UNIVERSITÀ DEGLI STUDI DI MILANO

FACOLTÀ DI SCIENZE E TECNOLOGIE

Corso di Laurea in Informatica

Apprendimento di insiemi fuzzy nell'ambito del web semantico

Relatore: Prof. Dario Malchiodi

Correlatore: Prof.ssa Anna Maria Zanaboni

Tesi di Laurea di: Alessia Cecere
Matr. 923563

Anno Accademico 2020/2021

Indice

Introduzione

Capitolo 1

Apprendimento di insiemi fuzzy

1.1 Gli insiemi fuzzy

1.2 Applicazione alla ricerca di assiomi in un insieme di formule

Capitolo 2

Elementi del problema

2.1 L'algoritmo di apprendimento

Per affrontare il problema sopra descritto, si è utilizzata una variante dell'algoritmo di Support vector clustering, descritta in [1]

Dati $\{x_1, \ldots, x_n\}$ un campione di elementi appartenenti a un dominio X e $\{\mu_1, \ldots, \mu_n\}$ i rispettivi gradi di appartenenza a un fuzzy set sconosciuto A, duplice obiettivo dell'algoritmo è trovare il fuzzy set A, e inferire i parametri della funzione di appartenenza μ_A. Per spiegare la tecnica, immaginiamo di mappare i punti x_i su una circonferenza di raggio R e centro a sconosciuti, e che vi sia un modo di far sì che il loro grado di membership dipenda dalla distanza dal centro a. A questo punto, la ricerca si trasformerebbe nella risoluzione di un problema di ottimizzazione vincolata, di cui possiamo dare una prima formulazione:

$$min R^2 + C \sum_i \xi_i$$

$$s.t.$$

$$||x_i - a||^2 \leq R^2 + \xi_i$$

$$\xi_i \ \geq 0, C > 0$$

Ovvero, il problema diventa trovare la circonferenza di raggio minimo, al variare di R^2 e a, che raccolga i punti rappresentanti gli elementi del campione. Le ξ_i corrispondono a variabili di slack che vengono aggiunte per far rispettare i vincoli di appartenenza, la cui sommatoria minimizzo nella funzione obiettivo; il valore C funge da bilanciamento tra questo adattamento e il rispetto del vincolo, identificando se sia più importante mantenere vincoli stringenti o mappare tutti i punti nella circonferenza. In realtà, l'obiettivo è che la distanza dal centro della circonferenza rappresenti un grado di appartenenza all'insieme fuzzy, e dunque andiamo a inserire le μ_i nella formulazione, che diventa, moltiplicando per μ_i:

$$minR^2 + C\sum_i(\xi_i + \tau_i)$$

$$s.t.$$

$$\mu_i||x_i - a||^2 \leq \mu_i R^2 + \xi_i$$

$$(1-\mu_i)||x_i - a||^2 \geq \mu_i R^2 - \tau_i$$

$$\xi_i \ \geq 0, \tau_i \ \geq 0, C > 0$$

La formulazione cattura in parte l'insieme fuzzy, infatti:

- Se $\mu_i = 1$, il secondo vincolo diventa ridondante, e si torna alla formulazione iniziale, nella quale si chiede di trovare il centro e il raggio della circonferenza più piccola che contiene tutti i punti con membership 1.
- Se $\mu_i = 0$, il primo vincolo diventa ridondante, e la formulazione è volta a cercare punti che stanno fuori dalla ciconferenza.
- Se $\mu_i = \frac{1}{2}$, moltiplicando entrambi i vincoli per 2 ottengo:

$$||x_i - a||^2 \leq R^2 + 2\xi_i$$

$$||x_i - a||^2 \geq R^2 - 2\tau_i$$

Dal momento che entrambe le variabili di slack devono essere il più possibile vicine e zero, questo sta a significare che per la membership $\frac{1}{2}$ i punti tendono a stare esattamente sulla circonferenza. Rimane il problema di modellazione delle membership intermedie, come si vedrà in seguito. Per affrontare quest'ultima formulazione del problema di ottimizzazione, si risolve il problema duale, attraverso il metodo di Wolfe. Per farlo, costruiamo la funzione lagrangiana, sottraendovi obiettivo e vincoli, moltiplicati per altrettante variabili:

$$L = R^2 + C\sum_i(\xi_i + \tau_i) - \sum_i \alpha_i(\mu_i + R^2 + \xi_i - ||\xi_i - a||^2) -$$

$$\sum_i \beta_i[(1-\mu_i)||x_i - a||^2 - (1-\mu_i)R^2 + \tau_i] - \sum_i \gamma_i\xi_i - \sum_i \delta_i\tau_i$$

I vincoli richiedono che vengano messe a zero tutte le derivate parziali della lagrangiana rispetto al problema originale. Sostituendo il pattern $\sum_i \alpha_i\mu_i - \beta_i(1-\mu_i)$ con ϵ_i, sapendo che $\sum_i \epsilon_i = 1$, a partire dal vincolo sulla derivata parziale rispetto ad a arriviamo all'equazione $\sum_i x_i \epsilon_i = a$. Dunque, nel momento in cui trovo le variabili ottimali ϵ_i, so di riuscire a identificare anche il centro della circonferenza cercata. La forma finale del problema duale, da massimizzare, è dunque:

$$max \sum_i \epsilon_i x_i x_i - \sum_{i,j} \epsilon_i \epsilon_j x_i x_j$$
$$s.t.$$
$$\sum_i \epsilon_i = 1$$
$$C \geq 0, 0 \leq \alpha i \ \leq C, 0 \leq \beta_i \ \leq C$$

Una volta risolto il problema e ottenute le ϵ_i^* ottimali, posso calcolare il valore:

$$R^2(x) = xx - \sum i \epsilon_i^* x_i x + \sum i, j \epsilon_i^* \epsilon_j^* x_i x_j$$

Si può dimostrare che questa quantità è esattamente uguale al quadrato della distanza tra a^* valore ottimare del raggio e x fornito. Inoltre, se prendo un i tale che $0 \leq \alpha i \ \leq C$ o $0 \leq \beta i \ \leq C$, allora $R^2(x) = R^*$, raggio ottimale della sfera. Perciò, dato un punto, sono in grado di stimare il suo grado di appartenenza alla sfera: se $R(x)^2 > R^*$ ha membership $> \frac{1}{2}$, se $R(x)^2 = R^*$ la membership è $\frac{1}{2}$, se $R(x)^2 < R^*$. Se trovo una funzione che si comporta in questo modo, posso calcolare la membership in ogni punto.

In realtà, nell'algoritmo i punti non vanno visti come inscritti in una circonferenza ma in una sfera, e mappati nello spazio tridimensionale tramite una trasformazione Φ, che viene definita kernel. Anche non conoscendo la funzione specifica, ma conoscendo la sua famiglia, posso ricavarne il prodotto scalare. Ad esempio, nel caso sia una funzione polinomiale di grado al massimo p, $\Phi(x_i) \cdot \Phi(x_j) = (1 + x_i + x_j)^p$. Di conseguenza, la formulazione finale del problema diventa:

$$max \sum_i \epsilon_i \Phi(x_i)\Phi(x_j) - \sum_{i,j} \epsilon_i \epsilon_j \Phi(x_i)\Phi(x_j)$$

2.1.1 Utilizzo della libreria mulearn

Qui descriverei mulearn e in particolare come è stata utilizzata durante gli esperimenti.

2.2 Il Kernel

Qui inizierei a descrivere il ruolo del kernel nella computazione.

2.2.1 Kernel precomputato

Qui descriverei il kernel precomputato utilizzato, passando per la matrice di Gram e per il relativo adjustment. Sempre in questo contesto spiegherei i problemi riscontrati durante il fitting, e la possibilità che siano stati causati proprio dal fatto che non sia esattamente un kernel.

2.2.2 Alternative al kernel precomputato

Qui descriverei le altre forme di computazione del kernel che sono state considerate.

Length-based similarity

Hamming similarity

Levenshtein similarity

Jaccard similarity

Capitolo 3

Esperimenti

Qui andrei a mostrare gli esperimenti e i relativi risultati conseguiti tramite cross validation e model selection.

3.1 Riproduzione degli esperimenti originali

3.2 Esperimenti sul kernel

3.2.1 Kernel alternativi

3.2.2 Possibili soluzioni al problema del fitting

Eliminazione combinatoria di formule

Eliminazione a campione di formule

Valori di similarità come vettori in input all'algoritmo

Conclusione

Bibliografia

[1] Dario Malchiodi and Witold Pedrycz. Learning membership functions for fuzzy sets through modified support vector clustering. 2013.