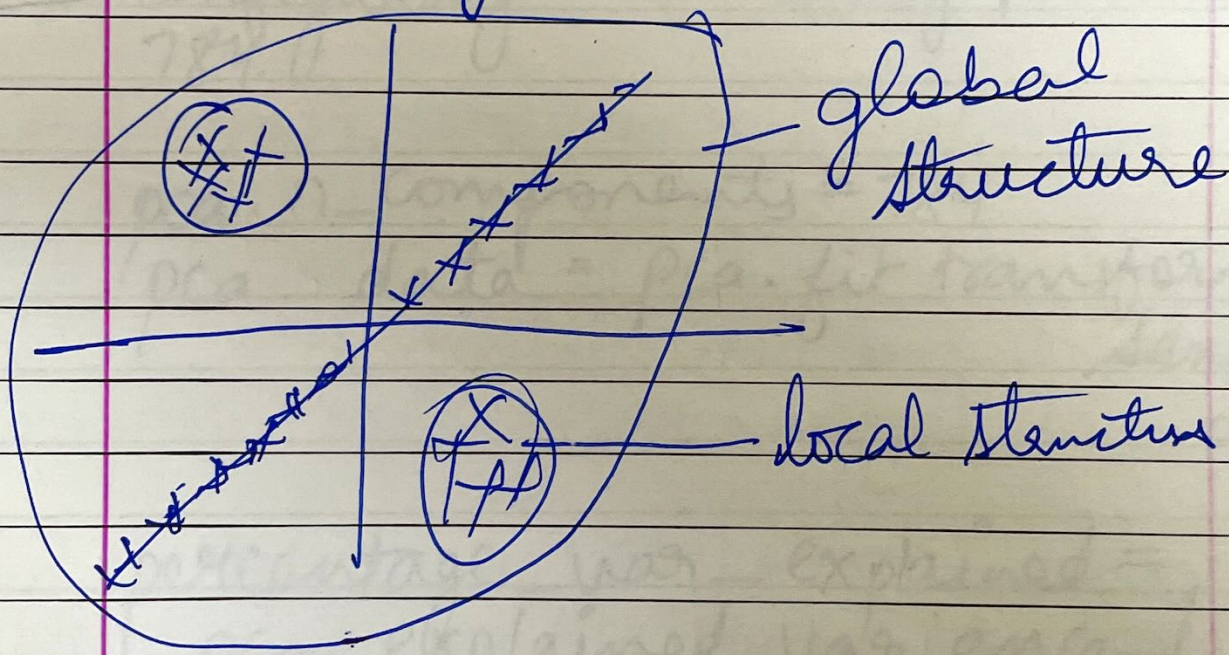
t-SNE- t-distributed Neighbourhood Embedding

DATE
PAGE

lec1 PCA - Basic
tSNE - Best dim reduction

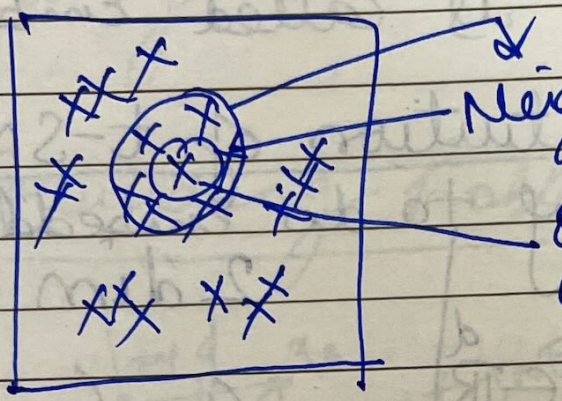
PCA - tries to preserve the global shape of data.



t-SNE - preserves local structure

DATE
PAGE

lec2 Neighbourhood of a point, Embeddi (-ng

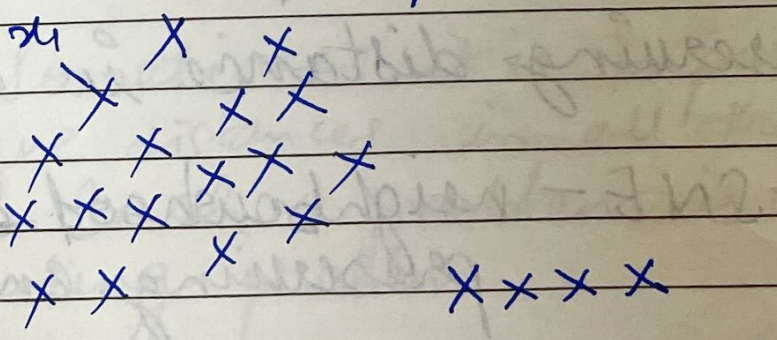


All points x_j

$N(x_i) = \{x_j$, s.t. x_i and x_j are geometrically close$\}$

$||x_i - x_j||^2 = \text{distance}^2$

* d-dimensional space:



2d: $x^{d_1} \times x_2^{d_2}$

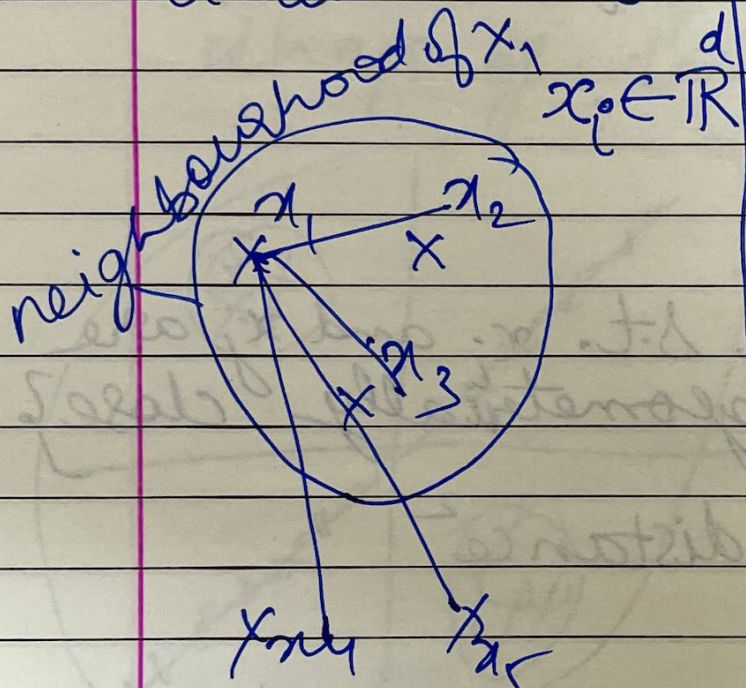
DATE _____
PAGE _____

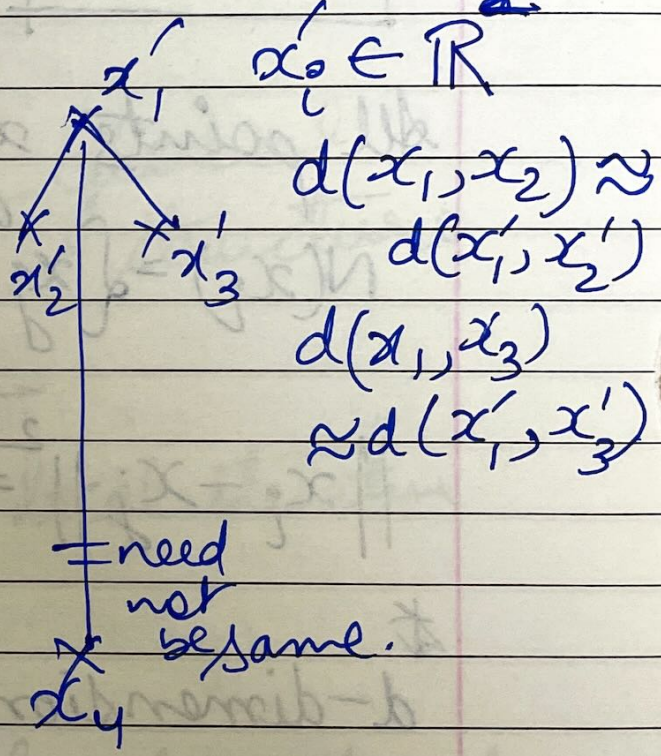
Every ~~Taking~~ points from high dim space, we have points in low-dim space is called Embedding

lec 3 Geometric Intuition of t-SNE

we are trying to do Embedding

d-dim | 2 dim





Preserving distance in neighbourhood.

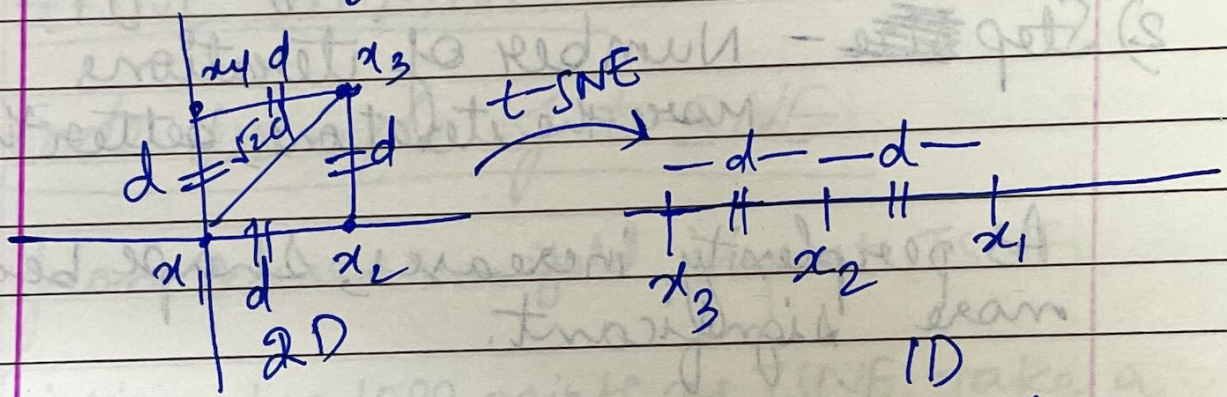
t-SNE - neighbourhood ~~embed~~ preserving embedding

DATE
PAGE

Mathematical formulation

fairly advanced

lec 4 Grouding Problem



Since $x_3 \notin N_d(x_1)$ we don't have to preserve it.

But placing x_4 makes distance b/w x_3 & x_4 as $3d$ which is not preserved.

So, sometimes it is impossible to preserve distances in all the neighbourhood. Such a problem is called Grouding problem.

DATE
PAGE

lec5 How to apply t-SNE and interpret its output

t-SNE works in iteration.

1) Perplexity – No. of points in the neighbourhood which are getting preserved.

2) Step – Number of iterations
More the iterations, better the sol^n

As perplexity increases, shape becomes more significant.

Always run t-SNE with multiple perplexity values.

If perplexity = Total datapoints you get a messy figure.
⇒ So perplexity < No. of datapoints

⇒ Always run iterations till the shape doesn't change much.

Stochastic means probabilistic

DATE
PAGE

(*) t-SNE expands dense clusters
" shrinks sparse clusters

t-SNE does not preserve distance b/w clusters.

lec6 t-SNE on MNIST

lec7 Code example of t-SNE

```
from sklearn.manifold import TSNE

# picking top 1000 points as TSNE take a lot of time for 15K points
data_1000 = standardized_data[0:1000, :]
label_1000 = labels[0:1000]
model = TSNE(n_components=2, random_state=0)
```

DATE
PAGE

Configuring the parameters

the no. of ~~points~~ components = 2
default perplexity = 30
default learning rate = 200
default Max. no. of iteration for the optimization = 1000

```
tsne_data = model.fit_transform(
                    data_1000)
```

Creating a new data frame which helps us in plotting the result data

```
tsne_data = np.vstack((tsne_data.T,
                    labels_1000)).T

tsne_df = pd.DataFrame(data=
                    tsne_data,
          columns=("Dim_1", "Dim_2",
                    "label"))
```

DATE
PAGE

Plotting the result of tsne

```
sns.FacetGrid(tsne_df, hue="label"
,size=6).map(plt.scatter,"Dim1",
"Dim2").add_legend()
plt.show()
```