

Імовірнісні ітеративні методи оптимізації

- Нехай критерій оптимальності, виражений у вигляді функціонала вектора \mathbf{c} ,

$$J(\mathbf{c}) = M_x \{Q(\mathbf{x}, \mathbf{c})\} \quad (3.1)$$

- не відомий в явній формі. Тобто не відома щільність розподілу $p(x)$, а відомі лише $Q(\mathbf{x}, \mathbf{c})$, які залежать від стаціонарних випадкових процесів (або послідовностей) \mathbf{x} і вектору \mathbf{c} . Крім того, будемо вважати, що не відомо, детерміновані або випадкові процеси розглядаються, не відомі типи розподілів (або їхні параметри). Тобто відсутня достатня апріорна інформація.
- Задача потребує визначення оптимального вектору \mathbf{c}^* , який доставляє екстремум (мінімум) функціоналу (3.1), використовуючи для цього як базовий регулярний ітеративний метод та імовірнісний підхід.

Імовірнісні ітеративні методи оптимізації (продовження)

- Тоді умову оптимальності (2.2), з урахуванням (3.1) можна записати як

$$\nabla J(\mathbf{c}) = M_x \{ \nabla_{\mathbf{c}} Q(\mathbf{x}, \mathbf{c}) \} = 0, \quad (3.2)$$

- де

$$\nabla_{\mathbf{c}} Q(\mathbf{x}, \mathbf{c}) = \frac{dQ(\mathbf{x}, \mathbf{c})}{dc_1}, \dots, \frac{dQ(\mathbf{x}, \mathbf{c})}{dc_N} \quad (3.3)$$

- представляє собою градієнт $Q(\mathbf{x}, \mathbf{c})$ по \mathbf{c}

- Тоді імовірнісний алгоритм оптимізації можна представити як

$$\mathbf{c}[n] = \mathbf{c}[n-1] - \Gamma[n] \nabla_{\mathbf{c}} Q(\mathbf{x}[n], \mathbf{c}[n-1]) \quad (3.4)$$

- У виразах (3.4), (3.5), (3.6) можна побачити аналогію з (2.4), (2.7), (2.8). Але вони істотно відрізняються, тому що тепер при $\mathbf{c} = \mathbf{c}^*$

$$\nabla_{\mathbf{c}} Q(\mathbf{x}, \mathbf{c}^*) \neq 0$$

Імовірнісні ітеративні методи оптимізації (продовження)

- Через це змінюються умови, які накладають на $\Gamma[n]$, щоб забезпечити збіжність. У найпростішому випадку обирають $\Gamma[n]$ у вигляді діагональної матриці

-
- $$\Gamma[n] = A\gamma[n] = \begin{pmatrix} \gamma_1[n] & 0 & \boxtimes & 0 \\ 0 & \gamma_2[n] & \boxtimes & 0 \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \gamma_N[n] \end{pmatrix} \quad (3.8)$$

- де A - оператор, що перетворює вектор $\Gamma[n]$ в діагональну матрицю.
- Випадок рівних компонент дозволяє, $A\gamma[n] = I\gamma[n]$ де дія оператора A відповідає множенню на одиничну матрицю.

Імовірнісні ітеративні методи оптимізації (продовження)

- У цьому випадку алгоритм (3.4) можна записати як
- $$\mathbf{c}[n] = \mathbf{c}[n-1] - \gamma[n] \nabla_{\mathbf{c}} Q(x[n], \mathbf{c}[n-1]) \quad (3.9)$$

- а пошуковий алгоритм у рекурентній формі, з урахуванням

- $$\left. \begin{aligned} Q_+(x, \mathbf{c}, a) &= (Q(x, \mathbf{c} + ae_1), \dots, Q(x, \mathbf{c} + ae_N)), \\ Q_-(x, \mathbf{c}, a) &= (Q(x, \mathbf{c} - ae_1), \dots, Q(x, \mathbf{c} - ae_N)), \\ Q_0(x, \mathbf{c}) &= (Q(x, \mathbf{c}), \dots, Q(x, \mathbf{c})), \end{aligned} \right\} \quad (3.12)$$

- де a - скаляр, e_i ($i = 1, 2, \dots, N$) - базисні вектори, градієнт оцінений приблизно, за допомогою розділених різниць

- $$\frac{Q_+(x, \mathbf{c}, a) - Q_-(x, \mathbf{c}, a)}{2a} = \tilde{\nabla}_{\mathbf{c}^+} Q(x, \mathbf{c}, a) \quad (3.13)$$

- буде

- $$\mathbf{c}[n] = \mathbf{c}[n-1] - \gamma[n] \tilde{\nabla}_{\mathbf{c}^+} Q(x[n], \mathbf{c}[n-1], a[n]) \quad (3.15)$$

Ітеративні методи оптимізації.

Про правило зупинки

- При практичному використанні алгоритмів виникає питання, щодо кількості кроків n_0 , завдяки якій можна вважати, що з достатнім ступенем точності отримано вектор \mathbf{c}^* .
- Для визначення цієї кількості кроків можна використовувати сковзне середнє

$$m_N[n] = \frac{1}{N} \sum_{k=n}^{n+N} \mathbf{c}[k] \quad (n = 0, 1, 2, \dots) \quad (3.38)$$

- Якщо, починаючи з якого-небудь номера k_0 для всіх $k \geq k_0$
- $\|m_N[kN] - m_N[(k+1)N]\| < \varepsilon$, (3.39)
- (де $\varepsilon > 0$ достатньо мала величина), то величина $n_0 = k_0 N$ визначає той момент часу, коли можна вважати, що

- $M\{\mathbf{c}[n_0]\} \approx \mathbf{c}^*$ (3.40)

Розпізнавання. Загальні положення

- Проблема розпізнавання є загальною. В системному аналізі вона виникає в зв'язку з необхідністю автоматизації процедур розпізнавання фігур (цифр, букв, простих зображень), звуків (голосів, шуму), діагностики захворювань або несправностей і таке інше.
- Розпізнавання є важливим ступенем обробки інформації, яку отримує людина (завдяки органам чуттів) чи система (завдяки датчикам та приладам).
- Спочатку розпізнають **предмети**, після цього – співвідношення між ними, тобто **ситуації**. Далі – мають бути розпізнаними зміни ситуацій, тобто **явища**.
- Саме це дає можливість визначити закономірності і **прогнозувати подальший хід явищ та їх розвиток**.

Розпізнавання.

Загальні положення (продовження)

- Задача розпізнавання складається з віднесення об'єкту до одного з класів, найчастіше спочатку ці класи невідомі. Об'єкти, які належать одному з класів певною мірою схожі. Те загальне, що об'єднує об'єкти в клас називають **образом**.
- Для вирішення задачі розпізнавання необхідно провести **навчання** шляхом показу образів, приналежність яких до певного класу відома.
- Геометрично така задача складається з **двох етапів**.
- **Перший етап** - побудова поверхні (функції, яка розділює), котра в якому-небудь змісті краще за всі інші розділює простір на області, які відповідають певним класам. Ця побудова (її звать навчанням) проводиться на основі показу певної кількості об'єктів (образів), які належать цим класам.
- **Другий етап** – складається з іспиту нового об'єкту, для котрого спочатку невідомо, до якого класу він належить, та визначення цього класу.

Розпізнавання.

Загальні положення (продовження)

- Позначимо функцію, яка розділює, $y = f(x)$, (4.1)
- де \mathbf{x} - 1-мірний вектор, який характеризує образ, а y - визначає клас, до якого цей образ належить.
- Ця функція має володіти властивістю
- $$\text{sign } f(\mathbf{x}) = \begin{cases} 1, & \mathbf{x} \in A, \\ -1, & \mathbf{x} \notin B, \end{cases} \quad (4.2)$$
- тобто знак $f(x)$ визначає приналежність \mathbf{x} до класу A або B.
- Як можна зрозуміти з формули (4.2), для класів, які легко розрізнити, існує багато функцій, які визначають цю поверхню розділу.
- Якщо це не так, зазвичай існує найкраща така функція.

Розпізнавання.

Загальні положення (продовження)

- Позначимо клас функцій, які приблизно такі, як треба для розділення класів через $\hat{f}(\mathbf{x}, \mathbf{c})$, де \mathbf{c} - вектор коефіцієнтів цієї функції розділу, які ще мають бути визначені, а міру відхилення визначимо як деяку опуклу функцію від $y = f(x)$
- та від $\hat{f}(\mathbf{x}, \mathbf{c})$, наприклад $F(y, \hat{f}(\mathbf{x}, \mathbf{c}))$.
- Оскільки \mathbf{x} відображає випадкову складову, це має бути відображене та враховане при виборі форми функціоналу, який представляє собою математичне очікування міри відхилення:

$$J(\mathbf{c}) = M\{F(y, \hat{f}(\mathbf{x}, \mathbf{c}))\}$$

(4.3)

- Найкраще приближення функції, яка розділює класи до $\mathbf{c} = \mathbf{c}^*$ ідеальної буде відповідати такому вибору вектора \mathbf{c} при котрому $J(\mathbf{c})$ досягає мінімуму.
 - В більшості випадків обирають
- $$J(\mathbf{c}) = M\{F(y - \hat{f}(\mathbf{x}, \mathbf{c}))\} \quad (4.4)$$