Slide 1

***Data ebedding*** vuol dire cercare di proiettare punti in uno spazio **d**-dimensionale grande, che non è visibile all’occhio umano, dato che risuciamo a vedere solo il 3D (che già può dare problemi).

Per noi il visibile sono il 2D ed il 3D, il nostro occhio riesce a vedere massimo 3 dimensioni.

Ci sono dei metodi che consentono di rappresentare più dimensioni in uno spazio 2D, massimo 3D.

E si cerca il modo migliore per proiettare questi punti.

Quello in slide è un esempio di proiezione. Il massimo che si può mostrare in slide è il 2D.

Slide 2

Prima idea: una sola dimensione → si proietta sull’asse delle x, il problema è che classi diverse si sommano e quindi si perde la caratteristiche dell’asse delle y.

Le classi blu si confondono, si mettono una sopra l’altra.

Slide 4

Uno dei metodi di *data embedding*.

Slide 5

Si cerca di avere una proiezione in cui i punti che erano vicini rimangano vicini ed i punti che erano lontani rimangano lontani.

Si vuole una proiezione che mantenga le distanze corrette.

Slide 8

In tante applicazoni si è disposti a perdere delle informazioni sulle distanze più grosse e mantenere informazioni sulle distanze più piccole.

Se si vede il malfunzionamento della PCA della slide di prima (classi che si sovrappongono sulla linea), il problema è dovuto al fatto che la PCA viene calcolata utilizzando delle proieizioni lineari.

Slide 9

Il fatto che venga **massimizzata** la varianza vuol dire che vengono considerati più importanti i punti più distanti dal punto medio, ossia l’origine.

Quando i punti blu proiettati sull’asse delle y sono tanto vicini ai punti gialli, la PCA, viste queste vicinanze che riducono la distanza globale tra i vari punti, riesce a schiacciarli un po’.

Si vorrebbe una tarsformazione che funzionasse come ha funzionato per i punti rossi.

Ciò che interssa è mantenere gli intorni: i punti che erano vicini nello spazio originario lo devono essere anche nello spazio nuovo, mentre i punti distanti devono rimanere distanti.

Slide 10

È un dataset di 24x24 pixel.

Un’immagine è un insieme di pixel. Un colore molto chiaro in un’immagine corrisponde ad un pixel che ha un valore pari a 255.

Un colore molto scuro corrisponde ad un pixel che ha il valore pari a 0.

Un’imagine RGB è un’immagine a 3 canali: rosso, verde e blu.

Un’immagine è una matrice di numeri, in cui ogni numero è il valore di un pixel.

Se si ha un’immagine con un solo canale è un’immagine a toni di grigio.

Indipendentemente da tutto ciò.

Se si prende un’immagine, ch è una matrice di 24x24 pixel.

Si raccolgono tutti i valori dell’immagine e si linearizza quindi l’immagine 24x24.

Quindi si hanno dei vettori (24x24=576) 576-dimensionale.

Si vuole trasformare l’immagine in vettori.

Comè un vettore linearizzato che rappresenta 1?

All’inizio ha tutti zeri, poi ad un certo punto ha un bel po’ di elementi consecutivamente uguali a 255 e poi ritorna a 0.

I vettori degli 0 si distanziano dai vettori che formano l’1.

Piacerebbe vedere se c’è una distribuzione che caratterizza i 2, i 6, gli 8…

Si può voler avere magari dei **classificatori**.

Quando si lavora su un dataset, è utile fare delle visualizzazioni per capire cosa c’è all’interno di tale dataset.

Slide 11

Quelle su cui i punti sono più distanti l’uno dall’altro.

Ogni colore indica il numero rappresentato nell’immagine (della slide di prima).

Qualcosa di buono la PCA fa.

Problema: si supponga di non avere classi.

Se non si hanno e si vuole vedere se c’è un **raggruppamento** è difficile.

Quindi se si eliminano i colori, si avrebbe un insieme di punti e basta.

Le classi servono per vedere come i dati sono raggruppati.

Slide 13

Ci interessa mantenere i vicinati.

Slide 14

**SNE** è il padre di **t-SNE**.

Slide 15

È uno **swiss roll**. Mostra una proiezione in uno spazio a dimensione ridotta.

La distanza tra A e B è la stessa che c’è tra A e C.

Se ci si mette in un qualsiasi punto nel centro, la distanza è la stessa.

Percettivamente l’umano segue la curva.

Tendiamo a seguire la continuità.

Secondo questo pincipio si dovrebbe misurare la distanza tra A e B come distanza geodetica, non euclidea.

Euclidea: è una distanza tra due punti, in particolare è una misura della lunghezza del segmento avente per estremi i due punti.

Geodetica: è la distanza che si percorre quando andiamo in montagna - la curva più breve che congiunge due punti di uno spazio.

t-SNE sfrutta questo concetto: dal momento che ciò che interessa è spesso il mantenimento delle sdistantze geodetiche, ciò che si vuole mantenere è la distanza tra gli intorni.

Se il punto A è tanto distante da B o tanto distante da C non importa; perché tanto quei punti non sono collegati tra di loro.

Si parte da un insieme di punti X e l’**xi**-esimo punto è uno spazio. Si vuole mappare ogni **xi**.

Slide SNE

*Stochastic* significa che c’è la statistica.

*Neighborhood*: questo metodo funziona facendo la mappatura degli interi intorni.

Quindi si dà una descrizione degli intorni.

Come si esprimono gli intorni in ogni punto?

Per ogni punto e **xi** si prendono tutti gli altri punti e si calcola uno *score* per il **j**-esimo punto.

Si hanno i punti **xi** ed **xj**.

Si sta considerando **xi**, per ogni altro punto **xj** si considera un raggio di **2sigmai**.

Per ogni **xi** si ha un sigma diverso (ecco perché c’è la **i** vicino a **sigma**).

Il denominatore è un fattore di normalizzazione poiché la sommatoria di tutti i possibili punti che hanno uno score dà come risultato 1.

Per ogni **xi** si costruisce una distribuzione di probabilità.

Slide 18

Si tratta di una **Gaussiana**.

Slide 19

Se si calcola la Gaussiana in un punto x che si discosta dalla media, la Gaussiana dà un valore crescente progressivamente.

La probabiltià di trovare **xj** se ci si centra su **xi** è praticamente 0.

Slide 20

Ad ogni punto **j** si dà un valore che dipende da **xi**.

Più **xj** è distante da **i** più lo score diminuisce.

Se **xj** dista più di **3sigmai** da **i** (n poisitivo o negativo) allora lo score è 0.

Se per ogni punto si considerano tutti gli altri punti, per ogni punto si sa dire quali sono gli altri punti nell’intorno.

Per ogni riga si può dire qual è la distanza/lo score del punto **j**-esimo rispetto al punto **i**-esimo.

Lo score dice qual è la probabilità che **j** sia nell’intorno di **xi**.

**xi** è una misura dell’intorno che si sta calcolando e varia da punto a punto.

Slide 21

Per ogni punto **yi** si fanno i medesimi calcoli fatti prima: si vede com’è fatto il suo intorno, quali sono **i** **j** che hanno la probabiltà di trovarsi nell’intorno di **i** (come prima).

Siccome si sta andando in un nuovo spazio, nello spazio di partenza i punti sono dati e non si può far nulla, nel nuovo spazio si vuole che i raggi siano uguali ossia si vuole che gli intorni siano uguali.

Slide 22

Se per ogni punto **xi** si prendono tutti gli altri punti **xj** nello spazio di partenza e si fa lo stesso nello spazio di arrivo.

Per tutti i **j** si hanno due distribuzioni di probabilità.

Si vuole trovare una mappatura che mantenga gli intorni.

Si sta dicendo che se nello spazio di arrivo, questa distribuzion esprime una distribuzione dell’introno **xi**, nel nuovo spazio la distribuzione che si calcola deve esser il più possibile simile alla distribuzione dello spazio di partenza: se è bassa nello spazio di partenza lo deve essere anche nel nuovo spazio.

Slide 23

Come si calcola la distanza tra due distribuzioni?

Con la *Kullback-libler divergence* (**KL**).

Si calcola prendendo tutti i punti della distribuzione in **j** presenti in **Pi** e **Qi**.

In rosso è la curva *Kullback* degli spazi di partenza e di arrivo.

Slide 24

La curva di *Kullback* tra P e Q è diversa; c’è un logaritmo al numero degli elementi di P e quindi un valore maggiore di 1 è un valore positivo…

Slide 25

Nella distribuzione di partenza se si ha uno score nello spazio di partenza alto e lo si mappa sullo score nello spazio .

Se il punto **xj** è molto vicino al punto **xi** si ha uno score molto alto.

Si vuole minimizzare la curva di *Kullback*.

La curva di Kullback per trasformazioni che mantengono le distanze, non le pesa.

Il fatto che il mantenimento delle distanze non pesi è corretto.

Ciò che non va bene in SNE è il fatto che un punto che è lontano rispetto ad **xi**, viene riavvicinato ad **xi** nel nuovo spazio.

Si è visto che SNE ciò che fa è cercare di mantenere gli intorni, andando a minizzare la distanza di curva *Kullback*.

Dà un “*reward*” a punti che inizialmente erano lontani e ora sono vicini.

**xi** si scegli diverso da un punto all’altro.

Slide 26

La perplessità è legata all’entropia della distribuzione che si sta calcolando.

L’entropia di una distribuzione è calcolata come **H(Pi)**.

Indipendentemente dalla formula, l’entropia dice quanto è “incasinata” una distribuzione.

I **pj** sono sempre minori di 1.

L’entropia dice quanto i valori di una distribuzione sono sparsi.

Dato che si sta facendo una somma di tutti i punti che si stanno calcolando, quindi di tutti gli **j** rispetto agli **i**, si definisce un **sigma**.

Si dà un bound al **sigmaj**.

L’entropia è una sorta di varianza della distribuzione.