

Accelerometer Based Gesture Recognition using HMMs

Andrea Tarocchi, Marco Magnatti

8 gennaio 2009



Outline

- 1 Introduzione
- 2 Quantizzazione dei dati
- 3 HMM
 - Definizione
 - Riconoscimento
 - Apprendimento
 - Decodifica
- 4 Decisore
- 5 Test e risultati
- 6 Bibliografia



Descrizione del problema

Da un device dotato di accelerometri si ricevono i dati relativi alle accelerazioni a cui è sottoposto durante il moto (lungo le tre coordinate spaziali).

Tale device può essere impugnato per eseguire gesture.

Il nostro obiettivo è realizzare un sistema che, dopo adeguato addestramento, sia in grado di riconoscere gesti effettuati tramite tale dispositivo.



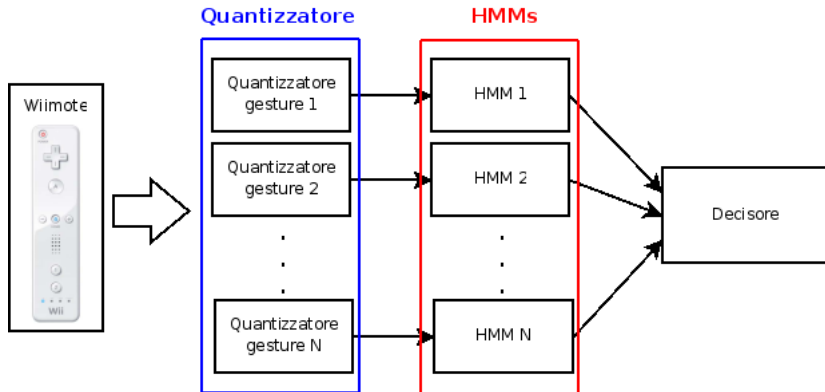
Strumenti utilizzati



- Device di acquisizione dati: Nintendo Wiimote
- Linguaggio di programmazione utilizzato: C/C++
- wiiuse: libreria C per comunicazione PC - Wiimote
- boost C++: librerie C++ con numerose strutture dati
- HMM (Hidden Markov Model): per il riconoscimento



Struttura generale



Quantizzazione con k-means

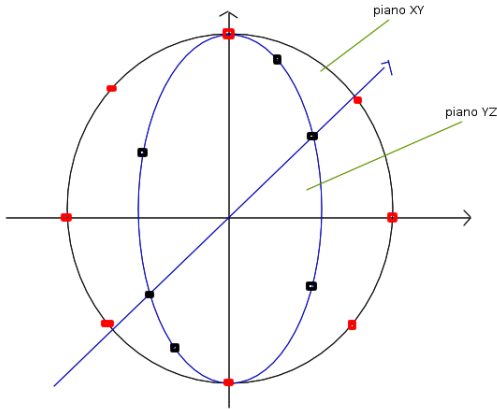
Poiché i dati in arrivo dal Wiimote sono valori in campo reale compresi tra $-4G$ e $+4G$, è necessaria una discretizzazione prima di poterli sottoporre ad un HMM che, per scelta implementativa, lavora in campo discreto.

La discretizzazione viene fatta tramite l'algoritmo k-means.

Per via sperimentale, si è stabilito la posizione iniziale e il numero dei centroidi da utilizzare (e di conseguenza il numero di differenti simboli che l'HMM può emettere).



Posizione dei centroidi



Possibili approcci alternativi:

- Inizializzazione random
- Inizializzazione lungo la direzione del vettore medio



Con HMM è possibile affrontare problemi del tipo “modellare la probabilità $P(X)$ di una sequenza di osservazioni X ”.
Nel nostro caso, le sequenze osservate sono costituite dai vettori di accelerazioni quantizzati in un numero finito di simboli.

Si assume che la sequenza osservata sia generata da una sequenza S di variabili nascoste, chiamate stati.

Altra assunzione fondamentale è che “il futuro è indipendente dal passato, dato il presente”, ovvero vale:

$$S[t+1] \perp S[1], S[2], \dots, S[t-1], X[1], X[2], \dots, X[t] | S[t]$$

$$X[t+1] \perp S[1], S[2], \dots, S[t], X[1], X[2], \dots, X[t] | S[t+1]$$



blah



blah



La procedura Concave-Convex(CCCP)

Il problema (3) non è $\frac{1}{2}$ convesso. CCCP è $\frac{1}{2}$ una procedura per ottimizzare problemi non convessi. Assumiamo che una funzione di costo $J(\Theta)$ possa essere riscritta come la somma di una parte convessa $J_{vex}(\Theta)$ e di una parte concava $J_{cav}(\Theta)$. Ogni iterazione della procedura CCCP approssima la parte concava con la sua tangente e minimizza la funzione convessa risultante.

$$\Theta^{t+1} = \arg \min_{\Theta} (J_{vex}(\Theta) + J'_{cav}(\Theta^t) \cdot \Theta). \quad (1)$$



CCCP-2

Algoritmo Iterativo

Si può $\frac{1}{2}$ dimostrare che:

- $J(\Theta^t)$ decresce ad ogni iterazione;
- la procedura converge ad un minimo locale.



CCCP per TSVMs

La Ramp Loss può essere riscritta come differenza tra due Hinge Loss

$$R_s(z) = H_1(z) - H_s(z) \quad (2)$$

É quindi possibile applicare la procedura CCCP al problema di minimizzazione TSVM (3). Il costo $J^s(\Theta)$ può essere decomposto in una parte convessa $J_{vex}^s(\Theta)$ e in una parte concava $J_{cav}^s(\Theta)$



CCCP per TSVMs-2

$$\begin{aligned}
 J^s(\Theta) &= \frac{1}{2} \|w\|^2 + C \sum_{i=1}^L H_1(yif_{\Theta}(\mathbf{x}_i)) + C^* \sum_{i=L+1}^{L+2U} R_s(yif_{\Theta}(\mathbf{x}_i)) = \\
 &= \underbrace{\frac{1}{2} \|w\|^2 + C \sum_{i=1}^L H_1(yif_{\Theta}(\mathbf{x}_i)) + C^* \sum_{i=L+1}^{L+2U} H_1(yif_{\Theta}(\mathbf{x}_i))}_{J_{vex}^s(\Theta)} - \\
 &\quad \underbrace{C^* \sum_{i=L+1}^{L+2U} H_s(yif_{\Theta}(\mathbf{x}_i))}_{J_{cav}^s(\Theta)}.
 \end{aligned}$$

(3)



CCCP per TSVMs-3

La decomposizione precedente permette di applicare la procedura CCCP. Il problema convesso alla base della procedura viene espresso utilizzando le variabili duali α . Dobbiamo anche implementare il vincolo di bilanciamento (4). Per semplificare l'approssimazione del primo ordine della parte concava, denotiamo:

$$\beta_i = \frac{\partial J_{cav}^s(\Theta)}{\partial f_{\Theta}(x_i)} = \begin{cases} C^* & \text{if } y_i f_{\Theta}(\mathbf{x}_i) \\ 0 & \text{altrimenti} \end{cases}, \quad (4)$$

per gli esempi non etichettati ($\beta_i = 0$ per $i \leq L$).



Introduzione
Quantizzazione dei dati
HMM
Decisore
Test e risultati
Bibliografia

Definizione
Riconoscimento
Apprendimento
Decodifica

CCCP per TSVMs-4



Complessità

Addestrare con la procedura CCCP-TSVM richiede la risoluzione di una serie di problemi di ottimizzazione SVM con $(L + 2U)$ variabili. Sebbene SVM abbia complessità $\mathcal{O}(L + 2U)^{\frac{1}{2}}$ nel caso peggiore, nel caso tipico scala quadraticamente e lo stesso vale per i sottoproblemi di TSVMs. Assumendo un numero costante di iterazioni TSVMs con CCCP dovrebbero scalare quadraticamente nella maggior parte dei casi.



Esperimenti eseguiti

Si sono condotti esperimenti sul database MNIST trattato come un problema a 10-classi. Il dataset ha 42k esempi di training, 18k di validation e 10k di test. Il training set \tilde{A} stato campionato per ottenere il punti etichettati ed $\tilde{A}_{\frac{1}{2}}$ stato usato il training set per gli esempi non etichettati. Abbiamo replicato il set-up del paper ovvero:

- $C = 10$ e $\gamma = 0.0128$ con kernel RBF;
- C^* variabile con il numero di esempi non etichettati e $s = 0.01$.

L'errore $\tilde{A}_{\frac{1}{2}}$ stato valutato sul validation test.







Commento dei risultati

Dal grafico dell'errore si nota che l'**aggiunta di dati non etichettati porta ad un significativo miglioramento delle prestazioni**. A meno di uno scarto medio dell' 1,71% l'andamento dell'errore degli esperimenti effettuati $\dot{\approx} \frac{1}{2}$ lo stesso di quelli riportati nel paper. Dal grafico dei tempi di vede chiaramente che il tempo di ottimizzazione ha un andamento **quadratico** con il numero dei dati non etichettati confermando le ipotesi fatte in precedenza.



SSL Vs Transductive Learning

Se si considera l'apprendimento trasduttivo come una versione di SSL dove i dati non etichettati sono il test set di interesse, $\tilde{\mathcal{D}}_{\frac{1}{2}}$ possibile comparare le prestazioni dei due approcci. L'esperimento prevede l'utilizzo di dati non etichettati diversi dal test set per l'SSL. Nello specifico abbiamo utilizzato 10000 punti non etichettati, un test set di 10000 punti e un training set di 1000 punti. I risultati ottenuti:

- Testing error SSL: 6,944%;
- Testing error Transductive: 6,350%.



Considerazioni

- Si è verificato che l'utilizzo di dati non etichettati migliora le prestazioni di un classificatore induttivo.
- Comunque il trade-off tra costi computazionali e benefici non è soddisfacente.
- Dal momento che l'algoritmo ottimizza problemi SVMs standard, ogni miglioramento delle tecniche di risoluzione di questi si riflette immediatamente sulle TSVMs.
- Interessante l'approccio trasduttivo in particolari situazioni.



Bibliografia



Collobert R. and Sinz F. and et al.
Large Scale Transductive SVMs.



Schlkopf, Bernhard and Smola, Alexander J.
Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond (Adaptive Computation and Machine Learning).
The MIT Press, 2001.



Chapelle, O. and Schölkopf, B. and Zien, A.
Semi-Supervised Learning (Adaptive Computation and Machine Learning).
The MIT Press, 2006.



Bibliografia



Yuille A. L. and Rangarajan A.
The concave-convex procedure (CCCP).
The MIT Press, 2002.



Vapnik, Vladimir N.
The Nature of Statistical Learning Theory.
Springer, 1999.



N. Cristianini and J. Shawe-Taylor.
Kernel Methods for Pattern Analysis.
Cambridge Univ. Press, 2004.

