

УДК 681:51

Л.М. Любчик, В.А. Колбасин

Национальный технический университет «ХПИ», Харьков

## РЕКУРРЕНТНОЕ ОЦЕНИВАНИЕ МАТРИЦЫ ГРАММА ПРИ ИДЕНТИФИКАЦИИ НЕЛИНЕЙНЫХ НЕСТАЦИОНАРНЫХ ДИНАМИЧЕСКИХ СИСТЕМ

*Рассмотрена задача рекуррентной идентификации нелинейной нестационарной динамической системы на основе ядерных методов в реальном масштабе времени. На основе метода скользящего контроля предложен алгоритм 2-го уровня для оценки параметра регуляризации. Предложен способ вычисления критерия качества для алгоритма второго уровня.*

**Ключевые слова:** ядерные методы, матрица Грамма, регуляризация, скользящий контроль, рекуррентные алгоритмы.

### Введение

Проблема идентификации нелинейных динамических систем по экспериментальным данным является в настоящее время предметом интенсивных исследований. Несмотря на обилие публикаций [11 – 13] она сохраняет свою актуальность, в первую очередь, за счет привлечения новых математических методов и расширения решаемых классов задач. Сохраняющиеся трудности в значительной мере связаны с проблемами структурной идентификации (с поиском структуры моделей), оптимальным образом согласованной с имеющейся априорной и текущей информацией [14 – 15]. Для нелинейных систем, характеризующихся сложным характером зависимостей между переменными, использование традиционного параметрического подхода приводит к существенным трудностям, связанным с вычислительными трудностями оценивания большого числа неизвестных параметров. Это, в свою очередь, стимулирует развитие непараметрических методов идентификации и методов «мягкой» идентификации на основе подходов вычислительного интеллекта. Эффективные подходы созданы на основе метода опорных векторов [15] и так называемых ядерных методов [7], обеспечивающих получение нелинейных версий алгоритмов идентификации, пригодных для использования в условиях малых выборок.

Ядерные методы, основанные на идее нелинейного преобразования исходных данных в новое пространство высокой размерности (пространство признаков), обеспечивают возможность идентификации нелинейных моделей высокой сложности. В соответствии с теоремой Мерсера, указанное преобразование выбирается таким образом, чтобы скалярные произведения в пространстве признаков имели вид положительно определенной функции-ядра. При этом идентифицируемая модель может быть представлена в непараметрической форме в виде взвешенной линейной комбинации функций-ядер, а соответствующие весовые коэффициенты могут быть вычислены без непосредственного использования векторов при-

знаков, что приводит к возможности получения весьма экономных и эффективных вычислительных процедур идентификации нелинейных систем [7 – 16].

Классические алгоритмы ядерной идентификации использовали для оценок полную выборку наблюдений и характеризовались ростом размерности модели с увеличением её длины. В настоящее время ведутся интенсивные поиски рекуррентных модификаций ядерных алгоритмов, пригодных для реализации в реальном масштабе времени и обеспечивающих ограничение сложности идентифицируемой модели [12]. В частности, разработаны ядерные аналоги рекуррентного метода наименьших квадратов [17] и метода скользящего среднего [11].

В настоящей работе рассматривается рекуррентный алгоритм ядерной идентификации со скользящим окном наблюдений. Также разработан алгоритм 2-го уровня, который позволяет оценивать параметр регуляризации в процессе рекуррентной идентификации нелинейной динамической системы.

### Постановка задачи

Рассмотрим временной ряд  $\{\eta_i\}_1^N$  – результат наблюдений переменной состояния некоторой сложной нелинейной динамической системы. Используя теорему Такенса [1, 2], будем предполагать, что для данной системы возможно построение модели в виде:

$$\eta_{k+1} = f(\eta_k, \eta_{k-1}, \dots, \eta_{k-m}) + \varepsilon_k, \quad k = 0, \dots, N,$$

где  $f(\cdot)$  – эволюционный оператор (некоторая нелинейная функция);  $\varepsilon_k$  – белый шум,  $E\{\varepsilon_k\} = 0$ ,

$E\{\varepsilon_k^2\} = \sigma^2$ ; величина  $m$ , называемая также размерностью вложения, неизвестна. Данное предположение является достаточно «мягким», т.к. к динамической системе предъявляется только требование гладкости функций в правых частях уравнений, описывающих систему. Таким образом, исходная задача может быть приведена к задаче структурной и параметрической идентификации модели вида

$$y_k = f(x_k) + \varepsilon_k, \quad k = 0, \dots, N$$

по наблюдаемой выборке  $\{x_k, y_k\}_1^N$ , где

$$x_k = (\eta_k, \dots, \eta_{k-m}) \text{ и } y_k = \eta_{k+1}.$$

Рассмотрим задачу, когда наблюдения поступают в реальном масштабе времени, т.е. в каждый момент времени  $k$  необходимо построить модель эволюционного оператора  $f(\cdot)$ .

Задача осложняется тем, что свойства системы со временем могут меняться: это может быть обусловлено либо изменением параметров, либо изменением структуры динамической системы.

Таким образом, необходимо построить рекуррентный алгоритм идентификации динамической системы по наблюдаемому временному ряду.

### Оценка размерности вложения

До решения задачи идентификации системы необходимо оценить ее размерность вложения  $m$ . Для этого разработаны некоторые оценки: метод ложных соседей [3], корреляционная размерность [4], метод главных компонент [5]. Однако обычно приходится перебирать разные значения размерности, начиная с малых и постепенно их увеличивая, пока не будет достигнута удовлетворительная точность модели. Поэтому подбор размерности вложения может становиться частью единого целого процесса моделирования, а не отдельной законченной первой стадией [6].

### Идентификация системы

В последнее время было разработано и предложено множество методов идентификации сложных динамических систем. Однако отсутствие априорной информации о системе вынуждает использовать методы, основанные на универсальных аппроксиматорах, к которым относятся искусственные нейронные сети, ядерные методы и другие.

Основным недостатком нейронных сетей является медленная сходимость, что создает проблемы при идентификации в реальном масштабе времени.

Далее будем рассматривать методы, основанные на ядерных аппроксимациях. В этом случае неизвестная функция  $f(\cdot)$  представляется в виде:

$$\hat{f}(x; \alpha) = \sum_{i=1}^r \alpha_i \kappa(x, x_i), \text{ где } \alpha - \text{вектор весов; } \kappa(\cdot, \cdot) -$$

ядерная функция;  $r$  – количество опорных векторов.

Наиболее часто в качестве ядерной функции выбирается гауссово ядро:

$$\kappa(x_1, x_2) = \exp(-\mu \|x_1 - x_2\|_2^2), \text{ где } \mu - \text{настроечный параметр. Это обусловлено хорошими свойствами данного ядра, например гладкостью [7].}$$

Таким образом, для построения модели необходимо оценить параметры  $\alpha, \lambda, \mu$  функции  $\hat{f}(\cdot)$ .

Параметры  $\alpha$  определяются методом минимизации эмпирического риска, что приводит к оценке:

$$\alpha = K^{-1}Y, \tag{1}$$

где  $K \in R^{N \times N}$  – матрица Грамма, построенная на  $g$  опорных векторах,  $Y = (y_1, \dots, y_N)^T$ .

Однако на практике зачастую матрица  $K$  плохо обусловлена, что приводит к неустойчивости решения, переобучению и плохой интерпретируемости результатов. Поэтому обычно применяется регуляризованная версия данного алгоритма:

$$\alpha = K(\lambda)^{-1}Y = (K + \lambda I)^{-1}Y, \tag{2}$$

где  $\lambda$  – параметр регуляризации,  $I$  – единичная матрица.

Для оценки настроечных параметров  $\lambda, \mu$  на практике обычно используются различные вариации алгоритмов перебора. Однако наиболее мощным подходом для поиска параметров оказался подход, основанный на 2-х уровневой оптимизации (уровень 1: поиск  $\alpha$ ; уровень 2: поиск  $\lambda, \mu$ ) [8]. На каждом уровне используется своя целевая функция: 1й уровень – функция эмпирического риска; 2й уровень – функция скользящего контроля (cross-validation). Т.е. алгоритм 1го уровня используется на 2м уровне как составляющая часть, которая возвращает функцию  $f(x; \lambda, \mu)$ .

### Алгоритм рекуррентного оценивания

Т.к. свойства системы со временем могут меняться, то целесообразно использовать алгоритм, который будет «забывать» старые измерения и использовать только  $g$  последних измерений. Это позволит алгоритму подстраиваться под изменения реальной системы, а также ограничит вычислительную сложность алгоритма и сложность модели.

Рекуррентная оценка для такого алгоритма может быть получена с использованием последовательных преобразований регуляризованной матрицы Грамма:

$$K_{n-1, r}^{-1}(\lambda) \rightarrow K_{n-1, r-1}^{-1}(\lambda) \rightarrow K_{n, r}^{-1}(\lambda) \tag{3}$$

где первый индекс – момент времени, для которого рассчитывается матрица Грамма; второй индекс – размерность матрицы Грамма, т.е. количество измерений используемых для оценки вектора параметров  $\alpha$ .

Тогда рекуррентный алгоритм примет вид:

$$\alpha_{n+1} = (\lambda^{-1}I_r + K_{n, r})^{-1}(\lambda^{-1}\alpha_n + Y_n)$$

$$K_{n-1, r-1}^{-1} = R_r K_{n-1, r}^{-1} R_r^T - (e_1^T K_{n-1, r}^{-1} e_1)^{-1} R_r K_{n-1, r}^{-1} e_1 e_1^T K_{n-1, r}^{-1} R_r^T;$$

$$K_{n, r}^{-1}(\lambda) = \begin{pmatrix} A & B \\ C & \delta_n^{-1} \end{pmatrix} \tag{4}$$

$$A = K_{n-1, r-1}^{-1} + \delta_n^{-1} \cdot K_{n-1, r-1}^{-1}$$

$$\cdot k_{n-1, r-1}(x_n) \cdot k_{n-1, r-1}^T(x_n) \cdot K_{n-1, r-1}^{-1}$$

$$B = -\delta_n^{-1} K_{n-1, r-1}^{-1} \cdot k_{n-1, r-1}(x_n);$$

$$C = -\delta_n^{-1} k_{n-1, r-1}^T(x_n) \cdot K_{n-1, r-1}^{-1};$$

$\delta_n = \lambda^{-1} + k_{n,n} - k_{n-1, r-1}^T(x_n) \cdot K_{n-1, r-1}^{-1} \cdot k_{n-1, r-1}(x_n)$ ,  
 где  $I_n$  – единичная матрица  $n \times n$ ,  $R_r = (0_r : I_{r-1})$ ,  
 $e_1 = (1 \ 0 \ \dots \ 0)^T$ ,  $\kappa(x, x')$  – выбранная ядерная  
 функция,  $k_{n-1, r-1}^T(x_n) = (\kappa(x_n, x_{n-r+1}) \dots \kappa(x_n, x_{n-1}))$ ,  
 $k_{n,n} = \kappa(x_n, x_n)$  [9].

К недостаткам приведенного алгоритма следует отнести следующий момент: в процессе подстройки модели к изменениям системы может возникнуть необходимость в подстройке параметров регуляризации  $\lambda$  и параметров ядра  $\mu$ . Как видно из соотношения (4), для каждого следующего шага  $n+1$  рекуррентно рассчитывается матрица  $K_{n+1, r}^{-1}(\lambda)$ , используя аналогичную матрицу, рассчитанную на предыдущем шаге. При этом в процессе рекуррентного пересчета в матрице  $K_{n, r}^{-1}(\lambda)$  будет затруднительно изменить параметр регуляризации  $\lambda$  и параметры ядра  $\mu$ . Это может потребоваться в нескольких случаях: изменение режима функционирования системы и увеличение плохой обусловленности матрицы  $K_{n, r}(\lambda)$  вследствие сильно-зависимых данных. Огранич наше рассмотрение только анализом параметра  $\lambda$ .

Далее будем предполагать, что параметр ядра будет рассчитан либо методом линейного поиска, либо с помощью критерия скользящего контроля на начальном этапе работы алгоритма.

Чтобы преодолеть указанный недостаток, касающийся параметра регуляризации, можно воспользоваться сингулярным разложением матрицы  $K(\lambda) = K + \lambda I = Q(\Lambda + \lambda I)Q^T$  и, следовательно:

$$K^{-1}(\lambda) = (K + \lambda I)^{-1} = Q(\Lambda + \lambda I)^{-1} Q^T. \quad (5)$$

Т.к. матрица  $(\Lambda + \lambda I)$  диагональная, то и обратная матрица также диагональная, что приводит к соотношению:  $\left( (\Lambda + \lambda I)^{-1} \right)_{ii} = \frac{1}{\Lambda_{ii} + \lambda}$ .

Таким образом, видно, что коэффициент регуляризации выступает в качестве стабилизирующего коэффициента: без регуляризации (когда  $\lambda = 0$ ) маленькие собственные числа  $\Lambda_{ii}$  матрицы  $K(\lambda)$  приводят к огромным собственным числам обратной матрицы  $K^{-1}(\lambda)$ . Увеличение коэффициента  $\lambda$  приводит к стабилизации получаемого решения.

Также можно получить интервал, в котором имеет смысл варьировать значение  $\lambda$ . Обычно это интервал  $[\Lambda_{\min}; \Lambda_{\max}]$ , т.к. слишком маленькие зна-

чения  $\lambda$  не стабилизируют систему, а слишком большие приведут к недообучению (underfitting).

### Алгоритм 2-го уровня

Как было сказано ранее, для обоснованной оценки настроечных параметров необходимо использовать критерий 2-го уровня, например алгоритм скользящего контроля. Обычно используется 1-кратный скользящий контроль (leave-one-out cross-validation), когда вся обучающая выборка длиной  $N$  разбивается  $N$  различными способами на две непесекающиеся подвыборки:

- обучающая подвыборка длиной  $N-1$
$$S^i =$$

$$= \{ (x_1, y_1), \dots, (x_{i-1}, y_{i-1}), (x_{i+1}, y_{i+1}), \dots, (x_N, y_N) \};$$
- контрольная подвыборка длиной 1:  $\{ (x_i, y_i) \}$ .

Далее минимизируется критерий:

$$V = \frac{1}{N} \sum_{i=1}^N LE_i = \frac{1}{N} \sum_{i=1}^N (y_i - LV_i), \quad (6)$$

где  $LE_i$  – ошибка вычисленная на контрольной подвыборке для  $i$ -го разбиения,  $LV_i$  – прогноз модели в точке  $x_i$  при обучении по выборке  $S^i$  [10].

Однако такая минимизация приведет к большому объему вычислений. Покажем, как избежать этого.

Определим вектор  $Y^i$  следующим образом:

$$y_j^i = \begin{cases} y_j, & j \neq i \\ LV_i, & j = i \end{cases} \quad (7)$$

Можно показать что  $LV_i = \left( K \cdot K^{-1}(\lambda) \cdot Y^i \right)_i$ . Тогда:

$$LV_i - f_S(x_i) = \sum_j (K \cdot K^{-1}(\lambda))_{ij} (y_j^i - y_j) =$$

$$= (K \cdot K^{-1}(\lambda))_{ii} (LV_i - y_i), \quad (8)$$

где  $f_S(x_i) = (K \cdot K^{-1}(\lambda) \cdot Y)_i$ , т.е. прогноз модели в точке  $x_i$  при обучении по всей обучающей выборке. Тогда, перегруппировав слагаемые, получаем, что

$$LV_i = \frac{(K \cdot K^{-1}(\lambda) \cdot Y)_i - (K \cdot K^{-1}(\lambda))_{ii} y_i}{1 - (K \cdot K^{-1}(\lambda))_{ii}} \quad (9)$$

или в векторном представлении:

$$LV = \frac{K \cdot K^{-1}(\lambda) \cdot Y - \text{diag}_m(K \cdot K^{-1}(\lambda)) \cdot Y}{\text{diag}_v(I - (K \cdot K^{-1}(\lambda)))}. \quad (10)$$

Здесь в (10) деление происходит поэлементно;  $\text{diag}_m(M)$  обозначает матрицу, на диагонали которой находятся числа  $M_{ii}$ ;  $\text{diag}_v(M)$  обозначает вектор, состоящий из элементов  $M_{ii}$ .

Соответственно вектор ошибок скользящего контроля будет иметь вид:

$$\begin{aligned}
 LE &= Y - LV = \\
 &= Y + \frac{\text{diag}_m(K \cdot K^{-1}(\lambda)) \cdot Y - K \cdot K^{-1}(\lambda) \cdot Y}{\text{diag}_v(I - (K \cdot K^{-1}(\lambda)))} = \quad (11) \\
 &= \frac{Y - K \cdot K^{-1}(\lambda) \cdot Y}{\text{diag}_v(I - (K \cdot K^{-1}(\lambda)))} = \frac{(I - K \cdot K^{-1}(\lambda)) \cdot Y}{\text{diag}_v(I - (K \cdot K^{-1}(\lambda)))}.
 \end{aligned}$$

Кроме того, используя сингулярное разложение матрицы  $K$ , получим оценку:

$$\begin{aligned}
 K \cdot K^{-1}(\lambda) &= Q\Lambda Q^T Q(\Lambda + \lambda I)^{-1} Q^T = \\
 &= Q(\Lambda + \lambda I - \lambda I)(\Lambda + \lambda I)^{-1} Q^T = \quad (12) \\
 &= I - \lambda K^{-1}(\lambda).
 \end{aligned}$$

Или

$$I - K \cdot K^{-1}(\lambda) = \lambda K^{-1}(\lambda). \quad (13)$$

Подставив (13) в (11), получим:

$$LE = \frac{\lambda K^{-1}(\lambda) Y}{\text{diag}_v(\lambda K^{-1}(\lambda))} = \frac{\alpha}{\text{diag}_v(K^{-1}(\lambda))}, \quad (14)$$

где, строго говоря,  $\alpha$  также зависит от параметра  $\lambda$ .

Таким образом, вычислительная сложность вычисления критерия 2-го уровня резко уменьшается.

### Итоговый алгоритм

Для рекуррентной идентификации нелинейных систем можно воспользоваться следующей схемой:

1. На первом этапе строится матрица  $K_{n,r}$ , а также её сингулярное разложение  $K = Q\Lambda Q^T$ .
2. Далее, используя соотношение (14), методом линейного поиска в интервале  $[\Lambda_{\min}; \Lambda_{\max}]$  оценивается наилучший параметр регуляризации  $\lambda$ .
3. По формулам (4) рекуррентно пересчитываем матрицу  $K_{n,r}(\lambda)$ .
4. В моменты времени, когда рекуррентный прогноз приводит к большим ошибкам, необходимо вначале пересчитывать параметр регуляризации, воспользовавшись пунктом 2, а затем корректировать матрицу  $K_{n,r}(\lambda)$ .

### Выводы

Предложенный подход позволяет получать устойчивые и эффективные в вычислительном отношении решения задачи рекуррентной идентификации в реальном масштабе времени.

Существенное влияние на работу алгоритма оказывают такие настроечные параметры:  $r$  – количество опорных векторов и  $m$  – размерность вложения, оценка которых должна производиться на подготовительном этапе работы алгоритма. С одной стороны, это сужает область применения разработанного алгоритма, с другой стороны, изменение этих параметров сигнализирует о существенном изменении структуры системы.

Также представляет интерес конструирование более точного и статистически обоснованного критерия для пересчета параметра регуляризации. В этом случае также появится возможность сегментирования рассматриваемого временного ряда в реальном масштабе времени.

### Список литературы

1. *Detecting strange attractors in turbulence*, / F Taken, D. Rang, L.S. Young (eds) // *Lecture Notes. Mathematics*, 1981. – № 898. – P. 366 – 381.
2. *Embedology* / T. Sauer, J.A. Yorke, M. Casdagli // *J. Stat. Phys.*, 1991. – № 6. – P. 579 – 616.
3. *Determining embedding dimension for phase – space reconstruction using a geometrical construction* / M.B. Kennel, R. Brown, H.D.I. Abarbanel // *Phys.Rev.*, 1992. – № 45 – P. 3403 – 3411.
4. *Procaccia I. Measuring the strangeness of strange attractors* / I. Procaccia // *Physica.*, 1983. – № 9.P. 189 – 208.
5. *Broomhead D.S. Extracting qualitative dynamics from experimental data* / D.S. Broomhead, G.P King // *Physica D.*, 1986 – №20 – P. 217 – 236.
6. Безручко П. *Современные проблемы моделирования по временным рядам* / П. Безручко, Д.А. Смирнов // *Вестник ХПИ*. – X.: НТУ ХПИ, 2007. – Вып. 145. – С. 34 – 45.
7. *Smola A. Learning with kernel* / A. Smola.// *Cambridge*. – MA: MIT Press, 2002.
8. *Model selection via bilevel optimization* / J. Hu, X. Ji, G. Kunapuli, and J.-S. Pang // *Proc. IJCNN0*, 2006. – P. 3588 – 3505.
9. *Lyubchik L., Kolbasin V. Kernel-based methods for non-stationary time-series identification and prediction* / L. Lyubchik, V.Kolbasin // *Information Technologies and knowledge*, 2009. – № 3.
10. *Воронцов В. Комбинаторный подход к оценке качества обучаемых алгоритмов* / В. Воронцов Под ред. О.Б. Луфанов. – М.: Физматлит, 2004. – Т.13. – С. 5 – 36.
11. *Javier V., Santamar I. Nonlinear System Identification using a New Sliding Window Kernel RLS Algorithm* / V. Javier, I. Santamar // *Journal of Communications*. – 2007. – № 2.
12. *Online learning with kernels* / J. Kivinen, A.J. Smola, R.C. Williamson // *IEEE Transactions on Signal Processing*. – 2004. – №52. – P. 2165 – 2176.
13. *Ljung L. System Identification: Theory for the User*. Prentice Hall / L. Ljung. – New Jersey, Second edition, 1999.
14. *Dorffner G. Neural networks for time series processing* / G. Dorffner // *Neural Network World* – 1996. – №6(4). – P. 447–468.
15. *Vapnik V. Statistical Learning Theory* / V. Vapnik. – Wiley, New-York, 1998.
16. *Kernel based partially linear models and nonlinear identification* / M. Espinoza, J.A.K. Suykens, and B. De Moor // *IEEE Transactions on Automatic Control*. – 2005. – №50 (10). – P. 1602–1606.
17. *Least Squares Support Vector Machines* / J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle // *World Scientific*, Singapore. – 2002.

Поступила в редакцию 30.06.2012

**Рецензент:** д-р техн. наук, проф. А.И. Стрелков, Харьковский университет Воздушных Сил им. И. Кожедуба, Харьков.

**РЕКУРЕНТНЕ ОЦІНЮВАННЯ МАТРИЦІ ГРАММА  
ПРИ ІДЕНТИФІКАЦІЇ НЕЛІНІЙНИХ НЕСТАЦІОНАРНИХ ДИНАМІЧНИХ СИСТЕМ**

Л.М. Любчик, В.А. Колбасин

*Розглянуто задачу рекурентною ідентифікації нелінійної нестационарної динамічної системи на основі ядерних методів в реальному масштабі часу. На основі методу ковзний контроль запропоновано алгоритм 2-го рівня для оцінки параметра регуляризації. Запропоновано спосіб обчислення критерію якості для алгоритму другого рівня.*

**Ключові слова:** ядерні методи, матриця Грамма, регуляризації, ковзний контроль рекурентні алгоритми..

**RECURRENT ESTIMATION OF TRANSFER GRAM MATRIX  
FOR IDENTIFICATION OF NONLINEAR NONSTATIONARY DYNAMIC SYSTEMS**

L. M. Lyubchyk, V.A. Kolbasin

*The problem of recursive identification of nonlinear time-dependent dynamical systems based on nuclear techniques in real time. On the basis of a sliding control algorithm level 2 to estimate the regularization parameter. The way of calculation of the quality criterion for the algorithm of the second level.*

**Keywords:** nuclear techniques Gram matrix, regularization, cross-validation, recursive algorithms.