Lecture Notes: Probabilistic Linear Classification

Ted Meeds[1,2]

[1] Informatics Institute, University of Amsterdam
[2] The Centre for Integrative Bioinformatics, Vrije University
tmeeds@gmail.com

Abstract In this note two probabilistic classification models are presented: discriminative models $P(C|\mathbf{x})$, aka **logistic regression**, and generative models $P(C, \mathbf{x})$. Using probabilistic models resolves many of the issues with least-squares classification and has other benefits related to decision theory. Though the models presented are probabilistic, we do not consider them Bayesian because their parameters are set to maximum likelihood estimators and no posterior distribution or predictive distributions are produced.

1 Probabilistic Models for Classification Data

The real problem with least-squares classification is that it makes a Gaussian assumptions for discrete data:

$$p(\mathcal{D}|\mathbf{W}) = \prod_{n=1}^{N}\prod_{k=1}^{K} \mathcal{N}\left(t_{nk}|\mathbf{w}_k^T\mathbf{x}, \sigma_k^2\right)$$

A more sensible approach is to model the class probabilities directly (just like we modeled the real-values conditional probabilities for regression):

$$p(\mathcal{D}|\mathbf{W}) = \prod_{n=1}^{N} P(c_n|\mathbf{x}_n)$$

where c_n is the index of the label turned on in $\mathbf{t}_n$ (we will alternate between representations for labels in this note). This can be rewritten as

$$p(\mathcal{D}|\mathbf{W}) = \prod_{n=1}^{N}\prod_{k=1}^{K} P(\mathcal{C}_k|\mathbf{x}_n)^{[t_{nk}]}$$

where $[\cdot] = 1$ is $\cdot$ is true, 0 other wise. The indicator has *selected* the observed class label. this notation is easier to work with.

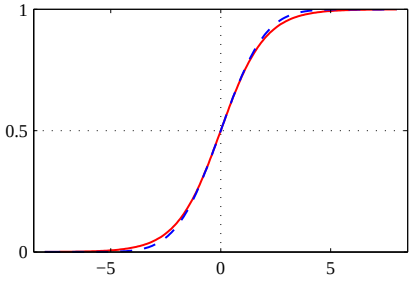
In this note two approaches for modeling $P(c_n|\mathbf{x}_n)$ are described. In this first, a discriminative model is learned directly. In the second, a fully generative model of $P(c, \mathbf{x})$ is learned and then Bayes rule is applied to produce $P(c_n|\mathbf{x}_n)$.

2 Probabilistic Discriminative Models

We will study a specific form for $P(C|\mathbf{x})$ that is called **logistic regression** which assumes a **logistic sigmoid**:

$$P(\mathcal{C}_1|\boldsymbol{\phi}, \mathbf{w}) = \sigma\left(\mathbf{w}^T\boldsymbol{\phi}\right) \tag{1a}$$

$$\sigma(a) = \frac{1}{1+\exp\left(-a\right)} \tag{1b}$$



Some useful properties of the sigmoid:

$$P(\mathcal{C}_1|\boldsymbol{\phi}, \mathbf{w}) = \sigma(\mathbf{w}^T\boldsymbol{\phi}) \tag{2a}$$

$$P(\mathcal{C}_2|\boldsymbol{\phi}, \mathbf{w}) = 1 - \sigma(\mathbf{w}^T\boldsymbol{\phi}) \tag{2b}$$

$$= \sigma(-\mathbf{w}^T\boldsymbol{\phi}) \tag{2c}$$

$$\frac{d\sigma}{da} = \sigma(1-\sigma) \tag{2d}$$

other properties: max of gradient at $1/2$, min value of 0 at $a = -\infty$, max value of 1 at $a = \infty$.

For binary classification, we let $y(\mathbf{x}) = \sigma(\mathbf{w}^T\boldsymbol{\phi})$, i.e. we directly model the class-conditional probability, and $t_n \in \{0, 1\}$ (1 for class 1, 0 for class 2).

$$\begin{aligned} p(\mathbf{t}|\mathbf{w}) &= \prod_{n=1}^{N} p(t_n|\mathbf{x}, \mathbf{w}) \\ &= \prod_{n=1}^{N} y_n^{t_n}(1-y_n)^{1-t_n} \end{aligned}$$

Log-likelihood and therefore MLE is straightforward:

$$\ln p(\mathbf{t}|\mathbf{w}) = \sum_{n=1}^{N} t_n \ln y_n + (1-t_n)\ln(1-y_n)$$

note this is known as the (negative) **cross-entropy** error (objective function).

2.1 Maximum Likelihood

We will maximize the log-likelihood, but equivalently could minimize the cross-entropy.

$$\begin{aligned}
\frac{\partial \ln p(\mathbf{t}|\mathbf{w})}{\partial \mathbf{w}} &= \sum_{n=1}^{N} \frac{t_n}{y_n}\frac{\partial y_n}{\partial a_n}\frac{\partial a_n}{\partial \mathbf{w}} + \frac{(1-t_n)}{(1-y_n)}\frac{\partial(1-y_n)}{\partial a_n}\frac{\partial a_n}{\partial \mathbf{w}} \\
&= \sum_{n=1}^{N} t_n(1-y_n)\boldsymbol{\phi}_n - (1-t_n)y_n\boldsymbol{\phi}_n \\
&= \sum_{n=1}^{N} t_n\boldsymbol{\phi}_n - t_n y_n\boldsymbol{\phi}_n - y_n\boldsymbol{\phi}_n + t_n y_n\boldsymbol{\phi}_n \\
&= \sum_{n=1}^{N} \underbrace{(t_n - y_n)}_{e_n}\boldsymbol{\phi}_n \\
&= \sum_{n=1}^{N} e_n\boldsymbol{\phi}_n
\end{aligned}$$

in solving this we have used $\frac{\partial y_n}{\partial a_n} = y_n(1-y_n)$, $\frac{\partial(1-y_n)}{\partial a_n} = -y_n(1-y_n)$, $\frac{\partial a_n}{\partial \mathbf{w}} = \boldsymbol{\phi}_n$.

2.2 Optimization

Notice the final form of the gradient: 1) the objective decomposes into a sum of terms over the entire data set, 2) the error term is the same as we had for linear regression, but 3) there is no closed form for $\mathbf{w}$ as there was for linear regression ($y = \mathbf{w}^T\mathbf{x}$) since for classification y is a non-linear function ($y = \sigma(\mathbf{w}^T\mathbf{x})$). This means we perform **gradient descent**, or possibly **stochastic gradient descent** (on the cross-entropy or negative log-likelihood).

2.3 Basis Function

Note on $\boldsymbol{\phi}$ versus $\mathbf{x}$: a) as with linear regression, using non-linear basis functions will result in non-linear decision regions in the input space, b) they cannot remove overlapping data (they will overlap in $\boldsymbol{\phi}$-space), and c) using fixed basis functions is limited – we will want to learn them (e.g. using NNs).

2.4 Overfitting

When the data is completely separable (no class overlap), the magnitude of $\mathbf{w}$ will go to ∞, and y becomes a Heavyside step function. Then every value of $P(\mathcal{C}_k|\mathbf{x})$ goes to 0 or 1 for every point. To fix this we need regularization, just like regression.

3 Probabilistic Generative Models

The main idea behind probabilistic models applied to classification is to first build a complete, generative, model of the data (both labels and input) $p(C, \mathbf{x})$, then second to apply Bayes rule to predict $P(C|\mathbf{x})$. As an aside, this is called a *generative* model because we can draw samples $\{t, \mathbf{x}\}$ from the model by first drawing $c \sim P(C)$, then drawing $\mathbf{x}$ $p(\mathbf{x}|C = c)$, something that we cannot do with a discriminative model. One argument for using a generative model versus a discriminative model for classification is that if we can faithfully reproduce the generative process for the data, then this model should be the best possible classifier. On the other hand, if we cannot generate data faithfully, we should probably use a discriminative classifier.

To begin we model the joint distribution using the following factorization $P(C, \mathbf{x}) = P(C)p(\mathbf{x}|C)$. Notice that this factorization is natural for classification since we can build a model of the inputs for each class k, $p(\mathbf{x}|C_k)$, and then weight them by the probability of the class $P(C_k)$.

Next we apply Bayes rule to get the "posterior" over classes:

$$P(C_k|\mathbf{x}) = \frac{P(C_k)p(\mathbf{x}|C_k)}{\sum_j P(C_k)p(\mathbf{x}|C_k)} \tag{3a}$$

For binary classification (K=2) we can rewrite the posterior as a **logistic sigmoid** and compare the result with logistic regression:

$$P(C_1|\mathbf{x}) = \frac{P(C_1)p(\mathbf{x}|C_1)}{P(C_1)p(\mathbf{x}|C_1) + P(C_2)p(\mathbf{x}|C_2)} \tag{4a}$$

$$= \frac{1}{1 + \underbrace{\frac{P(C_2)p(\mathbf{x}|C_2)}{P(C_1)p(\mathbf{x}|C_1)}}_{e^{-a}}} \tag{4b}$$

$$= \sigma(a) \qquad a = \ln\left(\frac{P(C_1)p(\mathbf{x}|C_1)}{P(C_2)p(\mathbf{x}|C_2)}\right) = \underbrace{\ln\left(\frac{P(C_1|\mathbf{x})}{P(C_2|\mathbf{x})}\right)}_{\text{log-odds}} \tag{4c}$$

For multi-class classification (K>3) we can rewrite the posterior in terms of the **softmax**:

$$P(C_k|\mathbf{x}) = \frac{\exp(a_k)}{\sum_j \exp(a_j)} \qquad a_k = \ln(P(C_k)p(\mathbf{x}|C_k)) \tag{5a}$$

(called "softmax" because if $a_k >> a_j$, $\forall j$, then $P(C_k|\mathbf{x}) \to 1$).

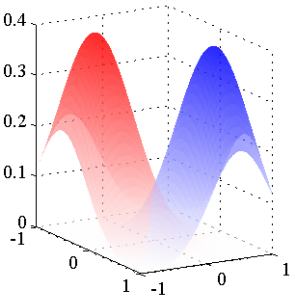
3.1 Gaussian Class-conditional Model

We can build a generative-based classifier on either continuous or discrete inputs $\mathbf{x}$. We will focus on the latter; for discrete models, a popular choice is **naive**

Bayes classifier (see Bishop and HW). The simplest class-conditional model is a Gaussian:

$$P(\mathbf{x}|C_k) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{6a}$$

$$= \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}_k|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right) \tag{6b}$$



We can plug the models for class 1 and class 2 into the logistic sigmoid representation. In particular, we can expand a, which for the case where $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}$ can be written $a = \mathbf{w}^T\mathbf{x} + w_0$. Note we will write $\pi_k = P(C_k)$.

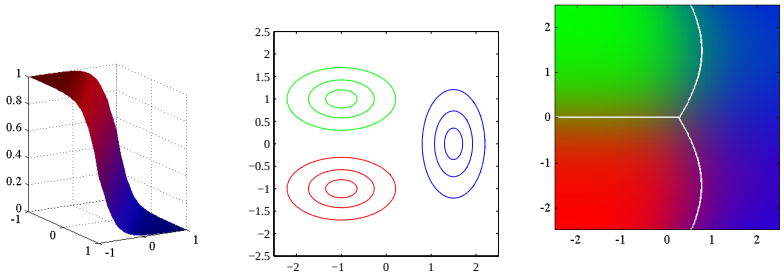
$$a = \ln\frac{\pi_1}{\pi_2} + \ln\frac{\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)}{\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)} \tag{7}$$

$$= \ln\frac{\pi_1}{\pi_2} + \frac{|\boldsymbol{\Sigma}_2|^{1/2}}{|\boldsymbol{\Sigma}_1|^{1/2}} - \frac{1}{2}\left[\mathbf{x}^T\left(\boldsymbol{\Sigma}_1^{-1} - \boldsymbol{\Sigma}_2^{-1}\right)\mathbf{x} - 2\mathbf{x}^T\left(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2\right) + \boldsymbol{\mu}_1^T\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2^T\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2\right] \tag{8}$$

$$= \mathbf{x}^T\left(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2\right) - \frac{1}{2}\mathbf{x}^T\left(\boldsymbol{\Sigma}_1^{-1} - \boldsymbol{\Sigma}_2^{-1}\right)\mathbf{x} + C \tag{9}$$

$$= \mathbf{x}^T \underbrace{\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)}_{\mathbf{w}} + w_0 \tag{10}$$

Where in the last step we assumes equal covariances (and therefore $w_0 = C$). This means that if there are equal covariances, there is a linear separation between classes; however, if we model then separately (as we probably should), the decision boundary will be nonlinear due to the quadratic term $-\frac{1}{2}\mathbf{x}^T\left(\boldsymbol{\Sigma}_1^{-1} - \boldsymbol{\Sigma}_2^{-1}\right)\mathbf{x}$ that does not cancel.



The result generalizes to $K > 2$; if we assume a common covariance, then $a_k(\mathbf{x}) = \mathbf{w}_k^T \mathbf{x} + w_{0k}$, where $\mathbf{w}_k = \boldsymbol{\Sigma}^{-1/2}\boldsymbol{\mu}_k$ and $w_{k0} = -\frac{1}{2}\boldsymbol{\mu}_k^T \boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k + \ln \pi_k$.

3.2 Maximum Likelihood

It can be confusing to keep track of notation for the labels; sometimes we use t_n and sometimes c_n. For binary classification, we can say $c_n \in \{1, 2\}$, corresponding to classes C_1 and C_2. But we will abuse notation and also use C_1 to represent the set of indices from 1 to N where $c_n = 1$ (for example). It is sometimes easier to develop the likelihood using c instead of t at first, then switch to t for convenience.

Our observations are pairs $\{c_n, \mathbf{x}_n\}$ and our likelihood is the density of all pairs and is not a conditional likelihood (as with logistic-regression). The parameters are $\boldsymbol{\theta} = \{\pi_1, \mu_1, \mu_2, \boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2\}$ (we do not need π_2). Let us distinguish between what is the model and what is the likelihood. The model for class labels is $P(C_1) = \pi_1$ and the likelihood for observation c_n is π_{c_n}. The conditional model for any input vector $\mathbf{x}$ for class 1 is $\mathcal{N}(\mathbf{x}|\mu_1, \boldsymbol{\Sigma}_1)$ and the likelihood for an observation input $\mathbf{x}_n$ is the model *selected* by its c_n: $\mathcal{N}(\mathbf{x}_n|\mu_{c_n}, \boldsymbol{\Sigma}_{c_n})$. Therefore the likelihood for an observation pair is $p(c_n, \mathbf{x}_n) = \pi_{c_n}\mathcal{N}(\mathbf{x}_n|\mu_{c_n}, \boldsymbol{\Sigma}_{c_n})$.

For the full set of observations, the likelihood is:

$$p(\mathbf{c}, \mathbf{X}|\boldsymbol{\theta}) = \prod_n \pi_{c_n}\mathcal{N}(\mathbf{x}_n|\mu_{c_n}, \boldsymbol{\Sigma}_{c_n}) \tag{11a}$$

$$= \left(\prod_{n\in C_1} \pi_1\mathcal{N}(\mathbf{x}_n|\mu_1, \boldsymbol{\Sigma}_1)\right)\left(\prod_{n\in C_2} \pi_2\mathcal{N}(\mathbf{x}_n|\mu_2, \boldsymbol{\Sigma}_2)\right) \tag{11b}$$

$$= \prod_n \left(\pi_1\mathcal{N}(\mathbf{x}_n|\mu_1, \boldsymbol{\Sigma}_1)\right)^{t_n} \left((1-\pi_1)\mathcal{N}(\mathbf{x}_n|\mu_2, \boldsymbol{\Sigma}_2)\right)^{1-t_n} \tag{11c}$$

Note: Bishop writes the likelihood as a function of $\mathbf{t}$ and does not include $\mathbf{X}$. Be aware that this really a likelihood over $\mathbf{t}$ and $\mathbf{X}$. Continuing to the log-likelihood (replacing $\mathbf{c}$ with $\mathbf{t}$):

$$\log p(\mathbf{t}, \mathbf{X}|\boldsymbol{\theta}) = \sum_n t_n \log \pi_1 + t_n \log \mathcal{N}(\mathbf{x}_n|\mu_1, \boldsymbol{\Sigma}_1) + (1-t_n)\log(1-\pi_1) + (1-t_n)\log \mathcal{N}(\mathbf{x}_n|\mu_2, \boldsymbol{\Sigma}_2) \tag{11d}$$

This is now in a form with which we can take derivatives and solve for $\boldsymbol{\theta}$.

Class probability $\boldsymbol{\pi_1}$:

$$\frac{\partial \log p(\mathbf{t}, \mathbf{X}|\boldsymbol{\theta})}{\partial \pi_1} = \sum_n \left(\frac{t_n}{\pi_1} + \frac{1-t_n}{1-\pi_1}(-1) \right) \tag{11e}$$

$$0 = (1-\pi) \underbrace{\sum_n t_n}_{N_1} - \pi_1 \underbrace{\sum_n (1-t_n)}_{N_2} \tag{11f}$$

$$\pi_1(N_1 + N_2) = N_1 \tag{11g}$$

$$\pi_1^{\text{MLE}} = \frac{N_1}{N} \qquad \pi_2^{\text{MLE}} = \frac{N_2}{N} \tag{11h}$$

Note for $K > 2$ we need to enforce $\sum_k \pi_k = 1$ by adding a Lagrange multiplier.

Class mean $\boldsymbol{\mu_1}$:

$$\frac{\partial \log p(\mathbf{t}, \mathbf{X}|\boldsymbol{\theta})}{\partial \boldsymbol{\mu}_1} = \sum_n \frac{t_n}{\mathcal{N}(\mathbf{x}_n|\mu_1, \boldsymbol{\Sigma}_1)} \frac{\partial \mathcal{N}(\mathbf{x}_n|\mu_1, \boldsymbol{\Sigma}_1)}{\partial \boldsymbol{\mu}_1} \tag{11i}$$

$$= \sum_n \frac{t_n}{\mathcal{N}(\mathbf{x}_n|\mu_1, \boldsymbol{\Sigma}_1)} \mathcal{N}(\mathbf{x}_n|\mu_1, \boldsymbol{\Sigma}_1) \boldsymbol{\Sigma}_1^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_1) \tag{11j}$$

$$= \sum_n t_n \boldsymbol{\Sigma}_1^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_1) = 0 \tag{11k}$$

$$\sum_n t_n \mu_1 = \sum_n t_n \mathbf{x}_n \tag{11l}$$

$$\mu_1^{\text{MLE}} = \frac{1}{N_1} \sum_n t_n \mathbf{x}_n \tag{11m}$$

and similarly for $\boldsymbol{\mu}_2$. Note this is the empirical mean for class 1.

Class covariance $\boldsymbol{\Sigma_1}$: Assuming separate covariances (Bishop assumes equal covariances).

$$\frac{\partial \log p(\mathbf{t}, \mathbf{X}|\boldsymbol{\theta})}{\partial \boldsymbol{\Sigma}_1} = \sum_n \frac{t_n}{\mathcal{N}(\mathbf{x}_n|\mu_1, \boldsymbol{\Sigma}_1)} \frac{\partial \mathcal{N}(\mathbf{x}_n|\mu_1, \boldsymbol{\Sigma}_1)}{\partial \boldsymbol{\Sigma}_1} \tag{11n}$$

$$= \sum_n \frac{t_n}{\mathcal{N}(\mathbf{x}_n|\mu_1, \boldsymbol{\Sigma}_1)} \mathcal{N}(\mathbf{x}_n|\mu_1, \boldsymbol{\Sigma}_1) \left(-\frac{1}{2} \boldsymbol{\Sigma}_1^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_1) \right) \tag{11o}$$

$$= \sum_n t_n \boldsymbol{\Sigma}_1^{-1} + \frac{1}{2} \boldsymbol{\Sigma}_1^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_1)(\mathbf{x}_n - \boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}_1^{-1} = 0 \tag{11p}$$

$$\sum_n t_n \mu_1 = \sum_n t_n \mathbf{x}_n \tag{11q}$$

$$\boldsymbol{\Sigma}_1^{\text{MLE}} = \frac{1}{N_1} \sum_n t_n (\mathbf{x}_n - \boldsymbol{\mu}_1)(\mathbf{x}_n - \boldsymbol{\mu}_1)^T \tag{11r}$$

and similarly for $\boldsymbol{\Sigma}_2$. Note this is the empirical covariance for class 1.

3.3 Prediction

Given the MLE parameters, we plug them in and use them for any test vector $\mathbf{x}_\star$:

$$P(C_1|\mathbf{x}_\star) = \frac{P(C_1)p(\mathbf{x}_\star|C_1)}{P(C_1)p(\mathbf{x}_\star|C_1) + P(C_2)p(\mathbf{x}_\star|C_2)} \tag{11s}$$

where we have implicitly included $\boldsymbol{\theta}^{\text{MLE}}$ where appropriate.

4 Summary

In this note we have presented two approaches to probabilistic classification: discriminative and generative. When is one appropriate and the other not? As mentioned, if the generative model faithfully represents the input data, then a classifier of this form should be the best (it has much more information about the data than a discriminative model). As a consequence of this, the generative model has many more parameters to set, so it might require much more data to fit these parameters. It also might not be necessary, since logistic regression may perform very well and the extra modeling capacity of a generative model is just wasted on a classification task. A generative model could be used for other tasks, however, such as outlier detection. Neural networks are natural extensions of logistic regression and equivalent extensions of generative models, though possible and exist, are much more computationally expensive and might not provide a big gain – again it depends how good the generative model is. A discriminative model might also be less sensitive to outliers (they will move the generative model, but may not affect a discriminative model). Though the logistic sigmoid forms for logistic regression and equal covariance generative models are the same, the solutions $\mathbf{w}$ and w_0 are not necessarily the same and can be very different.

Other issues to keep in mind: equal covariance assumption (in Bishop) is rarely appropriate, naive Bayes independence assumption never a reality (even though it may work well in practice).