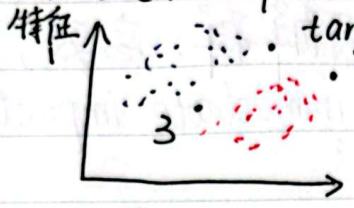
No.

Date . .

KNN : K-Nearest Neighbors

— Duck test: If it looks like a duck, swims like a duck, and quacks like a duck, then it probably is a duck

考虑一个Dataset:



two classes

那么新样本 target1 是 blue(B) or red(R)? 直观上是R，因为它距离红点"更近点"；target 2 & 3 呢？也许就要多看自己身边的样本点"们" label 是啥。

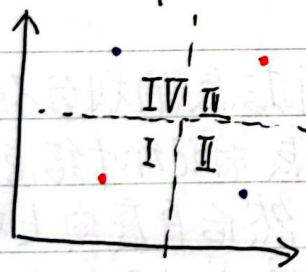
上一段有几个透露的关键：

① "近" => Distance Metric！如何衡量样本间在特征空间上的距离？Emperically：

Euclidean: $d(x, x') = \sqrt{\sum_{d=1}^{D} (x_d - x_d')^2}$

Manhattan: $d(x, x') = \sum_{d=1}^{D} |x_d - x_d'|$

② "们" => 看身边哪些点？Intuitively, k-nearest samples! KNN 算法 idea 便呼之欲出。



在左例中，黑线划分 Feature Space 为4份，构成 Voroni Diagram。划分依据：e.g., 在I中的点，身边最近的一个点是红点；III中是红。(KNN, k=1)

可视构成 Voroni Diagram 的过程为 training，i.e., training 是无error的，因为 $d(x, x) = 0$；在 Test 时，对于一个新 sample，找 K-Nearest Points，取 labels 然后 majority vote 决定它的 label

No.

Date

既然有 majority vote，在 Binary Classification 中，我便希望 ties 不会发生，因此可设 k 为奇数。当然也可：① 再看额外一个点 ② 移去 k 个中最远的 ③ 距离 ⇒ 权重 ④ another distance metric

在 K-NN 中，Distance 为欧式距离 $d: \mathbb{R}^M \times \mathbb{R}^M \to \mathbb{R}$

$$d(u,v) = \sqrt{\sum_{m=1}^{M} (u_m - v_m)^2}$$

当然其他 Distance 也可以。那么这个 algorithm complexity 为？

设 $x \in \mathbb{R}^M$，N samples for training，则：

train time: $O(1)$

Predict time (one sample): $O(MN)$ ↓ on average

可用 k-d tree 优化，则上述两者变为：$O(MN\log N)$；$O(2^M \log N)$

在 M 大时，GG！实践中常用 stochastic approximation

那么 KNN Performance 如何呢？有 Theoretical Guarantee：

(by Cover & Hart, 1967)：$h(x)$ 为 KNN ($k=1$) 的二分类器：

$$\text{error}_{\text{true}}(h) < 2 \times \text{Bayes Error Rate}.$$

BER 可近似理解为：the best you could probably do

☆ Inductive Bias：(归纳偏差) 是算法在进行归纳学习时所持有的先验假设。如决策树的 Inductive Bias 就是尝试找到 (最小的) 决策树 s.t. 训练误差↓且互信息↑

Occam's Razor：try to find the 'simplest' classifier that explains the training dataset

那么 KNN 的 Inductive Bias 便有：{ 1. similar point → similar label; 2. 所有维度 are created equally }

第2点 ⇒ 有一个大问题：特征不同的 scale 对 Distance 有影响！

因此假设应舍第2点；当然也可人为定义权重以纠正

KOKUYO

No.

Date . .

(Voroni Diagram中)

之前提到k的影响：那么一个重要现象是：k↑，boundary越平滑；ppt中的例子十分直观！

Extension: Model Selection & Experiment Design

一个model可以设很多不同 setting：如KNN，k=?；Decision Tree中，criterion？因此与其说是一个"model"，不如是"一类"model.

setting 有参数，也有超参。model training 的是为了什么？是当前 model structure、hyperparameter下的最优 parameters.

因此model selection，就是在众多 model中挑最好，而众多model不同的是超参！如何挑超参呢？需要实验！

Training exp: input: 超参 & T Data; output: Best parameters

Hyper exp: input: T & V Data; Output: Best hyperparameters

关于Training Exp，有Cross-Validate特殊技巧：E.g., Data Fold [1,2,3]，则可[1,2]训[3]验，[2,3]训[1]验 … 它们Loss的和可作为为实验设计训练Loss的重要参考！

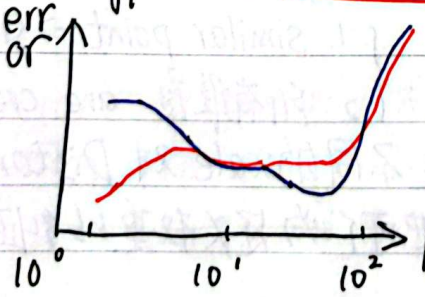
挑超参作实验规范流程：

① Data ⇒ [train] [val] [test]. 挑出来

② [train]上训，更换hyper；[val]上测的最好的hyper

③ 以这个hyper，在 [train]+[val] 上训，用[test]看表现

Eg: K-NN:



超参选择实验：Grid！

α, β, … 所有可能取值作组合一一实验。也可Random选α, β, 重复多次。

△：有限时间内，Random比Grid更有可能找到好的hyper，尤其高维空间中

No.

Date

那么：如果 data 并不 linearly separable 呢？那么 perceptron 不会收敛。但有定理发现：在 examples 上训一遍，依然能得出近似边界

Theorem：设 $\langle (x_1, y_1), \cdots, (x_m, y_m) \rangle$，$\|x_i\| \le R$，令 u 为任一单位向量，$\gamma > 0$. 定义每一个 example deviation 为：

$$d_i = \max\{0, \gamma - y_i(u \cdot x_i)\}$$

再定义：$D = \sqrt{\sum_{i=1}^{m} d_i^2}$，则 Perceptron 在这个集上 mistakes 数量：

$$\le \left(\frac{R+D}{\gamma}\right)^2$$

在 w^* 上投影　　点到 w^* 平面最短距离

* Proof: Claim1: $w_{t+1} \cdot w^* \ge w_t \cdot w^* + \gamma$

w^* 为 max-margin 权重向量，且 $\|w^*\| = 1$

Claim2: $\|w_{t+1}\|^2 \le \|w_t\|^2 + R^2$，因为 $w_{t+1} = w_t \pm x$，$\|x\| \le R$

则 $w_{M+1} \cdot w^* \ge \gamma M$，$\|w_{M+1}\| \le R\sqrt{M}$ 且 $w_{M+1} \cdot w^* \le \|w_{M+1}\|$

$\therefore \gamma M \le R\sqrt{M}$，$M \le \left(\frac{R}{\gamma}\right)^2$

KOKUYO