Nel package ***Manifold learning*** si hanno tanti metodi (**TSNE**, **LLE**…) che fanno *embedding* di spazi ad alta dimensionalità a spazi a bassa dimensionalità.



 Un’immagine è una matrice di punti: dove c’è il nero si ha 0, mentre si ha il

bianco quando si ha 255.

In **MNIST** si hanno le immagini.

Con lo **scatter plot** si fa in modo che ogni vettore sia un punto.

L’asse x indica la prima coordinata del vettore e l’asse y la seconda coordinata.

**PCA** impiega meno tempo: nell’ordine del secondo.

**t-SNE** impiega tanto tempo: nell’ordine dei 150 secondi.

Slide 16

Per ogni punto in quel cerchio si dà uno score: più è distante il punto più è piccolo lo score.

Il raggio è definito in modo da far sì che la perplessità riguardante il cerchio nel punto **xi** sia minore ad un punto riguardante la soglia.

t-SNE utilizza uno score che non è simmetrico.

Si fa in modo che gli score siano simmetrici.

Slide 17

**Sigmai** è il raggio.

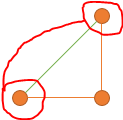
L’entropia esprime la varianza degli score, se si hanno tanti **j**.

Se si aumenta il cerchio, aumentano i punti e quindi aumenta la varianza degli score, aumenta l’entropia e aumenta quindi la perplessità.

Per intorni fitti si ha un **sigmai** piccolo, per gli altri intorni si ha un **sigmai** più alto.

Slide 19

Con t-SNE, lo score di ogni punto lo si disegna sulla base di una Gaussiana (funzione sulla DX).

La distanza tra i punti cerchiati è sempre la stessa per ciascun punto dal centro.

CROWDING PROBLEM

*Il "problema di affollamento" è definito come: "l'area della mappa bidimensionale che è disponibile per ospitare punti moderatamente distanti non sarà abbastanza grande rispetto all'area disponibile per ospitare punti vicini.”*

Supporre che il triangolo ↑ sia pieno e che si voglia distribuire i quadratini che compongono il triangolo tutti sull’asse delle x per ridurre la dimensionalità: si avrebbero dei quadratini che vanno a scontrarsi l’uno con l’altro.

Slide 20

Invece si vuole che questi quadratini si distribuiscano e occupino uno spazio maggiore sull’asse delle x, come mostrato in slide → per evitare il problema del ***crowding***.

Slide 22

Lo score nello spazio di partenza lo si calcola con la **Gaussiana** (che è un tipo di distribuzione).

Mentre nello spazio di arrivo si utilizza la **t-distribution**, che dà uno score minore a punti molti vicini (rispetto alla Gaussiana) e dà score più alti a punti che sono più lontani.

La *t-distribution* spalma la Gaussiana nello spazio.

Se prima la Gaussiana (o comunque punti) occupava questo spazio/larghezza sull’asse delle x 

, dopo la mappatura vengono spinti ad occupare uno spazio maggiore

.

In modo da evitare il *crowding problem*: non si vuole arrivare ad avere punti appiccicati tra di loro.

SNE mappava una Gaussiana in una Gaussiana, avendo così il *crowding problem*, come visto prima, ed utilizzava degli score non simmetrici.

t-SNE utilizza:

• score simmetrici → lo score che esprime la distanza da A a B è lo stesso valore dello score che esprime la distanza da B ad A.

• un sistema di *scoring* che evita il *crowding problem*.

Slide 23

O i punti nello spazio di arrivo sono più distanti di quanto lo erano nello spazio di partenza oppure si ha lo stesso score, quindi **qij**=**pij**.

La perplessità è un parametro che si fornisce in input alla funzione t-SNE.

**p** e **q** dipendono da come sono le nuove disposizioni dei punti.

Come si trovano le nuove disposizioni dei punti, ovvero gli **yi**? Da un processo iterativo, detto discesa del gradiente.

Slide 26

Quando si deve trovare il massimo ed il minimo di una funzione si calcola la derivata.

La derivata dà l’orientazione della retta del gradiente.

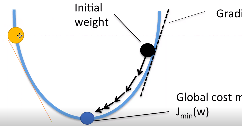
La retta del gradiente è quella retta che in ogni punto è tangente alal funzione e va nella direzione della funzione.

• Se **m** è negativo e la funzione in quel punto è discendente.

• Se **m** è positivo, quindi la derivata è positiva, la funzione sta salendo.

La retta del gradiente indica In che direzione in cui sta andando la funzione.

Si piazzano i punti totalmente a caso, si calcolano i valori **qij** nel nuovo spazio (**pij** si hanno già e non si muovono), si calcola la curva Kullback-leibler nel punto giallo:



Ogni punto è un valore della curva Kullback-leibler.

Si sfrutta il gradiente.

Si può fare un aggiornamento della posizione dei punti.

Se si spostano i punti in modo da seguire il gradiente della curva di Kullback-leibler, si raggiunge il minimo, la derivata minima (pallino in mezzo alla curva).

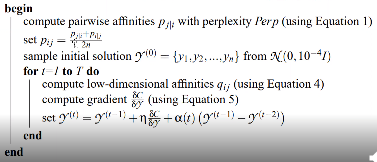
Slide 27

Il gradiente della curva Kullback-leibler rispetto alla nuova disposizione dei punti:

Calcola la distanza dei punti nel nuovo spazio.

Slide 28

***Learning rate*** → non ci si muove sempre di uno step rispetto al gradiente, ma si somma un po’ il gradiente per spostarsi di poco.

****- **Equation 1**: calcolo dello score basato sulla Gaussiana.

**- pij** → si fa in modo che gli score siano simmetrici

**2n** sono i punti totali.

***- Sample intiali solution*** significa che si sparano a caso i punti nel nuovo spazio, che è dato dalla dimensione che si vuole ottenere, ossia **t**.

- **T** indica l’iterazione ed è un parametro da fornire a t-SNE.

Slide 29

Per **MNIST**: alla fine, alla 100° iterazione, si hanno i punti mappati nel nuovo spazio.

Slide 31

Dato che t-SNE è lento, si utilizza PCA per fare una prima riduzione della dimensionalità.

Arrivati ad uno spazio 50-dimensionale si utilizza t-SNE.

Slide 32

La differenza tra due vettori è un altro vettore che parte da un punto **yi** (giallo) e raggiunge un altro punto **yj** (arancione).

Slide 33

Ogni punto subisce un update.

→ E’ l’update che subisce il punto **i**-esimo.

Ogni punto subisce forze attrattive o repulsive dagli altri punti.

Queste forze funzionano in base a come lavora t-SNE.

Slide 42

Nota sulla differenza tra vettori.