CS 536 - Mini Project 1

Sakib Jalal

February 21, 2018

1 Multivariate Gaussian

1.1

It suffices to show that

$$p(r|d) = \frac{S_d r^{d-1}}{(2\pi\sigma^2)^{\frac{d}{2}}} e^{-\frac{r^2}{2\sigma^2}}$$

because a small thickness $\epsilon << 1$ doesn't change radius r significantly. This probability is equal to the probability density function of the multivariate Gaussian (centered at $\mathbf{0}$) integrated over the shell of radius r. Since the covariance matrix is diagonal, every variable is independent of every other variable, and since all of the covariances are the same (σ^2), the probability distribution is equal at all points on the shell of radius r. Thus, the probability can be taken outside the integrals, and so the probability of being on the shell is equal to the probability of a point drawn from the multivariate Gaussian being on the shell multiplied by the surface area of the shell.

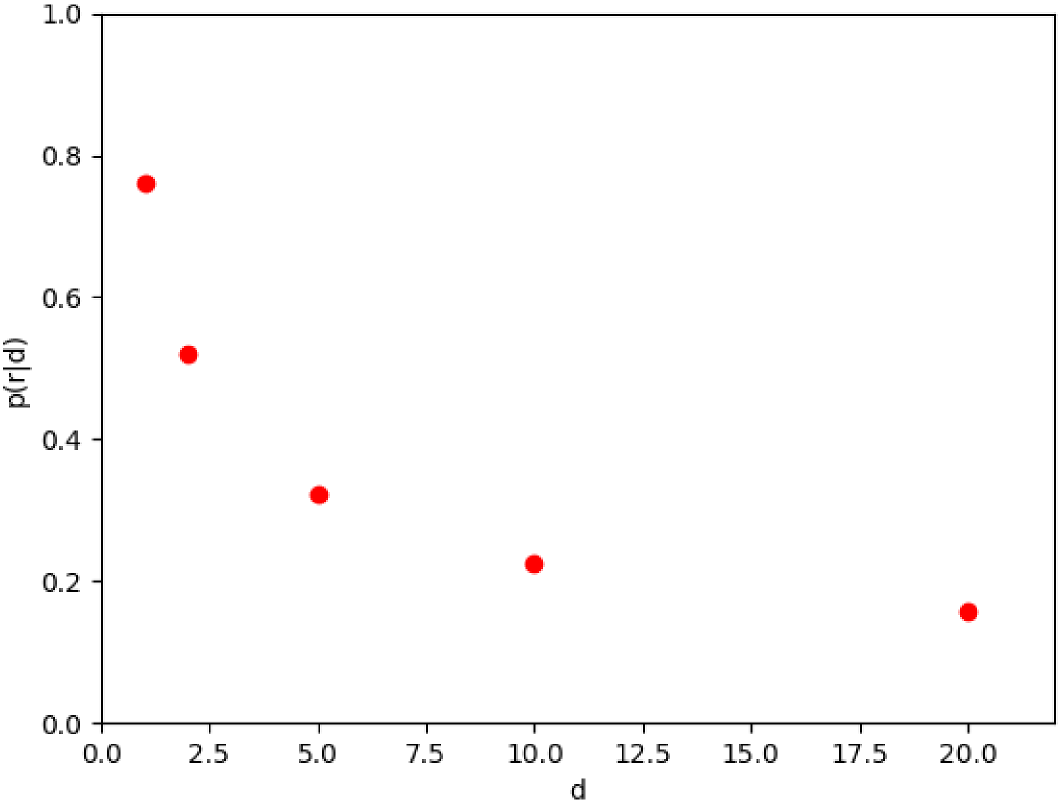
$$\begin{aligned}
p(\mathbf{r}|\mathbf{0}, \sigma^2 I) &= \frac{1}{(2\pi)^{\frac{d}{2}} |\sigma^2 I|^{\frac{1}{2}}} e^{-\frac{1}{2}\mathbf{r}^T(\sigma^2 I)^{-1}\mathbf{r}} \\
&= \frac{1}{(2\pi\sigma^2)^{d/2}} e^{-\frac{r^2}{2\sigma^2}} \\
\int_{x_1} ... \int_{x_d} p(\mathbf{r}|\mathbf{0}, \sigma^2 I) dx_1 ... dx_d &= p(\mathbf{r}|\mathbf{0}, \sigma^2 I) \int_{x_1} ... \int_{x_d} dx_1 ... dx_d \\
&= p(\mathbf{r}|\mathbf{0}, \sigma^2 I) S_d r^{d-1} \\
&= \frac{S_d r^{d-1}}{(2\pi\sigma^2)^{\frac{d}{2}}} e^{-\frac{r^2}{2\sigma^2}}
\end{aligned}$$

1.2

$$\begin{aligned}
\frac{d}{dr} p(r|d) &= \frac{d}{dr} \frac{S_d r^{d-1}}{(2\pi\sigma^2)^{\frac{d}{2}}} e^{-\frac{r^2}{2\sigma^2}} \\
&= \frac{(d-1) S_d r^{d-2}}{(2\pi\sigma^2)^{\frac{d}{2}}} e^{-\frac{r^2}{2\sigma^2}} + \frac{S_d r^{d-1}}{(2\pi\sigma^2)^{\frac{d}{2}}} \frac{-r}{\sigma^2} e^{-\frac{r^2}{2\sigma^2}} \\
&= p(r|d) \Big(\frac{d-1}{r} - \frac{r}{\sigma^2} \Big) \\
&= p(r|d) \Big(\frac{d-1}{\sqrt{d}\sigma} - \frac{\sqrt{d}\sigma}{\sigma^2} \Big) \\
&= p(r|d) \frac{1}{\sigma} \Big(\frac{d-1-d}{\sqrt{d}} \Big) \\
&= p(r|d) \frac{1}{\sigma} \Big(\frac{1}{\sqrt{d}} \Big)
\end{aligned}$$

$$\lim_{d\to\infty} \frac{d}{dr} p(r|d) = 0$$

1.3



1.4

Experiments show that as the dimension d increases, the probability mass in a thin shell away from the center keeps decreasing. This suggests that the density at the origin increases because as you add more dimensions (more univariate Gaussians), the probability density near the center of the hypersphere increases, and as you travel further from the center, the probability of being on the shell falls off faster than the surface area of the shell increases.

2 PAC-learning

The class of concepts consisting of concentric circles can be PAC learned if we demonstrate that there is a PAC learning algorithm for the class. We want to show that $P(error > \epsilon) \leq \delta$ where δ is the upper bound on the failure probability of the error in our learning algorithm exceeding ϵ after observing N data.

Let's say our learning algorithm wants to learn the target concept of a circle centered at $\mathbf{0}$ with radius r. We define a ring R inside this circle with an upper radius r_U and some lower radius r_L: $R = \{(x, y) : r_L^2 \leq x^2 + y^2 \leq r_U^2\}$. If we choose r_L such that the probability of a point landing inside the ring is $\geq \epsilon$, then we will not land inside ring R with probability at most $(1 - \epsilon)$.

Therefore, our generalization error exceeds ϵ if a point doesn't land in ring R. For N training data, our failure probability becomes at most $(1 - \epsilon)^N$. (Note that as ϵ shrinks, the ring must become tighter).

$$
\begin{gathered}
P(error > \epsilon) \leq (1-\epsilon)^N \leq \delta \\
1 - x \leq e^{-x} \\
P(error > \epsilon) \leq e^{-\epsilon N} \leq \delta \\
e^{\epsilon N} \geq \frac{1}{\delta} \\
N \geq \frac{1}{\epsilon} \ln \frac{1}{\delta}
\end{gathered}
$$

For linearly decreasing error probabilities, we note that N grows exponentially larger.

3 Optimal Estimator

$$\frac{d}{dy} E[L_q(t, y(\mathbf{x}))] = \frac{d}{dy} \int_{t,y} |t - y(\mathbf{x})|^q p(t|\mathbf{x}) dt d\mathbf{x} = \int_t |t - y(\mathbf{x})|^q p(t) dt$$

When $y^*(\mathbf{x})$ minimizes $\int_t |t - y(\mathbf{x})|^q$, the generalization error will be minimized.

3.1

$$
\begin{aligned}
\int_t (t - y(\mathbf{x}))^2 dt &= \int_t ((t - \mathbb{E}(t)) + (\mathbb{E}(t) - y(\mathbf{x})))^2 dt \\
&= \int_t (t - \mathbb{E}(t))^2 dt + \int_t 2(t - \mathbb{E}(t))(\mathbb{E}(t) - y(\mathbf{x})) dt + \int_t (\mathbb{E}(t) - y(\mathbf{x}))^2 dt
\end{aligned}
$$

$$\int_t 2(t - \mathbb{E}(t))(\mathbb{E}(t) - y(\mathbf{x})) dt = 2(\mathbb{E}(t) - y(\mathbf{x})) \int_t t - \mathbb{E}(t) dt = 0$$

The first term is always > 0, and the last term is minimized to 0 at $y^*(\mathbf{x}) = \mathbb{E}(t)$.

3.2

$$
\begin{aligned}
|t - y(\mathbf{x})| &= |t - median(t)| + \int_{median(t)}^{y(\mathbf{x})} p(t \leq z) - p(t > z) dz \\
&= |t - median(t)| + \int_{median(t)}^{y(\mathbf{x})} 2p(t \leq z) - 1 dz
\end{aligned}
$$

We note that because $2p(t \leq z)$ is a cumulative density, it never decreases, and also that $2p(t \leq median(t)) - 1 = 0$. Therefore, the whole integral is non-negative, which implies that

$$|t - y(\mathbf{x})| \geq |t - median(t)|$$

3.3

$$\int_t |t - y(\mathbf{x}))^0 p(t|\mathbf{x}| dt = \int_t p(t|\mathbf{x}) dt$$

By definition, $mode(t)$ maximizes $p(t|\mathbf{x})$ over t for each $\mathbf{x}$, so $mode(t)$ minimizes the amount of mismatches between t and $y(\mathbf{x})$, minimizing the generalization error.

4 LDA

With the PCA reduction factor set to .005, we keep 169 features per document and get 88.0% accuracy. With the PCA reduction factor set to .03, we kept 1014 features per document and get 89.8% accuracy, which is slightly improved, implying that much of the data variation was captured with just 169 features.

5 KNN Classifiers

5.1

Ran features.py.

5.2

Train Ratio	K								
	1	**3**	**5**	**7**	**9**	**11**	**15**	**19**	K **Selection**
0.1	.825	.850	.850	.850	.900	.850	.900	.800	15
0.2	.877	.877	.864	.827	.840	.877	.901	.889	15
0.3	.885	.885	.902	.869	.869	.877	.885	.902	19
0.4	.907	.907	.914	.901	.883	.901	.870	.889	5
0.5	.936	.916	.921	.911	.911	.906	.901	.911	1
0.6	.918	.906	.918	.910	.906	.893	.881	.898	5
0.7	.923	.923	.919	.912	.898	.887	.877	.877	3
0.8	.923	.920	.926	.911	.895	.895	.905	.886	5
0.9	.937	.929	.923	.913	.904	.896	.904	.902	1
1.0	.943	.933	.924	.926	.921	.916	.909	.919	1

My rule for selecting K was to first go for the value of K which yielded the highest accuracy, and then, upon ties, I would choose the larger K because larger K's tend to generalize better to unseen datasets. I noted that as the train ratio increased, the best choice of K generally decreased, demonstrating overfitting to the training dataset with the KNN model.

5.3

Train Ratio	K							
	1	**3**	**5**	**7**	**8**	**11**	**15**	**19**
0.1	.754	.755	.770	.769	.771	.763	.765	.773
0.2	.775	.786	.799	.806	.819	.826	.826	.819
0.3	.790	.803	.806	.823	.826	.834	.826	.841
0.4	.805	.814	.813	.826	.841	.836	.834	.843
0.5	.812	.812	.814	.826	.839	.837	.839	.845
0.6	.813	.817	.814	.829	.840	.837	.841	.849
0.7	.826	.831	.829	.845	.842	.842	.841	.845
0.8	.828	.837	.833	.841	.839	.851	.844	.844
0.9	.837	.848	.845	.837	.841	.854	.850	.848
1.0	.840	.846	.854	.848	.848	.851	.861	.857

Based on my test accuracies, it appears that my previous selections for K are not valid anymore, because regardless of the train ratio value, the accuracy seems to almost always increase when K gets larger (approaching 19). I would change the selection of the parameter K to heavily favor larger values of K.

5.4

Based on experimental results, the accuracy of the KNN model is generally higher when the number of training data is smaller, and lower when the number of training data is larger. Also, the accuracies peak with smaller values of K when the training data is small, and they peak with larger values of K when the training data is large. This suggests that the KNN model won't be very accurate as the number of training data balloons, nor would it be time or space efficient because it keeps every data point and parses every data point on every query. Better selection of K in the 5.2 experiment may be conducted by choosing larger values of K even if the accuracy doesn't peak at the larger values.