Lezione 1

Corso	Riconoscimento e recupero dell'informazione per Bioinformatica
Data	@October 4, 2021 1:48 PM
Rifacimento	☐
Status	Completed
Tipo	

Pattern Recognition:

Estrarre informazioni, sono tecniche che sono alla base di molti programmi.

Cos'è la pattern recognition?

Il procedimento che porta l'essere umano a rispondere a domande del tipo: individuare classi di oggetti, riconoscere oggetti, distinguere oggetti, è la pattern recognition.

Riconoscere, associare, distinguere e identificare. Sono tutte informazioni legate al concetto di ***categoria***. Il processo che ci porta a questo è la Pattern Recognition, legato al concetto di classe, categoria, gruppo → Prendo in ingresso un insieme di dati, effettuo un'analisi di tali dati per rispondere ad una domanda legata al concetto di categoria.

Pattern: immagine, suono, odore, è il dato che viene analizzato, l'entità di interesse.

> *Definizione*
>
> Il processo che prende in input dati grezzi ed effettua un'azione sulla base della categoria dei dati.

Il processo quindi interagisce con il pattern, come fa l'uomo automaticamente (ma lo fa tramite processi complicati che non sono ora completamente chiari)

Spostandosi nella prospettiva informatica: **voglio che questo sia automatico** → scrivere un algoritmo che è in grado di fare pattern recognition. Il problema è stato studiato a lungo, ma cose che per l'uomo sono banali per una macchina sono molto difficili.

Perché?

C'è molta variabilità → completamente diversi per dare una definizione corretta e universale.

Oggetti della stessa classe possono essere molto diversi ma oggetti di classi diversi possono anche essere molto simili.

Gli esseri umani hanno sensori migliori, anni e anni di evoluzioni hanno portato i nostri sensi a sviluppare il modo migliore per poter analizzare i dati → devo quindi avere dei **sensori** in grado di riconoscere i pattern. I "nasi elettronici" ad esempio sono lontanissimi dall'accuratezza del nostro naso: ad esempio ci sono dei macchinari in grado di riconoscere la qualità dell'olio, ma non si riesce ancora a riconoscere tutta la gamma degli odori.

Gli esseri umani inoltre hanno informazione d'insieme, globale. Il calcolatore no, vede ogni pixel, vede ogni numero, ma non riesce a vedere l'insieme. **Il calcolatore non vede un contesto, ma matrici di numeri e questo è estremamente difficile.**

Scrivere un algoritmo che riconosce le immagini è molto più complicato → devo considerare molte variabili.

Esempi classici

Distinguere diverse persone sulla base del volto, pattern: la parte dell'immagine che contiene la faccia. Riconoscere che anche cambiando le variabili di una persona (umore, espressione, ecc.): l'uomo ha la capacità di riconoscere i pattern anche tramite informazioni parziali, cosa che il calcolatore non ha.

Riconoscimento del parlato, delle impronte digitali o riconoscimento di gesti.

Esempio in cui il calcolatore è migliore dell'uomo: biometria basata sul riconoscimento delle impronte digitali. In realtà negli ultimi anni ci sono diversi casi, soprattutto perché il calcolatore non si stanca (task che richiedono tempo e fatica).

> Biometria è la scienza che permette di riconoscere un individuo sulla base di tratti caratteristici.

Dopo l'11 settembre c'è stato un boom di investimenti su quel versante.

Altro esempio: riconoscimento di scene o di categorie di luoghi. Classificazione di video, (primo es. quello della classificazione dei pixel in bianco e nero a seconda del si muovono o meno - secondo es. fare tracking dei movimenti, es. riconoscere se mi trovo in una situazione anomala tipo tentato furto di auto in un parcheggio).

💡 **Problema principale:**
Capire e modellare i diversi pattern di un problema (cercare di caratterizzarli in termini di classi/gruppi/categorie) → **paradigma principale** (come lo risolvo?): apprendimento da esempi. La conoscenza deriva da un insieme di esempi campionati dal problema (training set).

Per un calcolatore più che la definizione, funziona **l'esempio**. La cosa fondamentale è che, dati gli esempi, il calcolatore ha costruito la sua conoscenza, sa riconoscere anche pattern al di fuori del training set: ha imparato.

💡 **GENERALIZZAZIONE**: capacità di generalizzare e riconoscere anche oggetti non presenti nel training set.

Si fa solitamente un'operazione per costruire un modello (disegnare un algoritmo), si fa un'operazione di **ottimizzazione**.

Devo inventare una funzione:

> **model ← max E(T, P, Theta)** *dove T è training set, P è informazioni a priori e Theta è parametri.*
> Il tutto viene formulato tipicamente con un algoritmo.

Devo

- definire la E:
  - compromesso tra la capacità di spiegare il training set e la complessità

  es. la funzione E da trovare per fittare il polinomio, ad esempio trova la funzione minima che si avvicini ai punti del training set.)
- ottimizzare la E

- Aspetti pratici
  - accuratezza
  - requisiti computazionali (tempo e spazio)
  - flessibilità

- usabilità (quindi l'accettabilità è anch'essa un requisito fondamentale)

Problemi princpali in PR

- Classificazione
- Detection
- Clustering

Occorre costruire un modello a partire dai dati.

Aspetti principali

La realizzazione di un sistema di pattern recognition implica la soluzione di alcuni problemi:

- **Rappresentazione**

  Come rappresento in modo digitale gli oggetti? Devo trovare un modo per farlo.

- **Costruzione del modello**

  Supponendo di aver raccolto un training set, devo capire come costruire il modello

- **Testing**

Dal problema faccio **campionamento**, che mi permette di portare l'oggetto all'interno del calcolatore (foto, scansione, spettrometro,ecc.) → operazione di pulizia

Quando costruisco il modello, uso delle informazioni a priori, faccio un addestramento e ottengo dei modelli. Nel testing uso dei modelli addestrati per estrarre le informazioni necessarie.

Rappresentazione

> 💡 **Obiettivo**: trovare una rappresentazione digitale per gli oggetti del problema in esame.

Cos'è una *feature*?

Una singola misura, l'insieme delle misure è il pattern. (Es. un immagine che è pattern, ogni pixel è feature). Un'immagine è una matrice, un insieme di pixel (puntini) che posso vedere come un insieme di valori. Il valore mi indica quanta luce

colpisce quel punto. A seconda della risoluzione ho un numero di puntini, raffinando la rappresentazione posso estrarre sempre più feature (es. larghezza, altezza), estraendo queste feature ho molte più informazioni su cui confrontarmi → questa operazione si chiama **Preprocessing** (es. riduco le dimensioni)

Costruzione del modello

> 💡 **Problema**: costruire un modello in grado di spiegare i dati del training set (generalizzare).

Lo scopo fondamentale è astrarre.

Paradigma di apprendimento da esempi (basato su training set) si basa su:

- le misure

Training Set

Il training set deve essere:

- largo (molti pattern)
- completo (tutte le categorie sono rappresentate)
- variabile (deve tenere in considerazione la variabilità dei pattern nelle categorie)

Non basta un buon metodo, serve un training set adeguato.

Esempio: **classificazione**, nel training set so esattamente per ogni oggetto a che classe appartiene. Vado ad estrarre le feature per ogni training set e posso rappresentare il tutto nello spazio. Costruire un classificatore vuol dire essere in grado di dividere le due classi. Per poter poi classificare un oggetto, estraggo le feature nello stesso modo in cui facevo col training set e uso le feature come avevo fatto in precedenza, nel modello dato dal training set. pattern recognition supervisionata (ho info a priori)

L'obiettivo ovviamente è classificare correttamente.

Esempio: **detection**, voglio capire se appartiene o meno ad una determinata classe. La costruzione del modello è simile alla classificazione.

Esempio: **clustering**, insieme di oggetti su cui non ho alcuna informazione a priori, devo riuscire a distinguere delle classi tra quegli insiemi, scoprire i gruppi in cui

posso dividerli. Il problema è molto più difficile: pattern recognition non revisionata (no info a priori), non posso misurare la correttezza del risultato.

> Il concetto di cluster è molto vago: dipendentemente dalle misure di similarità utilizzate, così cambia il risultato. Ci sono varie sottocategorie che sono possibili e buone. Il problema è NP-completo.

L'informazione a priori è fondamentale per ottenere risultati che abbiano un senso.

Quindi la costruzione del modello può essere *supervisionato* o *non supervisionato*.

Interpretazione dei risultati

Una volta costruito il modello, devo usarlo.

> Il focus è l'interpretabilità: dei metodi e delle soluzioni. Soprattutto nel contesto della biomedicina, l'interpretabilità è fondamentale, bisogna sempre cercare di usare e produrre metodi e soluzioni interpretabili.

La soluzione che ha valore è la soluzione in cui l'utente è a proprio agio e capisce cosa sta succedendo, soprattutto nel contesto biomedicale.

Ad alto livello pattern recognition e machine learning sono la stessa cosa. In realtà ci sono delle differenze: pattern recognition viene dalla fisica (dati reali e focus sulle soluzioni), machine learning viene dalla matematica (il focus è sulle proprietà dei metodi).

Pattern Recognition e Bioinformatica

Perché ha senso usare queste tecniche nella bioinformatica?

💡 **LA MOTIVAZIONE PRINCIPALE**: la caratterizzazione di una popolazione in termini di gruppi/classi/categorie può essere utilizzata per inferire alcune proprietà di oggetti sconosciuti guardando ad oggetti conosciuti nello stesso gruppo.

- in bioinformatica ci sono molti problemi di clustering, classificazione e detection
- Possibilità di derivare modelli per i dati tramite esempi
- Ci sono problemi di classificazione che possono essere automatizzati

Problemi classici di bioinformatica:

- Microarray per capire l'espressione di un determinato gene

Si hanno poi delle matrici in cui ogni riga ha un diverso livello di espressione. Questa matrice viene analizzata tramite classificazione di campioni, oppure tramite clustering (ad esempio coregolazioni, ecc.)