EE660 Oct 31, 2018

Discussion 10

Today:
- Project comments
- Kernels (note on tf-idf at last page)

(check DNN related seminar at EEB Nov 29)
(confirm)

Project

1. Groups - be sure the workload is adequate
2. Used methods & techniques: at least 50% from EE 660
   - non-trivial feature extraction or preprocessing
   - "obvious" DNN problem
   - Recommender systems

3. Data handling

- EDA must be done on separate pre-training set

histogram, heat maps etc

- Remember to keep test set untouched till the end.

- CV vs. validation

If D huge ~~on~~ remember CV will take a lot of time

4. Few rejected proposals : resubmit and e-mail

5. Final report

Be sure to summarize main findings, ~~an~~ clearly state problem and goals, ~~try to~~ analyze results and try to justify choices

—//—

Kernels (Murphy 14)

In SVM, Kernels "naturally" appear.
Let's get a more general view:

. What is a Kernel?

Kernel functions are usually seen as a measure of similarity between vectors. In this case; for $x_1, x_2 \in X$

$K(x_1, x_2) \in \mathbb{R}$

$K(x_1, x_2) = K(x_2, x_1)$

$K(x_1, x_2) \geq 0$

. Some examples: (Murphy 14.2)

a) Gaussian Kernel

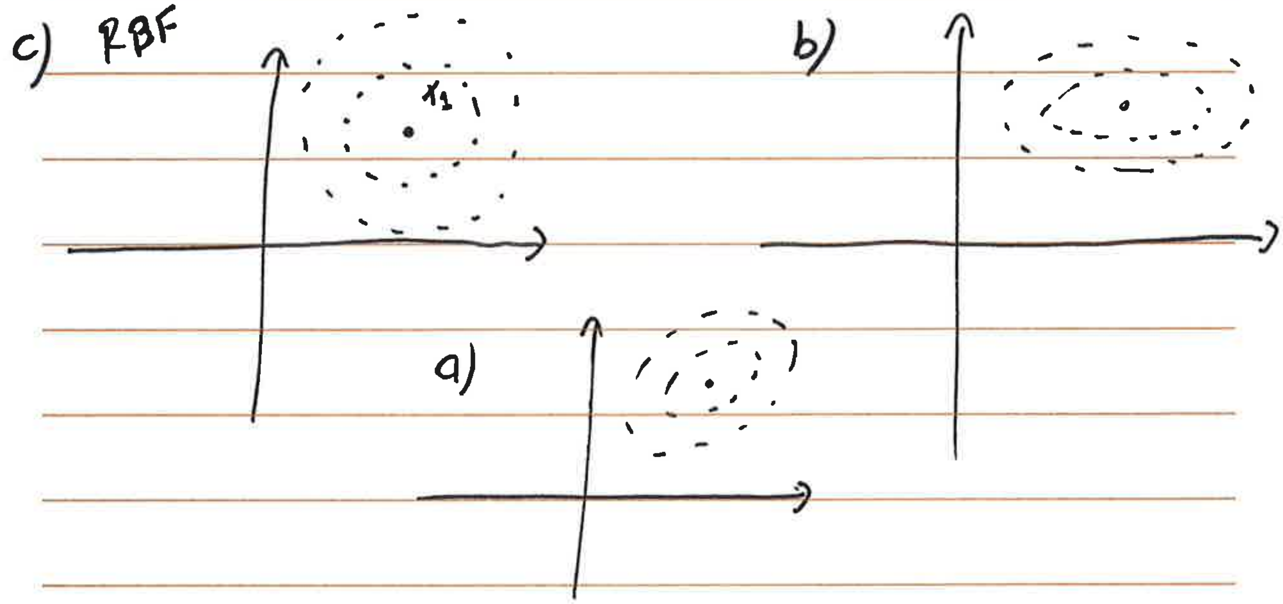
$$K(x_1, x_2) = \exp\left(-\frac{1}{2}(x_1 - x_2)^T \Sigma (x_1 - x_2)\right)$$

b) If Σ is diagonal

$$K(x_1, x_2) = \exp\left(-\frac{1}{2}\sum_{j=1}^{D}\frac{(x_{1,j} - x_{2,j})^2}{\sigma_j^2}\right)$$

c)- If $\sigma_j = \sigma \ \forall j$, RBF:

$$K(x_1, x_2) = \exp\left(-\frac{1}{2}\frac{\|x_1 - x_2\|^2}{\sigma^2}\right)$$



For comparing documents

. Cosine ~~sima~~ similarity

$$K(x_1, x_2) = \frac{x_1^T x_2}{\|x_1\|_2 \|x_2\|_2}$$

x_i is a bag of words vector

. Term frequency inverse document frequency (TF-IDF) i is the document; j is the word

$tf(x_{ij}) = \log(1 + x_{ij})$ (not unique definition)

$$idf(j) = \log \frac{N}{1 + \sum_{i=1}^{N} \mathbb{I}(x_{ij} > 0)}$$

✱ at end of document

indicator function

. Kernelized Feature Vector

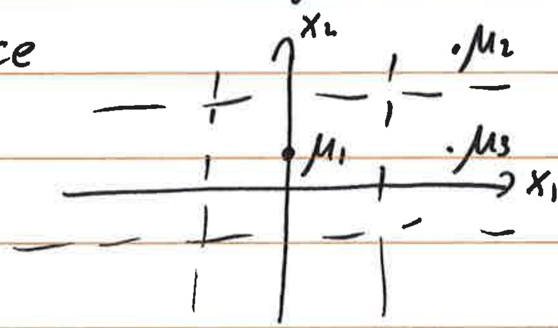
$\phi(x) = [K(x, \mu_1) \ldots \cancel{K(x, \mu_K)}]$

$K(x, \mu_K)$

μ_K are centroids

How to choose μ_K?

. Low dimensional input space: create "tiles" in input space



. Find clusters.

Main cons: still have to choose K

clustering is unsupervised, might not yield good results

. Make each x_i a prototype

$\phi(x) = [K(x, x_1) \ldots K(x, x_N)]$

$D' = N \Rightarrow$ now we need sparsity-promoting method. Most common $l_1 \rightarrow$ L1VM

7)

$$\text{tf-idf}(x_i) = \begin{bmatrix} tf(x_{i1}) \cdot idf(1) \\ \vdots \\ tf(x_{iM})\, idf(M) \end{bmatrix}$$

Then use cosine similarity.