Lecture #14

Unsupervised methods: no "training data", i.e. labelled examples from particular classes

ex: - k-means (often well-separated, compact clusters)
- DBSCAN: look for connected high density regions
- Spectral clustering: approximate the best graph cut

Supervised learning: Some (perhaps a lot) of labelled instances are available

Problems: • regress a function $f(x)$ given samples $\{(x_i, f(x_i))\}_{i=1}^n$.
• build a classifier $C(x) \in \{1, \ldots, K\}$ given labeled examples $\{(x_i, C(x_i))\}$,

• Of course, classification is a special case of regression.
• Many ways to do this; classical topics in statistics.
• We will focus on classification, though similar ideas will apply to regression.
• We will focus on three methods:
1.) Nearest neighbor classification: simple, inelegant, suffers obvious limitations
2.) Support vector machines: more complicated, elegantly handle nonlinear data through the "kernel trick"
3.) Neural networks: extremely complicated, poorly understood theoretically, amazing results in some settings

Remark: Battle between (kernel) SVM and neural networks is ongoing...

Neural networks have the decisive upper hand at the moment.

/

- Let $\{(x_i, y_i)\}_{i=1}^n$ be labelled data — $x_i \in \mathbb{R}^D$
  — $y_i \in \{1, \dots, K\}$

- The nearest neighbor classifier sets for a new point $x \in \mathbb{R}^D$
  the label $C(x) = \text{mode}\{x_i^{NN}\}_{i=1}^{K_{NN}}$, where 1) K_{NN} is some integer ≥ 1
  2) $\{x_i^{NN}\}_{i=1}^{K_{NN}}$ are the K_{NN} nearest neighbors of x, i.e.
  $\{x_i^{NN}\}_{i=1}^{K_{NN}}$ are the K_{NN} minimizers of $\|x - x_i\|_2$.

- This is simple in the important way: nothing to do before assigning a label — "lazy" classifier

ex:

If $K_{NN} = 4 \Rightarrow C(x) = 2$
$K_{NN} = 8 \Rightarrow C(x) = 1$

- Choosing K_{NN} is important, but it's the only parameter.
- It can, moreover, be <u>learned</u> through a general process in supervised learning called cross validation.

Indeed, one can take the training data $\{x_i\}_{i=1}^n$ and split it into a training and validation set.

- For KNN-classification, the whole training set can be used for cross-validation: choose k that maximizes accuracy on the training set, say by testing x_i on $\{x_j\}_{j \neq i}$ for all $i = 1, \dots, n$ (leave one out cross-validation).

- Remark: While simple (and often effective), KNN classification suffers from at least two serious problems.

Problem 1: Computation. Suppose we fix $x \in \mathbb{R}^D$. Who among the n training points are the k_{NN}-Nearest neighbors to x? Well, for each x_i need to compute the ℓ^2 distance in $\mathbb{R}^D$. Need to do this k_{NN} times $\rightsquigarrow O(n \cdot D \cdot k_{NN})$.

- Indexing structures (kd-trees, cover trees) attempt to improve on this.

- If you want $k_{NN} \to \ln(n)$, it may be faster to just sort the pairwise distances: $O(Dn \log(n))$.

Problem 2: Curse of Dimensionality. KNN-classification

tries to use pairwise distances to make decisions. So, there should be some information in the pairwise distances for this to work.

- But, in high dimensions, pairwise distances are uninformative!
- Intuitively, for the unit ball (volume = 1) or $[0,1]^D$, most of the volume concentrates near the boundary.

- Exercise (hard): Let $B_r^D(0) = \{x \in \mathbb{R}^D \mid \|x\|_2 \le r\} = B_r^D$ be the radius-r sphere in $\mathbb{R}^D$. Then the surface area of B^D is

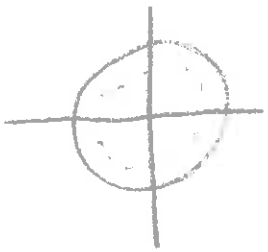
$$SA(B_r^D) = \frac{D\pi^{\frac{D}{2}} r^{D-1}}{\Gamma(\frac{D}{2}+1)}, \quad \Gamma(z) = \int_0^\infty t^{z-1} e^{-t}\, dt$$

- So, to go from surface area to volume, integrate over r:

$$\mathrm{Vol}(B_r^D) = \int_0^r \frac{D\pi^{\frac{D}{2}}}{\Gamma(\frac{D}{2}+1)} \rho^{D-1}\, d\rho$$

$$= \frac{\pi^{\frac{D}{2}}}{\Gamma(\frac{D}{2}+1)} \int_0^r D\rho^{D-1}\, d\rho$$

- So, how much volume between the spheres of radius 1 (unit sphere) and the sphere of radius $(1-\varepsilon)$? in $\mathbb{R}^2$:



$$\text{vol}(B_1^2) - \text{vol}(B_{1-\varepsilon}^2)$$
$$= \pi 1^2 - \pi(1-\varepsilon)^2$$
$$= \pi\left[1-(1-\varepsilon)^2\right]$$

(P) vol = area
SP = perimeter in $\mathbb{R}^2$...

As a proportion of the volume of B_1^2:

$$\frac{\text{vol}(B_1^2) - \text{vol}(B_{1-\varepsilon}^2)}{\text{vol}(B_1^2)}$$
$$= 1-(1-\varepsilon)^2$$
$$= 2\varepsilon - \varepsilon^2$$

- How about in $\mathbb{R}^D$? $\text{vol}(B_1^D) - \text{vol}(B_{1-\varepsilon}^D)$

$$= \frac{\pi^D}{\Gamma\left(\frac{D}{2}+1\right)} \int_{1-\varepsilon}^{1} D\rho^{D-1}\, d\rho$$
$$= \frac{\pi^D}{\Gamma\left(\frac{D}{2}+1\right)}\left[1-(1-\varepsilon)^D\right]$$

As a proportion of total volume:

$$\frac{vol(B_1^D) - vol(B_{1-\varepsilon}^D)}{vol(B_1^D)}$$

$$= 1 - (1-\varepsilon)^D$$

- So, if $D = 100$, $\varepsilon = \frac{1}{2}$, then $1 - \left(\frac{1}{2}\right)^{100} \approx 1$.
- This concentration of measure of phenomenon makes KNN bad when D is large, without a lot of samples.