Конспект лекции
«Линейные динамические системы. Фильтр Калмана.»
по спецкурсу «Структурные методы анализа изображений и сигналов» 2011

Ликбез: некоторые свойства нормального распределения.

Пусть $\boldsymbol{x} \in \mathbb{R}^d$ распределен по нормальному закону, т.е.

$$p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \Sigma) = \frac{1}{\sqrt{2\pi}^d \sqrt{\det \Sigma}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right).$$

Разобьем вектор $\boldsymbol{x}$ на две группы переменных $\boldsymbol{x}_a, \boldsymbol{x}_b$ и обозначим

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_a \\ \boldsymbol{x}_b \end{bmatrix}, \ \boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{bmatrix}, \ \Sigma = \begin{bmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{bmatrix}, \ \Lambda = \Sigma^{-1} = \begin{bmatrix} \Lambda_{aa} & \Lambda_{ab} \\ \Lambda_{ba} & \Lambda_{bb} \end{bmatrix}.$$

Тогда можно показать, что

$$\begin{aligned} &p(\boldsymbol{x}_a) = \mathcal{N}(\boldsymbol{x}_a|\boldsymbol{\mu}_a, \Sigma_{aa}), \\ &p(\boldsymbol{x}_a|\boldsymbol{x}_b) = \mathcal{N}(\boldsymbol{x}_a|\boldsymbol{\mu}_a - \Lambda_{aa}^{-1}\Lambda_{ab}(\boldsymbol{x}_b - \boldsymbol{\mu}_b), \Lambda_{aa}^{-1}). \end{aligned} \tag{1}$$

Это, в частности, означает, что у многомерного нормального распределения все маргинальные и маргинальные условные распределения также являются нормальными.

Рассмотрим величину $\boldsymbol{y} \in \mathbb{R}^D$, которая с точностью до нормального шума связана линейно с величиной $\boldsymbol{x}$, т.е.

$$p(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{y}|A\boldsymbol{x}, \Gamma).$$

Тогда можно показать, что

$$p(\boldsymbol{y}) = \mathcal{N}(\boldsymbol{y}|A\boldsymbol{\mu}, \Gamma + A\Sigma A^T), \tag{2}$$

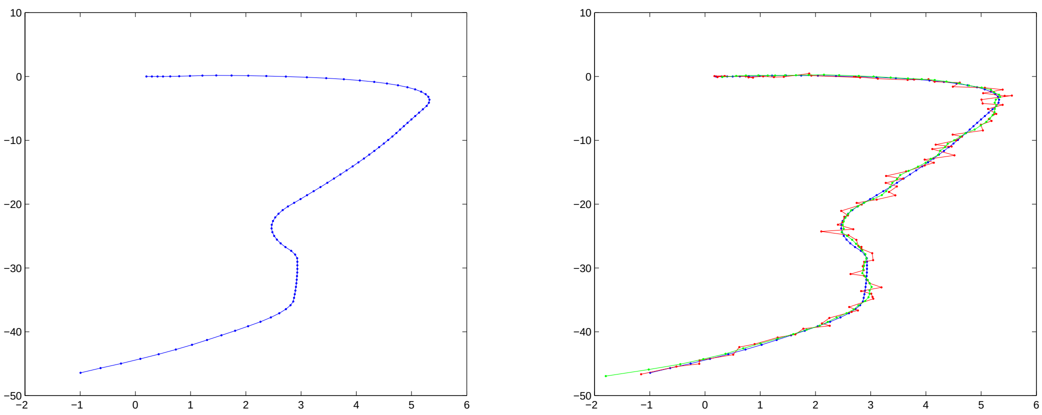
$$p(\boldsymbol{x}|\boldsymbol{y}) = \mathcal{N}(\boldsymbol{x}|P(A^T\Gamma^{-1}\boldsymbol{y} + \Sigma^{-1}\boldsymbol{\mu}), P), \ \ P = (\Sigma^{-1} + A^T\Gamma^{-1}A)^{-1}. \tag{3}$$

В частности, если $\Gamma = 0$, то результат (2) говорит о том, что любые линейные комбинации компонент нормального распределения также распределены нормально.

1. Модельный пример задачи фильтрации сигнала.

Рассмотрим модельную задачу сопровождения (трекинга) объекта. Пусть имеется некоторая траектория объекта в пространстве (см. рис. 1a). При этом координаты объекта в каждый момент времени измеряются с некоторой погрешностью (см. рис. 1b, красная кривая). Задача состоит в том, чтобы уточнить координаты объекта путем сглаживания наблюдаемой траектории (см. рис. 1b, зеленая кривая).

Обозначим через $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ наблюдаемые характеристики объекта в моменты времени $1, \ldots, N$, а через $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_N$ — скрытые (истинные) параметры объекты. Предположим, что динамика изменения параметров объекта во времени является марковским процессом, т.е. величина $\boldsymbol{t}_n$ зависит только от $\boldsymbol{t}_{n-1}$, а наблюдаемые характеристики $\boldsymbol{x}_n$ полностью определяются параметрами объекта $\boldsymbol{t}_n$ в момент времени n. Таким образом, мы получили байесовскую сеть, показанную на рис. 2, где $p(\boldsymbol{t}_n|\boldsymbol{t}_{n-1})$ — модель движения объекта, а $p(\boldsymbol{x}_n|\boldsymbol{t}_n)$ — модель сенсора.



(a) (b)

Рис. 1: Траектория движения некоторого объекта на плоскости. Синяя кривая показывает истинную траекторию объекта, красная кривая — наблюдаемая траектория, зеленая кривая — сглаженная траектория.

Рассмотрим в качестве параметров объекта координаты, скорости и ускорения по каждой координате $\boldsymbol{t}_n = [\xi_1(n), \dot{\xi}_1(n), \ddot{\xi}_1(n), \xi_2(n), \dot{\xi}_2(n), \ddot{\xi}_2(n)]$. Тогда моделировать движение объекта можно следующим образом:

$$\begin{aligned}
&\xi_i(n) = \xi_i(n-1) + \dot{\xi}_i(n-1)\Delta t + \ddot{\xi}_i(n-1)\frac{\Delta t^2}{2} + \varepsilon_{1i},\ i = 1, 2;\\
&\dot{\xi}_i(n) = \dot{\xi}_i(n-1) + \ddot{\xi}(n-1)\Delta t + \varepsilon_{2i},\ i = 1, 2;\\
&\ddot{\xi}_i(n) = \ddot{\xi}_i(n-1) + \varepsilon_{3i},\ i = 1, 2;\\
&\varepsilon_{ji} \sim \mathcal{N}(0, \gamma_{ji}),\ j = 1, 2, 3,\ i = 1, 2.
\end{aligned}$$

Аналогично, модель сенсора можно представить как

$$x_i(n) = \xi_i(n) + \nu_i,\ \nu_i \sim \mathcal{N}(0, \sigma_i),\ i = 1, 2.$$

Не ограничивая общности, можно считать, что $\Delta t = 1$. Тогда модель движения и модель сенсора можно записать в матричном виде следующим образом:

$$\begin{aligned}
&\boldsymbol{t}_n = A\boldsymbol{t}_{n-1} + \boldsymbol{\varepsilon},\ \boldsymbol{\varepsilon} \sim \mathcal{N}(0, \Gamma),\ \Leftrightarrow\ p(\boldsymbol{t}_n|\boldsymbol{t}_{n-1}) = \mathcal{N}(A\boldsymbol{t}_{n-1}, \Gamma),\\
&\boldsymbol{x}_n = C\boldsymbol{t}_n + \boldsymbol{\nu},\ \boldsymbol{\nu} \sim \mathcal{N}(0, \Sigma),\ \Leftrightarrow\ p(\boldsymbol{x}_n|\boldsymbol{t}_n) = \mathcal{N}(C\boldsymbol{t}_n, \Sigma),\\
&A = \begin{pmatrix} 1 & 1 & 0.5 & 0 & 0 & 0\\ 0 & 1 & 1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 1 & 0.5\\ 0 & 0 & 0 & 0 & 1 & 1\\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},\ C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix},\\
&\Gamma = \mathrm{diag}(\gamma_{11}, \ldots, \gamma_{32}),\ \Sigma = \mathrm{diag}(\sigma_1, \sigma_2).
\end{aligned}$$

2. Линейная динамическая система.

Линейной динамической системой (ЛДС) называется байесовская сеть, показанная на рис. 2, где $\boldsymbol{x}_n \in \mathbb{R}^d$, $\boldsymbol{t}_n \in \mathbb{R}^D$, и все атомарные распределения задаются линейной гауссовской моделью:

$$\begin{aligned}
&p(\boldsymbol{t}_n|\boldsymbol{t}_{n-1}) = \mathcal{N}(A\boldsymbol{t}_{n-1}, \Gamma),\\
&p(\boldsymbol{x}_n|\boldsymbol{t}_n) = \mathcal{N}(C\boldsymbol{t}_n, \Sigma),\\
&p(\boldsymbol{t}_1) = \mathcal{N}(\boldsymbol{\mu}_0, \Sigma_0).
\end{aligned} \tag{4}$$

Заметим, что байесовская сеть на рис. 2 соответствует также скрытой марковской модели. Основное отличие ЛДС от СММ заключается в том, что в ЛДС переменные $\boldsymbol{t}_n$ являются непрерывными, а в СММ — дискретными. Совместное распределение всех переменных в ЛДС задается как

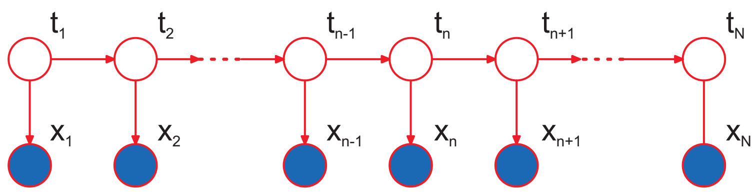


Рис. 2: Графическая модель линейной динамической системы.

$$p(X,T|A,\Gamma,C,\Sigma) = p(\boldsymbol{t}_1)p(\boldsymbol{x}_1|\boldsymbol{t}_1)\prod_{n=2}^{N} p(\boldsymbol{t}_n|\boldsymbol{t}_{n-1})p(\boldsymbol{x}_n|\boldsymbol{t}_n) \propto$$
$$\exp\left[-\frac{1}{2}\left(\sum_{n=1}^{N}(\boldsymbol{x}_n - C\boldsymbol{t}_n)^T\Sigma^{-1}(\boldsymbol{x}_n - C\boldsymbol{t}_n) + \sum_{n=2}^{N}(\boldsymbol{t}_n - A\boldsymbol{t}_{n-1})^T\Gamma^{-1}(\boldsymbol{t}_n - A\boldsymbol{t}_{n-1}) + (\boldsymbol{t}_1 - \boldsymbol{\mu}_0)\Sigma_0^{-1}(\boldsymbol{t}_1 - \boldsymbol{\mu}_0)\right)\right].$$

В показателе экспоненты стоит квадратичная функция относительно переменных модели. Следовательно, совместное распределение $p(X,T)$ является многомерным нормальным распределением. Тогда из свойства (1) следует, что распределение $p(T|X)$, а также все маргинальные распределения вида $p(\boldsymbol{t}_n|X)$, $p(\boldsymbol{t}_n|\boldsymbol{x}_1,\dots,\boldsymbol{x}_n)$ также являются нормальными.

У нормального распределения математическое ожидание совпадает с модой. Это означает, что в ЛДС наиболее вероятная конфигурация T при известном X определяется математическим ожиданием нормального распределения $p(T|X)$. Рассмотрим маргинальное распределение $p(\boldsymbol{t}_n|X)$. Из свойства (1) следует, что математическое ожидание $p(\boldsymbol{t}_n|X)$ определяется соответствующей компонентой математического ожидания распределения $p(T|X)$. Таким образом, в линейной динамической системе знание маргинальных распределений $p(\boldsymbol{t}_n|X)$ позволяет найти и наиболее вероятную конфигурацию всех скрытых переменных модели T. В результате, для ЛДС аналог алгоритма Витерби не требуется. Заметим, что в СММ, в отличие от ЛДС, наиболее вероятная конфигурация T не состоит, вообще говоря, из индивидуально наиболее вероятных состояний $\boldsymbol{t}_n^* = \arg\max p(\boldsymbol{t}_n|X)$.

3. Вывод в ЛДС: фильтр Калмана.

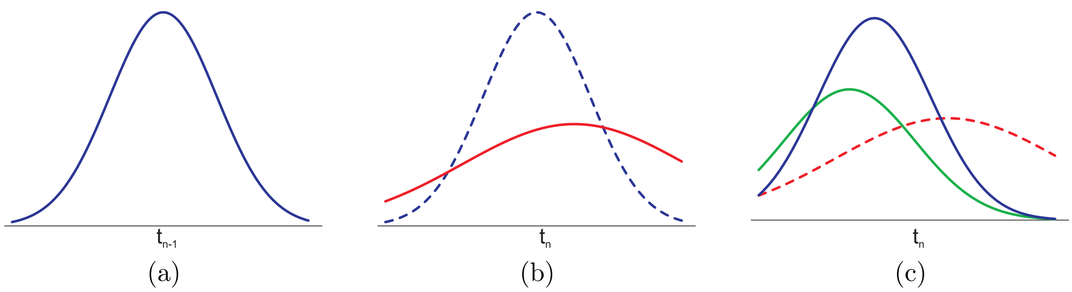


Рис. 3: Прогнозирование с помощью фильтра Калмана. На рис. a показано текущее распределение для $\boldsymbol{t}_{n-1}$ (синяя кривая), на рис. b показано прогнозное распределение для $\boldsymbol{t}_n$ (красная кривая), на рис. c показан уточненный прогноз для $\boldsymbol{t}_n$ после прихода значения $\boldsymbol{x}_n$.

Рассмотрим задачу фильтрации сигнала в реальном времени. Это соответствует поиску распределений $p(\boldsymbol{t}_n|\boldsymbol{x}_1,\dots,\boldsymbol{x}_n)$ для каждого момента времени $n = 1,\dots,N$. Как было показано выше, все эти распределения являются нормальными:

$$p(\boldsymbol{t}_n|\boldsymbol{x}_1,\dots,\boldsymbol{x}_n) = \mathcal{N}(\boldsymbol{t}_n|\boldsymbol{\mu}_n, V_n).$$

Пусть известно распределение $p(\boldsymbol{t}_{n-1}|\boldsymbol{x}_1,\dots,\boldsymbol{x}_{n-1})$ для момента времени $n-1$. Тогда прогноз значения $\boldsymbol{t}_n$ вычисляется следующим образом:

$$p(\boldsymbol{t}_n|\boldsymbol{x}_1,\dots,\boldsymbol{x}_{n-1}) = \int p(\boldsymbol{t}_n,\boldsymbol{t}_{n-1}|\boldsymbol{x}_1,\dots,\boldsymbol{x}_{n-1})d\boldsymbol{t}_{n-1} = \int p(\boldsymbol{t}_n|\boldsymbol{t}_{n-1})p(\boldsymbol{t}_{n-1}|\boldsymbol{x}_1,\dots,\boldsymbol{x}_{n-1})d\boldsymbol{t}_{n-1} =$$
$$\int \mathcal{N}(\boldsymbol{t}_n|A\boldsymbol{t}_{n-1},\Gamma)\mathcal{N}(\boldsymbol{t}_{n-1}|\boldsymbol{\mu}_{n-1},V_{n-1})d\boldsymbol{t}_{n-1} = \mathcal{N}(\boldsymbol{t}_n|A\boldsymbol{\mu}_{n-1},\Gamma + AV_{n-1}A^T).$$

Последнее равенство следует из свойства (2) для нормальных распределений. Таким образом,

$$p(\boldsymbol{t}_n|\boldsymbol{x}_1,\dots,\boldsymbol{x}_{n-1}) = \mathcal{N}(\boldsymbol{t}_n|\tilde{\boldsymbol{\mu}}_n,\tilde{V}_n),\ \ \tilde{\boldsymbol{\mu}}_n = A\boldsymbol{\mu}_{n-1},\ \ \tilde{V}_n = \Gamma + AV_{n-1}A^T. \tag{5}$$

После того, как значение $\boldsymbol{x}_n$ становится известным, можно уточнить прогноз для $\boldsymbol{t}_n$:

$$p(\boldsymbol{t}_n|\boldsymbol{x}_1,\dots,\boldsymbol{x}_n) = \frac{p(\boldsymbol{t}_n,\boldsymbol{x}_1,\dots,\boldsymbol{x}_n)}{p(\boldsymbol{x}_1,\dots,\boldsymbol{x}_n)} = \frac{p(\boldsymbol{x}_n|\boldsymbol{t}_n)p(\boldsymbol{x}_1,\dots,\boldsymbol{x}_{n-1}|\boldsymbol{t}_n)p(\boldsymbol{t}_n)}{p(\boldsymbol{x}_n|\boldsymbol{x}_1,\dots,\boldsymbol{x}_{n-1})p(\boldsymbol{x}_1,\dots,\boldsymbol{x}_{n-1})} =$$
$$= \frac{p(\boldsymbol{x}_n|\boldsymbol{t}_n)p(\boldsymbol{t}_n|\boldsymbol{x}_1,\dots,\boldsymbol{x}_{n-1})}{p(\boldsymbol{x}_n|\boldsymbol{x}_1,\dots,\boldsymbol{x}_{n-1})} \propto \mathcal{N}(\boldsymbol{x}_n|C\boldsymbol{t}_n,\Sigma)\mathcal{N}(\boldsymbol{t}_n|\tilde{\boldsymbol{\mu}}_n,\tilde{V}_n) = \mathcal{N}(\boldsymbol{t}_n|\boldsymbol{\mu}_n,V_n).$$

$$\begin{aligned}
\boldsymbol{\mu}_n &= \tilde{\boldsymbol{\mu}}_n + K_n(\boldsymbol{x}_n - C\tilde{\boldsymbol{\mu}}_n),\\
V_n &= (I - K_nC)\tilde{V}_n,\\
K_n &= \tilde{V}_nC^T(C\tilde{V}_nC^T + \Sigma)^{-1}.
\end{aligned} \tag{6}$$

Этот результат следует из свойства (3) для нормальных распределений.

Таким образом, фильтр Калмана состоит из двух шагов. Пусть имеется текущее (априорное) распределение $p(\boldsymbol{t}_{n-1}|\boldsymbol{x}_1,\dots,\boldsymbol{x}_{n-1})$ (см. рис. 3a). На первом шаге осуществляется прогноз значения $\boldsymbol{t}_n$ по формулам (5) (см. рис. 3b). При этом дисперсия прогноза (матрица $\tilde{V}_n$) увеличивается по сравнению с дисперсией для $\boldsymbol{t}_{n-1}$. Затем, на втором шаге, происходит коррекция прогноза для $\boldsymbol{t}_n$ с учетом новой информации $\boldsymbol{x}_n$ (формулы (6)). При этом дисперсия прогноза V_n уменьшается по сравнению с $\tilde{V}_n$ (см. рис. 3c).

4. Вывод в ЛДС: РТС уравнения.

Рассмотрим задачу фильтрации в случае, когда наблюдаемый сигнал X известен полностью до начала процедуры фильтрации. Эта задача соответствует поиску распределений $p(\boldsymbol{t}_n|\boldsymbol{x}_1,\dots,\boldsymbol{x}_N)$. Такие распределения также нужны для решения задачи обучения параметров ЛДС $(A, C, \Gamma, \Sigma, \boldsymbol{\mu}_0, V_0)$.

Как уже было замечено выше, графическая модель ЛДС совпадает с аналогичной для СММ. Следовательно, алгоритм вывода в СММ «вперед-назад» (реализация общего алгоритма SUM-PRODUCT для графической модели типа цепочка) может быть использован и для ЛДС с тем отличием, что суммы по $\boldsymbol{t}_n$ заменяются на интегралы:

$$\gamma(\boldsymbol{t}_n) = p(\boldsymbol{t}_n|X) = \hat{\alpha}(\boldsymbol{t}_n)\hat{\beta}(\boldsymbol{t}_n) = \mathcal{N}(\boldsymbol{t}_n|\hat{\boldsymbol{\mu}}_n,\hat{V}_n),$$
$$c_n\hat{\alpha}(\boldsymbol{t}_n) = p(\boldsymbol{x}_n|\boldsymbol{t}_n)\int\hat{\alpha}(\boldsymbol{t}_{n-1})p(\boldsymbol{t}_n|\boldsymbol{t}_{n-1})d\boldsymbol{t}_{n-1}, \tag{7}$$
$$c_{n+1}\hat{\beta}(\boldsymbol{t}_n) = \int\hat{\beta}(\boldsymbol{t}_{n+1})p(\boldsymbol{x}_{n+1}|\boldsymbol{t}_{n+1})p(\boldsymbol{t}_{n+1}|\boldsymbol{t}_n)d\boldsymbol{t}_{n+1}. \tag{8}$$

Проход вперед в этом алгоритме, т.е. вычисление c_n и $\hat{\alpha}(\boldsymbol{t}_n) = p(\boldsymbol{t}_n|\boldsymbol{x}_1,\dots,\boldsymbol{x}_n)$, в точности соответствует фильтру Калмана, рассмотренному в предыдущем пункте. Проход назад в случае ЛДС получил название РТС уравнений (по первым буквам фамилий авторов — Rauch, Tung, Striebel). Можно показать, что для прохода назад справедливы следующие формулы:

$$\begin{aligned}
p(\boldsymbol{t}_n|X) &= \mathcal{N}(\boldsymbol{t}_n|\hat{\boldsymbol{\mu}}_n,\hat{V}_n),\\
\hat{\boldsymbol{\mu}}_n &= \boldsymbol{\mu}_n + J_n(\hat{\boldsymbol{\mu}}_{n+1} - A\boldsymbol{\mu}_n),\\
\hat{V}_n &= V_n + J_n(\hat{V}_{n+1} - \tilde{V}_n)J_n^T.
\end{aligned}$$

По аналогии с алгоритмом «вперед-назад» для СММ, мы также можем эффективно вычислить условное распределение для соседних скрытых переменных $\boldsymbol{t}_{n-1}, \boldsymbol{t}_n$:

$$\xi(\boldsymbol{t}_{n-1},\boldsymbol{t}_n) = p(\boldsymbol{t}_{n-1},\boldsymbol{t}_n|X) = \frac{1}{c_n}\hat{\alpha}(\boldsymbol{t}_n)p(\boldsymbol{x}_n|\boldsymbol{t}_n)p(\boldsymbol{t}_n|\boldsymbol{t}_{n-1})\hat{\beta}(\boldsymbol{t}_n) = \mathcal{N}([\boldsymbol{t}_{n-1},\boldsymbol{t}_n]|[\gamma(\boldsymbol{t}_{n-1}),\gamma(\boldsymbol{t}_n)], J_{n-1}\hat{V}_n).$$

Объединяя все вышесказанное, получаем общую схему алгоритма «вперед-назад» для ЛДС (см. Алгоритм 1).

5. Обучение фильтра Калмана.

Рассмотрим задачу обучения параметров ЛДС по данным. Будем решать эту задачу с помощью метода максимального правдоподобия, т.е.

$$\log p(X|A,\Gamma,C,\Sigma,\boldsymbol{\mu}_0,V_0) = \log\int p(X,T|A,\Gamma,C,\Sigma,\boldsymbol{\mu}_0,V_0)dT \to \max_{A,\Gamma,C,\Sigma,\boldsymbol{\mu}_0,V_0}$$

Алгоритм 1: Алгоритм «вперед-назад» для ЛДС

Вход: $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N$ – наблюдаемый сигнал, $(A, \Gamma, C, \Sigma, \boldsymbol{\mu}_0, V_0)$ – параметры ЛДС
Выход: $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_N, V_1, \dots, V_N$ – параметры распределений $p(\boldsymbol{t}_n|\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)$;
$\hat{\boldsymbol{\mu}}_1, \dots, \hat{\boldsymbol{\mu}}_N, \hat{V}_1, \dots, \hat{V}_N$ – параметры распределений $p(\boldsymbol{t}_n|\boldsymbol{x}_1, \dots, \boldsymbol{x}_N)$;
$L = \log p(X|A, \Gamma, C, \Sigma, \boldsymbol{\mu}_0, V_0)$ – логарифм неполного правдоподобия.

// Проход вперед (фильтр Калмана)
$K_1 = V_0 C^T (C V_0 C^T + \Sigma)^{-1}$;
$c_1 = \mathcal{N}(\boldsymbol{x}_1 | C\boldsymbol{\mu}_0, C V_0 C^T + \Sigma)$;
$\boldsymbol{\mu}_1 = \boldsymbol{\mu}_0 + K(\boldsymbol{x}_1 - C\boldsymbol{\mu}_0)$;
$V_1 = (I - K_1 C) V_0$;
для $n = 2, \dots, N$
$\tilde{V}_{n-1} = A V_{n-1} A^T + \Gamma$;
$K_n = \tilde{V}_{n-1} C^T (C V_{n-1} C^T + \Sigma)^{-1}$;
$c_n = \mathcal{N}(\boldsymbol{x}_n | CA\boldsymbol{\mu}_{n-1}, C\tilde{V}_{n-1} C^T + \Sigma)$;
$\boldsymbol{\mu}_n = \boldsymbol{\mu}_{n-1} + K_n(\boldsymbol{x}_n - CA\boldsymbol{\mu}_{n-1})$;
$V_n = (I - K_n C)\tilde{V}_{n-1}$;

// Проход назад (РТС уравнения)
$\hat{\boldsymbol{\mu}}_N = \boldsymbol{\mu}_N$;
$\hat{V}_N = V_N$;
для $n = N-1, \dots, 1$
$J_n = V_n A^T \tilde{V}_n^{-1}$;
$\hat{\boldsymbol{\mu}}_n = \boldsymbol{\mu}_n + J_n(\hat{\boldsymbol{\mu}}_{n+1} - A\boldsymbol{\mu}_n)$;
$\hat{V}_n = V_n + J_n(\hat{V}_{n+1} - \tilde{V}_n) J_n^T$;

$L = \sum_{n=1}^N \log c_n$; // Вычисляем логарифм неполного правдоподобия

Это задача оптимизации неполного правдоподобия. Следовательно, здесь можно применить EM-алгоритм. Итерационная схема EM-алгоритма состоит из двух шагов:

E-шаг:
$$p(T|X, \Theta_{old}) = \frac{p(X, T|\Theta_{old})}{p(X|\Theta_{old})},$$

M-шаг:
$$\mathbb{E}_{T|X,\Theta_{old}} \log p(X, T|\Theta) \to \max_{\Theta}$$

Рассмотрим обучение параметров C, Σ. Запишем слагаемые $\mathbb{E} \log p(X, T|\Theta)$, которые зависят от C, Σ:

$$-\frac{1}{2}\mathbb{E}\left[\sum_{n=1}^N (\boldsymbol{x}_n - C\boldsymbol{t}_n)^T \Sigma^{-1} (\boldsymbol{x}_n - C\boldsymbol{t}_n)\right] - \frac{N}{2} \log\det\Sigma =$$
$$= -\frac{1}{2}\sum_{n=1}^N \left[\boldsymbol{x}_n^T \Sigma^{-1} \boldsymbol{x}_n - 2\mathbb{E}\boldsymbol{t}_n^T C^T \Sigma^{-1} \boldsymbol{x}_n + \mathrm{tr}(C^T \Sigma^{-1} C \mathbb{E}\left(\boldsymbol{t}_n \boldsymbol{t}_n^T\right))\right] - \frac{N}{2}\log\det\Sigma \to \max_{C,\Sigma}$$

Здесь $\mathbb{E}\boldsymbol{t}_n = \hat{\boldsymbol{\mu}}_n$, $\mathbb{E}\boldsymbol{t}_n\boldsymbol{t}_n^T = J_{n-1}\hat{V}_n + \hat{\boldsymbol{\mu}}_n\hat{\boldsymbol{\mu}}_n^T$. Вычисляя производные по C, Σ и приравнивая их к нулю, получаем следующие формулы пересчета:

$$C^{new} = \left(\sum_{n=1}^N \boldsymbol{x}_n \mathbb{E}\boldsymbol{t}_n^T\right)\left(\sum_{n=1}^N \mathbb{E}\boldsymbol{t}_n\boldsymbol{t}_n^T\right)^{-1},$$
$$\Sigma^{new} = \frac{1}{N}\sum_{n=1}^N \left[\boldsymbol{x}_n\boldsymbol{x}_n^T - C^{new}\mathbb{E}\boldsymbol{t}_n\boldsymbol{x}_n^T - \boldsymbol{x}_n\mathbb{E}\boldsymbol{t}_n^T (C^{new})^T + C^{new}\mathbb{E}\boldsymbol{t}_n\boldsymbol{t}_n^T (C^{new})^T\right].$$

Рассуждая аналогично для остальных параметров ЛДС, получаем следующие формулы пересчета:

$$\boldsymbol{\mu}_0^{new} = \mathbb{E}\boldsymbol{t}_1,$$
$$V_0^{new} = \mathbb{E}\boldsymbol{t}_1\boldsymbol{t}_1^T - \mathbb{E}\boldsymbol{t}_1\mathbb{E}\boldsymbol{t}_1^T,$$
$$A^{new} = \left(\sum_{n=2}^{N}\mathbb{E}\boldsymbol{t}_n\boldsymbol{t}_{n-1}^T\right)\left(\sum_{n=2}^{N}\mathbb{E}\boldsymbol{t}_{n-1}\boldsymbol{t}_{n-1}^T\right)^{-1},$$
$$\Gamma^{new} = \frac{1}{N-1}\sum_{n=2}^{N}\left[\mathbb{E}\boldsymbol{t}_n\boldsymbol{t}_n^T - A^{new}\mathbb{E}\boldsymbol{t}_{n-1}\boldsymbol{t}_n^T - \mathbb{E}\boldsymbol{t}_n\boldsymbol{t}_{n-1}^T(A^{new})^T + A^{new}\mathbb{E}\boldsymbol{t}_{n-1}\boldsymbol{t}_{n-1}^T(A^{new})^T\right].$$

6. Пример применения фильтра Калмана: задача стабилизации ориентации объекта.

Рассмотрим пример задачи стабилизации ориентации объекта в абсолютном пространстве. Такая задача часто встречается на практике, например, при стабилизации дула танка во время движения[1] или стабилизации направления взора видео-, кино- и тепловизорной камеры, установленной на подвижном основании[2].

В большинстве случаев для решения данной задачи на объект устанавливают датчики абсолютной угловой скорости (MEMS, оптоволоконные и др.), которые используются для оценки ориентации объекта. Непосредственно ориентация объекта может быть вычислена путем интегрирования угловых скоростей. Однако, эти датчики имеют существенный недостаток, связанный с тем, что в их показаниях присутствует случайная низкочастотная составляющая, называемая дрейфом. При интегрировании дрейф приводит к тому, что со временем показания датчика начинаются меняться даже при отсутствии движения. Показания датчика можно представить в виде:

$$\omega = \tilde{\omega} + \nu,$$

где ω — истинная проекция угловой скорости на ось чувствительности датчика, ν — составляющая дрейфа. Дрейф обычно достаточно хорошо моделируется броуновским движением:

$$\dot{\nu} = \varepsilon_\nu,$$

где ε_ν – нормальный шум. Дисперсия ε_ν является заводской характеристикой датчика и предполагается известной.

Для компенсации дрейфа на объект наряду с датчиками угловых скоростей устанавливают акселерометры. Показания акселерометра можно моделировать так:

$$\tilde{\boldsymbol{a}} = A\boldsymbol{g} + \boldsymbol{a} + \boldsymbol{\varepsilon}_a,$$

где $\boldsymbol{g}$ — ускорение свободного падения, $\boldsymbol{a}$ — истинное ускорение, $\boldsymbol{\varepsilon}_a$ – нормальный шум, дисперсия которого, как правило, полагается единичной. Матрица A задает ориентацию объекта в поле силы тяжести.

Возникает задача совмещения информации от акселерометров и датчиков угловых скоростей с целью компенсации дрейфа при определении ориентации объекта. Для решения данной задачи применяют фильтр Калмана.

В целях упрощения выкладок будем рассматривать задачу определения ориентации объекта в вертикальной плоскости. В этом случае ориентация задается одним углом φ. Движение объекта описывается одним кинематическим уравнением:

$$\dot{\varphi} = \omega.$$

Показания акселерометров имеют вид:

$$\begin{aligned}\tilde{a}_x &= g\sin\varphi + a_x + \varepsilon_x,\\ \tilde{a}_z &= -g\cos\varphi + a_z + \varepsilon_z,\end{aligned}$$

Полагая, что стабилизация происходит в окрестности горизонтального положения объекта, линеаризуем уравнения показаний акселерометров. Тогда вертикальная составляющая становится неинформативной, горизонтальная же имеет вид:

$$\tilde{a}_x = g\varphi + a_x + \varepsilon_x.$$

[1] http://www.youtube.com/watch?v=Lj75U7TkBBY
[2] http://www.robycam.ru

Проекция истинного ускорения a_x является неизвестной, и обычно ее моделируют случайным процессом, спектральные характеристики которого отражают ускорения при эксплуатации объекта. В большинстве случаев, для моделирования спектра хватает уравнения второго порядка:

$$k_1\ddot{a}_x + k_2\dot{a}_x + a_x = r_x,$$

где k_1, k_2 — коэффициенты, определяющие форму спектральной характеристики, r_x — нормальный шум, чья дисперсия определяет интенсивность ожидаемых значений ускорения.

Общая система может быть записана в матричном виде:

$$\begin{aligned}\dot{\xi} &= C\xi + Br,\\ \tilde{a} &= H^T\xi + \varepsilon_x,\end{aligned}$$

где

$$\xi = \begin{pmatrix}\varphi\\ \nu\\ a_x\\ \dot{a}_x\end{pmatrix},\qquad r = \begin{pmatrix}\tilde{\omega}\\ \varepsilon_\nu\\ r_x\end{pmatrix},\qquad C = \begin{pmatrix}0 & 1 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & -k_2/k_1 & -1/k_1\end{pmatrix},$$

$$B = \begin{pmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 0\\ 0 & 0 & 1\end{pmatrix},\qquad H = \begin{pmatrix}g\\ 0\\ 0\\ 1\end{pmatrix}.$$

Для данной системы может быть построен фильтр Калмана, доставляющий оценку φ. На этом принципе основаны алгоритмы оценок в ряде систем, в частности, в существующей системе для съемок на подвижном основании RobyCam[3].

7. Ограничения фильтра Калмана. Расширенный фильтр Калмана.

Фильтр Калмана выводится в предположениях нормальных линейных моделей (4). В реальной ситуации сигналы зачастую имеют нелинейную динамику и ненормальный шум. Рассмотрим, в каких ситуациях идеи фильтра Калмана могут быть обобщены на более сложные случаи.

Необходимым условием для существования точного алгоритма вывода в графической модели на рис. 2 является возможность аналитического вычисления интегралов в выражениях (7) и (8). Другим требованием к алгоритму является «неусложнение» модели для $\hat{\alpha}(\boldsymbol{t}_n)$ и $\hat{\beta}(\boldsymbol{t}_n)$ с ростом n. Пусть, например, модель сенсора $p(\boldsymbol{x}_n|\boldsymbol{t}_n)$ представляет собой смесь из K нормальных распределений, а остальные атомарные распределения в ЛДС являются нормальными линейными. Тогда $\hat{\alpha}(\boldsymbol{t}_1)$ является гауссианой, $\hat{\alpha}(\boldsymbol{t}_2)$ — смесью из K гауссиан, $\hat{\alpha}(\boldsymbol{t}_3)$ — смесью из K^2 гауссиан и т.д. Таким образом, здесь приходится иметь дело с экспоненциальным количеством слагаемых, что не позволяет реализовать метод на компьютере. В результате точный алгоритм фильтрации типа Калмана возможен только при наличии линейного перехода между мат.ожиданиями и моделью шума из т.н. экспоненциального семейства распределений.

Рассмотрим задачу нелинейной фильтрации с гауссовскими шумами:

$$\begin{aligned}\boldsymbol{t}_n &= \boldsymbol{f}(\boldsymbol{t}_{n-1}) + \boldsymbol{\varepsilon},\ \boldsymbol{\varepsilon} \sim \mathcal{N}(0,\Gamma),\\ \boldsymbol{x}_n &= \boldsymbol{g}(\boldsymbol{t}_n) + \boldsymbol{\nu},\ \boldsymbol{\nu} \sim \mathcal{N}(0,\Sigma).\end{aligned}$$

Здесь $\boldsymbol{f}$ и $\boldsymbol{g}$ – известные вектор-функции.

Для такой задачи можно построить приближенный алгоритм фильтрации в реальном времени. Пусть в момент времени $n-1$ найдено текущее распределение $p(\boldsymbol{t}_{n-1}|\boldsymbol{x}_1,\dots,\boldsymbol{x}_{n-1})$ вида $\mathcal{N}(\boldsymbol{t}_{n-1}|\boldsymbol{\mu}_{n-1},V_{n-1})$. Приблизим вектор-функцию $\boldsymbol{f}$ линейной функцией в окрестности точки $\boldsymbol{\mu}_{n-1}$:

$$\boldsymbol{t}_n \simeq \boldsymbol{f}(\boldsymbol{\mu}_{n-1}) + \nabla\boldsymbol{f}(\boldsymbol{\mu}_{n-1})\boldsymbol{t}_{n-1} + \boldsymbol{\varepsilon}.$$

Тогда мы можем осуществить прогноз для $\boldsymbol{t}_n$ по формулам фильтра Калмана (5), где в качестве матрицы A выступает матрица $\nabla\boldsymbol{f}(\boldsymbol{\mu}_{n-1})$:

$$p(\boldsymbol{t}_n|\boldsymbol{x}_1,\dots,\boldsymbol{x}_{n-1}) = \mathcal{N}(\boldsymbol{t}_n|\tilde{\boldsymbol{\mu}}_n,\tilde{V}_n).$$

Аналогично, приблизим вектор-функцию $\boldsymbol{g}$ линейной функцией в окрестности точки $\tilde{\boldsymbol{\mu}}_n$:

$$\boldsymbol{x}_n \simeq \boldsymbol{g}(\tilde{\boldsymbol{\mu}}_n) + \nabla\boldsymbol{g}(\tilde{\boldsymbol{\mu}}_n)\boldsymbol{t}_n + \boldsymbol{\nu}.$$

[3] http://www.robycam.ru/ru/video.html

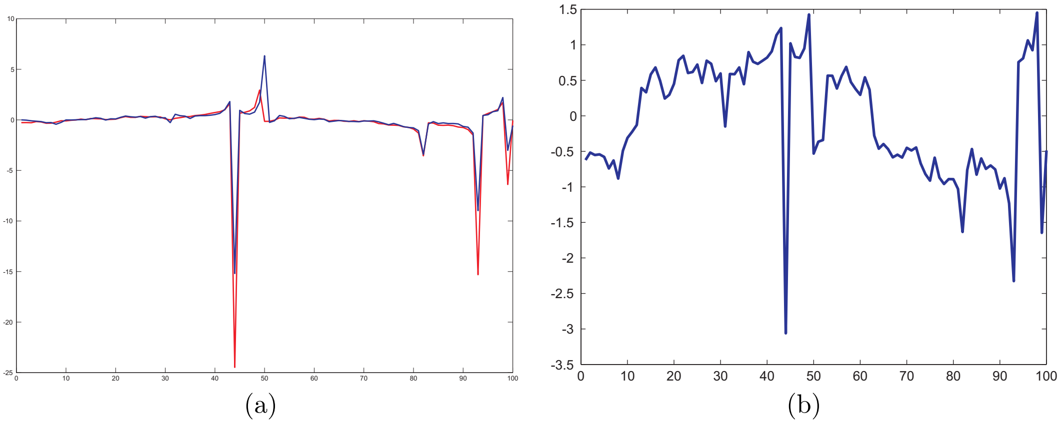


Рис. 4: Пример применения расширенного фильтра Калмана для нелинейной фильтрации сигналов. На рис. a показана истинная скрытая переменная (синяя кривая) и восстановленная скрытая переменная (красная кривая). На рис. b показан наблюдаемый сигнал.

Тогда мы можем провести коррекцию по формулам (6), где в качестве матрицы C выступает $\nabla \boldsymbol{g}(\tilde{\boldsymbol{\mu}}_n)$. Такой алгоритм фильтрации получил название расширенного фильтра Калмана. В том случае, если дисперсии шумов не слишком велики (т.е. линейная аппроксимация является адекватной), применение расширенного фильтра Калмана дает решение задачи с высокой точностью (см. пример на рис. 4).

В том случае, когда шумы не являются гауссовскими, расширенный фильтр Калмана применять нельзя. В этом случае обычно применяют фильтр частиц, в котором используются численные методы взятия интегралов на основе методов Монте Карло с марковскими цепями.