Machine Learning Technique Using the Signature Method for Automated Quality Control of Argo Profiles

Nozomi Sugiura1 and Shigeki Hosoda1

1Research and Development Center for Global Change, JAMSTEC, Yokosuka, Kanagawa, Japan

Abstract A profile from the Argo ocean observation array is a sequence of three-dimensional vectors composed of pressure, salinity, and temperature, appearing as a continuous curve in three-dimensional space. The shape of this curve is faithfully represented by a path signature, which is a collection of all the iterated integrals. Moreover, the product of two terms of the signature of a path can be expressed as the sum of higher-order terms. As a result of this algebraic property, a nonlinear function of the profile shape can always be represented by a weighted linear combination of the iterated integrals, which enables machine learning of a complicated function of the profile shape. In this study, we performed supervised learning for existing Argo data with quality control flags by using the signature method and demonstrated the estimation performance by cross validation. Unlike rule-based approaches, which require several complicated and possibly subjective rules, this method is simple and objective in nature because it relies only on past knowledge regarding the shape of profiles. This technique is critical for realizing automatic quality control for Argo profile data.

1. Introduction

Argo is an international effort involving the collecting of high-quality temperature and salinity profiles, typically from the upper 2,000 m of the global ocean (Gould et al., 2004). The data are obtained from battery-powered autonomous floats that drift predominantly at a depth where they are stabilized at a constant pressure level. Typically at 10-day intervals, the floats rise to the surface for approximately 6 hr while measuring temperature and salinity. On surfacing, the satellites position the floats and receive the transmitted data. Currently, the array of over 3,000 floats provides 100,000 temperature/salinity profiles annually, distributed over the global oceans at an average 3° spacing. Quality control (QC) of the massive Argo profile data (ARGO, 2019) must be systematic to keep the quality of the observational data homogeneous and to utilize human resources efficiently. In addition, accurately quantifying the relationship between the profile shape and the effect it has on oceanic processes is essential for understanding the ocean state through the profile observation. Conventionally, significant time and effort are expended assigning a quality control flag to each Argo profile.

Regarding attempts for advanced automatic QC procedures on oceanographic profiles, several studies have been conducted on the Argo CTD profile because of the huge amount of data accumulated for 200 million profiles over 20 years. For example, Udaya Bhaskar et al. (2013) developed a semiautomatic QC procedure using objective mapping to remove anomalous values from the profiles. Udaya Bhaskar et al. (2017) demonstrated automatic QC by defining the convex hulls from the climatological data set. Meanwhile, Ono et al. (2015b) attempted to apply a machine learning method to the delayed-mode QC of Argo profiles toward a possible automatic QC system for an Argo data stream. Similarly, an integrated Argo data flow using machine learning was presented as an automated system with an improved QC ability (presented by Maze, 2017 in the report of the 18th Argo Data Management Meeting). Thus, QC procedures for oceanographic data have been gradually improved by many researchers using advanced tools or methods.

The discrimination procedures involved in the automation of the QC have been performed mainly in a rule-based manner (e.g., Hayashi et al., 2016; Kamikawaji et al., 2016; Ono et al., 2015a). As an alternative and more flexible approach, this study attempts to automate the process via supervised learning of the human judgment process. In doing so, it is essential to quantify the profile shape so that the function
that yields the quality control flag can be expressed as a linear combination of the numerical values that represent the profile shape. The machine learning thereby reduces to a linear optimization problem that can be easily solved. The key tool that enables this quantification is the signature, which is the set of all iterated integrals (Chevyrev & Kormilitzin, 2016; Levin et al., 2013), proposed in the theory of rough path by Lyons et al. (2007).

In this paper, we propose a procedure in which the vector sequence of each Argo profile is first converted into a sequence of real numbers that represents its shape, and then a nonlinear function of the shape is expressed in the form of a linear combination of these numbers; this conversion facilitates machine learning of the nonlinear function. A machine learning experiment involving learning of the function was performed and applied to automatic assignment of QC flags to the profiles.

2. Theoretical Background

The central concept in this study is the signature, proposed in the theory of rough path by Lyons et al. (2007). In what follows, we briefly introduce the concept of signature and the notations used in this paper. For more details, refer to Chevyrev and Kormilitzin (2016).

As perceived from a reexamination of controlled differential equations (refer to Appendix A), characteristics of a data sequence can be represented by the signature, which comprises the iterated integrals. Note that in this paper, the subscript notation $X_s$ is used to denote dependence on the parameter $s \in [0, t]$. $A^{s*}$ and $A^{s* n}$ denote the $n$th power and $n$ times tensor product, respectively; otherwise, a superscript denotes a component.

Suppose we have a sequence of $d$-dimensional vectors $X_u$ ($0 \leq u \leq t$). Let the time order be $0 < t_1 < \cdots < t_n < t$. We define the iterated integral for indices $i_1, \ldots, i_n = 1, \ldots, d$ as

$$X^{(i_1 \cdots i_n)} = \int_{t_{i_n}=0}^{t} \cdots \int_{t_{i_1}=0}^{t} dX_{i_1} \cdots dX_{i_n},$$

(1)

where we should be careful about the difference between the font for a sequence of vectors $X^{(i_1 \cdots i_n)}$ and that for an iterated integral $X^{(i_1 \cdots i_n)}$. By treating all the index values together, we obtain a tensor of order $n$:

$$X_n = \int_{0 < t_1 < \cdots < t_n < t} dX_{t_1} \otimes \cdots \otimes dX_{t_n} \quad n = 1, 2, \ldots,$$

(2)

and $X_n$ has a constant value of 1. Moreover, by combining the tensors of order less than or equal to $n$, we obtain the signature up to degree $n$:

$$S^n(X) = (X_0, X_1, X_2, \ldots, X_n),$$

(3)

which has $(d^{n+1} - 1)/(d - 1)$ components. For instance, the signature up to degree 2 for a two-dimensional sequence is

$$S^2(X) = \left( X^{0}, \left[ X^{(1)}_{2} \right], \left[ X^{(12)}_{1} \right], X^{(21)}_{1} \right) = \left( 1, \left[ X^{0}_{0} \right], \left[ X^{1}_{0} \right] , \left[ \frac{1}{2} (X^{1}_{0} X^{2}_{0}) \right], \int_{0}^{1} X^{1}_{0} dX^{2}_{0} \right),$$

(4)

where $X^{0}_{0} \equiv X_{0}^{0}$ and $X^{1}_{0} \equiv X_{0}^{1}$; the second power, and $X^{0}_{0} = X^{1}_{0} = 1$. Note that the order of integrands matters in $\int_{0}^{1} X^{1}_{0} dX^{2}_{0}$ and $\int_{0}^{1} X^{2}_{0} dX^{1}_{0}$. In general, it is important for the signature to encode the order in which each component changes along the path.

Suppose we have two paths, $X = \{X_t\}_{0 \leq t \leq s}$ and $Y = \{Y_t\}_{s \leq t \leq t}$. Their concatenation is the path defined by

$$(X \circ Y)_t = \begin{cases} X_t & \text{if } t \in [0, s] \\ X_s + Y_t - Y_s & \text{if } t \in [s, t]. \end{cases}$$

(5)

On the other hand, regarding their signatures, $S^n(X) = (X_0, X_1, \ldots, X_n)$ and $S^n(Y) = (Y_0, Y_1, \ldots, Y_n)$, we can define the product as

$$S^n(X) \otimes S^n(Y) \equiv (Z_0, Z_1, \ldots, Z_n),$$

(6)
Here, we describe the basic concept of the signature method and how to apply it to Argo profiles. We also explain how to construct a procedure for supervised learning using the signature and how to verify the results.

### 1. The First-order Iterated Integrals

The first-order iterated integrals are

\[ X^{(p)} = \int_{t=0}^{t} dP_t = P_t - P_0, \quad X^{(s)} = S_t - S_0, \quad X^{(t)} = T_t - T_0. \]

which are profile depth, sea surface salinity, and sea surface temperature, respectively.

### 2. The Second-order Iterated Integrals

The second-order iterated integrals include

\[ X^{(pp)} = \frac{1}{2}(P_t - P_0)^2, \quad X^{(sp)} = \int_{t=0}^{t} (S_t - S_0) dP_t, \quad X^{(tp)} = \int_{t=0}^{t} (T_t - T_0) dP_t, \]

which represent the square of profile depth, total salinity content, and total heat content, respectively.

In this manner, the set of signatures has a group structure in the free tensor algebra with respect to the group operation (6).

In the context of geophysics, we can show that some diagnoses for oceanographic conditions are written in terms of iterated integrals. Consider a vertical sequence of vector \((P_t, S_t, T_t)\) (pressure, salinity, and temperature) in the ocean.

1. The first-order iterated integrals are

2. The second-order iterated integrals include

We find another example in Appendix B.

Note that \(P\) is treated equally to \(S, T\) in the above because the seemingly redundant parameter \(\tau\) is essential to ensure that the path has no self-intersection and the signature is invariant under the reparameterization of \(\tau\). If one parameterizes \(T, S\) with \(P\), the path would be drawn on a two-dimensional \(T, S\) surface, which loses considerable information on the shape of the sequence.

### 3. Method

The data used in this research were observed by the global array of Argo floats (ARGO, 2019), each of which floats and sinks from the sea surface to a depth of approximately 2,000 m.

Because the shape of a vector sequence \((P_t, S_t, T_t)\) is only perceived in a certain reference frame, it is convenient to make the original quantities dimensionless; \(P\) in dbar, \(S\) in psu, and \(T\) in °C into \(\hat{P} = P/2000, \hat{S} = S/2,\) and \(\hat{T} = T/20\), where divisor (2000, 2, 20) is chosen as a typical scale of the components. For simplicity, henceforth, we omit the hat symbol for the component. Figures 1 and 2 show examples of the vertical profiles of temperature, salinity, and pressure, along with the corresponding iterated integrals. By virtue of quality control procedures with manual judgment, the quality control flags are already assigned to all of the data.

Here, we describe the basic concept of the signature method and how to apply it to Argo profiles. We also explain how to construct a procedure for supervised learning using the signature and how to verify the results.
Figure 1. An example of (a) profile shape obtained from Argo observation and (b) their first few iterated integrals. For example, \((T,P)\) denotes the iterated integral
\[ X(T,P) = \int_0^T \int_0^P dP dt. \]
\(T, P, S, \) and \(S\) are divided in advance by 2,000 dbar, 2 psu, and 20°C, respectively, and thus dimensionless. The profile was observed at 122.0°W, 19.1°S on 20 December 2009.

3.1. Representing the Argo Profile Shape by Signature

3.1.1. Computation of Signature

Suppose we have \(d\)-dimensional profile data \(\{X_{\tau}\}_{0 \leq \tau \leq t}\) that can be seen as a line graph connecting points \(X_0 = X_{u_0}, X_{u_1}, \ldots, X_{u_d} = X_t\); then, we can compute its iterated integrals as follows:

1. For line segment \(X_{seg} \overset{def}{=} \{X_i^0 + \chi_i^{j-1} X_j^i\}_{0 \leq \tau \leq t}\), which has starting point \(X_i^0\) and slope \(\chi_i^{j-1}\), the iterated integrals are calculated as

\[
X(j) = X(j)^0, \quad X(j)^i = \frac{1}{2!} X(j)^0 X(j)^i, \quad X(j)^{i_1 i_2 \cdots i_h} = \frac{1}{3!} X(j)^0 X(j)^{i_1} X(j)^{i_2} X(j)^{i_3},
\]

and the 0th iterated integral is constant 1. In this case, the signature (up to degree \(n\)) is nothing but a commutative exponential function for the vector \(X_{u_d}^0\):

\[
S^n(X_{seg}) = \sum_{h=0}^{n} \frac{1}{h!} \sum_{i_1 i_2 \cdots i_h} \prod_{k=1}^{h} X(j)^{i_k} e_{i_k},
\]

where \(e_{i_k}\) is the \(i_k\)-th unit vector.

Figure 2. Another example of (a) profile shape obtained from Argo observation and (b) their first few iterated integrals. The profile was observed at 163.3°E, 42.5°N on 14 November 2001.
2. Let the time order be \( s \leq u \leq t \). By concatenating a path \( X_{su} \) from time \( s \) to \( u \) with a path \( X_{u,t} \) from time \( u \) to \( t \), we obtain a path \( X_{su,t} \) from time \( s \) to \( t \), whose signature is the product of the signatures:

\[
S^n(X_{su,t}) = S^n(X_{su}) \otimes S^n(X_{u,t}),
\]

which is due to Chen’s identity (10).

3. By concatenating the paths successively using Equation 13, we can compute the signature for the whole line graph.

The numerical computation of the signature in this study was performed by using the Python library \( \text{Esig} \) (Kormilitzin, 2017).

### 3.1.2. Lead-Lag Transformation

Suppose we have a sequence of \( d_0 \)-dimensional \((d_0 = 3)\) vectors with length \( L + 1 \):

\[
X = (X_0, X_1, \cdots, X_L) = \begin{bmatrix}
P_0 \\ S_0 \\ T_0 \\
SP_1 \\ SS_1 \\ ST_1 \\
\vdots \\
PL \\ SL \\ TL
\end{bmatrix}
\]

To more precisely grasp the shape of the line graph, we perform a lead-lag transformation (Chevyrev & Kormilitzin, 2016), which defines a sequence of \( d(=2d_0) \)-dimensional vectors with length \( L \cdot d + 1 \):

\[
\begin{bmatrix}
P_0 \\ S_0 \\ T_0 \\
P_1 \\ S_1 \\ T_1 \\
\vdots \\
P_L \\ S_L \\ T_L
\end{bmatrix}
\]

The transition rule for the lead-lag transformation is as follows:

1. Take two copies of \( X_0 \) and use it as the initial condition.
2. Update only one component among \( d \) components at once.
3. Use the previous value instead if the present value is missing.

### 3.2. Machine Learning Procedure for QC Process

Suppose we have a set of profile data \( X(m) \Def \{X_\tau(m)\}_{0 \leq \tau \leq t} \) for \( m = 1, 2, \cdots, M \), whose signature is denoted as \( X(m) \Def S(X(m)) \). Let us consider the problem of assigning the discriminant values to each profile depending on whether a profile matches the quality standard.

1. We first make a model for the rule of QC as a functional form; that is, a linear combination of the iterated integrals \( X^I \) for all combinations of indices \( I = () , (i_1), (i_1i_2), \cdots , (i_1 \cdots i_n) \) yields the discriminant value.

\[
y = \sum_I w^I X^I + \epsilon, \tag{14}
\]

where \( \epsilon \) is the error. As each index in \( I \) runs over \( 1, 2, \cdots , d \), the sequence of iterated integrals in Equation 14 has \( \sum_{|I|=n} d^* = (d^{n+1} - 1)/(d - 1) \) terms. Note that \( X^() \) represents the constant 1. Such a representation is possible because its nonlinearity is unraveled thanks to the property of shuffle product; for a fixed path \( X \), the product of iterated integrals for indices \( A \) and \( B \) is expressed by the iterated integral with respect to the shuffle product \( A \shuffle B \):

\[
X^A X^B = X^{A\shuffle B}. \tag{15}
\]

For example, \( X^{(a)}X^{(b)} = X^{(ba)} + X^{(ab)} + X^{(a)ab} \). This means that a product of iterated integrals is always reduced to the sum of higher-order iterated integrals. Moreover, by virtue of the Stone-Weierstrass theorem, any nonlinear function of the shape of a path can be represented as a linear combination of the iterated integrals.
Table 1

Confusion Matrix With Cutoff $y_c$

| Estimated $\hat{y}$ | True $y$ | 0 | 1 |
|---------------------|----------|---|---|
| $\hat{y} < y_c$    | True-positive $N_{TP}$ | False-positive $N_{FP}$ |
| $y_c \leq \hat{y}$| False-negative $N_{FN}$ | True-negative $N_{TN}$ |

2. Suppose we have pairs $(X(m), y(m))$, where each $X(m)$ is a profile sequence, and $y(m) = 0, 1$ is the discrimination value, which is already given to each sample $m = 1, 2, \cdots, M$ as training data. Learning these data is simply deriving the weights $w^I$ that minimize an $L_1$-regularized cost function:

$$J(w) = \frac{1}{2M} \sum_{m=1}^{M} \left( y(m) - \sum_I w^I X(m)^I \right)^2 + \alpha \sum_I |w^I|.$$  \hspace{1cm} (16)

Because the terms in $\sum_I |w^I|$ are not quadratic but linear, they have the effect of selecting significant terms under the summation over the set labeled by $I$. This is the notion of least absolute shrinkage and selection operator (LASSO) (Tibshirani, 1996), which can help prevent overfitting. The larger the value of $\alpha$ is, the smaller the number of selected terms with $w^I \neq 0$ is. To set an appropriate number of terms, several values of $\alpha$ will be tested.

3. Using the coefficients $w$ derived in (16), and substituting into Equation 14 the iterated integrals for a profile not used for training, we obtain $\hat{y}$, an estimate for $y$, as follows.

$$\hat{y}(m) \stackrel{def}{=} \sum_I w^I X(m)^I,$$  \hspace{1cm} (17)

where $w^I$'s are estimated from the minimization of cost (16).

4. The minimization problem is efficiently solved by the coordinate descent (CD) method (Friedman et al., 2007).

For the $L_1$-regularization term to apply evenly, each iterated integral $X^I$ is preprocessed by subtracting the ensemble mean $\mu_{\text{train}}^I$ of the training ensemble and dividing by the standard deviation $\sigma_{\text{train}}^I$ of the training ensemble:

$$X(m)^I \leftarrow \frac{X(m)^I - \mu_{\text{train}}^I}{\sigma_{\text{train}}^I}.$$  \hspace{1cm} (18)

The same operation is performed for the iterated integrals in cross validation. The minimization problem was solved by using the Python library scikit-learn (Pedregosa et al., 2011).

3.3. Assessment of Learning Results

The performance of the binary classifier can be quantitatively assessed by visualizing it with the receiver operating characteristic (ROC) curve (Egan, 1975). We refer to the profiles that pass the quality criterion as negative (normal) $y = 1$ and the others as positive (bad) $y = 0$. By shifting the cutoff value $y_c$, one can count the number of positive profiles with $\hat{y} < y_c$ and that of negative profiles with $y_c \leq \hat{y}$. Then, the samples fall into the four categories in Table 1.

The true-positive rate is defined as $N_{TP}/(N_{TP} + N_{FN})$, and the false-positive rate as $N_{FP}/(N_{FP} + N_{FN})$. The ROC curve is the two-dimensional plot of false-positive rate versus true-positive rate, by changing the cutoff $y_c$. It has better performance if the trajectory approaches the upper left corner. Therefore, the area under the ROC curve indicates the performance.

Note that, to improve the readability of the histograms, we use a modified estimation value:

$$\jmath(m) \stackrel{def}{=} 1 - \left( 1 - \sum_I w^I X(m)^I \right),$$  \hspace{1cm} (19)

where we apply transformation $y \leftrightarrow 1 - |1 - y|$ so that $\jmath(m) \leq 1$.  

SUGIURA AND HOSODA
3.4. Experiment Using Principal Component Analysis

Alternatively, principal component analysis (PCA) (e.g., Thomson & Emery, 2014) can be applied to represent the normal profiles. In that case, the experiment for estimating the QC flag is performed as follows.

1. We apply the same nondimensionalization to the $T$ and $S$ sequences as in the signature method and then perform nearest-neighbor interpolation at points $2,000 \overline{P} = 5, 15, \ldots, 1995$, which are placed every 10 dbar. Accordingly, the sequences are transformed into a sequence $X = (X_1, \ldots, X_L)^T$ with $L = 400$.

2. Let $\mathcal{P} = \{X(m) | m = 1, 2, \ldots, M\}$ be the set of all profiles. We randomly choose the training ensemble $\mathcal{P}_{tr} \subset \mathcal{P}$, which comprises negative (normal) samples $\mathcal{P}_{tr,n} \subset \mathcal{P}_{tr}$, and positive (bad) samples $\mathcal{P}_{tr,p} \subset \mathcal{P}_{tr}$.

3. Training is performed by computing the principal components (PCs) for negative training samples $\mathcal{P}_{tr,n}$. Let

$$U = \begin{bmatrix} U_1^1 & \cdots & U_1^L \\ \vdots & \ddots & \vdots \\ U_{N_{pc}}^1 & \cdots & U_{N_{pc}}^L \end{bmatrix}$$

be the truncated PCs, and $\overline{X} = (\overline{X}_1, \ldots, \overline{X}_L)^T$ be the ensemble mean for the training ensemble $\mathcal{P}_{tr,n}$.

4. For the $m$-th profile $X(m) \in \mathcal{P}_{tr}$ (or $\in \mathcal{P} \setminus \mathcal{P}_{tr}$ for cross validation), the mean square residual for representing it by the first $N_{pc}$ PCs is computed as

$$r(m) = \frac{1}{2L} \left| (I_L - U^T U)(X(m) - \overline{X}) \right|^2,$$

where $I_L$ is an $L$-dimensional identity matrix. The estimated $\tilde{y}$ is thereby defined as $\tilde{y}(m) = -r(m)$.

5. For a fixed threshold value $y_c$, we assign negative to the $m$th profile if $\tilde{y}(m) > y_c$ and positive otherwise. The ROC curve is drawn by plotting false-positive rates versus true-positive rates for various $y_c$.

4. Results and Discussion

We used a data set observed at the locations shown in Figure 3. Each profile is assigned a delayed-mode QC flag by Japan Agency for Marine-Earth Science and Technology (JAMSTEC). We treated profiles with depth spans (the difference between the minimum and maximum depths) of more than 1,000 m, and each profile had approximately $L \approx 100$ observation points. The number of profiles was $M = 8.2 \times 10^4$, and the training data were randomly chosen from these profiles. After applying the lead-lag transformation, each profile was converted into the signature up to degree $n = 6$.

The machine learning results are summarized by the histogram of estimated values $\tilde{y}$ for normal samples ($y = 1$), and the histogram for bad samples ($y = 0$).
Figure 4. Histogram for discriminant analysis (identification of the training data). Proportion of training data: 0.4, \( \alpha = 10^{-5} \). The horizontal axis is the estimated value \( \hat{y} \), and the vertical axis is its frequency. Blue: histogram for positive data (bad samples with flag \( y = 0 \)), red: histogram for negative data (normal samples with flag \( y = 1 \)), cyan: true-positive rate, and orange: false-positive rate.

Figures 4 and 5 show the histograms when 40\% of the data are used for training and the remaining 60\% are used for cross validation. We can see that learning is performed properly because there is little difference between the identification of training data and the cross validation. In particular, this approach never misidentifies negative (normal) profiles if the appropriate cutoff \( y_c \) is used, but it may accept positive (bad) profiles with a probability of 0.6 when \( y_c = 0.5 \). This property is also reflected in the tendency of the ROC curve (Figure 6) to be almost tangential to the horizontal axis when \( x \) is small but not tangential to \( y = 1 \) when \( y \) is large. The histogram for positive samples has two clear peaks, which suggests that the ambiguity is not caused by the judgment by the machine learning but by the fact that the original QC flag had a criterion that cannot be decided only by the shape. For example, the original QC, encoded in \( y \), may have a criterion about deviation from climatological variation, which cannot always be detected from profile shape. Moreover, the original QC is partly done through visual checking, for which the criteria can fluctuate between checks. Obviously, both are not represented by the signature, which is static and shape oriented.

Figures 7 and 8 show the histograms when 2.5\% of the data are used for training and the remainder are used for cross validation. In this case, there is a clear tendency of overlearning, which indicates that the number of learning samples, 2.5\%, is insufficient.

Figure 5. Histogram for discriminant analysis (cross validation). Proportion of training data: 0.4, \( \alpha = 10^{-5} \). The horizontal axis is the estimated value \( \hat{y} \), and the vertical axis is its frequency. Blue: histogram for positive data (bad samples with flag \( y = 0 \)), red: histogram for negative data (normal samples with flag \( y = 1 \)), cyan: true-positive rate, and orange: false-positive rate.
Figure 6. ROC curves for various regularization parameters. Identification of the training data (broken curves) and cross validation (solid curves) are depicted. The horizontal axis is the false-positive rate, and the vertical axis is the true-positive rate.

The performance of a method can be measured by the area under the ROC curves (AUC). Comparing that for the experiments with various ratios of learning samples, we notice that overlearning occurs when the ratio is less than 20\% (Figure 9).

We also compared the results of the experiments with various weights $\alpha$ of the regularization term by the AUC. The number of terms under the summation over the set labeled by $I$ in Equation 16 is dependent on $\alpha$. Therefore, if we increase the degrees of freedom of the coefficients $w$ by using a smaller $\alpha$, the performance of the reproduction capability increases. However, the estimation capability begins to saturate at approximately 6,700 degrees of freedom (Figure 10), where $\alpha = 10^{-5}$ is used. At that point, the complexity appears to become appropriate.

To confirm the efficacy of lead-lag transformation, we performed a similar experiment as in Figures 4 and 5, except without lead-lag transformation. We set $\alpha = 10^{-5}$ and the proportion of training data to 0.4. Figure 11

Figure 7. Histogram for discriminant analysis (identification of the training data). Proportion of training data: 0.025, $\alpha = 10^{-5}$. The horizontal axis is the estimated value $\hat{y}$, and the vertical axis is its frequency. Blue: histogram for positive data (bad samples with flag $y = 0$), red: histogram for negative data (normal samples with flag $y = 1$), cyan: true-positive rate, and orange: false-positive rate.
Figure 8. Histogram for discriminant analysis (cross validation). Proportion of training data: 0.025, $\alpha = 10^{-5}$. The horizontal axis is the estimated value $\hat{y}$, and the vertical axis is its frequency. Blue: histogram for positive data (bad samples with flag $y = 0$), red: histogram for negative data (normal samples with flag $y = 1$), cyan: true-positive rate, and orange: false-positive rate.

shows the ROC curves for the experiment. The curves for the case with lead-lag are to the upper left of those for the case without lead-lag, which indicates that the lead-lag transformation helps improve the estimation of the QC flag.

As a reference case using another representation of the shape, we performed PCA experiments with $N_{\text{PC}} = 50$ and 100 PCs. The proportion of training data was set to 0.4, the same as for the signature case. Figure 12 depicts the ROC curves for the PCA experiments. Although the PCA method also exhibits a considerable skill, the curves stay to the lower right of those for the signature method, which indicates that the signature method is more effective than the PCA method in estimating the QC flag. Meanwhile, the computational cost for the signature method (elapsed time of 573 min by serial processing) is higher than that for the PCA method (elapsed time of 101 min by serial processing), but the required time can be reduced by parallelization or some other algorithmic means, if necessary. Note that the former additionally requires converting each data sequence into the truncated signature, which can be performed separately for each profile as preprocessing (elapsed time of 0.12 min per profile).

Figure 9. Learning curves for different proportions of training data. Red: identification of the training data; blue: cross-validation.
Further, comparison with the data from the ARGO intercomparison project was performed. The performances of the real-time assignment of QC flags (Wong et al., 2020) by several institutes are shown in Wedd et al. (2015), when the corresponding assignments by the delayed-mode QC are regarded as the ground truth. Because the sets of profile data differ from those in our case, a direct comparison is not strictly relevant but will still serve as a measure of the performance. Figure 13 shows the false-positive versus true-positive rates for those samples in comparison to the signature case. Apart from the pressure data, all the points for real-time QC data lie on the bottom right side of our ROC curve. This suggests that the signature method may assign the QC flags more efficiently than the real-time QC procedure does, provided that the past assignment
Figure 12. ROC curves for the cases using signature and PCA. The cases using the signature method (red and blue) are compared to the cases using the PCA method with 50 PCs (brown) and 100 PCs (purple); identification of the training data (broken curves) and cross validation (solid curves) are depicted. The horizontal axis is the false-positive rate, and the vertical axis is the true-positive rate.

Figure 13. Performance points for real-time QC on ROC graph. The red and blue curves are for the signature method. The points denote temperature (circle), salinity (cross), and pressure (triangle), each of which represents the false-positive versus true-positive rates for real-time QC data from the Coriolis data center (CRS; green), the Australian Bureau of Meteorology (BoM; brown), the United Kingdom Met Office (UKMO; cyan), and the Fleet Numerical Meteorology and Oceanography Centre (FNMOC; magenta), taken from Table 5 of Wedd et al. (2015).
results are ready for use. Another advantage of the signature method is that it assigns the flags consistently to all the components \(P, S, T\) with higher reliability.

Overall, we found that machine learning using the signature method can learn the existing QC flags of Argo profiles and automatically assign the flags to new profiles, but it sometimes overlooks bad samples because of the ambiguity inherent in the original QC flag. Comparative study shows the signature method has a higher skill in estimating the flag than other conventional methods, including the one with the PCA representation and the operational assignments of real-time QC.

5. Conclusions

In this study, we first demonstrated that the shape of a profile from the Argo ocean observing array can be represented by the iterated integrals. Then, we constructed a model for the function that assigns a QC flag to the shape of a profile, which is expressed as a weighted sum of the iterated integrals. Subsequently, we performed supervised learning for the weights using the existing QC flags for training data and demonstrated via cross validation that it has good performance in estimating flags for unknown data.

A comparative experiment conducted using the PCA method showed that the signature method, in combination with lead-lag transformation, outperforms the PCA method in estimating the QC flag. This suggests that the signature method is superior to the conventional machine learning technique.

This algorithm can potentially enable automatic assignment of QC flags to new Argo data. The significance of the algorithm is that it objectively and automatically assigns the QC flag only on the basis of past knowledge about the quality of data without imposing any ad hoc rules. Hence, it should enable more objective and efficient QC compared to traditional manual methods or rule-based machine learning.

The signature method is quite effective for expressing the shape of an Argo profile and its nonlinear function quantitatively. The rationale for this advantage is that a nonlinear and complicated function for assigning QC flags can be transformed into a linear combination of the iterated integrals through algebraic transformation (shuffle product) without introducing any errors. This is superior to conventional multivariate regression models, which approximately regard nonlinear dependencies as linear dependencies. Along this line, we can express, as a function of signature, not only QC flags but also any oceanic phenomena.

One application of the signature method is assimilation of the signature of observational data into a general ocean circulation model. For example, we can convert a vertical sequence of observational data and that of model data into iterated integrals. We then construct a cost function that compares the signatures for model and observation, rather than directly comparing the state vectors composed of temperature and salinity at each depth. Although a cost term is for a single horizontal and temporal point, data assimilation, in particular the four-dimensional variational method, can combine the effects from multiple terms via model integration and adjoint integration. By doing so, we gain the advantage that the projection of a vertical profile onto any ocean phenomena attains a linear form, which will result in efficient data assimilation. This is expected because many diagnoses for oceanic conditions are written in terms of iterated integrals, as illustrated in section 2 and Appendix B.

Appendix A: Picard Iteration

To understand the notion of signature, consider how the theory of rough path treats a data sequence acting on a system. Suppose we have a system of ordinary differential equations with respect to \(Y_t\) forced by a path \(X_t\):

\[
dY^i_t = \sum_{j,k} F^i_{jk} Y^j_t dX^k_t,
\]

where \(Y^i_t\) is the \(i\)th component of vector \(Y_t\), and \(F^i_{jk}\) is the \(i,j,k\)th component of three-dimensional tensor \(F\). Performing the Picard iteration yields a solution:

\[
Y^i_t = \sum_{n=0}^{\infty} \sum_{i_1, \ldots, i_n} F^{i_1}_{i_{n+1}, i_2} \cdots F^{i_n}_{i_1, i_n} X^{i_1, i_2, \ldots, i_n} Y^i_0.
\]
where $X_{n}^{k_{1}k_{2}...k_{n}} \overset{\text{def}}{=} \int_{0}^{\tau} \int_{\tau_{1}}^{\tau_{2}} ... \int_{\tau_{n-1}}^{\tau_{n}} dX_{\tau_{1}}^{k_{1}} dX_{\tau_{2}}^{k_{2}} ... dX_{\tau_{n}}^{k_{n}}$ is a component of the $n$-th iterated integral (2). By omitting the indices, we can simply write the solution as $Y_{n} = \sum_{m=0}^{\infty} F_{n}^{m} X_{n} Y_{0}$. Notice that the convergence of the series is guaranteed because the magnitude of each iterated integral is uniformly bounded: $|X_{n}^{k_{1}k_{2}...k_{n}}| < \frac{1}{n!}$, where $L$ is the path length. This form of solution suggests that the action of $X$ on $Y$ can be well summarized by the iterated integrals, and an approximate solution is reproduced by a truncated series of iterated integrals $(X_{0}, X_{1}, ..., X_{n})$, which is called a truncated signature up to degree $n$.

The point is that the effect of a forcing on a system is asymptotically approximated by the truncated path signature but not by the partial sequence of state vectors.

It has been proven that a path that never crosses itself, as in the case of Argo profiles, is completely determined by its signature (Hambly & Lyons, 2010). A function of a path, say $\phi$, can thus be regarded as that of its signature and compactly approximated by that of a truncated signature:

$$\phi \left( \left\{ X_{j} \right\}_{0 \leq j \leq n} \right) \approx f \left( X_{0}, X_{1}, ..., X_{n} \right). \quad (A3)$$

A further advantage of such treatment is that the function $f$ can always be expressed as a linear combination of iterated integrals, owing to the shuffle product property, which is explained later.

Appendix B: Thermal Wind Flow in Terms of Iterated Integrals

As an example of higher-order iterated integrals, we show here that thermal wind flow can be written with iterated integrals.

The thermal wind relation is written in vertical $P$ coordinates as

$$f \frac{\partial u}{\partial P} = - \frac{\partial}{\partial y} (\rho^{*+1}) \bigg|_{P=\text{const.}}, \quad f \frac{\partial v}{\partial P} = \frac{\partial}{\partial x} (\rho^{*+1}) \bigg|_{P=\text{const.}}. \quad (B1)$$

where $f$ is the Coriolis parameter, $u,v$ are velocity, $\rho$ is density, and $x,y$ are the longitudinal and latitudinal coordinates, respectively. For a fixed latitude $y$, by performing integrations along the $x$ direction and then the $P$ direction, we obtain an estimate for the meridional velocity as

$$f \int_{x'=x_{0}}^{x_{1}} \frac{\partial}{\partial P} dx' = \rho(x_{1}, P^{*+1}) - \rho(x_{0}, P^{*+1}) = \left[ \rho(x, P^{*+1}) \right]_{x=x_{0}}^{x=x_{1}}. \quad (B2)$$

$$f \int_{x'=x_{0}}^{x_{1}} v(x', P)dx' = \left[ \int_{P'=P_{0}}^{P} \rho(x, P')^{*+1} dP' \right]_{x=x_{0}}^{x=x_{1}}, \quad (B3)$$

where we set $v(x', P_{0}) = 0$ as the layer of no motion. Integrating again along the $P$ direction, we obtain the meridional flow rate as

$$Q_{y} := -g^{*+1} \left[ \int_{P'=P_{0}}^{P_{1}} \int_{x'=x_{0}}^{x_{1}} v(x', P') dx' dP' \right]_{x=x_{0}}^{x=x_{1}}, \quad (B4)$$

where the unit is in $[\text{kg s}^{-1}]$ because of the $P$ coordinate.

Let $\tau \in [0,1]$ be a parameter for the order of observational points in a profile. Evaluating the density in Equation B4 with the state equation $\rho$, we have

$$\rho(x, P^{*+1}) = \rho(T(x, \tau), S(x, \tau), P(x, \tau)^{*+1} = \rho \left( \int_{0}^{\tau} dT_{\tau}, \int_{0}^{\tau} dS_{\tau}, \int_{0}^{\tau} dP_{\tau} \right)^{*+1}, \quad (B5)$$

which has iterated integrals as independent variables. Notice that the shuffle product property transcribes this as a linear combination of iterated integrals. Substituting this into Equation B4 finally yields

$$Q_{y} = -(gf)^{*+1} \left[ \int_{\tau_{1}=0}^{\tau_{1}=1} \int_{\tau_{2}=0}^{\tau_{1}} \rho \left( \int_{\tau_{1}=0}^{\tau_{1}} dT_{\tau_{1}}, \int_{\tau_{1}=0}^{\tau_{1}} dS_{\tau_{1}}, \int_{\tau_{1}=0}^{\tau_{1}} dP_{\tau_{1}} \right)^{*+1} dP_{\tau_{2}} dP_{\tau_{1}} \right]_{x=x_{0}}^{x=x_{1}}. \quad (B6)
This shows that the meridional flow rate $Q_y$ is represented as a linear combination of iterated integrals with respect to $T$, $S$, and $P$.

**References**

ARGO (2019). Argo float data and metadata from Global Data Assembly Centre (Argo GDAC). SEANOE, https://doi.org/10.17882/42182

Chen, K.-T. (1958). Integration of paths—A faithful representation of paths by noncommutative formal power series. *Transactions of the American Mathematical Society*, 89(2), 395–407.

Chevyyev, I., & Kormilitzin, A. (2016). A primer on the signature method in machine learning. ArXiv e-prints.

Friedman, J., Hastie, T., Höfling, H., & Tibshirani, R. (2007). Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2), 302–332.

Gould, I., & Lyons, T. J. (2007). Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 69(3), 282–302.

Hambly, B., & Lyons, T. (2016). The signature method in machine learning. *https://github.com/kormilitzin/*

Hayashi, S., Ono, S., Hosoda, S., Numao, M., & Fukui, K. (2016). Decision tree-based feature function design in conditional random field applied to error detection of ocean observation data. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)* (pp. 1–8). https://doi.org/10.1109/SSCI.2016.7849862

Kamikawaji, Y., Matsuyma, H., Fukui, K., Hosoda, S., & Ono, S. (2016). A preliminary study on quality control of oceanic observation data by machine learning methods. In *H. Handa, H. Ishibuchi, Y.-S. Ong, & K. C. Tan, (Eds.) Intelligent and Evolutionary Systems* (Vol. 1, pp. 679–693). Springer International Publishing. https://doi.org/https://doi.org/10.1007/978-3-319-13359-1_52

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

Sugiura, N. (2020). nozomi-sugiura/argo_signature: Argo signature method first release. https://doi.org/10.5281/zenodo.3863034

Tompson, R., & Emery, W. (2014). *Data analysis methods in physical oceanography*. Newnes.

Tibshirani, R. (1996). Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1), 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

Udaya Bhaskar, T., Rao, E. P. R., Reddem, V., & Devender, R. (2013). A note on three way quality control of Argo temperature and salinity profiles—A semi-automated approach at INCOIS. *International Journal of Earth Sciences and Engineering*, 5, 1510–1514.

Udaya Bhaskar, T. V. S., Venkat Shesu, R., Boyer, T. P., & Pattabhi Rama Rao, E. (2017). Quality control of oceanographic in situ data from Argo floats using climatological convex hulls. *MethodsX*, 4, 469–479. https://doi.org/10.1016/j.jmx.2017.11.007

Wedd, R., Stringer, M., & Haines, K. (2015). Argo real-time quality control intercomparison. *Journal of Operational Oceanography*, 8(2), 108–122. https://doi.org/10.1080/1755876X.2015.1087186

Wong, A., Keeley, R., & Carval, T. (2020). Argo quality control manual for CTD and trajectory data.