MGT-448 | **Midterm Project**
SVM and Generative models

Gloria Dal Santo

November 16^{th}, 2020

1 Using SVM for Spam Classification

1.1 SVM with linear kernels

The effect of the parameter C is to control the extent by which the outliers of the training samples are penalized during the classification. Increasing C the boundary becomes more tilted towards the outlier at coordinates $(0.09, 4.10)$, in particular, for $C = 100$ (Fig.1) the boundary classifies the training set perfectly but it is most likely to classify badly different data sets whose elements are closer to it. In conclusion, even tough lower values of C misclassify the outlier, they would set the decision boundary more properly for similar data sets, as it can be seen in Fig.2.

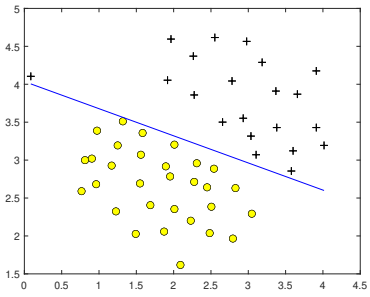

Figure 1: Linear classification, $C = 100$

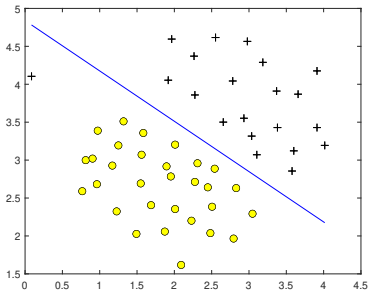

Figure 2: Linear classification, $C = 1$

1.2 SVM with Gaussian kernels

The Gaussian kernel for a pair of samples $(x^{(i)}, x^{(j)})$ is defined as follows:

$$K_g\left(x^{(i)}, x^{(j)}\right) = \exp\left(-\frac{\sum_{k=1}^{n}\left(x^{(i)} - x^{(j)}\right)^2}{2\sigma^2}\right)$$

Also in this case, parameter C controls the cost of misclassification on the training data while σ represents the radius of influence of samples selected by the model as support vectors. For large σ we have that the class of the support vectors have strong influence on deciding the class of the other vectors even if they are far apart. Fig. 3 shows the boundary obtained for $C = 10$ and $\sigma = 0.1$. The obtained result represent a good fit of the data set. For the case with $\sigma = 0.15$, depicted in Fig.4, the region of influence of the support vector relative to the negative examples (yellow dots) includes a larger training set area.

1.3 Spam Classification

To train the SVM for spam classification I set $C = 0.1$. With this parameter, the accuracy that I obtained are 99.83% and 98.9% respectively for 'spamTrain.mat' and 'spamTest.mat' data sets.

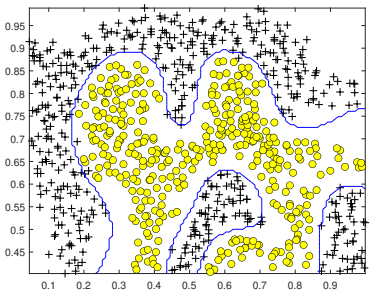

Figure 3: Non linear classification, $\sigma = 0.1$

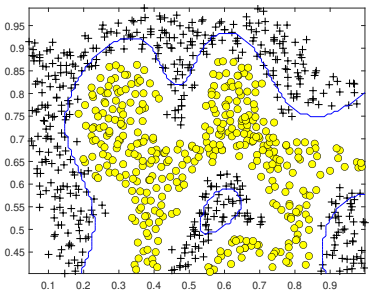

Figure 4: Non linear classification, $\sigma = 0.15$

2 Generative models with model mismatch

2.1 GDA with different covariance matrices

The generative model with Gaussian prior assuming different means and covariance matrices is defined as follows:

$$\begin{aligned} y &\sim \text{Bernoulli}(\phi) \\ x|y=0 &\sim \mathcal{N}(\mu_0, \Sigma_0) \\ x|y=1 &\sim \mathcal{N}(\mu_1, \Sigma_1) \end{aligned}$$

The log-likelihood of the data with the assumed distribution is given by

$$\begin{aligned} \ell(\phi, \mu_0, \mu_1, \Sigma_0, \Sigma_1) &= \log \prod_{i=1}^{N} p(x^{(i)}|y^{(i)}; \mu_0, \mu_1, \Sigma_0, \Sigma_1) p(y^{(i)}; \phi); \\ &= \sum_{i=1}^{N} 1\{y^{(i)} = 0\} \log \left(\frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_0|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} \left(x^{(i)} - \mu_0\right)^\top \Sigma_0^{-1} \left(x^{(i)} - \mu_0\right) \right) \cdot (1-\phi) \right) \\ &+ \sum_{i=1}^{N} 1\{y^{(i)} = 1\} \log \left(\frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_1|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} \left(x^{(i)} - \mu_1\right)^\top \Sigma_1^{-1} \left(x^{(i)} - \mu_1\right) \right) \cdot \phi \right) \end{aligned}$$

Where d is the dimension of x, which in this case is 2, and N is the number of training samples.
To find the maximum likelihood estimate of all parameters, the log-likelihood ℓ must be maximized with respect to each parameter. First, the estimated parameter $\hat{\phi}$ can be found by setting to zero the relative partial derivative of ℓ and solving the equation for ϕ as follows

$$\frac{\partial \ell}{\partial \phi} = \sum_{i=1}^{N} \left(1\{y^{(i)} = 1\} \frac{1}{\phi} - 1\{y^{(i)} = 0\} \frac{1}{1-\phi} \right) = 0$$

which gives

$$\hat{\phi} = \frac{\sum_{i=1}^{N} 1\{y^{(i)} = 1\}}{N} \tag{1}$$

The other parameters can be estimated through the same procedure.
For the means we have

$$\frac{\partial \ell}{\partial \mu_{0/1}} = \sum_{i=1}^{N} \left(1\{y^{(i)} = 0/1\} \cdot \Sigma_{0/1}^{-1} \left(x^{(i)} - \mu_{0/1}\right) \right) = 0$$

which gives the following estimators:

$$\hat{\mu}_0 = \frac{\sum_{i=1}^{N} 1\{y^{(i)} = 0\} x^{(i)}}{\sum_{i=1}^{N} 1\{y^{(i)} = 0\}} \quad \text{and} \quad \hat{\mu}_1 = \frac{\sum_{i=1}^{N} 1\{y^{(i)} = 1\} x^{(i)}}{\sum_{i=1}^{N} 1\{y^{(i)} = 1\}} \tag{2}$$

For the covariance matrices, recalling that $\frac{\partial \log|\Sigma|}{\partial \Sigma} = \Sigma^{-1}$ and $\frac{\partial\left(a^\top \Sigma^{-1} b\right)}{\partial \Sigma} = -\Sigma^{-1} a b^\top \Sigma^{-1}$, we have

$$\frac{\partial \ell}{\partial \Sigma_{0/1}} = \sum_{i=1}^{N} 1\{y^{(i)} = 0/1\} \left(-\frac{1}{2}\Sigma_{0/1}^{-1} + \frac{1}{2}\Sigma_{0/1}^{-1} \left(x^{(i)} - \mu_{0/1}\right) \left(x^{(i)} - \mu_{0/1}\right)^\top \Sigma_{0/1}^{-1} \right) = 0$$

$$\hat{\Sigma}_0 = \frac{\sum_{i=1}^{N} 1\{y^{(i)} = 0\} \left(x^{(i)} - \mu_0\right) \left(x^{(i)} - \mu_0\right)^\top}{\sum_{i=1}^{N} 1\{y^{(i)} = 0\}}$$

$$\hat{\Sigma}_1 = \frac{\sum_{i=1}^{N} 1\{y^{(i)} = 1\} \left(x^{(i)} - \mu_1\right) \left(x^{(i)} - \mu_1\right)^\top}{\sum_{i=1}^{N} 1\{y^{(i)} = 1\}}$$

The parameters estimated with this generative model and trained with the given data, are defined as follows:

$$\begin{aligned} &\hat{\phi} = 0.2850 \\ &\hat{\mu}_0 = [-2.8832 - 0.1274]^\top \quad \hat{\mu}_1 = [2.6390 - 0.3274]^\top \\ &\hat{\Sigma}_0 = \begin{bmatrix} 10.4831 & -8.1718 \\ -8.1718 & 9.8865 \end{bmatrix} \quad \hat{\Sigma}_0 = \begin{bmatrix} 10.4453 & 8.3203 \\ 8.3203 & 10.2377 \end{bmatrix} \end{aligned} \tag{3}$$

2.2 GDA with shared covariance matrix

The generative model with Gaussian prior assuming different means but same covariance matrix for both labels is defined as follows:

$$\begin{aligned} y &\sim \text{Bernoulli}(\phi) \\ x|y = 0 &\sim \mathcal{N}(\mu_0, \Sigma) \\ x|y = 1 &\sim \mathcal{N}(\mu_1, \Sigma) \end{aligned}$$

The log-likelihood of the data with the assumed distribution is given by

$$\begin{aligned} \ell(\phi, \mu_0, \mu_1, \Sigma) &= \log \prod_{i=1}^{N} p(x^{(i)}|y^{(i)}; \mu_0, \mu_1, \Sigma) p(y^{(i)}; \phi); \\ &= \sum_{i=1}^{N} 1\{y^{(i)} = 0\} \log \left(\frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} \left(x^{(i)} - \mu_0\right)^\top \Sigma^{-1} \left(x^{(i)} - \mu_0\right) \right) \cdot (1 - \phi) \right) \\ &+ \sum_{i=1}^{N} 1\{y^{(i)} = 1\} \log \left(\frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} \left(x^{(i)} - \mu_1\right)^\top \Sigma^{-1} \left(x^{(i)} - \mu_1\right) \right) \cdot \phi \right) \end{aligned}$$

The estimator of the Bernoulli parameter $\hat{\phi}$ has the same expression as in eq.1. Also the estimators of the means does not change, and they can be assumed equal to those described in eq.2. For what concerns the covariance matrix, we have

$$\begin{aligned} \frac{\partial \ell}{\partial \Sigma} &= \sum_{i=1}^{N} 1\{y^{(i)} = 0\} \left(-\frac{1}{2}\Sigma^{-1} + \frac{1}{2}\Sigma^{-1} \left(x^{(i)} - \mu_0\right) \left(x^{(i)} - \mu_0\right)^\top \Sigma^{-1} \right) \\ &+ \sum_{i=1}^{N} 1\{y^{(i)} = 1\} \left(-\frac{1}{2}\Sigma^{-1} + \frac{1}{2}\Sigma^{-1} \left(x^{(i)} - \mu_1\right) \left(x^{(i)} - \mu_1\right)^\top \Sigma^{-1} \right) = 0 \end{aligned}$$

which gives

$$\hat{\Sigma} = \frac{1}{N} \sum_{i=1}^{N} \left(x^{(i)} - \mu_{y^{(i)}}\right) \left(x^{(i)} - \mu_{y^{(i)}}\right)^\top$$

where $\mu_{y^{(i)}}$ is the estimated mean value of the label $y^{(i)}$ (i.e. μ_0 if $y^{(i)} = 0$, μ_1 otherwise).
The values of the estimated covariance matrix are

$$\hat{\Sigma} = \begin{bmatrix} 10.4723 & -3.4716 \\ -3.4716 & 9.9866 \end{bmatrix}$$

2.3 Generative model with Laplace distribution

The generative model assuming independent Laplace distribution for each variable is defined as follows:

$$\begin{aligned} y &\sim \text{Bernoulli}(\phi) \\ x_j|y=0 &\sim \text{Laplace}(\mu_j^0, b_j^0)\ j=1,2 \\ x_j|y=1 &\sim \text{Laplace}(\mu_j^1, b_j^1)\ j=1,2 \end{aligned}$$

Assuming that the entries of each feature x are independent, the log-likelihood can be written as follows:

$$\begin{aligned} \ell(\phi,\mu_0,\mu_1,b_0,b_1) &= \log\prod_{i=1}^{N} p(x^{(i)}|y^{(i)};\mu_0,\mu_1,b_0,b_1)p(y^{(i)};\phi) \\ &= \sum_{i=1}^{N} 1\{y^{(i)}=0\}\log\left(p\left(x_1^{(i)}|y^{(i)}=0\right)p\left(x_2^{(i)}|y^{(i)}=0\right)\cdot(1-\phi)\right) \\ &+ \sum_{i=1}^{N} 1\{y^{(i)}=1\}\log\left(p\left(x_1^{(i)}|y^{(i)}=1\right)p\left(x_2^{(i)}|y^{(i)}=1\right)\cdot\phi\right) \\ &= \sum_{i=1}^{N} 1\{y^{(i)}=0\}\log\left(\frac{1}{2b_1^0}\cdot\exp\left(-\frac{|x_1^{(i)}-\mu_1^0|}{b_1^0}\right)\cdot\frac{1}{2b_2^0}\cdot\exp\left(-\frac{|x_2^{(i)}-\mu_2^0|}{b_2^0}\right)\cdot(1-\phi)\right) \\ &+ \sum_{i=1}^{N} 1\{y^{(i)}=1\}\log\left(\frac{1}{2b_1^1}\cdot\exp\left(-\frac{|x_1^{(i)}-\mu_1^1|}{b_1^1}\right)\cdot\frac{1}{2b_2^1}\cdot\exp\left(-\frac{|x_2^{(i)}-\mu_2^1|}{b_2^1}\right)\cdot\phi\right) \end{aligned}$$

Also in this case, it turns out that the estimation of the Bernoulli parameter $\hat{\phi}$ has the same expression as in eq.1. However the estimators of the means now are different and the partial derivative of ℓ is defined as follows:

$$\begin{aligned} \frac{\partial\ell}{\partial\mu_j^{0/1}} &= \frac{\partial}{\partial\mu_j^{0/1}}\left(\sum_{i=1}^{N} 1\{y^{(i)}=0/1\}\left(-\frac{|x_j^{(i)}-\mu_j^{0/1}|}{b_j^{0/1}}\right)\right) \\ &= \sum_{i=1}^{N} 1\{y^{(i)}=0/1\}\text{sign}\{x_j^{(i)}-\mu_j^{0/1}\}=0 \quad j=1,2 \end{aligned} \tag{4}$$

To find the correct estimator we have to separate the case when $\sum_{i=1}^{N} 1\{y^{(i)}=0/1\}=K$ is even and when K is odd. If K is odd, then in order to have $(K-1)/2$ of '1's, $(K-1)/2$ of '-1's from the sign function, $\mu_j^{0/1}$ must be equal to the median of the features whose label is either 0, for μ^0, or 1 for μ^1. On the other hand, if K is even then eq.4 can be minimized by choosing the mean of the two features that approximate the median.
Finally, parameters b_j can be estimated with the following expression:

$$\begin{aligned} \frac{\partial\ell}{\partial b_j^{0/1}} &= \sum_{i=1}^{N} 1\{y^{(i)}=0/1\}\left(\frac{|x_j^{(i)}-\mu_j^{0/1}|}{(b_j^{0/1})^2}-\frac{1}{b_j^{0/1}}\right)=0 \quad j=1,2 \\ &\Leftrightarrow \hat{b}_j^{0/1} = \frac{\sum_{i=1}^{N} 1\{y^{(i)}=0/1\}|x_j^{(i)}-\mu_j^{0/1}|}{\sum_{i=1}^{N} 1\{y^{(i)}=0/1\}} \quad j=1,2 \end{aligned}$$

which corresponds to the mean absolute deviation from the median. The parameters estimated with this generative model and trained with the given data, are defined as follows:

$$\begin{aligned} \hat{\mu}_0 &= [-2.8091\ -0.1443]^\top \qquad \hat{\mu}_1 = [2.3191\ -0.4649]^\top \\ \hat{b}_0 &= [2.6171\ 2.5143]^\top \qquad\quad \hat{b}_1 = [2.5404\ 2.4987]^\top \end{aligned}$$

2.4 Logistic regression

In the logistic regression the posterior probability distribution is of the form

$$p\left(y=1|x;\theta\right)=\frac{1}{1+\exp(-\theta^{\top}\tilde{x})}$$
$$p\left(y=0|x;\theta\right)=1-\frac{1}{1+\exp(-\theta^{\top}\tilde{x})}$$

where $\tilde{x}=[1,x]$ and $\theta\in\mathbb{R}^3$.
To find the parameter θ which maximizes the log-likelihood I applied the stochastic gradient ascent update rule, which can be described by the following algorithm:

Algorithm 1 Stochastic gradient Ascent for LR

procedure GRADIENTASCENTLR($\mathbf{X}, \mathbf{y}, \alpha, \theta$, numIt) ▷ estimated parameters θ
  for `k` $= 1 :$ `numIt` **do**
    for `i` $= 1 : N$ **do**
      $h_\theta = 1/(1+\exp(-\theta^{\top}x^{(i)}))$
      for `j` $= 1 : 3$ **do**
        $\theta_j = \theta_j + \alpha\cdot(\mathbf{y^{(i)}} - h_\theta)\mathbf{x_j^{(i)}}$
      end for
    end for
  end for
  return θ
end procedure

2.5 Results

The classification errors evaluated by testing the models with the given sets of data are summarized in Table 1. From the obtained results it can be seen that the more accurate model is the generative model with gaussian prior and different covariance matrices, that is the one that represents the most the generative model of the given training set. The estimated parameters in eq.(3) are indeed close to those of the given data set. On the other hand, the less accurate is the generative model with independent Laplace distribution, due to the fact that it is the one that differs the most from the probability distribution of the data sets. It can be noticed that the results obtained by applying the generative model with shared covariance matrix and the logistic regression are very similar and they are less accurate than the first one because they try to fit the data approximating the covariance. As we have seen in class, the probability $p(y|x)$ defined through GDA (Gaussian Discriminant Analysis) follows the logistic function, that is

$$p(y=1|x;\phi,\Sigma,\mu_0,\mu_1)=\frac{1}{1+\exp(-\theta^{\top}x)}$$

The similarity between the obtained results is thus a direct effect of this.

Model	Training Error [%]	Testing Error [%]
Generative - GDA I	8.30	9.60
Generative - GDA II	14.40	13.20
Generative - Laplace	17.20	16.40
Discriminative - LR	14.40	13.50

Table 1: Classification error of the tested models