Методы аппроксимации обратной ковариационной матрицы для эффективной оптимизации правдоподобия гауссовского процесса

Даниил Кононенко
1. Институт Проблем Передачи Информации
127994, г.Москва, ГСП-4, Большой Каретный Переулок, 19, стр.1
2. DATADVANCE
105064, г.Москва, Садово-Черногрязская улица, 13/3
3. МФТИ
141700, г.Долгопрудный, Институтский переулок, 9
daniil.kononenko@datadvance.net

Аннотация

Восстановление регрессии на основе гауссовских процессов — байесовский непараметрический метод, показывающий хорошие результаты во многих приложениях. Параметры модели настраиваются с помощью метода максимума правдоподобия. Каждый раз при подсчете правдоподобия и его производных необходимо выполнять подсчет обратной матрицы ковариации и ее детерминанта, что занимает порядка $O(N^3)$ операций, где N — размер обучающей выборки. В работе предлагается метод аппроксимации обратной матрицы ковариации и ее детерминанта за $O(N^2)$ операций. Проведен ряд вычислительных экспериментов, которые показывают значительное ускорение предложенного метода по сравнению со стандартными подходами.

1. Введение

При построении моделей на основе данных часто необходимо решать задачу аппроксимации неизвестной зависимости. Одним из наиболее популярных методов решения этой задачи является регрессия на основе гауссовских процессов. Наиболее распространенным способом выбора параметров ковариационной функции гауссовского процесса по выборке является метод максимума правдоподобия [1],[7].

Одной из проблем при практической реализации данного метода является значительное время обучения при больших размерах выборок. В стандартных реализациях одна оценка правдоподобия и его производных требует порядка $O(N^3)$ операций, где N — размер обучающей выборки [7]. При использовании большинства классических методов оптимизации для нахождения максимума правдоподобия, например, градиентных методов, необходимо несколько сотен оценок целевой функции, что делает время обучения достаточно большим уже при $N \approx 500$ [4].

Цель данной работы — предложить алгоритм, уменьшающий время обучения без значительной потери качества аппроксимации. В разделе 2 приводится постановка задачи аппроксимации на основе гауссовских процессов. В разделе 3 делается обзор сущетсвующих методом ускорения вычисления правдоподобия и его производных. В разделе 4 описывается предлагаемый алгоритм обучения, и в разделах 5, 6 обсуждаются результаты вычислительного эксперимента.

2. Гауссовские процессы

Пусть $f(x)$ — некоторая неизвестная функциональная зависимость, $f(x) \in \mathbb{R}$, $x \in \mathbb{X} \subset \mathbb{R}^n$, $f(x)$ — гладкая функция. Пусть зависимость $y(x)$ порождена моделью: $y(x) = f(x) + \varepsilon(x)$, $\varepsilon(x) \sim \mathcal{N}(0, \sigma^2)$ — гауссовский белый шум. Пусть также известна обучающая выборка $D = (\mathbf{x}, \mathbf{y}) = \left\{(x_i, y_i = y(x_i))\right\}_{i=1}^{N}$. Задача состоит в построении аппроксимации неизвестной зависимости $y = \hat{f}(x)$. Оценка качества полученной аппроксимации — средняя квадратичная ошибка на независимой тестовой выборке $D' = \left\{(x'_j, y'_j = f(x'_j))\right\}_{j=1}^{N'}$:

$$Q(\hat{f}) = \frac{1}{N'} \sum_{j=1}^{N'} (y'_j - \hat{f}(x'_j))^2. \qquad (1)$$

Построение аппроксимации с помощью гауссовских процессов основано на следующих вероятностных предложениях. Пусть задано случайное поле $f(x, \omega)$, $\omega \in \Omega$ — случайное событие в Ω. Предполагается, что для произвольного $x \in \mathbb{X}$ существуют

первый и второй моменты:

$$\mathsf{M}(x) = \mathbb{E}f(x),$$
$$\mathsf{K}(x_1,x_2) = \mathbb{E}(f(x_1) - \mathbb{E}f(x_1))(f(x_2) - \mathbb{E}f(x_2)),$$

а также условное математическое ожидание

$$\mathbb{E}(f(x)|f(x_1),f(x_2),\ldots,f(x_l)).$$

Далее будем предполагать без ограничения общности, что что среднее значение нулевое $\mathsf{M}(x) = 0$. Предположим также, что случайное поле — гауссовское. Для такого поля совместное распределение $f(x_1),f(x_2),\ldots,f(x_l)$ — нормальное и, следовательно, определяется математическим ожиданием и ковариационной функцией. Пусть ковариационная функция $K_0(x,x')$ гауссовского поля $f(x)$ принадлежит некоторому параметрическому семейству

$$K_0(x,x') = K_0(x,x'|\Theta), \tag{2}$$

где Θ — некоторый набор параметров. Тогда ковариационная функция процесса $y(x)$ имеет вид:

$$K(x,x'|\Theta,\sigma^2) = K_0(x,x'|\Theta) + \sigma^2\delta(x,x'). \tag{3}$$

При таких предположениях условное распределение $y(x)$ в произвольной точке x не из обучающей выборки имеет вид [7]:

$$Law(y(x)|y(x_1),y(x_2),\ldots,y(x_n)) \sim \mathscr{N}(\hat{f}(x),\hat{\sigma}^2(x))).$$

Математическое ожидание используются для построения аппроксимации [7]:

$$\hat{f}(x) = \mathbf{k}(x)\mathrm{K}^{-1}\mathbf{y},$$

оценка дисперсии имеет вид [7]:

$$\hat{\sigma}^2(x) = K_0(x,x) + \sigma^2 - \mathbf{k}(x)\mathrm{K}^{-1}\mathbf{k}(x)^T,$$

где $\mathbf{k}(x) = \{K(x,x_i)\}_{i=1}^N$, $\mathrm{K} = \{K(x_i,x_j)\}_{i,j=1}^N$.

Значения неизвестных параметров Θ,σ^2 восстанавливаются по обучающей выборке D с помощью метода максимума правдоподобия. Логарифм правдоподобия имеет вид [7]:

$$l(\mathbf{a}) = \log p(\mathbf{y}|\mathbf{x},\mathbf{a}) = -\frac{1}{2}\mathbf{y}^T\mathrm{K}^{-1}\mathbf{y} - \frac{1}{2}\log|\mathrm{K}| - \frac{N}{2}\log 2\pi, \tag{4}$$

где $\mathbf{a} = \{\Theta,\sigma^2\}$

3. Практическая реализация

Оценки параметров

$$\hat{\mathbf{a}} = \arg\max_{\mathbf{a}} l(\mathbf{a}) \tag{5}$$

предлагается искать с помощью метода сопряженных градиентов с мультистартом. Производная правдоподобия по параметру a_i [7]:

$$\frac{\partial l}{\partial a_i} = -\mathrm{tr}\left(\mathrm{K}^{-1}\frac{\partial \mathrm{K}}{\partial a_i}\right) + \mathbf{y}^T\mathrm{K}^{-1}\frac{\partial \mathrm{K}}{\partial a_i}\mathrm{K}^{-1}\mathbf{y}. \tag{6}$$

Чтобы вычислить правдоподобие и его производные в некоторой точке, необходимо знать значения следующих величин:

$$\mathrm{K}^{-1}\mathbf{y},\ \mathrm{tr}\left(\mathrm{K}^{-1}\frac{\partial \mathrm{K}}{\partial a_i}\right),\ \log(\det(\mathrm{K})). \tag{7}$$

Стандартным подходом является использование разложения Холецкого $\mathrm{K} = \mathrm{U}^T\mathrm{U}$, здесь U — верхняя треугольная матрица. Матрица K имеет размерность $N \times N$, поэтому вычисление разложения Холецкого требует порядка $O(N^3)$ операций [7].

Существует несколько подходов для приближенного вычисления величин (7).

Одним из возможных методов является переход к разреженным гауссовским процессам [7]. Их основной идеей является аппроксимация матрицы K матрицей неполного ранга с помощью введения некоторого (меньшего, чем число прецедентов) числа новых входов, например, из числа прецедентов. Если число новых прецедентов $M < N$, то сложность вычисления величин (7) становится равной $O(M^2N)$. Однако, чтобы добиться значительного ускорения, необходимо в несколько раз уменьшать число входов (например $M \sim \frac{N}{3}$), что значительно уменьшает точность аппроксимации. Кроме того, такой метод может быть нестабилен в силу итеративного характера процедуры оптимизации правдоподобия.

Другие подходы используют аппроксимацию непосредственно величин (7). Основным методом аппроксимации величины $\mathrm{K}^{-1}\mathbf{y}$ является численная минимизация квадратичной функции [3]

$$g(u) = \frac{1}{2}u^T\mathrm{K}u - u^T\mathbf{y}. \tag{8}$$

Точный минимум $u^* = \mathrm{K}^{-1}\mathbf{y}$. Используя градиентный метод оптимизации, например, метод сопряженных градиентов, приближенное решение может быть найдено за $O(N^2)$ операций [3], [4].

Аппроксимация $\mathrm{tr}\left(\mathrm{K}^{-1}\frac{\partial \mathrm{K}}{\partial a_i}\right)$ может быть осуществлена с помощью метода Монте-Карло [3]:

$$\mathrm{tr}\left(\mathrm{K}^{-1}\frac{\partial \mathrm{K}}{\partial a_i}\right) = E\left[d^T\mathrm{K}^{-1}\frac{\partial \mathrm{K}}{\partial a_i}d\right], \tag{9}$$

где d — случайный вектор, имеющий стандартное нормальное распределение.

В работах [5], [6], [8], [9] предложен альтернативный метод аппроксимации величин (7). Итеративный характер любой численной процедуры оптимизации правдоподобия (5) ведет, в частности, к тому, что правдоподобие (4) и его производные (6) на последовательных итерациях j и $(j+1)$ вычисляются в достаточно близких друг к другу точках $\mathbf{a}_j$ и $\mathbf{a}_{j+1}$. Предлагается итеративный метод аппроксимации обратной ковариационной матрицы K_{j+1}^{-1} по

известной матрице K_j^{-1}. Для этого квадратичная форма (8) оптимизируется квазиньютоновским методом [4]. Его преимущество в том, что обратная ковариационная матрица обновляется итеративно вместе с оптимизацией по u:

$$\begin{aligned} u_{j+1} &= u_j - \eta_j \mathrm{K}_j^{-1} \bigtriangledown g(u_j), \\ \mathrm{K}_{j+1}^{-1} &= \mathrm{K}_j^{-1} + \triangle \mathrm{K}_j^{-1}. \end{aligned} \tag{10}$$

Здесь j и $(j+1)$ — последовательные итерации оптимизации квадратичной функции (8), u_j и K_j^{-1} — приближения целевой переменной и обратной матрицы на этих итерациях, $\bigtriangledown g(u) = \mathrm{K}u - \mathbf{y}$, η_j — длина шага линейного поиска. Предлагаемый метод обновления обратной матрицы $\triangle \mathrm{K}_j^{-1}$ — формула BFGS [4]:

$$\triangle \mathrm{K}_j^{-1} = \left(1 + \frac{q_j^T \mathrm{K}_j^{-1} q_j}{q_j^T p_j}\right) \frac{p_j p_j^T}{p_j^T q_j} - \frac{p_j q_j^T \mathrm{K}_j^{-1} + \mathrm{K}_j^{-1} q_j p_j^T}{q_j^T p_j}, \tag{11}$$

где $p_j = u_{j+1} - u_j$, $q_j = \bigtriangledown g(u_{j+1}) - \bigtriangledown g(u_j)$.

При вычислении правдоподобия в точке a_{j+1} в качестве начальных приближений u_0 и K_0^{-1} берутся значения, полученные в предыдущей точке a_j. Таким образом, величины $\mathrm{K}^{-1}\mathbf{y}$ и K^{-1} могут быть найдены итеративно. Каждое обновление (10) требует $O(N^2)$ операций. Теоретически метод сходится за N итераций [4], и тогда сложность подсчета правдоподобия равна $O(N^3)$. Однако достаточная точность подсчета величин u_j и K_j^{-1} часто достигается раньше. Авторы предлагают два критерия остановки оптимизации (8):

$$||\bigtriangledown g(u_j)||_\infty \leq \frac{\varepsilon}{N} \text{ или } j > J,$$

где ε, J — фиксированные константы. Тогда сложность метода порядка $JN^2 = O(N^2)$ операций.

Т.к. предыдущее приближение используется в качестве инициализации для нового, то ошибка аппроксимации может накапливаться. Также предыдущее значение $\mathbf{a}_j$ может достаточно сильно отличаться от $\mathbf{a}_{j+1}$. Тогда достаточная точность аппроксимации обратной матрицы может быть не достигнута. Предлагается следующий критерий качества:

$$\frac{|\mathrm{tr}(\tilde{\mathrm{K}}^{-1}\mathrm{K}) - N|}{N} < \varepsilon_1, \tag{12}$$

где $\tilde{\mathrm{K}}^{-1}$ — построенная аппроксимация, ε_1 — фиксированная константа. Если этот критерий не выполнен, то производится рестарт: для данного значения параметров $\mathbf{a}_j$ обратная матрица вычисляется точно с помощью преобразования Холецкого.

Таким образом, в ходе оптимизации (5) при подсчете правдоподобия и его производных может быть 2 варианта:

1. Критерий (12) выполнен, аппроксимация достаточно точна. Тогда сложность равна $O(N^2)$ операций.
2. Критерий не выполнен, тогда производится рестарт и сложность становится равной $O(N^3)$ операций.

В работе [9] предлагается дополнение к методу обновления (11): на каждой итерации выбирается случайная размерность $i \in \{1, \ldots, N\}$ матрицы K, вычисляется величина $s = \mathrm{K}_j^{-1}(i,:) \times \mathrm{K}(:,i)$ и на нее делится текущее приближение: $\mathrm{K}_j^{-1} := \frac{\mathrm{K}_j^{-1}}{s}$. Таким образом предлагается точнее аппроксимировать информацию об амплитуде матрицы K.

Для аппроксимации члена $\log(\det(\mathrm{K}))$ в работах [6], [8] предлагается следующий подход.

$$\begin{aligned} &\log(\det(\mathrm{K})) = \mathrm{tr}(\log(K)), \\ &\log(\mathrm{K}) = -\sum_{i=1}^{\infty} \left(\frac{\mathrm{K}^i}{i}\right) \text{ — матричный логарифм.} \end{aligned} \tag{13}$$

Используется двухуровневая схема:

1. Берется некоторое конечное число членов в сумме (13): $\log(\mathrm{K}) = -\sum_{i=1}^{L} \left(\frac{\mathrm{K}^i}{i}\right)$
2. Каждый элемент вида $\mathrm{tr}(\mathrm{K}^i)$ оценивается с помощью метода Монте-Карло (9).

Каждый из двух этапов вносит ошибку в итоговую аппроксимацию, и авторами предлагается ряд компенсирующих схем, уменьшающих эти ошибки.

4. Предлагаемый метод аппроксимации

В данной работе развивается подход, предложенный в [5], [6], [8], [9]. При использовании квазиньютоновских методов для аппроксимации K^{-1} с помощью (10) используются обновления низкого ранга: $\mathrm{K}_{j+1}^{-1} = \mathrm{K}_j^{-1} + uv^T$, где u, v — некоторые вектора. Тогда, зная детерминант матрицы K_j^{-1}, можно точно вычислить детерминант матрицы K_{j+1}^{-1}, используя лемму об определителе матрицы:

$$\det(A + uv^T) = \det(A)(1 + v^T A^{-1} u). \tag{14}$$

Если использовать формулу BFGS (11) для обновления обратной матрицы (10), то из (14), получаем

$$\det(\mathrm{K}_{j+1}^{-1}) = \det(\mathrm{K}_j^{-1}) \frac{p_j^T \mathrm{K}_j p_j}{q_j^T p_j}, \tag{15}$$

где p_j, q_j имеют тот же смысл, что и в (11).

Видно, что для того, чтобы итеративно обновлять детерминант матрицы, необходимо знать текущее приближение для необращенной матрицы K_j. Для этого на каждой итерации необходимо обновлять еще и значение K_j, что влечет дополнительные

временные затраты. Поэтому предлагается использовать формулу DFP [4]:

$$\triangle \mathrm{K}_j^{-1} = \frac{p_j p_j^T}{q_j^T p_j} - \frac{\mathrm{K}_j^{-1} q_j q_j^T (\mathrm{K}_j^{-1})^T}{q_j^T \mathrm{K}_j^{-1} q_j}. \qquad (16)$$

Тогда формула для обновления значения детерминанта имеет вид

$$\det(\mathrm{K}_{j+1}^{-1}) = \det(\mathrm{K}_j^{-1}) \frac{q_j^T p_j}{q_j^T \mathrm{K}_j^{-1} q_j}, \qquad (17)$$

в который уже не входит необращенная матрица K_j.

Для более точной аппроксимации информации об амплитуде матрицы K предлагается следующий метод: на каждой итерации матрица K_j^{-1} делится на среднее значение диагонали $s = \mathrm{diag}(\mathrm{K}_j^{-1} \times \mathrm{K})$. Тогда значение критерия (12) всегда становится равным нулю. Вместо него предлагается использовать следующие два критерия для рестарта:

1. Если начальная норма градиента $||\nabla g(u_0)|| > h$, где h — некоторый порог, то выполняется рестарт. Рассматривается l_1-норма $||\nabla g(u_0)|| = \max\limits_{i=1,\ldots,N} g_i(u_0)$
2. Если в ходе итеративной процедуры норма градиента начинает увеличиваться (что означает, что процесс начинает расходиться в силу численных проблем), то выполняется рестарт.

При вычислении правдоподобия в точке a_{j+1} в качестве начальных приближений для u_0, K_0^{-1} и $\det(\mathrm{K}_0^{-1})$ предлагается брать значения в одной из предыдущих точек $a_1, \ldots, a_j$, ближайшей к a_{j+1} в смысле некоторой метрики $\rho(a,a')$. Вид метрики зависит от вида ковариационной функции (2), (3). Например, для экспоненциальной ковариационной функции

$$K_0(x,x') = \sigma_0^2 \exp\left(-\sum_{i=1}^{n} \theta_i^2 (x_i - x_i')^2 \right),$$

где $a = \{\sigma_0^2, \theta_1^2, \ldots, \theta_n^2, \sigma^2\}$, предлагается использовать следующую метрику:

$$\rho(a,a') = \sum_{i=1}^{n} (\exp(-\theta_i) - exp(-\theta_i'))^2 + \left(\frac{\sigma^2}{\sigma_0^2} - \frac{\sigma'^2}{\sigma_0'^2} \right)^2.$$

Приведем весь алгоритм аппроксимации величин (7):

1. Инициализируем $\mathrm{K}_0^{-1} = \mathrm{C}^{-1}$, $u_0 = \alpha$, $\det(\mathrm{K}_0^{-1}) = d$ где C^{-1}, α, d — величины K^{-1}, $\mathrm{K}^{-1}\mathbf{y}$, $\det(\mathrm{K}^{-1})$, подсчитанные на одной из предыдущих итераций оптимизации (5) в точке, ближайшей к текущей в смысле метрики $\rho(a,a')$.
2. Если $||\nabla g(u_0)|| > h$, то не пробуем аппроксимировать величины (7), а сразу вычисляем точно с помощью преобразования Холецкого.
3. Иначе для $j := 0, \ldots, J-1$
   (a) Если $||\nabla g(u_j)|| \le \frac{\varepsilon}{N}$, то аппроксимация достаточно точная, выходим из цикла по j.
   (b) Оптимальный шаг $\eta_j = -\frac{(\nabla g(u_j))^T \mathrm{K} \nabla g(u_j)}{(\nabla g(u_j))^T \mathrm{K}_j^{-1} \mathrm{K} \mathrm{K}_j^{-1} \nabla g(u_j)}$
   (c) $u_{j+1} = u_j - \eta_j \mathrm{K}_j^{-1} \nabla g(u_j)$
   (d) $\mathrm{K}_{j+1}^{-1} = \mathrm{K}_j^{-1} + \triangle \mathrm{K}_j^{-1}$, $\triangle \mathrm{K}_j^{-1}$ определяется по формуле (16).
   (e) $\det(\mathrm{K}_{j+1}^{-1}) = \det(\mathrm{K}_j^{-1}) \frac{q_j^T p_j}{q_j^T \mathrm{K}_j^{-1} q_j}$, где q_j, p_j определены в (11).
   (f) $m = \mathrm{mean}(\mathrm{diag}(\mathrm{K}_{j+1}^{-1} \times \mathrm{K}))$, где $mean(\mathbf{v}) = \frac{1}{n} \sum\limits_{i=1}^{n} v_i$ — среднее значение вектора.
   (g) $\mathrm{K}_{j+1}^{-1} := \frac{\mathrm{K}_{j+1}^{-1}}{m}$
   (h) $\det(\mathrm{K}_{j+1}^{-1}) := \frac{\det(\mathrm{K}_{j+1}^{-1})}{m^N}$
   (i) Если $||\nabla g(u_{j+1})|| > ||\nabla g(u_j)||$, то выходим из цикла по j и вычисляем (7) точно с помощью преобразования Холецкого.

5. Вычислительный эксперимент

В вычислительных экспериментах сравнивалось качество и время работы алгоритма с использованием аппроксимации, предложенной в данной работе, и алгоритма без аппроксимации, в котором величины (7) всегда вычисляются точно с помощью разложения Холецкого.

Для демонстрации экспериментальных результатов был использован большой набор тестовых функций, которые применяются для тестирования задач оптимизации [10], [11]. Размерности функций — от 2 до 8. Всего тестирование проводилось на 211 различных функциях, для каждой из которых генерировались случайные обучающие выборки размером 320 и 1000 точек. Результаты сравнивались по времени обучения и по среднеквадратичной ошибке (1) на больших контрольных выборках из 10000 точек. Для удобства результаты представлены в виде кривых Долан-Мора [2] на рисунках 1, 2, 3, 4. Чем выше кривая находится на графике, тем выше качество или меньше время работы соответствующего алгоритма. Видно, что точность работы двух алгоритмов практически не отличается, а время обучения алгоритма, использующего предложенную аппроксимацию, значительно меньше: ускорение до 8 раз при размере выборки 320 и до 16 раз при размере выборки 1000.

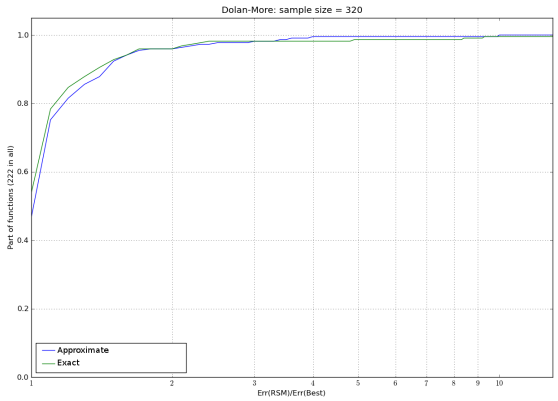

Рис. 1. Размер выборки 320, точность аппроксимации на тестовой выборке

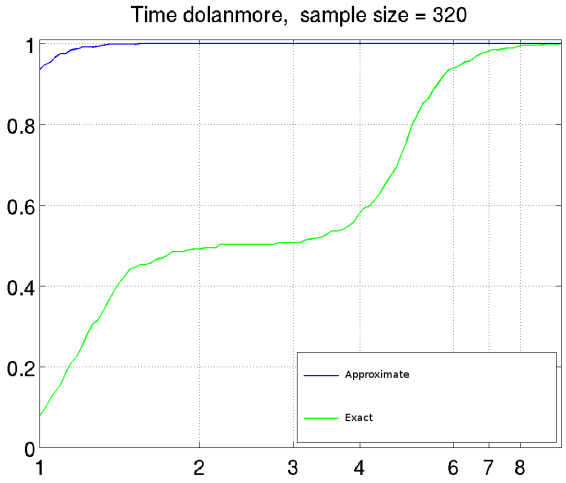

Рис. 2. Размер выборки 320, время обучения

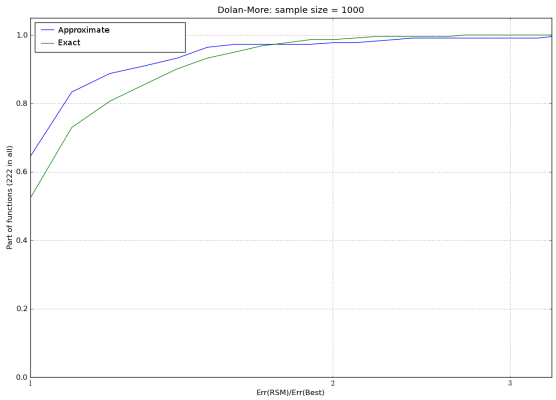

Рис. 3. Размер выборки 1000, точность аппроксимации на тестовой выборке

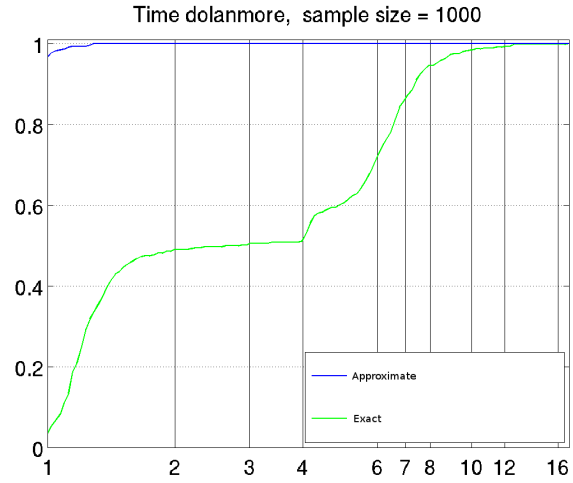

Рис. 4. Размер выборки 1000, время обучения

6. Выводы

Вычислительный эксперимент показал, что предложенный алгоритм дает значительное ускорение по времени обучения без потери точности аппроксимации. Предложенный метод позволяет избежать громоздких схем для аппроксимации $\log(\det(\mathrm{K}))$, предложенных в [6], [8]. Полученное значение аппроксимации детерминанта в предложенном алгоритме в точности совпадает со значением детерминанта аппроксимации обратной матрицы:

$$\tilde{d} = \frac{1}{\det(\tilde{\mathrm{K}}^{-1})},$$

где $\tilde{d}$ — аппроксимация величины $\det(\mathrm{K})$, $\tilde{\mathrm{K}}^{-1}$ — аппроксимация величины K^{-1}. Т.к. качество аппроксимации $\tilde{\mathrm{K}}^{-1}$ контролируется и при необходимости производятся рестарты, то автоматически контролируется качество аппроксимации $\tilde{d}$, в отличие от методов из статей других авторов.

Список литературы

[1] A. S. Alexander I. J. Forrester and A. J. Keane. *Engineering Design via Surrogate Modelling: A Practical Guide*. Wiley, 2008.

[2] E. Dolan and J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2002.

[3] M. Gibbs and D. MacKay. Efficient implementation of gaussian processes. 1997.

[4] S. J. W. Jorge Nocedal. *Numerical Optimization*. Springer, 1999.

[5] W. Leithead and Y. Zhang. O (n 2)-operation approximation of covariance matrix inverse in gaussian process regression based on quasi-newton bfgs method. *Communications in Statistics—Simulation and Computation*, 36(2):367–380, 2007.

[6] W. Leithead, Y. Zhang, and D. Leith. Efficient

gaussian process based on bfgs updating and logdet approximation. In *the 16th IFAC world congress*, 2005.

[7] C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. The MIT Press, 2006.

[8] Y. Zhang and W. Leithead. Approximate implementation of the logarithm of the matrix determinant in gaussian process regression. *journal of Statistical Computation and Simulation*, 77(4):329–348, 2007.

[9] Y. Zhang, W. Leithead, and D. Leith. Random scaling of quasi-newton bfgs method to improve the o (n2)-operation approximation of covariance-matrix inverse in gaussian process. In *Intelligent Control, 2007. ISIC 2007. IEEE 22nd International Symposium on*, pages 452–457. IEEE, 2007.

[10] GDR MASCOT-NUM Toy Functions benchmark. `http://gdr-mascotnum.math.cnrs.fr/data2/benchmarks/jm.pdf`

[11] Lappeenranta University of Technology: evoluationary computation pages - the function testbed. `http://www.it.lut.fi/ip/evo/functions/functions.html`