

Cuantización vectorial con aprendizaje

Diego Milone
Inteligencia Computacional
Departamento de Informática

FICH-UNL

Organización: LVQ

- ¿Qué es un cuantizador vectorial?
- ¿Cómo obtenemos el diccionario de cuantización?
- Algoritmo LVQ1
- Problemas con la velocidad de aprendizaje
- Algoritmo LVQ1 optimizado (LVQ1-O)

Cuantización vectorial con aprendizaje (LVQ)

Conceptos básicos:

- Cuantizador escalar: señales cuantizadas

Cuantización vectorial con aprendizaje (LVQ)

Conceptos básicos:

- Cuantizador escalar: señales cuantizadas
- Cuantizador vectorial:
 - centroides o prototipos
 - diccionario o code-book
 - el proceso de cuantización: de vectores a números enteros

Cuantización vectorial con aprendizaje (LVQ)

Conceptos básicos:

- Cuantizador escalar: señales cuantizadas
- Cuantizador vectorial:
 - centroides o prototipos
 - diccionario o code-book
 - el proceso de cuantización: de vectores a números enteros
- Ideas de como entrenarlo:
 - algoritmo k -means etiquetado para clasificación
 - SOM etiquetado como clasificador...
 - algoritmo supervisado LVQ

Algoritmo LVQ1

1. Inicialización aleatoria

Algoritmo LVQ1

1. Inicialización aleatoria
2. Selección:

$$c(n) = \arg \min_i \{ \|\mathbf{x}(n) - \mathbf{m}_i(n)\| \}$$

Algoritmo LVQ1

1. Inicialización aleatoria
2. Selección:

$$c(n) = \arg \min_i \{ \|\mathbf{x}(n) - \mathbf{m}_i(n)\| \}$$

3. Adaptación:

$$\mathbf{m}_c(n+1) = \mathbf{m}_c(n) + s(c, d, n) \alpha [\mathbf{x}(n) - \mathbf{m}_c(n)]$$

$$s(c, d, n) = \begin{cases} +1 & \text{si } c(n) = d(n) \\ -1 & \text{si } c(n) \neq d(n) \end{cases}$$

Algoritmo LVQ1

1. Inicialización aleatoria
2. Selección:

$$c(n) = \arg \min_i \{ \|\mathbf{x}(n) - \mathbf{m}_i(n)\| \}$$

3. Adaptación:

$$\mathbf{m}_c(n+1) = \mathbf{m}_c(n) + s(c, d, n) \alpha [\mathbf{x}(n) - \mathbf{m}_c(n)]$$

$$s(c, d, n) = \begin{cases} +1 & \text{si } c(n) = d(n) \\ -1 & \text{si } c(n) \neq d(n) \end{cases}$$

4. Volver a 2 hasta satisfacer error de clasificación

LVQ1: observaciones

- Interpretación gráfica
 - Caso de clasificación correcta
 - Caso de clasificación incorrecta

LVQ1: observaciones

- Interpretación gráfica
 - Caso de clasificación correcta
 - Caso de clasificación incorrecta
- No hay arquitectura neuronal

LVQ1: observaciones

- Interpretación gráfica
 - Caso de clasificación correcta
 - Caso de clasificación incorrecta
- No hay arquitectura neuronal
- Se puede ver al cuantizador como SOM lineal, sin entorno y supervisado

LVQ1: observaciones

- Interpretación gráfica
 - Caso de clasificación correcta
 - Caso de clasificación incorrecta
- No hay arquitectura neuronal
- Se puede ver al cuantizador como SOM lineal, sin entorno y supervisado
- Velocidad de aprendizaje
 - ¿Existe un α_c óptimo para cada centroide?
 - ¿Se debe considerar un $\alpha_c(n)$ óptimo para cada instante de tiempo?

LVQ1: velocidad de aprendizaje

$$\mathbf{m}_c(n+1) = \mathbf{m}_c(n) + s(n)\alpha(n) [\mathbf{x}(n) - \mathbf{m}_c(n)]$$

LVQ1: velocidad de aprendizaje

$$\mathbf{m}_c(n+1) = \mathbf{m}_c(n) + s(n)\alpha(n) [\mathbf{x}(n) - \mathbf{m}_c(n)]$$

$$\mathbf{m}_c(n+1) = \mathbf{m}_c(n) + s(n)\alpha(n)\mathbf{x}(n) - s(n)\alpha(n)\mathbf{m}_c(n)$$

LVQ1: velocidad de aprendizaje

$$\mathbf{m}_c(n+1) = \mathbf{m}_c(n) + s(n)\alpha(n) [\mathbf{x}(n) - \mathbf{m}_c(n)]$$

$$\mathbf{m}_c(n+1) = \mathbf{m}_c(n) + s(n)\alpha(n)\mathbf{x}(n) - s(n)\alpha(n)\mathbf{m}_c(n)$$

$$\mathbf{m}_c(n+1) = [1 - s(n)\alpha(n)] \mathbf{m}_c(n) + s(n)\alpha(n)\mathbf{x}(n)$$

LVQ1: velocidad de aprendizaje

$$\mathbf{m}_c(n+1) = \mathbf{m}_c(n) + s(n)\alpha(n) [\mathbf{x}(n) - \mathbf{m}_c(n)]$$

$$\mathbf{m}_c(n+1) = \mathbf{m}_c(n) + s(n)\alpha(n)\mathbf{x}(n) - s(n)\alpha(n)\mathbf{m}_c(n)$$

$$\mathbf{m}_c(n+1) = [1 - s(n)\alpha(n)] \mathbf{m}_c(n) + s(n)\alpha(n)\mathbf{x}(n)$$

$$\mathbf{m}_c(n+1) =$$

$$= [1 - s(n)\alpha(n)]$$

$$\{\mathbf{m}_c(n-1) + s(n-1)\alpha(n-1) [\mathbf{x}(n-1) - \mathbf{m}_c(n-1)]\}$$

$$+ s(n)\alpha(n)\mathbf{x}(n)$$

$\mathbf{x}(n-1)$ es afectado dos veces por α

LVQ1: velocidad de aprendizaje

Siendo $\alpha < 1$ la importancia relativa de los primeros patrones de entrenamiento siempre será menor que la de los últimos.

LVQ1: velocidad de aprendizaje

Siendo $\alpha < 1$ la importancia relativa de los primeros patrones de entrenamiento siempre será menor que la de los últimos.

Si queremos que α afecte por igual a todos los patrones deberemos hacerlo decrecer con el tiempo.

LVQ1: velocidad de aprendizaje

Siendo $\alpha < 1$ la importancia relativa de los primeros patrones de entrenamiento siempre será menor que la de los últimos.

Si queremos que α afecte por igual a todos los patrones deberemos hacerlo decrecer con el tiempo.

Se debe cumplir que:

$$\alpha_c(n) = [1 - s(n)\alpha_c(n)] \alpha_c(n - 1)$$

LVQ1-O: velocidad de aprendizaje

Demostrar que:

$$\alpha_c(n) = \frac{\alpha_c(n-1)}{1 + s(n)\alpha_c(n-1)}$$

(cuidado, no sobrepasar $\alpha > 1$)