

PUMA – Luglio 2021

ORLANDO LEOMBRUNI

Dove eravamo rimasti...

- ▶ Siamo giunti a buon punto con la ricostruzione di un encefalo
 - ▶ 2 parametri: T_1 e T_2
 - ▶ dizionario di ~170mila elementi
 - ▶ Segnale e dizionario normalizzati, "defasati" e portati a valori reali

Metodologia di base (1)

- ▶ Avevamo sviluppato questa metodologia: partiamo aggiungendo del noise al dizionario (in "quantità" variabile)
- ▶ Per ogni entry del dizionario, prendiamo le componenti e le **discretizziamo** mediante binning
 - ▶ Dividiamo lo spazio di ciascuna componente in X bin
 - ▶ Ogni componente viene mappata sull'indice (n. intero) del bin corrispondente
 - ▶ Risultato: una entry diventa una stringa di interi -> un **pattern**
- ▶ La discretizzazione porta inesorabilmente al fatto che più entry del dizionario verranno convertite nel medesimo pattern
 - ▶ L'ideale sarebbe che entry "simili" (corrispondenti a valori vicini dei parametri) vengano convertite nello stesso pattern

Metodologia di base (2)

- ▶ Per ogni pattern generato come spiegato prima, manteniamo la lista delle entry del dizionario che vengono convertite in quel pattern
- ▶ I voxel del segnale vengono discretizzati con lo stesso procedimento
- ▶ Per ogni voxel convertito, eseguiamo il pattern matching e troviamo uno (full match) o più (partial match) pattern della banca che corrispondono al voxel
- ▶ Alla fine, facciamo il prodotto scalare "standard" sulle entry legate al pattern scelto anziché sull'intero dizionario e troviamo il massimo

Metodologia di base (3)

- ▶ In questo modo, idealmente eseguiamo una minima frazione dei prodotti scalari altrimenti richiesti e troviamo lo stesso risultato
- ▶ Nella realtà, non sempre il risultato è lo stesso del prodotto scalare su tutto il dizionario
- ▶ Se l'errore è piccolo (ossia siamo caduti in un pattern adiacente che non contiene la entry esatta, ma una sua "vicina") possiamo accettarlo in virtù del risparmio in termini di operazioni

Ricostruzione dell'encefalo

- ▶ All'inizio: scarsi risultati
 - ▶ Dizionario e segnale "non si parlavano"
- ▶ Bassa efficienza (<70% voxel ricostruiti)
- ▶ Bassa precisione (deviazione standard su T_1 : ~1)
- ▶ Svoltata: viewsharing

Ricostruzione con viewsharing

- ▶ Risultati migliori ottenuti:
 - ▶ Applicando il noise con i seguenti parametri:
 - ▶ 5% noise massimo, 5 livelli di noise (1%, 2%, 3%, 4%, 5%), 30 ripetizioni -> 150 applicazioni
 - ▶ Usando 15 bin per la discretizzazione
- ▶ Voxel ricostruiti: 18790/18829 -> 99.83%
- ▶ Errore medio su T1: 3.03% (deviazione standard: 0.1131)
- ▶ Errore medio su T2: 3.70% (deviazione standard: 0.0301)
- ▶ N. prodotti scalari: 91M anziché 3238M
 - ▶ ~1/36esimo

Passi successivi

- ▶ Dopo aver ottenuto questi risultati incoraggianti, abbiamo portato avanti due strade parallele:
 1. Testare quanto appena visto sull'hardware fisico (vedi slide Alberto)
 2. Provare ad estendere la metodologia ad un caso più generale, con 4 parametri

Il secondo segnale

- ▶ Un ginocchio
- ▶ 4 parametri: T1, B0, B1, Fat Fraction
- ▶ Dizionario di ~800000 elementi

Difficoltà

- ▶ Al nuovo segnale (e dizionario) non possono essere applicate le ipotesi semplificative di prima
- ▶ In particolare, le componenti di ogni voxel/entry vanno trattate come numeri complessi
- ▶ La metodologia precedente non porta buoni risultati

Vecchia metodologia sul ginocchio

- ▶ Con 10 bin: ~97% efficienza, ma precisione bassissima (std T1 = 1.5)
- ▶ Con 15 bin: precisione un po' più alta (std T1 = 1), bassa efficienza (77%)
- ▶ È chiaro che non bisogna togliere la fase e soprattutto mantenere i vettori complessi

Nuova metodologia, nuovo simulatore

- ▶ Abbiamo modificato il simulatore della AM Board creato precedentemente per estenderlo col supporto a vettori complessi
- ▶ Stesso processo di discretizzazione (binning) delle componenti, ma...
 - ▶ Applicato separatamente al vettore delle parti reali e a quello delle parti immaginarie
 - ▶ Quindi, se prima avevamo (ad es.) 10 componenti esse venivano convertite in 10 numeri interi, mentre ora ne abbiamo 20: 10 per la parte reale, 10 per la parte immaginaria
- ▶ Per il resto, il procedimento è lo stesso (es. applicazione del noise)

Difficoltà

- ▶ Lavorando con numeri complessi e con il dizionario con molte più entry, i tempi di simulazione aumentano a dismisura
 - ▶ In particolare, l'applicazione del noise e la simulazione della AM vera e propria
 - ▶ Si passa da circa 20 minuti totali per il caso del cervello a oltre 16 ore per il ginocchio
- ▶ Aumenta anche la memoria (RAM) richiesta dal simulatore
- ▶ Risultato: minore capacità di individuare errori e di stimare l'euristica per i parametri

Risultati

- ▶ Miglior risultato a cui siamo arrivati:
 - ▶ 30 ripetizione di 5 livelli di noise, max 5%, 10 bin
 - ▶ Efficienza: 92.3% (18242/19771 voxel ricostruiti)
 - ▶ Deviazione standard su T1: 0.4555
 - ▶ Prodotti scalari: 1/14 rispetto allo standard
- ▶ Domanda: Esiste la possibilità di avere il ginocchio in viewsharing? (O simili)