

Clustering e co-clustering

Macchine a Supporto Vettoriale e Informazione di Bregman

Vincenzo Russo (vincenzo.russo@neminis.org)

Università degli Studi di Napoli Federico II

5 maggio 2007



Contenuti

Sommario

Clustering, co-clustering, SVM e Bregman Framework

Co-clustering

Introduzione

Bregman framework

Vantaggi

Macchine a Supporto Vettoriale

Introduzione

Approccio

Parametri ed esempi

Vantaggi e svantaggi

Strumenti per il Co-clustering

Dhillon software

Strumenti per il Support Vector Clustering

SVM classiche

Least Squares SVM

Obiettivi generali

1. L'obiettivo generale del lavoro di tesi che si sta portando avanti è quello di avere una soluzione allo stato dell'arte nell'ambito del *data clustering*.
2. Il nostro lavoro si è concentrato sinora in due direzioni
 - 2.1 Co-clustering con divergenze di Bregman
 - 2.2 Approccio con *Macchine a Supporto Vettoriale* (nel seguito anche SVM, da *Support Vector Machine*) al clustering convezionale
3. Si auspica la creazione di un caso particolare di co-clustering con divergenze di Bregman, utilizzando le SVM e la distanza euclidea.

Parte I

Co-clustering e Informazione di Bregman

Il co-clustering

1. Data una matrice Z multidimensionale, il co-clustering è il clustering **simultaneo** lungo tutte le dimensioni
2. Nel caso bidimensionale, data una matrice $Z \in S^{m \times n}$, si tratta del clustering simultaneo di righe e colonne.
3. In genere, le righe della matrice rappresentano gli oggetti¹ di cui effettuare il clustering, mentre le colonne sono gli attributi (o *feature*) che rappresentano gli oggetti
 - 3.1 L'idea generale del co-clustering è produrre gruppi di attributi contestualmente al clustering dei dati stessi

¹Documenti di testo nelle applicazioni di text mining, microarray in bioinformatica, etc.

Il co-clustering - Vantaggi

1. Migliori prestazioni in caso di dati sparsi
2. Riduzione della dimensione dei dati in input
3. Permette di osservare l'interazione degli oggetti con i relativi cluster di attributi.

I primi due punti rendono il co-clustering un'ottima scelta anche nel caso si fosse interessati a un clustering convenzionale su una sola dimensione.

Il framework di Bregman

1. L'approccio al co-clustering seguito è quello che si basa sul Bregman framework [3, 2], il quale si fonda sul principio della *minima informazione di Bregman*.
2. Al contrario di altri lavori², il Bregman framework permette di
 - 2.1 astrarre dalla particolare interpretazione della matrice
 - 2.2 astrarre dalla particolare funzione di distorsione che misura la qualità del co-clustering
 - 2.3 generalizzare più approcci (preesistenti e non) in unico framework più astratto

²Approccio con teoria dei grafi [10], approccio basato sull'entropia relativa [12], etc.

Divergenze di Bregman

Definition

Sia $\phi : S \rightarrow \mathbb{R}$, $S = \text{dom}(\phi)$, una funzione strettamente convessa definita su un insieme convesso $S \subseteq \mathbb{R}^d$ tale che ϕ sia differenziabile in $\text{int}(S)$ ³, che si assume essere non vuoto. La **divergenza di Bregman** $d_\phi : S \times \text{int}(S) \rightarrow [0, \infty)$ è definita come

$$d_\phi(x, y) = \phi(x) - \phi(y) - \langle x - y, \nabla\phi(y) \rangle \quad (1)$$

dove $\nabla\phi(y)$ rappresenta il gradiente del vettore di ϕ valutato in y e dove $\langle x, y \rangle$ è il *prodotto scalare* (o *prodotto interno*) canonico

$$\langle x, y \rangle = \sum_{i=1}^d x_i y_i$$

³ $\text{int}(S)$ con S insieme, rappresenta l'*interno* dell'insieme stesso.

Divergenze di Bregman - Esempio

Example

La **distanza euclidea** è la più semplice e più usata divergenza di Bregman. La funzione che sta alla base $\phi(x) = \langle x, x \rangle$ è strettamente convessa, differenziabile in \mathbb{R}^d e

$$\begin{aligned}d_{\phi}(x, y) &= \langle x, x \rangle - \langle y, y \rangle - \langle x - y, \nabla \phi(y) \rangle = \\ &= \langle x, x \rangle - \langle y, y \rangle - \langle x - y, 2y \rangle = \\ &= \langle x - y, x - y \rangle = \|x - y\|^2\end{aligned}$$

Informazione di Bregman

Definition

Sia X una variabile casuale che assume valori nell'insieme $\mathcal{X} = \{x_i\}_{i=1}^n \subset S \subseteq \mathbb{R}^d$ (S è convesso) seguendo una distribuzione di probabilità discreta ν . Sia d_ϕ una divergenza di Bregman. L'**informazione di Bregman** della variabile casuale X per la divergenza di Bregman d_ϕ è definita come

$$I_\phi(X) = \min_{s \in \text{int}(S)} E_\nu[d_\phi(X, s)] = \min_{s \in \text{int}(S)} \sum_{i=1}^n \nu_i d_\phi(x_i, s) \quad (2)$$

Il vettore ottimale s che permette di ottenere il valore ottimo è chiamato *rappresentante di Bregman* (o semplicemente *rappresentante*) di X e in [3] è dimostrato che tale rappresentante esiste sempre ed è unico.

Informazione di Bregman - Esempio

Example

Sia $X = \{x_i\}_{i=1}^n$ un insieme in \mathbb{R}^d e consideriamo una distribuzione uniforme su X , ovvero $v_i = \frac{1}{n}$. L'informazione di Bregman di X con la distanza euclidea come divergenza di Bregman è data da

$$I_\phi(X) = \sum_{i=1}^n v_i d_\phi(x_i, \mu) = \frac{1}{n} \sum_{i=1}^n \|x_i - \mu\|^2$$

che altro non è che la **varianza**, con μ la media degli elementi in X come rappresentante di Bregman.

Formulazione del problema

In termini generali, nel framework di Bregman un clustering ottimo è un'applicazione che mappa gli elementi in input nei rispettivi cluster, **minimizzando la perdita di Informazione di Bregman** nel passaggio dall'insieme di dati originale a una sua approssimazione ottenuta in base al clustering stesso.

Formulazione del problema - Co-clustering di Bregman I

1. Sia $Z \in S^{m \times n}$ con S convesso
2. Consideriamo Z una variabile casuale funzione di altre due variabili casuali, X e Y , che assumono valori rispettivamente nell'insieme degli indici di riga $\{1, \dots, m\}$ e nell'insieme degli indici di colonna $\{1, \dots, n\}$.
3. Sia ν una distribuzione di probabilità (in genere uniforme)
4. Sia d_ϕ una divergenza di Bregman.

Effettuare un co-clustering $k \times l$ della matrice Z equivale a determinare una coppia di applicazioni

$$\rho^* : \{1, \dots, m\} \rightarrow \{1, \dots, k\}$$

$$\gamma^* : \{1, \dots, n\} \rightarrow \{1, \dots, l\}$$

tale che

Formulazione del problema - Co-clustering di Bregman II

$$(\rho^*, \gamma^*) = \operatorname{argmin}_{(\rho, \gamma)} E[d_\phi(Z, \hat{Z})] = \sum_{i=1}^m \sum_{j=1}^n \nu_{ij} d_\phi(z_{ij}, \hat{z}_{ij}) \quad (3)$$

dove \hat{Z} un'approssimazione di Z univocamente determinata dal co-clustering (ρ^*, γ^*) .

Dato un co-clustering, si possono avere più approssimazioni \hat{Z} determinate univocamente da esso. Senza addentrarsi nei dettagli, basti sapere che le diverse approssimazioni vengono determinate dal riferimento che si usa per calcolare la distanza (e non dalla distanza di Bregman scelta).

Ad esempio, in [9] la distanza euclidea viene usata in due modi diversi:

- ▶ Misurando la distanza tra gli elementi originali e la media del co-cluster
- ▶ Misurando la distanza tra gli elementi originali e la relativa media di riga e media di colonna

Formulazione del problema - Co-clustering di Bregman III

Data la distanza euclidea come divergenza di Bregman, il problema del co-clustering ottimo (vedi eq. 3) in questo caso può essere formulato come segue

$$\begin{aligned}(\rho^*, \gamma^*) &= \operatorname{argmin}_{(\rho, \gamma)} E[(Z - \hat{Z})^2] = \\ &= \operatorname{argmin}_{(\rho, \gamma)} \|Z - \hat{Z}\|^2 = \\ &= \operatorname{argmin}_{(\rho, \gamma)} \sum_{\hat{X}, \hat{Y}} \sum_{i \in \hat{X}, j \in \hat{Y}} (z_{ij} - \hat{z}_{ij})^2\end{aligned}$$

Un meta algoritmo

1. Si inizia con un co-clustering arbitrario (ρ^0, γ^0) . Si inizializza $t = 0$ e si calcola la matrice approssimata \hat{Z}^t .
2. Si ripete uno dei due seguenti passi finché non si verifica la condizione di convergenza⁴
 - 2.1 Fissato il clustering di colonna γ^t , si calcola il nuovo clustering di riga ρ^{t+1} . Sia $\gamma^{t+1} = \gamma^t$. Si calcoli la matrice approssimata \hat{Z}^{t+1} .
 - 2.2 Fissato il clustering di riga ρ^t , si calcola il nuovo clustering di colonna γ^{t+1} . Sia $\rho^{t+1} = \rho^t$. Si calcoli la matrice approssimata \hat{Z}^{t+1} .

Si tratta di uno schema generale di minimizzazione per il problema del co-clustering di Bregman che può essere utilizzato in svariati casi particolari, sia nuovi che già precedentemente sperimentati.

⁴La condizione di convergenza può variare di caso in caso; un esempio potrebbe essere stabilire una soglia minima per il decremento utile della funzione obiettivo.

Perché il framework di Bregman?

È chiaro che è possibile elaborare un algoritmo di co-clustering a prescindere⁵ dal framework di Bregman. Perché dunque questa scelta?

Theorem

L'algoritmo generale per il co-clustering di Bregman converge a una soluzione localmente ottima per il problema del co-clustering di Bregman (eq. 3)

⁵Il già citato lavoro [9], nonostante sia un'istanza del problema del co-clustering di Bregman, è stato elaborato prima dell'introduzione di tale framework

Parte II

Clustering con Macchine a Supporto Vettoriale

Introduzione

1. Le *Support Vector Machines* (SVM) sono state formalizzate da Vladimir Vapnik [22].
2. Le SVM nascono come insieme di metodi di apprendimento supervisionato.
3. Sono considerate lo stato dell'arte in termini di classificatori

Le SVM sono state utilizzate con successo e ottimi risultati in applicazioni di classificazione, per le quali sono state create. Tuttavia, lavori recenti [1, 4, 5, 6, 7, 15, 23, 24] hanno utilizzato con successo le SVM anche in applicazioni di clustering.

Punti chiave dell'approccio

1. I dati sono mappati, tramite una trasformazione non lineare, in uno spazio delle feature multidimensionale usando un kernel Gaussiano
2. In questo spazio, si cerca la sfera di minor raggio che racchiuda l'immagine dei dati
3. Trovata la sfera, questa viene rimappata sullo spazio dati originale; questo forma un insieme di contorni che racchiudono i punti (i dati)
4. I punti racchiusi nello stesso contorno sono associati allo stesso cluster
5. Il numero di contorni e quindi di cluster pu' essere controllato tramite la larghezza del kernel
6. È possibile fare uso di una costante di *soft margin* per trattare con gli *outliers*⁶
7. È possibile trattare agevolmente con cluster che si sovrappongono condividendo molti valori

⁶Elementi difficili o impossibili da classificare

Struttura dell'algoritmo

La struttura dell'algoritmo proposto da Vapnik et al. per il Support Vector Clustering è costituito di due parti:

1. Problema di programmazione quadratica, per la determinazione del contorno dei cluster (addestramento)
2. Ricerca delle componenti connesse di un grafo, per l'assegnazione dei cluster (*cluster labeling*)

La prima parte è una leggera variante dell'addestramento nel caso supervisionato, la seconda parte è invece specifica dell'utilizzo non supervisionato delle SVM.

Contorni del cluster I

Sia $\{x_j\} \subseteq \chi$ un insieme di dati di N punti, con $\chi \subseteq \mathbb{R}^d$ lo spazio dei dati. Si usa una trasformazione non lineare Φ da χ a uno spazio delle feature multidimensionale, nel quale si cerca la sfera a raggio minimo R che racchiuda l'immagine dei dati

$$\begin{aligned} & \min_{R,a} R^2 \\ & \text{s.t.} \\ & \| \Phi(x_j) - a \|^2 \leq R^2, \forall j \end{aligned}$$

dove a è il centro della sfera.

Per includere i vincoli *soft* si aggiungono variabili di slack ξ_j ; i vincoli diventano

$$\| \Phi(x_j) - a \|^2 \leq R^2 + \xi_j, \forall j$$

con $\xi_j \geq 0$.

Contorni del cluster II

- ▶ Per risolvere il problema si introduce la funzione Lagrangiana

$$L = R^2 - \sum_j (R^2 + \xi_j - \|\Phi(x_j) - a\|^2) \beta_j - \sum \xi_j \mu_j + C \sum \xi_j \quad (4)$$

dove $\beta_j \geq 0$ e $\mu_j \geq 0$ sono i moltiplicatori di Lagrange, C la costante di margine, il cui valore determina un clustering più o meno *soft* e $C \sum \xi_j$ è il costo da pagare se $\xi_j > 0$.

Contorni del cluster III

Imponendo che le tre derivate di L siano zero, otteniamo i nuovi vincoli

$$\frac{\partial L}{\partial R} = 0 \implies \sum_j \beta_j = 1 \quad (5)$$

$$\frac{\partial L}{\partial a} = 0 \implies a = \sum_j \beta_j \Phi(x_j) \quad (6)$$

$$\frac{\partial L}{\partial \xi_j} = 0 \implies \beta_j = C - \mu_j \quad (7)$$

Le condizioni di Karush-Kuhn-Tucker

$$\xi_j \mu_j = 0 \quad (8)$$

$$(R^2 + \xi_j - \|\Phi(x_j) - a\|^2) \beta_j = 0 \quad (9)$$

implicano l'esistenza di tre tipi di vettori

Contorni del cluster IV

1. per $\beta_i = C$ abbiamo i *Bounded Support Vectors* (BSV) che giacciono al di fuori della sfera
2. per $0 < \beta_i < C$ abbiamo i *Support Vectors* (SV) che giacciono sulla superficie della sfera
3. per $\beta_i = 0$ abbiamo i vettori interni alla sfera

Convertendo la funzione Lagrangiana nella forma duale di Wolfe e usando il kernel Gaussiano⁷ $K(x, y) = e^{-q\|x-y\|^2}$ per rappresentare il prodotto scalare $\Phi(x) \cdot \Phi(y)$, la 4 diventa

$$W = \sum_j K(x_j, x_j)^2 \beta_j - \sum_{i,j} \beta_i \beta_j K(x_i, x_j) = 1 - \sum_{i,j} \beta_i \beta_j K(x_i, x_j) \quad (10)$$

La distanza dell'immagine di un punto x dal centro della sfera è

Contorni del cluster V

$$d(x) = \|\Phi(x) - a\|^2 = K(x, x) - 2 \sum_j \beta_j K(x_j, x) + \sum_{i,j} \beta_i \beta_j K(x_i, x_j) \quad (11)$$

Il raggio della sfera chiaramente equivale alla distanza dei SV dal centro della sfera

$$R = \{d(x_i) \mid 0 < \beta_i < C\} \quad (12)$$

I contorni che racchiudono i punti nello spazio dei dati sono definiti dall'insieme

$$\{x \mid d(x) = R\} \quad (13)$$

ovvero, i contorni del cluster sono delimitati dai SV, mentre i BSV risultano essere degli outlier. I restanti vettori sono punti interni.

⁷Il parametro q è la larghezza del kernel ed equivale a $\frac{1}{2\sigma^2}$, dove σ rappresenta la varianza all'interno della distribuzione Gaussiana.

Assegnazione dei cluster I

La 13 descrive i contorni dei cluster ma non permette l'assegnazione di un punto a un cluster. Pertanto

- ▶ Data una coppia di punti che appartengono a cluster diversi, il percorso che li collega deve uscire dalla sfera nello spazio delle feature; tale percorso contiene, dunque, almeno un punto y tale che $d(y) > R$.

Questo porta alla definizione della seguente matrice

$$A_{ij} = \begin{cases} 1 & \text{se, per ogni } y \text{ sul percorso tra } x_i \text{ e } y_i, d(y) \leq R \\ 0 & \text{altrimenti} \end{cases} \quad (14)$$

I cluster sono definiti come la componente connessa del grafo indotto da A . I BSV sono outlier: si può scegliere di lasciarli non classificati oppure di assegnarli al cluster più vicino.

La forma dei contorni

La forma dei contorni dei cluster è determinata da due parametri: la larghezza del kernel q e la costante di margine C .

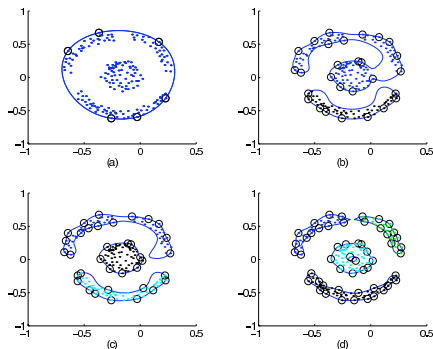


Figura: Clustering di un insieme di dati contenente 183 punti. La costante C è posta uguale a 1. I SV sono indicati da piccoli cerchi. (a) : $q = 1$ (b) : $q = 20$
(c) : $q = 24$ (d) : $q = 48$

I parametri

- ▶ La costante $C \geq 1$ assume che non ci siano BSV (vedi vincolo 5). In realtà spesso gli outlier impediscono la separazione dei contorni (e quindi il delinearci dei cluster) se $C \geq 1$; pertanto, variando tale parametro si fa uso dei BSV, rendendo più facile la separazione dei contorni, al costo di qualche outlier. Fissato un valore di q , diminuire C equivale ad aumentare il numero di BSV, poiché alcuni SV diverranno BSV. Un numero molto alto di BSV permette di trattare con cluster che si sovrappongono fortemente.
- ▶ Il parametro q determina il numero di SV. L'algoritmo di Support Vector Clustering viene usato iterativamente, aumentando man mano il valore di q . In genere il valore iniziale è

$$q = \frac{1}{\max_{i,j} \|x_i - x_j\|^2}$$

e comporta l'esistenza di un solo cluster.

Pro e contro

▶ Pro

- ▶ Possibilità di trattare con cluster di forma irregolare⁸.
- ▶ Mapping implicito nello spazio delle feature
- ▶ Controllo sul livello di clustering

▶ Contro

- ▶ Complessità computazionale $O(N^2m)$
 - ▶ Problema di programmazione quadratica (eq. 4): $O(N^2)$
 - ▶ Assegnazione dei cluster: $O(N^2m)$
- ▶ Il valore ottimo di q varia a seconda dell'insieme di dati ed è difficile regolarlo al meglio⁹

⁸Vantaggio unico di questa tecnica

⁹In [16] si fornisce un algoritmo per la determinazione di valori di q che portano a buoni risultati

Da esplorare

- ▶ In [15] viene presentato un metodo alternativo¹⁰ per l'assegnazione dei cluster con complessità lineare nella dimensione di N .
- ▶ In [21] viene presentata una variante della formulazione delle SVM: nelle LS-SVM (*Least Squares SVM*) la soluzione segue dalla risoluzione di un sistema lineare, anziché dalla risoluzione del problema di programmazione quadratica come accade per le SVM di Vapnik.
- ▶ Gli autori hanno espressamente esteso la loro versione delle SVM anche per l'apprendimento non supervisionato [21, cap. 7]

¹⁰La complessità viene abbassata sfruttando una caratteristica topologica del grafo; la soluzione calcolata resta ottima

Parte III

Strumenti software

Bregman Co-clustering

- ▶ Dhillon et. al [11] rendono disponibile liberamente codice C++ sviluppato da loro stessi per effettuare gli esperimenti delle proprie ricerche
- ▶ Il software implementa tre diversi algoritmi di co-clustering, tutti istanze del problema del co-clustering di Bregman
 - ▶ *Information-theoretic co-clustering* [12]: Entropia relativa, mutua informazione
 - ▶ Due versioni del *Minimum sum-squared residue co-clustering* [9]: Distanza euclidea, varianza
- ▶ In aggiunta, sono disponibili i dati utilizzati per gli esperimenti in [9]:
 - ▶ Quattro insiemi diversi
 - ▶ Per ognuno viene fornito: insieme di dati originale, insieme di dati ridotto, dati selezionati, risultati del clustering

SVM Software

- ▶ La lista di software che implementano le SVM di Vapnik è molto lunga [13].
- ▶ I software più famosi in assoluto sono quelli sviluppati da T. Joachims [14]:
 - ▶ *SVM^{light}*, usato largamente per classificazione e regressione.
 - ▶ *SVM^{struct}*, per modellare dati di output strutturati e multivariati, come alberi, sequenze, insiemi.
 - ▶ etc.
- ▶ Altro software molto popolare è LibSVM [8], che di recente fornisce anche un archivio di dati per problemi di classificazione e regressione direttamente nel proprio formato

Il problema è che tutte le soluzioni software attualmente in circolazione sono nate per gli obiettivi originali delle SVM: classificazione e regressione.

Algoritmo SVC

L'algoritmo SVC proposto da Vapnik et al. è costituito essenzialmente di due parti:

1. Problema di programmazione quadratica (eq. 4)
 2. Assegnazione dei cluster
- ▶ Il primo può essere risolto applicando modifiche minori (proposte in [19]) all'algoritmo *Sequential Minimal Optimization*, proposto per il training delle SVM nel normale utilizzo supervisionato. Tale algoritmo è implementato nel software open source Weka [18].
 - ▶ La seconda parte dell'algoritmo SVC è tipica dell'utilizzo non supervisionato delle SVM ed è quindi naturale che non esistano ancora, nei software di maggiore diffusione, implementazioni dei primi algoritmi di *cluster labeling* proposti in letteratura ([6, 15, 17]). Ulteriori ricerche di software sviluppato ad hoc non ha attualmente prodotto risultati.




Least Squares SVM Software

- ▶ Gli autori delle LS-SVM [21] rendono disponibile liberamente sul proprio sito [20] il software *LS-SVMlab*, una toolbox per Matlab.
- ▶ Se il software rispecchia completamente il materiale contenuto in [21], potrebbero essere presenti anche degli strumenti per l'utilizzo delle LS-SVM nell'apprendimento non supervisionato.





Parte IV

Bibliografia





Bibliografia I

-  D. Anguita, S. Ridella, F. Riviuccio, and R. Zunino.
Unsupervised clustering and the capacity of support vector machines.
In Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on, volume 3, pages 2023–2028, 2004.
-  A. Banerjee, I. S. Dhillon, J. Ghosh, S. Merugu, and D. Modha.
A generalized maximum entropy approach to bregman co-clustering and matrix approximation.
Technical report, UTCS TR04-24, UT, Austin, 2004.
-  A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh.
Clustering with bregman divergences.
Journal of Machine Learning Research, 6:1705 – 1749, 2005.

Bibliografia II

-  A. Ben-Hur, D. Horn, H. Siegelmann, and V. Vapnik.
A support vector method for hierarchical clustering.
In Fourteenth Annual Conference on Neural Information Processing Systems, Denver, Colorado, November 2000.
-  A. Ben-Hur, D. Horn, H. T. Siegelmann, and V. Vapnik.
A support vector method for clustering.
In NIPS, pages 367–373, 2000.
-  A. Ben-Hur, D. Horn, H. T. Siegelmann, and V. Vapnik.
Support vector clustering.
Journal of Machine Learning Research, 2:125–137, 2001.
-  A. Ben-Hur, H. T. Siegelmann, D. Horn, and V. Vapnik.
A support vector clustering method.
icpr, 02:2724, 2000.

Bibliografia III

-  C.-C. Chang and C.-J. Lin.
Libsvm - a library for support vector machines, 2007.
-  H. Cho, I. Dhillon, Y. Guan, and S. Sra.
Minimum sum squared residue co-clustering of gene expression data.
In Proceedings of the Fourth SIAM International Conference on Data Mining, pages 114–125, April 2004.
-  I. S. Dhillon.
Co-clustering documents and words using bipartite spectral graph partitioning.
In Knowledge Discovery and Data Mining, pages 269–274, 2001.
-  I. S. Dhillon, H. Cho, Y. Guan, and S. Sra.
Co-clustering software, version 1.1, 2004.

Bibliografia IV



I. S. Dhillon, S. Mallela, and D. S. Modha.

Information-theoretic co-clustering.

In *Proceedings of The Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(KDD-2003)*, pages 89–98, 2003.



O. Ivanciuc.

Svm - support vector machines, 2005.



T. Joachims.

Svmlight - support vector machine, 2004.



D. Lee.

An improved cluster labeling method for support vector clustering.

IEEE Trans. Pattern Anal. Mach. Intell., 27(3):461–464, 2005.

Member-Jaewook Lee.

Bibliografia V



S.-H. Lee and K. M. Daniels.

Gaussian kernel width generator for support vector clustering.

In M. He, G. Narasimhan, and S. Petoukhov, editors, *Advances in Bioinformatics and Its Applications*, volume 8, pages 151–162, 2005.



S.-H. Lee and K. M. Daniels.

Cone cluster labeling for support vector clustering.




In *Proceedings of 6th SIAM Conference on Data Mining*, pages 484–488, May 2006.



T. U. of WAIKATO.

Weka machine learning project, 2006.

Bibliografia VI

-  B. Scholkopf, R. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt.
Support vector method for novelty detection.
In In Advances in Neural Information Processing Systems 12: Proceedings of the 1999 Conference, 2000.
-  J. Suykens, T. V. Gestel, J. D. Brabanter, B. D. Moor, and J. Vandewalle.
Ls-svmlab, 2006.
-  J. A. K. Suykens, T. V. Gestel, J. D. Brabanter, and J. V. B. De Moor.
Least squares Support Vector Machines.
World Scientific Pub. Co., Singapore, 2002.

Bibliografia VII



V. N. Vapnik.

The Nature of Statistical Learning Theory.
Springer, 1995.



D. Wang, L. Shi, D. S. Yeung, P.-A. Heng, T.-T. Wong, and
E. C. C. Tsang.

Support vector clustering for brain activation detection.
In *Proceedings of the 8th International Conference on Medical Image
Computing and Computer Assisted Intervention (MICCAI 2005)*,
pages 572–579, Palm Springs, California, USA, July 2005.



J. Yang, V. Estivill-Castro, and S. K. Chalup.

Support vector clustering through proximity graph modelling.
In *Neural Information Processing, 2002. ICONIP '02. Proceedings of
the 9th International Conference on*, volume 2, pages 898–903,
November 2002.