Verso le Città Cognitive.
Un esempio di classificatore fuzzy:
il bot parcheggiatore

Francesco Saverio Cannizzaro, Giovanni Capizzi,
Francesco Landolina, Maria Grazia Palumbo
Dipartimento di Matematica e Informatica,
Università degli Studi di Palermo

Marco Elio Tabacchi
Gruppo di Ricerca SCo2 - Dipartimento di Matematica e Informatica,
Università degli Studi di Palermo
Istituto Nazionale di Ricerche Demopolis, Italia
marcoelio.tabacchi@unipa.it

1. Introduzione

Le città cognitive (Portmann, Seising e Tabacchi, 2017) sono una possibile evoluzione delle smart cities (Portmann e Fingers, 2016). Nella progettazione di una città cognitiva si tiene conto, oltre che della rete di sensori ed attuatori che contribuiscono alla condivisione dei dati, anche del rapporto tra la città ed il cittadino; a questo fine sono utilizzate una serie di tecnologie proprie della Computational Intelligence (Kacprzyk e Pedrycz, 2015), quali Metaeuristiche, Algoritmi evolutivi e genetici e metodologie Soft Computing per includere nel dialogo non solo vaste moli di dati, ma la possibilità di analisi introspettive che utilizzino come interfaccia da e verso gli utenti i linguaggi naturali e le logiche imprecise (Perticone e Tabacchi, 2016, D'Asaro et al., 2017). In questo lavoro presentiamo un esempio schematico di assistente virtuale basato sui classificatori a regole fuzzy (Magdalena, 2015), che utilizza ed integra gli open data prodotti dalla smart city con metodi linguistici per la classificazione al fine di indicare all'utente la convenienza nella scelta di un parcheggio in una realtà non conosciuta.

2. Il bot parcheggiatore

Chi visita una città straniera ed utilizza una macchina a nolo per le lunghe distanze (ad es. il tragitto aeroporto-città) si trova spesso a dover scegliere un parcheggio senza disporre delle informazioni che i residenti ricavano dall'esperienza quotidiana. Mentre la smart city può fornire molte informazioni logistiche utili, ad es. la localizzazione degli stalli liberi e gli orari di parcheggio libero e a pagamento, altri tipi di informazioni dipendenti dalla conoscenza delle dinamiche cittadine rimangono esclusivamente nella disponibilità degli umani: esempi sono la sicurezza (statistica) di un parcheggio rispetto ai furti, e la vicinanza con mezzi pubblici e categorie di punti di interesse. In questa sezione presentiamo la struttura di un bot di Telegram (2017) che ricevendo le coordinate fisiche di uno stallo di sosta libero ed utilizzando gli open data resi disponibili dal Comune di Palermo (Italia) fornisce in risposta un grado di convenienza nel parcheggiare in quello stallo. In Tabella 1 sono elencati i dati in ingresso forniti al bot. Il bot è programmato in Python (van Rossum, 1995).

Tipo di dato	Valori	Parametro base
Incidenti Stradali (IL, IG, IM)	{pochi, abbastanza, molti}	Numero Medio
Parcheggiatori Abusivi (PA)	{pochi, abbastanza, molti}	Numero Medio
Rimozioni (RI)	{poche, abbastanza, molte}	Numero Medio
Fermate Bus (FB)	{poche, abbastanza, molte}	Numero Medio, frequenza attesa <15min.

Tabella 1: dati in ingresso

L'algoritmo utilizzato per la classificazione è il seguente:

- Ognuno dei dati (IL, IG, IM, PA, RI, FB) viene convertito in insieme Fuzzy, secondo la seguente logica:
    - Si sovrappone alla mappa della città una griglia con risoluzione di 100mt.
    - Si calcolano i dati relativi ad ogni zona e se ne derivano minimi, massimi e tutte le informazioni necessarie alla definizione degli insiemi fuzzy, distinguendo i dati per categorie (Parcheggiatori Abusivi, Incidenti Stradali, Fermate AMAT).
    - Gli estremi dell'insieme fuzzy vengono calcolati come segue:
        - "poco" = {minimo, minimo, medio}
        - "abbastanza" = {minimo, medio, massimo}
        - "molto" = {medio, massimo, massimo}
    - Vengono definite le regole di classificazione del sistema fuzzy (Kacprzyk e Pedrycz, 2015, Rule-Based Systems) che andranno a definire la risposta per l'utente (ad es. "Se vi sono **pochi** IS, ed

abbastanza FB, allora è **molto** conveniente parcheggiare").
    - Si assegna per ogni regola un certo grado di confidenza.
    - Le regole con i gradi di confidenza maggiori vengono inserite nell'insieme di regole del sistema.
- Ad ogni interrogazione, il bot calcola i dati per la posizione, e li fornisce al sistema, ricavando una risposta per l'utente. Per FB i dati sono calcolati considerando soltanto le fermate per le quali un bus transita entro 15 min. Per i luoghi di interesse, la presenza o meno (su richiesta dell'utente) influirà sulla risposta del bot.

Un esempio dell'interazione è mostrato in Figura 1.

Figura 1: interazione tipo con il bot

In futuro il bot potrà essere migliorato, ottimizzando le euristiche scelte per i parametri, ed utilizzando una classificazione con apprendimento automatico che utilizzi gli open data per generare le regole senza (o con un limitato) intervento umano, ed aumentando il numero di parametri da tenere in considerazione grazie alla maggior diffusione delle informazioni generate dalla città cognitiva.

Bibliografia

D'Asaro, F. A., Di Gangi, M. A., Perticone, V., and Tabacchi, M. E. (2017). Computational intelligence and citizen communication in the smart city. Informatik-Spektrum,

40(1):25–34.
Kacprzyk, J., e Pedrycz, W. (Eds.). (2015). Springer handbook of computational intelligence. Springer.
Magdalena, L. (2015). Fuzzy rule-based systems. In Springer Handbook of Computational Intelligence (pp. 203-218). Springer.
Perticone, V. e Tabacchi, M. E. (2016). Towards the improvement of citizen communication through computational intelligence. In Portmann, E. and Fingers, M., editors, Towards Cognitive Cities, volume 63 of Studies in Systems, Decision and Control, pages 83–100. Springer International Publishing.
Portmann, E. e Fingers, M. (Eds.). (2016), Towards Cognitive Cities, volume 63 of Studies in Systems, Decision and Control. Springer.
Portmann, E., Seising, R e Tabacchi, ME. (Eds.). (2017), Designing Cognitive Cities. Studies in Systems, Decision and Control. Springer (in press).
van Rossum, G. (1995) Python tutorial, Technical Report CS-R9526, Centrum voor Wiskunde en Informatica (CWI).
Telegram (2017), retrived online, https://telegram.org