向上と安定した予測性能が確認できた。

参考文献

- [1] JoShua S. Richiman and J. Randall Moorman, Am.J. Physiol. 278: H2039-H2049(2000)
- [2] Lean Yu, Shouyang Wang, and Kin Keung Lai, Springer Science+Business Media, LLC(2007)