ASSIGNMENT 1: [IFT6390]

LÉA RICARD & JOSEPH D. VIVIANO

1. Probabilities

$P(D)$ = the probability of having disease.
$P(T)$ = the probability of having a positive test result.

- 1.5% of women in their 40s have breast cancer, therefore $P(D) = 0.015$.
- 87% true positive rate, therefore $P(T|D) = 0.87$.
- 9.6% false positive rate, therefore $P(T|\neg D) = 0.096$.

We want to know:

$$P(D|T) = \frac{P(T|D)P(D)}{P(T)} \tag{1.1}$$

Which means we need to calculate $P(T)$ which is $P(T|D) + P(T|\neg D) = 0.966$

$$P(D|T) = \frac{0.87 \times 0.015}{0.966} \approx 0.0135 \tag{1.2}$$

2. Curse of Dimensionality

(A) *Consider a hypercube in dimension d with side length c. What is the volume V ?*

In the 2-dimensional case, $area = c^2$. In the 3 dimensional case, $V = c^3$. In the n-dimensional case, $V = c^d$.

(B) *X is a random vector of dimension d ($x \in d$) distributed uniformly within the hypercube (the probability density $p(x) = 0$ for all x outside the cube). What is the probability density function $p(x)$ for x inside the cube? Indicate which property(ies) of probability densities functions allow you to calculate this result.*

For all probability distributions:

$$\int_{-\inf}^{\inf} p(x) = 1 \tag{2.1}$$

Date: Sept 2018.

We know $p(x) = 0$ for all points outside of the hypercube. Therefore, $p(x) = 1$ for x inside the cube (since $1 - 0 = 1$).

(C) *Consider the outer shell (border) of the hypercube of width 3% of c (covering the part of the hypercube extending from the faces of the cube and 0.03c inwards). For example, if c = 100cm, the border will be 3cm (left, right, top, etc ...) and will delimit this way a second (inner) hypercube of side 100 − 3 − 3 = 94cm. If we generate a point x according to the previously defined probability distribution (by sampling), what is the probability that it falls in the border area? What is the probability that it falls in the smaller hypercube?*
Let b be the amount to remove from the border on one side (i.e., left) of the outer hypercube.
$p(x_{large}) = 1$. Therefore in the general case the probability we are in the smaller hypercube is:

$$p(x_{small}) = (c - 2b)^d / c^d \tag{2.2}$$

$$p(x_{border}) = 1 - p(x_{small}) \tag{2.3}$$

Therefore for the above example:

$$p(x_{small}) = (100 - 2 \times 3)^d / 100^d = 94^d / 100^d \tag{2.4}$$

And as before:

$$p(x_{border}) = 1 - p(x_{small}) \tag{2.5}$$

And the probability we are in the border is:

(D) *Calculate the above for* $d = 1, 2, 3, 5, 10, 100, 1000$.

$$1 - 94^1 / 100^1 = 0.06 \tag{2.6}$$

$$94^2 / 100^2 = 0.1163 \tag{2.7}$$

$$94^3 / 100^3 = 0.1694 \tag{2.8}$$

$$94^5 / 100^5 = 0.2661 \tag{2.9}$$

$$94^{10} / 100^{10} = 0.4614 \tag{2.10}$$

$$94^{100}/100^{100} = 0.9980 \tag{2.11}$$

$$94^{1000}/100^{1000} \approx 1(i.e., tresgrand). \tag{2.12}$$

(E) When the dimension grows, the probability that x falls into the narrow border at the edge of the hypercube becomes more likely, which is contrary of our intuitions at lower dimensions.

3. Parametric Gaussian vs Parzen Window Density Estimation

3.1. **Isotropic Gaussian Distribution.** (A) The named parameters are μ, a d-long vector of means, and Σ, a $d \times d$ covariance matrix, where n is the number of data points.

(B)

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i. \tag{3.1}$$

$$\Sigma = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^{\mathrm{T}} \tag{3.2}$$

(C) For the μ parameter, the algorithm complexity is in $\mathcal{O}(n)$, since it is summing over the vectors (x_i).
For the Σ parameter, since the Gaussian density function is isotropic, the algorithm complexity is also in $\mathcal{O}(n)$. μ can be calculated by $\Sigma = \sigma I^2$.

(D)

$$p(x) = \frac{1}{(2\pi)^{fracd2}\sqrt{|\Sigma|}} \mathrm{e}^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)} \tag{3.3}$$

(E) The cost of this operation is $\mathcal{O}(1)$, since this is a straight calculation.

3.2. **Parzen windows with Isotropic Gaussian Kernels.** (A) The only thing learned during training of Parzen Window Density Estimation using Gaussian kernels is σ, since the peak of each Gaussian is each data point. Therefore, if σ is fixed by the user, nothing is learned during training (the Gaussians are simply centered on each training data point).

(B)

$$p(x) = \frac{1}{n}\sum_{i=1}^{n}\mathcal{N}_{x,\sigma}(x) \tag{3.4}$$

Expanded becomes:

$$p(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{(2\pi)^{\frac{d}{2}}\sigma^{d}}\mathrm{e}^{-\frac{1}{2}\frac{d(x_{test},x_{train})^2}{\sigma^2}} \tag{3.5}$$

(C) $\mathcal{O}(n)$, since we have to calculate the distance between x_{test} and all of the n data points x_{train}.

3.3. **Capacity/Expressivitiy.** (A) The Parzen Gaussian is more expressive, because it can store information for every data point. The capacity of the algorithm grows as we give it more data points, this isn't true for the Gaussian distribution, which averages over all data points, so it has a fixed capacity for a given dimensionality, no matter how many training data the algorithm is shown.

(B) Parzen windows with Isotropic Gaussian Kernels, in the case that we used a large number of training examples with a small σ would result in extreme memorization of the noise in the training data (i.e., overfitting).

(C) Because in parametric Gaussian density estimation, σ is learned from the data, while it is fixed for all data points when using Parzen windows.

4. Softmax Activation Function

Université de Montréal
Email address: joseph@viviano.ca