Filtragem Adaptativa

Charles Casimiro Cavalcante

charles@gtel.ufc.br

Grupo de Pesquisa em Telecomunicações Sem Fio – GTEL
Programa de Pós-Graduação em Engenharia de Teleinformática
Universidade Federal do Ceará – UFC
http://www.gtel.ufc.br/∼charles

"Filtros adaptativos, os quais têm como meta transformar os sinais portadores de informação em versões 'limpas' ou 'melhoradas', ajustam suas características de acordo com os sinais encontrados. Eles formam os exemplos mais simples de algoritmos no campo de aprendizado de máquinas."

Philip A. Regalia, 2005

IEEE Control Systems Magazine, Agosto de 2005

Conteúdo do curso

1. Introdução
2. Revisão de Processos Estocásticos
3. Filtragem Linear Ótima
4. Algoritmos Recursivos no Tempo
5. Método dos Mínimos Quadrados
6. Estruturas Alternativas de Filtragem Adaptativa
7. Tópicos Avançados

Parte VI

Método dos Mínimos Quadrados

Algoritmo RLS

- O algoritmo *Recursive Least Squares* (mínimos quadrados recursivos) é uma implementação recursiva da solução ótima
- **Idéia**: otimizar o filtro para os erros quadrados até o instante atual
- Melhor adequação para sistemas que apresentam rápidas variações

Formalismo

Dadas n amostras de $x(n)$, $d(n)$, $y(n)$ e $e(n)$, busca-se $\mathbf{w}(n)$ que minimiza

$$\sum_{i=0}^{n} \lambda^{n-i} \cdot e^2(i) \tag{139}$$

em que $0 \ll \lambda \leq 1$ é o fator de esquecimento (ponderação)

Solução:

$$\left.\begin{aligned} \mathbf{w}(n) &= \mathbf{R}_D^{-1}(n) \cdot \mathbf{p}_D(n) \\ \mathbf{R}_D(n) &= \sum_{i=0}^{n} \lambda^{n-i} \mathbf{x}(i)\mathbf{x}^T(i) \\ \mathbf{p}_D(n) &= \sum_{i=0}^{n} \lambda^{n-i} \mathbf{x}(i)d(i) \end{aligned}\right\} \text{ Equações normais} \tag{140}$$

Algoritmo

$$\begin{aligned}\mathbf{w}(n) &= \mathbf{R}_D^{-1}(n) \cdot \mathbf{p}_D(n) \\ \mathbf{w}(n+1) &= \mathbf{R}_D^{-1}(n+1) \cdot \mathbf{p}_D(n+1)\end{aligned} \tag{141}$$

Podemos então escrever

$$\mathbf{w}(n+1) = \mathbf{R}_D^{-1}(n+1)\left[\lambda \mathbf{p}_D(n) + d(n+1)\mathbf{x}(n+1)\right] \tag{142}$$

Mas

$$\begin{aligned}\mathbf{p}_D(n) &= \mathbf{R}_D(n) \cdot \mathbf{w}(n) \\ &= \frac{1}{\lambda}\left[\mathbf{R}_D(n+1) - \mathbf{x}(n+1)\mathbf{x}^T(n+1)\right]\mathbf{w}(n)\end{aligned} \tag{143}$$

Algoritmo - cont.

Assim, temos

$$\mathbf{w}(n+1) = \mathbf{R}_D^{-1}(n+1)\left[\mathbf{R}_D(n+1)\mathbf{w}(n) - \mathbf{x}(n+1)\mathbf{x}^T(n+1)\mathbf{w}(n) + d(n+1)\mathbf{x}(n)\right] \tag{144}$$

Logo, temos

$$\boxed{\mathbf{w}(n+1) = \mathbf{w}(n) + \mathbf{R}_D^{-1}(n+1)\mathbf{x}(n+1)e(n+1)} \tag{145}$$

em que

$$e(n+1) = d(n+1) - \mathbf{w}^T(n)\mathbf{x}(n+1) \quad \text{(erro } \textit{a priori}\text{)}$$

$$\mathbf{R}_D(n+1) = \lambda\mathbf{R}_D(n) + \mathbf{x}(n+1)\mathbf{x}^T(n+1)$$

Algoritmo - cont.

Problema: inversão de matriz é uma operação custosa.

Mas temos um resultado importante

Lema da inversão de matrizes

Seja $\mathbf{A} = \mathbf{B} + \mathbf{C}\mathbf{D}\mathbf{C}^T$

$$\mathbf{A}^{-1} = \mathbf{B}^{-1} - \mathbf{B}^{-1}\mathbf{C}\left[\mathbf{C}^T\mathbf{B}^{-1}\mathbf{C} + \mathbf{D}^{-1}\right]^{-1}\mathbf{C}^T\mathbf{B}^{-1} \quad (146)$$

No nosso caso

$$\mathbf{A} = \mathbf{R}_D(n+1); \quad \mathbf{B} = \lambda\mathbf{R}_D(n);$$
$$\mathbf{C} = \mathbf{x}(n+1); \quad \mathbf{D} = \mathbf{I}$$

Algoritmo - cont.

Assim, temos

$$\mathbf{R}_D^{-1}(n+1) = \frac{1}{\lambda}\left[\mathbf{R}_D^{-1}(n) - \frac{\mathbf{R}_D^{-1}(n)\mathbf{x}(n+1)\mathbf{x}^T(n+1)\mathbf{R}_D^{-1}(n)}{\lambda + \mathbf{x}^T(n+1)\mathbf{R}_D^{-1}(n)\mathbf{x}(n+1)}\right] \tag{147}$$

e definindo-se o ganho de adaptação:

$$\mathbf{g}(n) = \mathbf{R}_D^{-1}(n)\mathbf{x}(n) \tag{148}$$

Algoritmo - cont.

Assim temos as seguintes equações

$$\mathbf{g}(n+1) = \frac{\mathbf{R}_D^{-1}(n)\mathbf{x}(n+1)}{\lambda + \mathbf{x}^T(n+1)\mathbf{R}_D^{-1}(n)\mathbf{x}(n+1)} \tag{149}$$

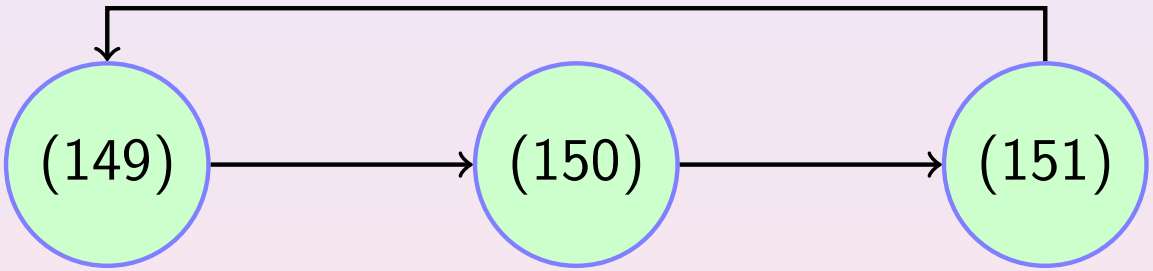
$$\mathbf{R}_D^{-1}(n+1) = \frac{1}{\lambda}\left[\mathbf{R}_D^{-1}(n) - \mathbf{g}(n+1)\mathbf{x}^T(n+1)\mathbf{R}_D^{-1}(n)\right] \tag{150}$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mathbf{g}(n+1)e(n) \tag{151}$$

Condições iniciais: $\mathbf{R}_D^{-1}(0)$ e λ, outros parâmetros nulos, recebe-se $x(n+1)$ e $d(n+1)$

Algoritmo RLS - cont.

Então, o algoritmo RLS é dado pela sequência cíclica das seguintes equações:



Algoritmo RLS - cont.

Algoritmo RLS - resumo

$$\mathbf{g}(n+1) = \frac{\mathbf{R}_D^{-1}(n)\mathbf{x}(n+1)}{\lambda + \mathbf{x}^T(n+1)\mathbf{R}_D^{-1}(n)\mathbf{x}(n+1)}$$

$$\mathbf{R}_D^{-1}(n+1) = \frac{1}{\lambda}\left[\mathbf{R}_D^{-1}(n) - \mathbf{g}(n+1)\mathbf{x}^T(n+1)\mathbf{R}_D^{-1}(n)\right]$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mathbf{g}(n+1)e(n)$$

Comparação RLS *versus* LMS

	LMS	**RLS**
complexidade	$\sim M$	$\sim M^2$
velocidade de convergência	depende de $\mathbf{R_x}$	não depende das estatísticas
desajuste	$\sim \mu$	pode convergir para zero quando $\mathbf{g}(n) \to 0$
estabilidade	estável para μ apropriado	possível estabilidade