Model Zoo 3

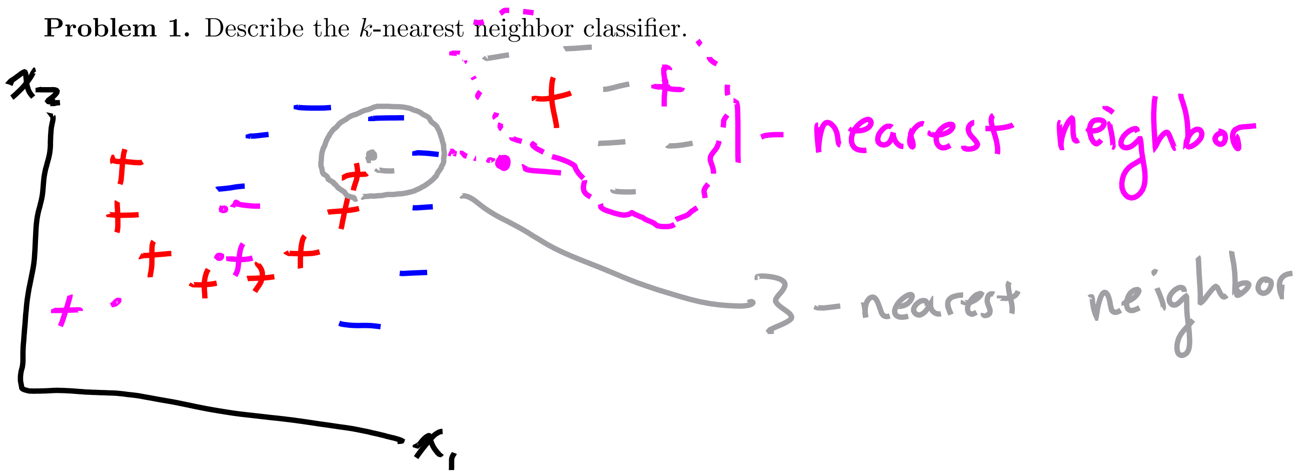
The textbook has detailed explanations of all of the models we've covered in the model zoo in the online supplements located at

`https://amlbook.com/eChapters.html`

The online supplements are all password protected with password `Paraskavedekatriaphobia`. For the most part, these online supplements are technical and provide relatively little insight into practical applications. The section on nearest neighbor algorithms (e-Chapter 6), however, is relatively readable.

Nearest Neighbor Methods

Problem 1. Describe the k-nearest neighbor classifier.



$k \uparrow \Rightarrow$ noise is less of a problem

Runtime

Brute force: runtime $\Theta(dN)$

scikit learn: ball tree, kdtree $O(2^d N)$

clever data structure:

goal: each search take time $O(\log N)$

practice: $O(2^d \log N)$

curse of dimensionality

Fact 1. The in sample error of 1-nearest neighbor is always 0.

$\Rightarrow$ $d_{vc} = \infty$

$\Rightarrow$ $|E_{in} - E_{out}| = O\left(\sqrt{\frac{d_{vc}}{N}}\right)$

is useless for nearest neighbor

Theorem 1 (informal)**.** Let h be the 1-NN hypothesis. Then for $\boxed{\text{“well behaved” data distributions,}}$ we have with high probability that

$$\boxed{E_{\text{out}}(h) \leq 2E_{\text{out}}(f)} + \boxed{4\sqrt{d}\,\boxed{N^{-\frac{1}{d+1}}}} \quad \text{exp. in } d \tag{1}$$

If h is the k-nearest neighbor classifier, then with high probability,

$$\boxed{E_{\text{out}}(h) \geq (1 + 1/k)E_{\text{out}}(f).} \tag{2}$$

Proof. See Theorem 19.3 of *Understanding Machine Learning: From Theory to Algorithms.* □

① $E_{out}(h) \approx 2\,E_{out}(f)$

if E_{out} is very small $\Rightarrow$ 1NN might be good

if " large $\Rightarrow$ 1NN definitely bad

② gen err $= \sqrt{\frac{d_{vc}}{N}}$ ~ linear $= d$

$N^{-\frac{1}{d+1}} = \varepsilon$

$\underbrace{N = \left(\frac{1}{\varepsilon}\right)^{d+1}}$

exponential in d

curse of dimensionality

As $d\uparrow \Rightarrow N\uparrow$ exponentially

practice rule of thumb

$d \lesssim 10 \Rightarrow$ okay

$d > 10 \Rightarrow$ very concerned

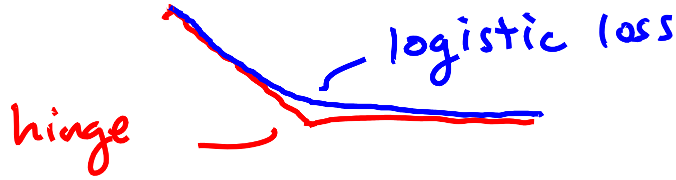
Problem 2. Recall that the ImageNet dataset has an estimated Bayes Error around 5%. (This estimate follows from Karpathy's observed human error rate of 5% and the observed labelled errors from the automatic collection process.) What is the best out of sample error we can expect from a 1-NN classifier?

$\approx 10\%$

Problem 3. Recall that the MNIST digit classification dataset has $N = 60000$ training image, and each image has $d = 28 \times 28 = 784$ dimensions. There are 13 known mislabeled images. (You can see them at `https://labelerrors.com/`.)

1. Why 1-NN is a bad choice of algorithm for this dataset?

2. Would it make more sense to use PCA or the polynomial feature map on this dataset before performing 1-NN?

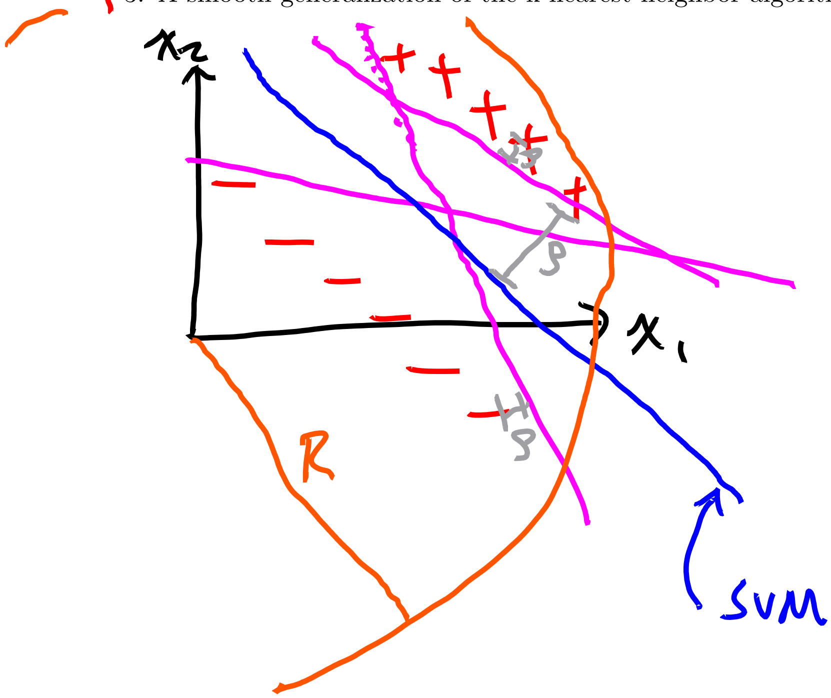


Support Vector Machines (SVMs)

convex, surrogate, most robust

There are three standard and equivalent interpretations of SVMs:

1. The hypothesis class $\mathcal{H}_{\text{perceptron}}$ with the hinge loss $\ell(\mathbf{x}; \mathbf{w}) = \max(1 - \mathbf{w}^T\mathbf{x}, 0)$.
2. The large margin classifier.
3. A smooth generalization of the k-nearest neighbor algorithm.



all valid perceptrons PLA might return any of these lines

Note 1. The following theorem is copied directly from Theorem 8.5 in Chapter 8 of the *Learning from Data* textbook. Most of this chapter is more technical than needed for this course, but the explanation of the theorem is straightforward and worth reading.

Theorem 2. Suppose the input space is the ball of radius R in $\mathbb{R}^d$. That is, $\|\mathbf{x}\|_2 \leq R$. Then,

$$d_{\text{VC}} \leq \boxed{\lceil R^2/\rho^2 \rceil + 1.} \quad \text{no dep. on } d \tag{3}$$

SVM is a linear classifier, it also satisfies $d_{VC} \leq d+1$

Problem 4. Describe the *dual learning problem* and the *kernel trick*.

Common sample kernel functions include:

kernel name	$K(\mathbf{x}_1, \mathbf{x}_2)$	feature dimensions ($\tilde{d}$)
linear	$\mathbf{x}_1^T \mathbf{x}_2$	d
polynomial	$(\gamma \mathbf{x}_1^T \mathbf{x}_2 + r)^Q$	$\Theta(d^Q)$
gaussian	$\exp(-\gamma \\|\mathbf{x}_1 - \mathbf{x}_2\\|_2^2)$	∞
sigmoid	$\tanh(\gamma \mathbf{x}_1^T \mathbf{x}_2 + r)$	∞