LECTURE NOTES

PROF. ALAN YUILLE

1. Non-Parametric Learning

In previous lectures, we described ML learning for parametric distributions – in particular, for exponential models of form $p(x|\lambda) = (1/Z[\lambda]) \exp\{\lambda \cdot \phi(x)\}$.

In this lecture we describe non-parametric methods.

We have already seen two non-parametric methods for learning distributions from a dataset $\mathcal{X} = \{x_i : i = 1, ..., N\}$:

(1) The empirical distribution $f(x) = (1/N) \sum_{i=1}^{N} I(x = x_i)$.

(2) The histogram. Divide the domain of x Ω (or $\mathbf{X}$) into B sub-domains $\Omega = \bigcup_{b=1}^{B} \Omega_b$, with $\Omega_i \bigcap \Omega_j = \emptyset$ if $i \neq j$ (e.g. each data point x lies in one, and only one, sub-domain). Examples of these sub-domains, or bins, are shown in figure (1). Then we represent the distribution by:

$$p(x \in \Omega)b) = n_b/N, \quad \text{where } n_b + \sum_{i=1}^{N} I(x_i \in \Omega_b).$$

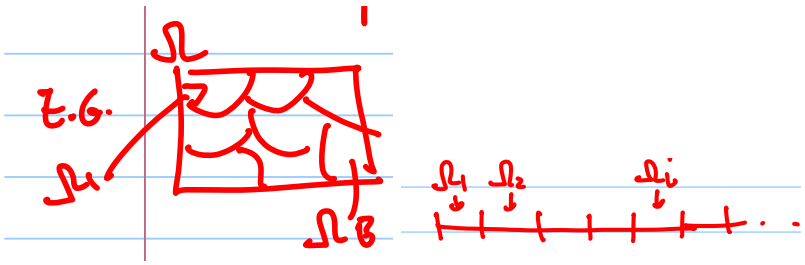

FIGURE 1. Examples of histogram. In one dimension (left) and in two dimensions (right)

Note: technically the histogram can also be thought of as a parametric model. The indicators $\sum_{i=1}^{N} I(x_i \in \Omega_b)$ are the sufficient statistics, and the counts n_b/N are the statistics of the data.

In general, non-parametric models are expressed in form:

$$p(\underline{x}) = \frac{1}{n}\sum_{i=1}^{n} w_n(\underline{x} - \underline{x}_i)$$

where $w_n(.)$ is a window function.
For example, we recover the empirical distribution by setting $w_n(\underline{x} - \underline{x}_i) = I(\underline{x} = \underline{x}_i)$.

2. Why non-parametric models?

It is hard to develop parametrized probability models for some data.

Example: estimate the distribution of the annual rainfall in the U.S.A.

Goal - model $p(x, y)$ - the probability that a raindrop falls at a position (x,y) (E.G. low in the Mojave desert, high in Hawaii).

It is hard to see how to model a multi-model distribution like this, see figure (2).

- Intuitively

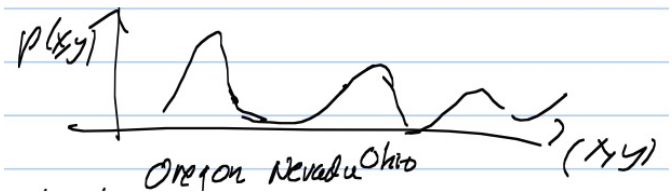

Figure 2. The distribution of rainfall in the US will have many modes. For example, you except high rainfall (peaks) in Oregon and Ohio. But extremely low rainfall (valleys) in Nevada.

(But see later lectures on models with hidden variables.)

3. Intuition for window function

We need to make some assumptions about the distribution – or else, we are stuck with the empirical distribution (which just memorizes the data and is terrible at generalizing). This is an ill-posed problems. Parametric methods address it by assuming a specific form for the distribution (i.e. the parametric model). Non-parametric models address it by assuming that the distribution is spatially smooth, see figure (3).

Note: this "smoothness" requirement relates to functional approximation (more about this in later lectures).

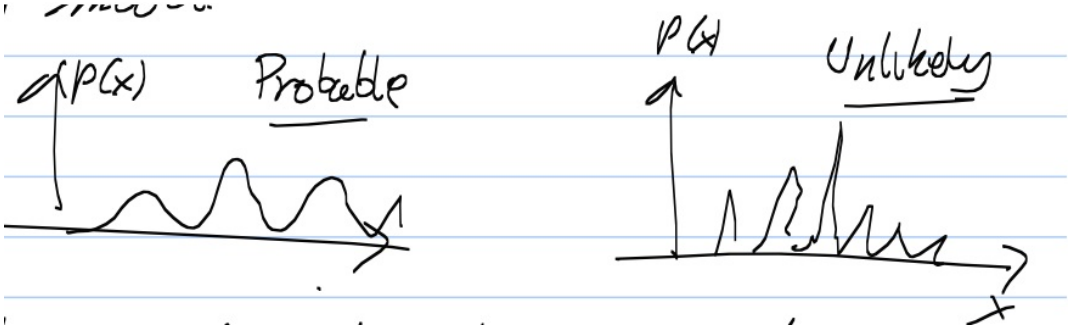

Figure 3. Window methods assume that smooth probability distribution (left) are more likely than jagged distributions (right). More pragmatically, if you do not assume smoothness then you may not have enough data to estimate the distribution.

Method 1 : Windows based on points $\underline{x}$ in space.

For each point $\underline{x}$, form a window centered on $\underline{x}$ with volume V_n. Count the number of samples K_n that fall in the window, see figure (**??**).

Estimate the probability density to be $p_n(x) = \frac{K_n}{nV_n}$. Note: this assumes smoothness at the scale of the window.

E.g., $K_n = 3, V_n = \pi r^2$, in figure (**??**) where r is the radius of the circle.

4. Strategy for Non-Parametric Models

We want to design methods that converge to the correct distributions – i.e. the unknown distribution $p(x)$ that generates the observed dataset $\mathcal{X} = \{x_1, ..., x_N\}$ – as the number of samples $N \mapsto \infty$. This requires *asymptotic consistency*. It is unclear if this is really

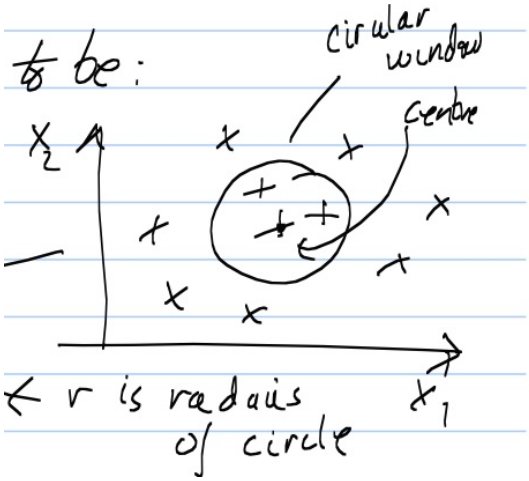

FIGURE 4. Count the number of data points within a window radius r centered on x.

necessary, because we never have enough data to get close to this asymptotic regime, but it is a reasonable requirement (and one that can be studied).

Goal is to design a sequence of windows $V_1, \ldots, V_n$ so that at each point x, $p_n(x) \to p(x)$ as $n \to \infty$ (recall, n is the no. samples) (and $p(x)$ is the true distribution).

Conditions for window design:

(i) Increasing spatial resolution

$\lim_{n\to\infty} V_n = 0$

(ii) Many samples at each point

$\lim_{n\to\infty} K_n = \infty$, (provided $(p(x) \neq 0)$

(iii) $\lim_{n\to\infty} \frac{K_n}{n} = 0$

i.e. K_n grows slower than n

LECTURE NOTES

5. Two Design Strategies: I Parzen Windows

There are two commonly used design strategies for non-parametric methods.
(A) Parzen Windows :

Fix the window size : $V_n = \frac{1}{\sqrt{n}}$ (same size for each point in space).

(B) K-NN :

: Fix no. samples in window (adaptive) : $K_n = \sqrt{n}$

(A) Parzen Window

uses a window function $\phi(\underline{u})$

s.t. $\phi(\underline{u}) \geq 0, \int \phi(\underline{u}) \delta \underline{u}) = 1.$

Examples :

(i) Unit hypercube : $\phi(\underline{u}) = 1$, if $|\underline{u}| < \frac{1}{2}$ and $\phi(\underline{u}) = 0$, otherwise.

(ii) Gaussian in d-dimensions.

$\phi_d(\underline{u}) = \frac{1}{(2\pi)^{d/2}} e^{-\frac{\underline{u}^T \cdot \underline{u}}{2}}$

No. of samples in the hypercube centered on x is $K_n = \sum_{i=1}^{n} \phi(\frac{\underline{x} - \underline{x}_i}{h_n})$ (i.e. we weight the samples), where h_n is the scale factor.

The volume is $V_n = h_n^d$

The estimated density is:

$$p_n(\underline{x}) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{V_n} \phi(\frac{\underline{x} - \underline{x}_i}{h_n})$$

6. Parzen Window Example: Gaussian window

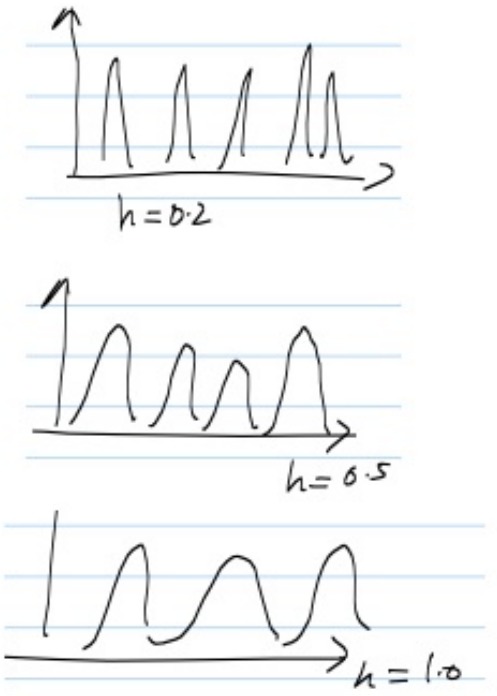

FIGURE 5. Suppose we have data samples from the distribution with three modes in figure (2) and we try to estimate it using Parzen windows. Then if h is too small – e.g. $h = 0.2, 0.5$ – then we get a poor reconstruction with too many nodes. We get a better result with a bigger window $h = 1.0$.

The true distribution has three modes, see figure (2). Suppose we have a few samples from this distribution and try to learn the model using Parzen windows. If h is too small we get a bad reconstruction with too many modes, but for larger h we get a better reconstruction – see figure (5).

For small h (=0.2,1.0), the Parzen window is too small and yields a distribution with too many modes. But a bigger h is not always better. If we use an h which is too big, then we may estime a distribution with a single mode only.

Parzen Window Convergence Theorem

$\lim_{n\to\infty} P_n(x) = P(x)$ (True Density)

Hence the Parzen window estimator converges to the true density at each point x with increasing no. of samples.

Comment: it is good to have consistency as $n \to \infty$, but behavior for small n is more important in practice because we only have a finite amount of data.

7. Proof of Convergence Theorem

Parzen density estimate is $P_n(\underline{x}) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{\sqrt{n}}\phi(\frac{\underline{x}-\underline{x}_i}{h_n})$

This is a random variable which depends on the observed samples $\mathcal{X} = \underline{x}_1, \ldots, \underline{x}_n$ from $P(\underline{x})$.

$\hat{P}_n(\underline{x}) = E\{P_n(\underline{x})\} = \frac{1}{n}\sum_{i=1}^{n} E\{\frac{1}{V_n}\phi(\frac{\underline{x}-\underline{x}_i}{h_n})\}$

$= \int d\underline{y} P(\underline{y}) \frac{1}{V_n}\phi(\frac{\underline{x}-\underline{y}}{h_n})$

As $n \to \infty \quad \frac{1}{V_n}\phi(\frac{\underline{x}-\underline{y}}{h_n}) \to \delta(\underline{x}-\underline{y})$

*$\delta(\underline{x}-\underline{y})$: Dirac delta function

So, $\lim_{n\to\infty} E\{P_n(\underline{x})\} = P(\underline{x})$, which is asymptotic consistency.

To complete the proof, we must show that the variance of the estimate of $P_n(\underline{x})$ tends to zero as $n \to \infty$

$\bar{\sigma_n}^2 = \sum_{i=1}^{n} E\{(\frac{1}{nV_n}\phi(\frac{\underline{x}-\underline{x}_i}{h_n}) - \frac{1}{n}\hat{P}_n(\underline{x}))^2\}$

$= \frac{1}{n}\{E\{(\frac{1}{V_n^2}\phi^2(\frac{\underline{x}-\underline{x}_i}{h_n})\} - (E\{P_n(\underline{x})\})^2\}$

$= \frac{1}{nV_n}\int \frac{1}{V_n}\phi^2(\frac{\underline{x}-\underline{y}}{h_n})p(\underline{y})d\underline{y} - \frac{1}{n}(E\{P_n(\underline{x})\})^2$

$\leq sup \frac{\phi(.)\bar{P}_n(\underline{x})}{nV_n} \rightarrow 0, n \rightarrow \infty$ (note: the second to third line is only true as $n \mapsto \infty$.

8. Parzen Window in Practice

In practice, we do not have an infinite number of samples. The choice of window shape and size is important. It interpolates the data.

How to estimate the best size h? Split the dataset into two. Estimate the distribution using h on half of the dataset and evaluate it on the other half. Pick the value of h which gives the best evaluation. This is an example of cross-validation.

If the window shape and size fits the local structure of the true probability density, then Parzen windows are effective. See figure (6).

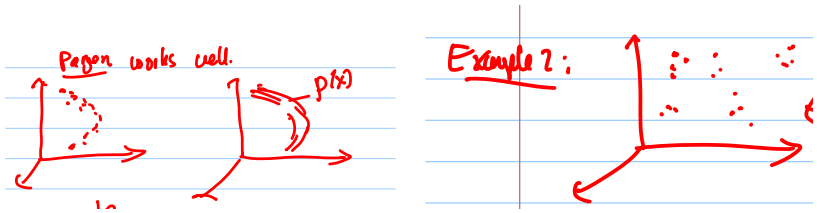

Figure 6. Parzen windows are good for some types of distributions (left) but not for others (right).

9. Two Design Startegies II: K-Nearest Neighbors

K Nearest Neighbors allows the windows to adapt in size to the data. It fixes the number of samples inside each window, so the window size depends on the local density of the data.
$K_n = \sqrt{n}, V_n = V_n(\underline{x})$- so the size is a function of $\underline{x}$

$P_n(\underline{x}) = \frac{K_n/n}{V_n} = \frac{1}{V_n(\underline{x})\sqrt{n}}$

Advantages & Disadvantages.

Pro's

The adaptive size of the window means that $P_n(\underline{x})$ will never be zero. This is an advantage in high dimensions.

Con's

There may be great variability in the size of the windows. Also the distribution may not be normalized.

E.g., for n=1, $P_n(\underline{x}) = \frac{1}{2|\underline{x}-\underline{x}_1|}$, so $P_n(\underline{x})$ is not normalizable. $P_n(\underline{x})$ will remain unnormalizable as the number of samples increases.

10. The Nearest Neighbor Decision Rule

Now consider the decisions which result from learning distributions using a non-parametric model with nearest neighbors.

Suppose we have n data samples $X = \{\underline{x}_1, \ldots, \underline{x}_n\}$ and c classes, $\omega_1, \omega_2, \ldots, \omega_c$, with n_i samples in class ω_i. $\sum_{i=1}^{c} n_i = n$

The likelihood function $p(x|\omega_i)$, or conditional distribution, for the data x (conditioned on the classes) are: $P(\underline{x} \mid \omega_i, \mathcal{X}) = \frac{k_i/n_i}{V_n}$.

Here there are a total number $k = \sum_i k_i$ samples in each window. A window V_n at $\underline{x}$ contains k_i samples in class ω_i.

The *prior* $p(\omega_i|\mathcal{X}) = n_i/n)$ is the normalized frequency of each class.

We assume that the loss function penalizes all errors equally. In this case, Bayes Decision Theory reduces to maximum a posteriori (MAP) estimation.

The posterior probability is:

$P(\omega_i \mid \underline{x}, \mathcal{X}) = \frac{P(\underline{x}|\omega_i,\mathcal{X})P(\omega_i\mathcal{X})}{\sum_{i=1}^{c} P(\underline{x}|\omega_j,X)P(\omega_j\mathcal{X})}$

By substituting for the likelihood and the prior we obtain:

$$P(\omega_i \mid \underline{x}, \mathcal{X}) = k_i/k$$

Hence Bayes Decision Rule reduces to taking the majority vote of the k nearest neighbors.

$$\omega^*(x) = \arg\max_i \{k_1, \ldots k_c\}$$

11. The k-Nearest Neighbor Classifier

The previous section showed that we could go directly from learning the distributions (using k-nn non-parametric model) to an intuitive decision rule. So we could bypass the distribution, and go directly to the k-NN decision rule.

This gives the nearest neighbor NN decision rule.

Partition the space into c disjoint subspaces:

$\Omega = \cup_{i=1}^{c} \Omega_i, \quad \Omega_i \cap \Omega_j = \emptyset, i \neq j$
(The Ω_i's need not be simply connected)

NN decision rule:

Let $\{(\underline{x}_1, \omega(\underline{x}_1)), \ldots, (\underline{x}_n, \omega(\underline{x}_n))\}$ be the labelled samples.

$\omega_{NN}(\underline{x}) = \omega(\underline{x}^*)$, where $\underline{x}^* = \arg_j \min\{|\underline{x} - \underline{x}_j| : j = 1, ..., n\}$

($\omega(\underline{x}^*)$ is the class of $\underline{x}^*$).

12. Partitioning the space: Voronoi Diagram

NN-partitions (with $N = 1$) the space into a Voronoi diagram, where each sample $\underline{x}_i$ occupies a cell, see figure (7).

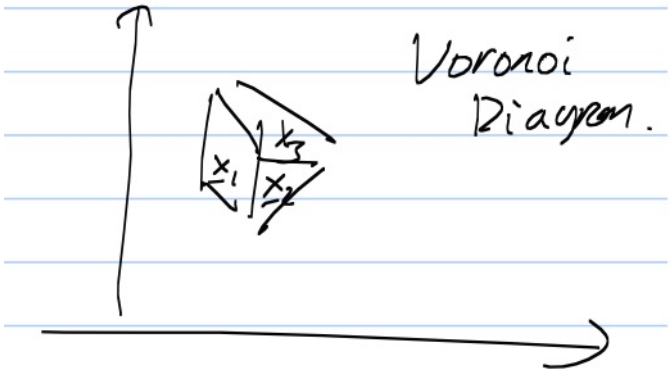

FIGURE 7. Each sample occupies a cell consisting of the set of points which are closest to it. All points in the cell will be classified by the label of the sample.

The NN decision rule is very intuitive. It labels an unknown point ? by the label of the closest data point, see figure (8).

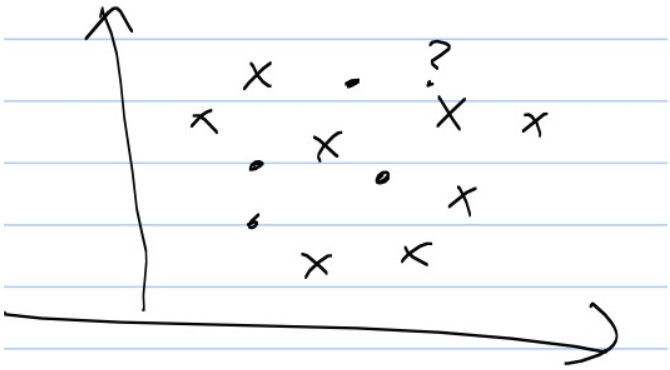

FIGURE 8. An unknown point ? is assigned the label of the closest data point.

To improve NN - go to k-nearest neighbors k-NN. For $\underline{x}$, assign the labels that is most common of the k-nearest samples. Find $k_1, \ldots, k_c$ s.t. $\sum_{i=1}^{c} k_i = k$ counts of nearest samples in each class. Find $j = \arg_i \max k_i$. Set $\hat{\omega}(x) = \omega_j$.

13. Asymptotic Analysis of NN

For large N, the performance of NN can be calculated. It is worse than the optimal Bayes classifier by a fixed amount.

Let $P_n(e \mid x)$ be the error rate at x based on a NN classifier with n samples.

Then $P_n(e \mid x) = \int P_n(e, x^* \mid x)dx^* = \int P_n(e \mid x^*, x)P(x^* \mid x)dx^*$

Where x^* is the point in the samples which is closest to x. x^* is a random variable which depends on the samples, so we must average over $p(x^* \mid x)$.
As $n \to \infty$ $\quad P(x^* \mid x) = \delta(x - x^*)$, so the nearest sample to x is arbitrarily close.

Now

$P_n(e \mid x^*, x) = 1 - \sum_{i=1}^{c} P(\omega_i \mid x^*)P(\omega_i \mid x)$

(an error occurs if x^*& x have different labels.)

We can write

$P_n(e \mid x) = \int\{1 - \sum_{i=1}^{c} P(\omega_i \mid x^*)P(\omega_i \mid x)\}P(x^* \mid x)dx^*$

$\lim_{n\to\infty} P_n(e \mid x) = \int[1 - \sum_{i=1}^{c} P(\omega_i \mid x^*)P(\omega_i \mid x)]\delta(x - x^*)dx^*$

$= 1 - \sum_{i=1}^{c} P^2(\omega_i \mid x)$

The expected error rate is:

$$P = \lim_{n\to\infty} \int P_n(e \mid x)P(x)dx = \int\{1 - \sum_{i=1}^{c} P^2(\omega_i \mid x)\}P(x)dx$$

Now we want to bound this error in terms of the best (Bayes) error rate P^*

Claim:

$P^* \leq P \leq P^*(2 - \frac{c}{c-1}P^*)$

To justify this claim,

let $\omega_m = \omega_{Bayes}(x)$, so $P^*(e \mid x) = 1 - P(\omega_m \mid x)$

Write:

$$\sum_{i=1}^{c} P^2(\omega_i \mid x) = P^2(\omega_m \mid \underline{x}) + \sum_{i \neq m} P^2(\omega_i \mid x) = \{1 - P^*(e \mid x)\}^2 + \sum_{i \neq m} P^2(\omega_i \mid x)$$

We bound this by minimizing $\sum_{i \neq m} P^2(\omega_i \mid x)$ subject to the constraint that $\sum_{i \neq m} P(\omega_i \mid x) = P^*(e \mid x)$. This minimization occurs with $P(\omega_i \mid x) = \frac{P^*(e|x)}{c-1}$, for all i.

Hence

$$\sum_{i=1}^{c} P^2(\omega_i \mid x) \geq (1 - P^*(e \mid x))^2 + \frac{{P^*}^2(e \mid x)}{c-1}$$

which implies $1 - \sum_{i=1}^{c} P^2(\omega_i \mid x) \leq P^*(e \mid x)\{2 - \frac{cP^*(e|x)}{c-1}\}$. The claim follows after integrating.(using $\int \{P^*(e \mid x)\}^2 P(x)dx \geq \{\int P^*(e \mid x)P(x)dx\}^2$

Comment:

The error bound of NN-rule reaches P^* in two extreme cases :

(1) When $P = P^* = \frac{c-1}{c}$, No information

(2) When $P = P^* = 0$, No uncertainty

14. Performance of k-NN as k increases.

The asymptotic performance of k-NN gets closer to the Bayes Risk as k increases, see figure (9)

But, recall again, that in most situations we do not have an infinite amount of data. Performance for small n is important, but very hard to analyze. In practice, determine the

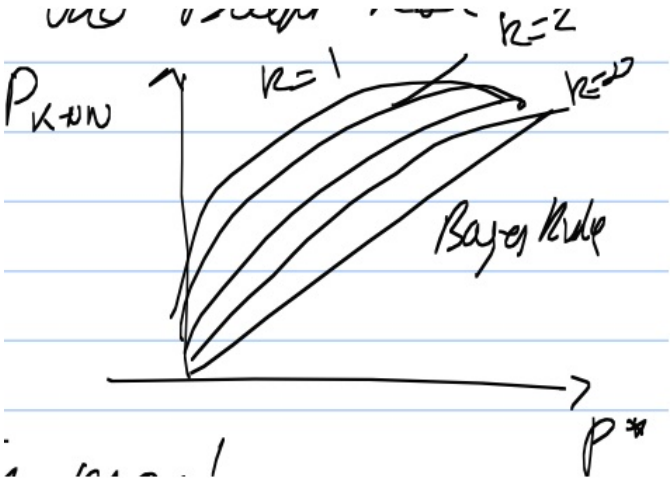

FIGURE 9. The asymptotic performance of k nearest neighbor gets closer to the Bayes Risk as k increases.

best value of k by cross-validation – comparison of performance on training and testing datasets.

15. K-NN for large datasets

When datasets get very large, we need algorithms for rapidly finding the nearest neighbors. These typically represent the dataset hierarchically, so that we can use use a coarse-to-fine strategy to detect the nearest neighbors. Need a reference for this!

16. Distance Measure for NN

What distance measure should be used for nearest neighbors? Here are some examples.

Minkowski :

$$D(x,y) = \langle \sum_{i=1}^{d} \mid x_i - y_i \mid^k \rangle^{1/k}$$

Tanimoto metric for sets, see figure (10) for n_1, n_2, n_{12}.

$D(s_1, s_2) = \frac{n_1+n_2-2n_{12}}{n_1+n_2-n_{12}}$

FIGURE 10. The Tanimoto distance between two sets is a function of the number of points n_1, n_2 in each set, and the number of points n_{12} in the overlap region.

Transform Distance:

Suppose we want to find a distance measure between handwritten digits, see figure (11). If we use a similarity measure based on the positions of points in the images, then the two 5's may be very different and one of the 5's is more similar to the 8.

FIGURE 11. Transform similarity.

One solution is to apply a set of transformations G – e.g. rotation, scaling, translation – to the digits and define a measure $D(x, y) = \min_{a \in G} \|f(x : a) - y\|$, which measures similarity after making the best transformation on the digits.

Alternatively use the Tangent Distance:

$$D(x, y) = \min_{a \in G} \|x + T_a - y\|$$

which is a linear expansion.