

Dal clustering al co-clustering: una panoramica

Vincenzo Russo (vincenzo.russo@neminis.org)

26 aprile 2007

Sommario

Il clustering è la forma più comune di *unsupervised learning*. Esso è da tempo oggetto di ricerca assidua, grazie alla quale sono nati diversi approcci al problema. Studi recenti hanno da un lato lavorato per unificare diverse classi di algoritmi apparentemente diversi e dall'altro prodotto tecniche per affrontare più efficacemente problemi che coinvolgono dati sparsi e/o di grandi dimensioni. In questo documento, pertanto, si farà una panoramica sulle tecniche di clustering, evidenziando in particolare il lavoro svolto per formulare il clustering in termini di Informazione di Bregman e l'approccio del co-clustering.

Indice

1	Il clustering	3
1.1	Notazione	3
1.2	Formulazione del problema	4
2	Introduzione al clustering con le divergenze di Bregman	4
2.1	Divergenze di Bregman	5
2.2	Informazione di Bregman	5
2.3	Formulazione del problema	7
3	Il co-clustering	8
3.1	Applicazioni	8
3.2	Approcci al problema	9
3.3	Co-clustering e divergenze di Bregman	9
3.3.1	La distanza euclidea come divergenza di Bregman . . .	11
3.3.2	Un meta algoritmo	12
3.4	Perché il Bregman framework	13
3.5	Co-clustering con Support Vector Machine	14
3.5.1	Support Vector Clustering	14

1 Il clustering

L'obiettivo primario del clustering è suddividere un insieme di dati in sottoinsiemi, detti appunto *cluster*. L'intento è creare cluster che siano internamente consistenti, ma chiaramente differenti tra loro. In altre parole, un generico *oggetto* (sia esso un documento di testo o altro) in un cluster dovrà essere simile a tutti gli altri del medesimo cluster, mentre dovrà essere sostanzialmente diverso da quelli presenti in altri cluster.

Il clustering è la forma più comune di *apprendimento non supervisionato* [MRS07]. L'assenza di supervisione significa che non viene eseguito alcun processo di assegnazione manuale di oggetti alle classi per costruire un *training set*. Nel clustering sono la distribuzione e la struttura dei dati a determinare l'appartenenza o meno a un determinato cluster.

Il data clustering può essere suddiviso in più tipologie, da diversi punti di vista. Innanzitutto distinguiamo *Flat clustering* [MRS07, cap. 16] e *Hierarchical clustering* [MRS07, cap. 17], quest'ultimo ulteriormente suddivisibile in *Agglomerative clustering* (approccio *bottom-up*) e *Divisive clustering* (approccio *top-down*).

Da una prospettiva diversa è possibile suddividere gli algoritmi di clustering in *Partitional algorithms* e non *Non-partitional algorithms*. I primi hanno la peculiarità di calcolare una *partizione* dell'insieme di dati iniziale, ovvero di effettuare l'assegnazione di un oggetto a uno e un solo cluster; si parla in tal caso anche di *Hard clustering* e i cluster sono sottoinsiemi non vuoti e mutuamente disgiunti. La seconda classe di algoritmi invece prevede maggiore tolleranza e un oggetto può essere assegnato a più cluster. Questo comportamento, detto anche *Soft clustering*, può essere utile in diversi contesti, come la stessa gestione dei testi: spesso documenti, articoli di giornale, siti web possono appartenere a più categorie.

1.1 Notazione

Definiamo X l'insieme di dati di cui effettuare il clustering come

$$X = \{\vec{x}_i | \vec{x}_i = (x_{i_1}, \dots, x_{i_d})\}_{i=1..n}$$

con n la cardinalità dell'insieme X .

Ogni elemento $\vec{x}_i \in X$ è un vettore (o *punto*) che rappresenta un oggetto. Ogni componente $x_{i_l} \in X_l$, con $l = 1..d$, è detta *attributo* (o *feature*, *keyword*). Il valore d indica il numero di attributi.

Un dato viene dunque rappresentato come un vettore di attributi¹; pertanto l'insieme dei dati X può essere visto come una matrice di dimensioni $n \times d$, dove ogni riga rappresenta un dato e ogni colonna un attributo.

¹La selezione delle feature non verrà affrontata in questa sede, ma per maggiori informazioni è possibile consultare [MRS07, par. 13.5]

1.2 Formulazione del problema

L'obiettivo finale del clustering è quello di assegnare i punti di un insieme X di dati a un numero finito k di cluster [Ber02]. Si tratta dunque di organizzare i dati in k cluster, secondo opportuni criteri. In generale i cluster prodotti non si intersecano, ma come già accennato, questa assunzione può essere violata²; l'unione dei sottoinsiemi ci fornisce l'insieme di dati originale, con alcune possibili eccezioni dovute ad elementi non classificabili, indicati col nome di *outliers*.

Formalmente, possiamo definire l'obiettivo di un algoritmo di clustering³ come segue. Dati

- un insieme $D = \{d_1, d_2, \dots, d_n\}$ di dati
- il numero di cluster desiderati, k
- una funzione che valuta la qualità del clustering

si vuole determinare un'applicazione $\gamma : D \rightarrow \{1, \dots, k\}$ che minimizza⁴ la suddetta funzione, nel rispetto di opportuni vincoli.

La funzione che valuta la qualità del clustering è spesso definita in termini di *similitudine* o *distanza* tra i dati ed è di frequente indicata anche come *funzione di distorsione*. La misura della similitudine è il parametro chiave per un algoritmo di clustering.

2 Introduzione al clustering con le divergenze di Bregman

Si è detto che il parametro di maggior importanza per un algoritmo di clustering è la misura della similitudine tra dati. In letteratura sono state esplorate diverse soluzioni in questo senso. Tra gli algoritmi di Hard Clustering e Soft Clustering è stata utilizzata la *distanza euclidea* nel popolare algoritmo *k-means* [MRS07, par. 16.4] e nei più recenti esperimenti di clustering con *Support Vector Machine* [BHHSV01]. Più giovani sono gli approcci che poggiano le proprie basi sulla teoria dell'informazione, dove possiamo incontrare un utilizzo della divergenza di Kullback-Leibler (conosciuta anche come *entropia relativa*) come misura di similitudine [DMK03]. Ancora, sono state usate la distanza di Itakura-Saito, la distanza di Mahalanobis e la *I-divergence*.

Studi recenti [BMDG05] hanno dimostrato come tutte queste funzioni di misura della similitudine fossero casi speciali di una più vasta classe di divergenze, le *divergenze di Bregman* [BMDG05, par. 2], in presenza delle quali

²E ciò accade di frequente soprattutto nelle ultime applicazioni nel contesto Web

³Nella fattispecie *Partitional flat clustering* [MRS07]

⁴In altri casi potrebbero essere richiesta esplicitamente la massimizzazione.

il problema del clustering può essere formulato in termini di minimizzazione della perdita di *Informazione di Bregman* [BMDG05, par. 3.1].

2.1 Divergenze di Bregman

Di seguito presenteremo la definizione di *divergenza di Bregman* e alcuni esempi che mostreranno come le funzioni di distorsione citate nel paragrafo precedente appartengano in realtà alla classe delle divergenze di Bregman.

Definizione 1. 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 $\text{int}(S)$ ⁵, che si assume essere non vuoto. La *divergenza di Bregman* $d_\phi : S \times \text{int}(S) \rightarrow [0, \infty)$ è definita come

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

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

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

Esempio 1. 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

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

Altri esempi in [BMDG05].

2.2 Informazione di Bregman

In questa sezione introduciamo il concetto di *Informazione di Bregman* di una variabile casuale, basata sulla *Teoria tasso-distorsione* di Claude Shannon. Il problema tasso-distorsione consiste nel trovare uno schema di codifica con un dato tasso (ovvero, il numero di bit per simbolo) tale che la distorsione attesa tra la variabile casuale sorgente e la variabile casuale decodificata sia minimizzata. La distorsione ottenuta è detta *funzione tasso-distorsione* ed è la minima distorsione che è possibile ottenere dato un certo tasso.

⁵ $\text{int}(S)$ con S insieme, rappresenta l'*interno* dell'insieme stesso.

Consideriamo ora una variabile casuale X 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 . Misuriamo inoltre la distorsione con una divergenza di Bregman d_ϕ . Prendiamo in considerazione un semplice schema di codifica che rappresenta una variabile casuale con un vettore costante s , ovvero il cifrario è uno (o equivalentemente, il tasso è zero). La soluzione al problema tasso-distorsione in questo caso è una banale assegnazione e la relativa funzione di tasso-distorsione è data da $E_\nu[d_\phi(X, s)]$ che dipende dalla scelta del rappresentante s e può essere ottimizzata scegliendo il rappresentante corretto. La funzione di tasso-distorsione ottima è detta *Informazione di Bregman* della variabile casuale X per la divergenza di Bregman d_ϕ ed è denotata come segue

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

Il vettore ottimale s che permette di ottenere la distorsione minima è chiamato *rappresentante di Bregman* (o semplicemente *rappresentante*) di X .

La seguente proposizione afferma che il rappresentante esiste sempre, è unico e non dipende dalla divergenza di Bregman. Infatti, non si tratta altro che del valore atteso della variabile casuale X .

Proposizione 1. *Sia X una variabile casuale che assume valori in $\mathcal{X} = \{x_i\}_{i=1}^n \subset S \subseteq \mathbb{R}^d$ seguendo una funzione di probabilità positiva ν tale che $E_\nu[X] \in \text{int}(S)$.⁶ Dato una divergenza di Bregman $d_\phi : S \times \text{int}(S) \rightarrow [0, \infty)$, il problema*

$$\min_{s \in \text{int}(S)} E_\nu[d_\phi(X, s)]$$

ha un unico rappresentante $s = \mu = E_\nu[X]$.

Per una dimostrazione, consultare [BMDG05, par. 3.1].

Esempio 2. Sia $X = \{x_i\}_{i=1}^n$ un insieme in \mathbb{R}^d e consideriamo una distribuzione uniforme su X , ovvero $\nu_i = \frac{1}{n}$. L'informazione di Bregman di X con la distanza Euclidea come divergenza di Bregman è data da

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

⁶L'assunzione che $E_\nu[X] \in \text{int}(S)$ non è restrittiva, dato che una violazione può presentarsi soltanto quando $\text{co}(X) \subset \text{bd}(S)$, ovvero quando l'involuppo convesso di X è sulla frontiera di S .

che altro non è che la *varianza*.

2.3 Formulazione del problema

Sia X una variabile casuale che assume valori in $\mathcal{X} = \{x_i\}_{i=1}^n$ seguendo una distribuzione di probabilità ν . Quando X possiede un'informazione di Bregman molto grande, non è sufficiente un singolo rappresentante per codificare X , poiché il nostro obiettivo è quello di avere un errore minimo nella quantificazione. In queste situazioni è naturale suddividere l'insieme \mathcal{X} in k partizioni disgiunte $\{X_h\}_{h=1}^k$, ognuna con il proprio rappresentante di Bregman, tale che una variabile casuale M definita sui rappresentati delle partizioni, agisca come una quantificazione appropriata per X . Sia $\mathcal{M} = \{\mu_h\}_{h=1}^k$ l'insieme dei rappresentanti delle partizioni e sia $\pi = \{\pi_h\}_{h=1}^k$ con $\pi_h = \sum_{x_i \in X_h} \nu_i$ la distribuzione di probabilità indotta su \mathcal{M} . Pertanto, la variabile casuale indotta M assume valori in \mathcal{M} seguendo la distribuzione π .

Possiamo misurare la qualità della quantificazione M tramite la perdita di informazione di Bregman dovuta alla quantificazione stessa, ovvero tramite l'espressione $I_\phi(X) - I_\phi(M)$. Per $k = n$, la scelta migliore è $M = X$, che non comporta alcuna perdita di informazione; per $k = 1$, la quantificazione migliore è scegliere $E_\nu[X]$ con probabilità 1, incorrendo in una perdita di $I_\phi(X)$. Per valori intermedi di k , la soluzione è meno ovvia.

Definiamo quindi il *problema del (hard) clustering di Bregman* come il problema di trovare un partizionamento di \mathcal{X} o, in maniera equivalente, il problema di trovare una variabile casuale M tale che la *perdita di informazione di Bregman* a causa della quantificazione, $L_\phi(M) = I_\phi(X) - I_\phi(M)$, sia minima.

Teorema 1. *Sia X una variabile casuale che assume valori in $\mathcal{X} = \{x_i\}_{i=1}^n \subset S \subseteq \mathbb{R}^d$ secondo una distribuzione di probabilità positiva ν . Sia $\{X_h\}_{h=1}^k$ un partizionamento di \mathcal{X} e sia $\pi_h = \sum_{x_i \in X_h} \nu_i$ la distribuzione di probabilità π indotta sulle partizioni. Sia inoltre X_h la variabile casuale che assume valori in X_h secondo la distribuzione $\frac{\nu_i}{\pi_h}$ per $x_i \in X_h$, per $h = 1, \dots, k$. Siano infine $\mathcal{M} = \{\mu_h\}_{h=1}^k$, con $\mu_h \in \text{int}(S)$, l'insieme dei rappresentanti di $\{X_h\}_{h=1}^k$ e M una variabile casuale che assume valori nell'insieme \mathcal{M} seguendo π . Allora*

$$L_\phi(M) = I_\phi(X) - I_\phi(M) = E_\pi[I_\phi(X_h)] = \sum_{h=1}^k \pi_h \sum_{x_i \in X_h} \frac{\nu_i}{\pi_h} d_\phi(x_i, \mu_h)$$

Per una dimostrazione, consultare [BMDG05, par 3.2].

Usando il teorema appena enunciato, il problema di minimizzare la perdita di informazione di Bregman può essere riscritto come segue

$$\min_M (I_\phi(X) - I_\phi(M)) = \min_M \left(\sum_{h=1}^k \sum_{x_i \in X_h} \nu_i d_\phi(x_i, \mu_h) \right)$$

In generale gli algoritmi di clustering assumono una distribuzione uniforme $\nu_i = \frac{1}{n}$, $\forall i$, che è un caso speciale della formulazione presentata.

3 Il co-clustering

Sebbene l'idea del *co-clustering*⁷ risalga ai primi anni '70 [Har72], la maggior parte del lavoro in letteratura si è concentrato sul clustering su una dimensione, ovvero il raggruppamento delle sole righe (gli oggetti) della matrice di dati X (cfr. 1.1 a pagina 3), basato sulla somiglianza delle stesse rispetto alle colonne (gli attributi).

Il problema più sentito nel clustering classico è la grande dimensione dei dati e/o la loro sparsità. Per affrontare queste problematiche la soluzione più naturale è quella di cercare di ridurre la dimensione dei dati riducendo il numero di attributi con i quali vengono rappresentati gli oggetti. Questo obiettivo può essere raggiunto in vari modi, alcuni dei quali semplici ma inefficaci, come eliminare un certo numero di attributi: ciò porta nella maggior parte dei casi al solo danneggiare la rappresentazione di alcuni dati. Meno banale, ma più efficace è raggruppare gli attributi. Il primo modo di effettuare questo raggruppamento è quello di eseguire un *feature clustering* prima del clustering dei dati, come in [DMK03]. L'altro approccio è appunto il co-clustering, che oltre a produrre vantaggi nelle prestazioni finali proprio grazie alla già citata riduzione della dimensione, comporta anche altri benefici, come il poter osservare l'interazione degli oggetti con i cluster di attributi.

L'idea generale del co-clustering è produrre gruppi di attributi contestualmente al clustering dei dati stessi: è a tutti gli effetti il clustering simultaneo di punti e attributi, che sfrutta la dualità tra le righe e le colonne della matrice di dati.

3.1 Applicazioni

Il co-clustering è una tecnica di data mining molto potente con applicazioni in svariati contesti, come l'analisi del testo, analisi di microarray in biologia molecolare e in molte altre importanti applicazioni nelle quali i dati vengono di frequente rappresentati come matrici di dati.

⁷Il co-clustering è conosciuto in letteratura anche come *biclustering*, *simultaneous clustering*, *block clustering*, *conjugate clustering*, *distributional clustering*, *bi-dimensional clustering*, *two-mode clustering*.

3.2 Approcci al problema

Negli ultimi anni i lavori sul co-clustering si sono moltiplicati, grazie anche al sempre più frequente utilizzo di questa tecnica in bioinformatica [CDGS04]. A seconda dell'interpretazione della matrice di dati, è stato possibile sviluppare diversi approcci. Recenti sono i lavori che si basano sulla teoria dei grafi [Dhi01], dove la matrice è vista come una matrice di adiacenza (o incidenza); sulla teoria dell'informazione (con approccio *Maximum Entropy*) [DMM03], dove l'interpretazione della matrice è quella di una *tabella di contingenza*; sull'Informazione di Bregman [BDG⁺04a, BDG⁺04b], dove si astrae dalla particolare interpretazione della matrice, cercando, come già sperimentato nel clustering classico, di unificare i vari approcci in unico framework più astratto.

Il nostro principale intento è quello di trasportare nel co-clustering il recente approccio al clustering con Support Vector Machine [BHHSV01], più usate in genere nell'apprendimento supervisionato. Per raggiungere il nostro obiettivo è molto utile basarsi sul framework delle divergenze di Bregman citato in precedenza, una breve introduzione del quale segue nella prossima sezione.

3.3 Co-clustering e divergenze di Bregman

L'approccio tramite divergenze di Bregman permette di astrarre dalla particolare interpretazione della matrice. Il co-clustering di Bregman è formulato in termini della divergenza di Bregman tra una matrice di dati Z e una sua approssimazione \hat{Z} ottenuta in funzione del co-clustering.

Sia $Z \in S^{m \times n}$ una matrice di dati i cui elementi assumono valori nell'insieme convesso $S = \text{dom}(\phi)$. Con un abuso di notazione, possiamo considerare Z una variabile casuale che è funzione deterministica di altre due variabili casuali, X e Y , le quali assumono valori, rispettivamente, nell'insieme degli indici di riga $\{1, \dots, m\}$ e nell'insieme degli indici di colonna $\{1, \dots, n\}$. In aggiunta, sia $\nu = \{\nu_{xy}\}_{\substack{x=1, \dots, m \\ y=1, \dots, n}}$ la funzione di probabilità congiunta della coppia (X, Y) , generalmente una funzione di distribuzione uniforme e sia infine d_ϕ una divergenza di Bregman.

L'obiettivo è quello di effettuare il clustering simultaneo delle righe in (al massimo) k cluster disgiunti e delle colonne in (al massimo) l cluster disgiunti, ovvero effettuare un co-clustering $k \times l$ della matrice Z . Ciò equivale a determinare una coppia di applicazioni

$$\begin{aligned}\rho &: \{1, \dots, m\} \rightarrow \{1, \dots, k\} \\ \gamma &: \{1, \dots, n\} \rightarrow \{1, \dots, l\}\end{aligned}$$

L'applicazione che misura la qualità del co-clustering in termini di accuratezza dell'approssimazione \hat{Z} può essere definita come

$$E[d_\phi(Z, \hat{Z})] = \sum_{i=1}^m \sum_{j=1}^n \nu_{ij} d_\phi(z_{ij}, \hat{z}_{ij})$$

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

Dato un co-clustering (ρ, γ) , ci possono essere più matrici di approssimazione da esso univocamente determinate, a seconda dell'informazione che si sceglie di mantenere. Siano \hat{X} e \hat{Y} variabili casuali rispettivamente per il clustering di riga e di colonna. La prima assume valori nell'insieme $\{1, \dots, k\}$ e la seconda nell'insieme $\{1, \dots, l\}$, tale che $\hat{X} = \rho(X)$ e $\hat{Y} = \gamma(Y)$. Il co-clustering coinvolge, allora, quattro variabili casuali, X, Y, \hat{X}, \hat{Y} , corrispondenti ai diversi partizionamenti della matrice Z . Possiamo dunque ottenere diverse approssimazioni di Z basandoci soltanto sulle combinazioni non banali di $\{X, Y, \hat{X}, \hat{Y}\}$

$$\Gamma = \{\{X, \hat{Y}\}, \{\hat{X}, Y\}, \{\hat{X}, \hat{Y}\}, \{X\}, \{Y\}, \{\hat{X}\}, \{\hat{Y}\}\}$$

Ogni elemento dell'insieme potenza di Γ , $\pi(\Gamma)$, è un insieme di vincoli che portano a una possibile differente approssimazione della matrice Z . Riportiamo di seguito quattro esempi di insiemi di vincoli non banali in $\pi(\Gamma)$

$$\mathcal{C}_1 = \{\{\hat{X}\}, \{\hat{Y}\}\}$$

$$\mathcal{C}_1 = \{\{\hat{X}, \hat{Y}\}\}$$

$$\mathcal{C}_1 = \{\{\hat{X}, \hat{Y}\}, \{X\}, \{Y\}\}$$

$$\mathcal{C}_1 = \{\{X, \hat{Y}\}, \{\hat{X}, Y\}\}$$

Dato un co-clustering (ρ, γ) e un insieme di vincoli $\mathcal{C} \in \pi(\Gamma)$ esiste un insieme di possibili approssimazioni di Z

$$M(\rho, \gamma, \mathcal{C}) = \{Z' \in S^{m \times n} : \forall C \in \mathcal{C}, E[Z'|C] = E[Z|C]\}$$

In [BDG⁺04b] si dimostra che la migliore approssimazione \hat{Z} in $M(\rho, \gamma, \mathcal{C})$ è la matrice con la minima l'informazione di Bregman

$$\hat{Z} = \operatorname{argmin}_{Z' \in M(\rho, \gamma, \mathcal{C})} I_\phi(Z') \quad (1)$$

Per la caratterizzazione della soluzione a tale problema di minimizzazione, consultare [BDG⁺04a, par. 2.1].

Ora che siamo in grado di quantificare la qualità del co-clustering, possiamo fornire la definizione completa del problema del co-clustering di Bregman.

Definizione 2. Dati k e l , una divergenza di Bregman d_ϕ , una matrice

di dati $Z \in S^{m \times n}$, un insieme di vincoli $\mathcal{C} \in \pi(\Gamma)$ e una distribuzione di probabilità uniforme ν , si desidera trovare il co-clustering (ρ^*, γ^*) tale che

$$(\rho^*, \gamma^*) = \operatorname{argmin}_{(\rho, \gamma)} E[d_\phi(Z, \hat{Z})] \quad (2)$$

dove $\hat{Z} = \operatorname{argmin}_{Z' \in M(\rho, \gamma, \mathcal{C})} I_\phi(Z')$.

3.3.1 La distanza euclidea come divergenza di Bregman

È bene cercare di comprendere meglio quanto spiegato nella sezione precedente, abbandonando tal livello di astrazione a favore di una trattazione più specifica dove la matrice dei dati sia $Z \in \mathbb{R}^{m \times n}$ e la divergenza di Bregman sia la distanza Euclidea.

Innanzitutto, osserviamo le soluzioni al problema della minima informazione di Bregman nel qual caso si consideri la distanza euclidea. Facendo riferimento ai quattro insiemi di vincoli riportati nella sezione precedente, avremo quanto riportato in Tabella 1.

Vincoli \mathcal{C}	Approssimazione \hat{Z}
\mathcal{C}_1	$E[Z \hat{X}] + E[Z \hat{Y}] - E[Z]$
\mathcal{C}_2	$E[Z \hat{X}, \hat{Y}]$
\mathcal{C}_3	$E[Z X] + E[Z Y] + E[Z \hat{X}, \hat{Y}] - E[Z \hat{X}] - E[Z \hat{Y}]$
\mathcal{C}_4	$E[Z X, \hat{Y}] + E[Z \hat{X}, Y] - E[Z \hat{X}, \hat{Y}]$

Tabella 1: Soluzioni al problema della minima informazione di Bregman nel caso di distanza euclidea come divergenza di Bregman

In [CDGS04], Dhillon et al. hanno usato la misura Euclidea in due diversi modi:

1. Misurando la distanza tra ogni elemento nel co-cluster e la media del co-cluster
2. Misurando la distanza tra ogni elemento nel co-cluster e la relativa media di riga e media di colonna

Questi due diversi modi di misurare la distanza comportano due diverse approssimazioni \hat{Z}

1. $\hat{Z} = [\hat{z}_{ij}]$, tale che $\hat{z}_{ij} = z_{\hat{X}\hat{Y}}$
2. $\hat{Z} = [\hat{z}_{ij}]$, tale che $\hat{z}_{ij} = z_{i\hat{Y}} + z_{\hat{X}j} - z_{\hat{X}\hat{Y}}$ ⁸

dove

⁸La media del co-cluster viene sottratta alla quantità per garantire la simmetria

- $z_{\hat{X}\hat{Y}} = \frac{\sum_{i \in \hat{X}, j \in \hat{Y}} z_{ij}}{|\hat{X}| |\hat{Y}|}$ è la media di tutti gli elementi nel co-cluster
- $z_{i\hat{Y}} = \frac{\sum_{j \in \hat{Y}} z_{ij}}{|\hat{Y}|}$ è la media di tutti gli elementi nella riga i i cui indici di colonna sono nell'insieme \hat{Y}
- $z_{\hat{X}j} = \frac{\sum_{i \in \hat{X}} z_{ij}}{|\hat{X}|}$ è la media di tutti gli elementi nella colonna j i cui indici di riga sono nell'insieme \hat{X}
- $|\hat{X}|$ e $|\hat{Y}|$ denotano la cardinalità, rispettivamente, degli insiemi \hat{X} e \hat{Y}

È possibile constatare che la prima misura della distanza corrisponde all'insieme di vincoli \mathcal{C}_2 , mentre la seconda corrisponde all'insieme \mathcal{C}_4 ; di conseguenza, le due approssimazioni sopra riportate sono soluzioni di due istanze del problema dell'informazione minima di Bregman, rispettivamente la generica istanza con insieme di vincoli \mathcal{C}_2 e la generica istanza con insieme dei vincoli \mathcal{C}_4 .

Le diverse approssimazioni, dunque, non dipendono dalla scelta della divergenza di Bregman, ma da come si sceglie di applicare la stessa, e quindi, come si affermava nella sezione precedente, dall'informazione che si intende mantenere.

Il problema del co-clustering ottimo (ρ^*, γ^*) in questo caso può essere formulato come segue

$$\begin{aligned}
(\rho^*, \gamma^*) &= \operatorname{argmin}_{(\rho, \gamma)} E[(Z - \hat{Z})^2] = \\
&= \operatorname{argmin}_{(\rho, \gamma)} \|Z - \hat{Z}\|^2 = \\
&= \operatorname{argmin}_{(\rho, \gamma)} \sum_{\hat{X}, \hat{Y}} \sum_{i \in \hat{X}, j \in \hat{Y}} (z_{ij} - \hat{z}_{ij})^2
\end{aligned} \tag{3}$$

3.3.2 Un meta algoritmo

Il framework proposto da Dhillon et. al [BDG⁺04b, BDG⁺04a] porta alla creazione di un meta-algoritmo, ovvero 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.

Il meta-algoritmo si articola nei seguenti passi

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 risolvendo il problema della minima informazione di Bregman (equazione 1) rispetto al co-clustering t .
2. Si ripete uno dei due seguenti passi finché non si verifica la condizione di convergenza⁹

⁹La condizione di convergenza può variare di caso in caso; un esempio potrebbe essere stabilire una soglia minima per il decremento utile della funzione obiettivo.

- (a) 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} risolvendo il problema della minima informazione di Bregman rispetto al co-clustering $t + 1$. Sia $t = t + 1$.
- (b) 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} risolvendo il problema della minima informazione di Bregman rispetto al co-clustering $t + 1$. Sia $t = t + 1$.

È chiaro che i punti chiave dello schema sopra riportato sono:

- La risoluzione del problema della minima informazione di Bregman, per il calcolo di \hat{Z}
- L'aggiornamento dei cluster di riga e di colonna

Per quanto riguarda il primo punto, già nel paragrafo 3.3 abbiamo omesso la caratterizzazione della soluzione al problema della minima informazione di Bregman, poiché i lavori già citati ([BDG⁺04a, CDGS04]) ci forniscono i risultati mostrati in tabella 1 e in generale nel paragrafo 3.3.1 per quel che riguarda il caso particolare della distanza euclidea come divergenza di Bregman.

Per ciò che concerne il secondo punto, seguiremo la stessa linea: in questa sede non mostreremo il metodo generale per aggiornare i cluster di riga e di colonna durante l'esecuzione di un generico algoritmo di co-clustering basato sul framework delle divergenze di Bregman, poiché il nostro obiettivo è quello di costruire un algoritmo di co-clustering che faccia uso delle Support Vector Machine (*SVM*).

3.4 Perché il Bregman framework

Da quanto presentato nelle ultime sezioni si intuisce che dovrebbe essere possibile derivare un algoritmo di co-clustering che faccia uso di SVM senza necessariamente fare ricorso al Bregman framework¹⁰.

È chiaro dunque che tale scelta debba comportare dei vantaggi; primo fra tutti è il seguente teorema.

Teorema 2. *L'algoritmo generale per il co-clustering di Bregman converge a una soluzione localmente ottima per il problema del co-clustering di Bregman (eq. 2).*

Per una dimostrazione, consultare [BDG⁺04b, par. 4.4].

¹⁰Il già citato lavoro [CDGS04], nonostante sia un'istanza di un problema del co-clustering di Bregman, è stato elaborato prima dell'introduzione di tale framework

Dunque è dimostrato che lo schema generale di minimizzazione¹¹ raggiunga un ottimo locale, pertanto seguendo tale schema correttamente si avrà tale sicurezza anche nelle soluzioni elaborate per casi particolari.

3.5 Co-clustering con Support Vector Machine

Il lavoro svolto in [CDGS04], dove viene elaborata una generalizzazione del *k*-means applicato al co-clustering, ci fornisce un ottimo schema per elaborare un algoritmo di co-clustering che faccia uso delle Support Vector Machine per la fase di aggiornamento dei cluster di riga e di colonna e che usi la distanza euclidea come misura della qualità del co-clustering. Nel paragrafo 3.3.1 si è già mostrato come, data la distanza euclidea come divergenza di Bregman, sia possibile calcolare approssimazioni della matrice di dati \hat{Z} . Ciò che manca, quindi, è soltanto il modo utilizzare le SVM nella fase di aggiornamento dei cluster di riga e di colonna.

3.5.1 Support Vector Clustering

Nonostante le SVM siano uno strumento utilizzato per lo più nella *classificazione* (quindi *apprendimento supervisionato*), lavori recenti [ARRZ04, BHHSV01, BHSV00, BHHSV00] hanno utilizzato con successo le SVM anche in applicazioni di clustering.

Nel clustering con SVM la funzione obiettivo fa uso della distanza euclidea per misurare la distanza tra un oggetto e il centro di una sfera, il tutto dopo essersi “spostati” in uno spazio delle feature multi-dimensionale tramite una trasformazione non lineare [BHHSV01, par. 2.1].

Riferimenti bibliografici

[ARRZ04] D. Anguita, S. Ridella, F. Riveccio, 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. 3.5.1

[BDG⁺04a] A. Banerjee, I. Dhillon, J. Ghosh, S. Merugu, and D. Modha. A generalized maximum entropy approach to bregman co-clustering and matrix approximation. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(KDD)*, pages 509–514, August 2004. 3.2, 3.3, 3.3.2, 3.3.2

¹¹Per una versione più dettagliata dello schema generale riportato nella sezione precedente, consultare [BDG⁺04b, par. 4.4]

- [BDG⁺04b] 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. 3.2, 3.3, 3.3.2, 3.4, 11
- [Ber02] Pavel Berkhin. Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA, 2002. 1.2
- [BHHSV00] Asa Ben-Hur, David Horn, Hava T. Siegelmann, and Vladimir Vapnik. A support vector method for clustering. In *NIPS*, pages 367–373, 2000. 3.5.1
- [BHHSV01] A. Ben-Hur, D. Horn, H. T. Siegelmann, and V. Vapnik. Support vector clustering. *Journal of Machine Learning Research*, 2:125–137, 2001. 2, 3.2, 3.5.1
- [BHSHV00] Asa Ben-Hur, Hava T. Siegelmann, David Horn, and Vladimir Vapnik. A support vector clustering method. *icpr*, 02:2724, 2000. 3.5.1
- [BMDG05] Arindam Banerjee, Srujana Merugu, Inderjit S. Dhillon, and Joydeep Ghosh. Clustering with bregman divergences. *Journal of Machine Learning Research*, 6:1705 – 1749, 2005. 2, 2.1, 2.2, 2.3
- [CDGS04] 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. 3.2, 3.3.1, 3.3.2, 10, 3.5
- [Dhi01] Inderjit S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Knowledge Discovery and Data Mining*, pages 269–274, 2001. 3.2
- [DMK03] I. Dhillon, S. Mallela, and R. Kumar. A divisive information-theoretic feature clustering algorithm for text classification. *Journal of Machine Learning Research*, 3:1265–1287, March 2003. 2, 3
- [DMM03] 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. 3.2
- [Har72] J. A. Hartigan. Direct clustering of a data matrix. *Journal of the American Statistical Association*, 67(337):123–129, 1972. 3

- [MRS07] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2007. 1, 1, 2, 3