INF8953CE - Fall 2020

Machine Learning

- Sarath Chandar

8. Probabilistic Discriminative Models.

8. Probabilistic Discriminative Models.

In generative modeling,

$$P(C_1|x) = \sigma(w^T x + w_0) \quad [\text{for 2-class problem}]$$

$$P(c|x) = \text{Softmax}(w^T x + w_0) \quad [\text{for k-class problem}]$$

where w and w_0 are functions of mean, variance, and prior probabilities.

Discriminative approach: Assume (w, w_0) as a vector of parameters and learn them directly by using maximum likelihood.

Advantages: 1. Fewer number of parameters than the generative approach.

2. Improved predictive performance when the class-conditional density assumptions give a poor approximation to the true distributions.

Logistic Regression Model :-

$$P(C_1|x) = y(x)$$

$$= \sigma(\omega^T x)$$

$$P(C_2|x) = 1 - P(C_1|x)$$

σ - logistic Sigmoid function.

Note: The model is called logistic regression. But it is used for classification!

Compared to GDA style models, this model has only $M+1$ parameters. If there are more features, then there is a clear advantage in working with the logistic regression model more directly.

Note: Derivative of $\sigma(a)$ is given by

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

Maximum likelihood for logistic regression :-

Consider a dataset $\{x^{(n)}, t^{(n)}\}_{n=1}^{N}$

where $t^{(n)} \in \{0, 1\}$

$$p(t|w) = \prod_{n=1}^{N} (y^{(n)})^{t^{(n)}} (1 - y^{(n)})^{1 - t^{(n)}}$$

where $t = (t^{(1)} \ldots t^{(N)})^T$ and $y^{(n)} = P(C_1 | x^{(n)})$

error function = negative log likelihood (NLL)

$$E(w) = -\ln p(t|w)$$

$$= -\sum_{n=1}^{N} \left\{ t^{(n)} \ln y^{(n)} + (1 - t^{(n)}) \ln (1 - y^{(n)}) \right\}$$

where $y^{(n)} = \sigma(a^{(n)})$ where $a^{(n)} = w^T x^{(n)}$.

$$\frac{\partial E}{\partial w} = \sum_{n=1}^{N} \frac{\partial E}{\partial y^{(n)}} \frac{\partial y^{(n)}}{\partial a^{(n)}} \frac{\partial a^{(n)}}{\partial w} \quad \text{---} (1)$$

$$\frac{\partial E}{\partial y^{(n)}} = -\frac{t^{(n)}}{y^{(n)}} + \frac{1 - t^{(n)}}{1 - y^{(n)}}$$

$$= \frac{(1 - t^{(n)}) y^{(n)} - t^{(n)} (1 - y^{(n)})}{y^{(n)} (1 - y^{(n)})}$$

$$= \frac{y^{(n)} - y^{(n)} t^{(n)} - t^{(n)} + y^{(n)} t^{(n)}}{y^{(n)}(1-y^{(n)})}$$

$$\frac{\partial E}{\partial y^{(n)}} = \frac{y^{(n)} - t^{(n)}}{y^{(n)}(1-y^{(n)})} \quad \text{——} \ (2)$$

$$\frac{\partial y^{(n)}}{\partial a^{(n)}} = \frac{\partial}{\partial a^{(n)}} \sigma(a^{(n)})$$

$$= \sigma(a^{(n)})\left(1 - \sigma(a^{(n)})\right)$$

$$\frac{\partial y^{(n)}}{\partial a^{(n)}} = y^{(n)}\left(1-y^{(n)}\right) \quad \text{——} \ (3)$$

$$\frac{\partial a^{(n)}}{\partial w} = x^{(n)} \quad \text{——} \ (4)$$

Sub. (2), (3), (4) in (1).

$$\frac{\partial E}{\partial w} = \sum_{n=1}^{N} \frac{y^{(n)} - t^{(n)}}{y^{(n)}(1-y^{(n)})} \cdot y^{(n)}(1-y^{(n)}) \cdot x^{(n)}$$

$$= \sum_{n=1}^{N} \left(y^{(n)} - t^{(n)}\right) x^{(n)}$$

The contribution to the gradient from data point 'n'

is given by the error $(y^{(n)} - t^{(n)})$ between the target value and the prediction of the model, times the input vector $x^{(n)}$.

↳ this takes precisely the same form as the gradient of the sum of squares error function for the linear regression model.

Maximum likelihood and least squares:-

Consider linear regression. Assume that the target variable 't' is given by a deterministic function $y(x; w)$ with additive Gaussian noise so that

$$t = y(x; w) + \epsilon$$

where ϵ is a zero-mean Gaussian random variable with precision (inverse variance) β.

$$p(t \mid x, w, \beta) = \mathcal{N}(t \mid y(x; w), \beta^{-1})$$

Now consider a dataset $\{x^{(n)}, t^{(n)}\}_{n=1}^{N}$

$$p(t \mid x, w, \beta) = \prod_{n=1}^{N} \mathcal{N}(t^{(n)} \mid w^T x^{(n)}, \beta^{-1})$$

$$\ln p(t|x,w,\beta) = \sum_{n=1}^{N} \ln \mathcal{N}(t^{(n)} | w^T x^{(n)}, \beta^{-1})$$

$$= \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi) - \beta E_D(w)$$

where $E_D(w) = \frac{1}{2} \sum_{n=1}^{N} \{ t^{(n)} - w^T x^{(n)} \}^2$

Thus maximizing log-likelihood w.r.t. w is equivalent to minimizing $E_D(w)$ w.r.t. w.

Note 1: For squarred loss function, the optimal prediction is given by Conditional mean of the target variable.

$$E[t|x] = \int t\, p(t|x)\, dt$$

$$= y(x; w)$$

The Gaussian noise assumption implies that the conditional distribution of t given x is unimodal, which may be inappropriate for some applications.

Note 2: Failure of least squares for classification can be explained now. Binary target vectors clearly have a distribution that is far from

Gaussian and hence bad performance.

Iterative reweighted least squares (IRLS):

In the case of least squares with linear regression model, we got a closed-form solution.

why? Error function was a quadratic function of the parameters.

In the case of Maximum likelihood for logistic regression, there is no closed-form solution.

why? Error function is a non-linear function which is not quadratic (due to the sigmoid function).

Error function for logistic regression is not quadratic but still a convex function! So there is an unique minimum.

More efficient method than Gradient descent:

Newton-Raphson method.

→ It is also an iterative scheme.

→ It uses a local quadratic approximation to the error function. Specifically, it uses a second-order Taylor series expansion to approximate $E(w)$ near some point w_0 ignoring the derivatives of higher order.

$$E(w) \approx E(w_0) + (w-w_0)^T \nabla_w E(w) + \frac{1}{2}(w-w_0)^T H (w-w_0)$$

where H = Hessian of E w.r.t. w evaluated at w_0.

Differentiate this function and set it to zero.

$$0 + \nabla_w E(w) + (w-w_0)^T H = 0$$

$$(w-w_0)^T H = -\nabla_w E(w)$$

$$(w-w_0)^T = -H^{-1} \nabla_w E(w)$$

$$w = w_0 - H^{-1} \nabla_w E(w)$$

$$\boxed{w^{(new)} = w^{(old)} - H^{-1} \nabla E(w)}$$

Note: If the error function is quadratic fn, then the Taylor series expansion is exact and Newton-Raphson

method will find the solution in one step!

Example: linear regression.

$$E_D(w) = \frac{1}{2} \sum_{n=1}^{N} \left\{ w^T \phi(x^{(n)}) - t^{(n)} \right\}^2$$

$$\nabla E(w) = \sum_{n=1}^{N} \left(w^T \phi^{(n)} - t^{(n)} \right) \phi^{(n)}$$

$$= \Phi^T \Phi w - \Phi^T t$$

$$H = \nabla \nabla E(w) = \Phi^T \Phi$$

Newton-Raphson update rule:-

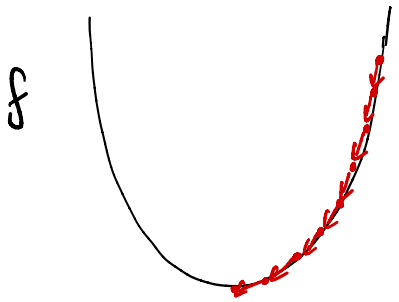
$$w^{(new)} = w^{(old)} - (\Phi^T \Phi)^{-1} (\Phi^T \Phi w^{(old)} - \Phi^T t)$$

$$= w^{(old)} - w^{old} + (\Phi^T \Phi)^{-1} \Phi^T t$$

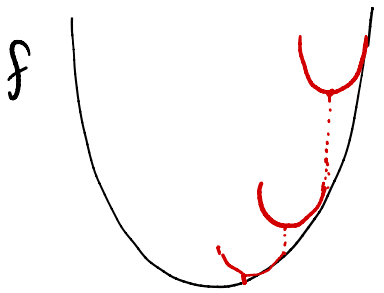
$$w^{(new)} = (\Phi^T \Phi)^{-1} \Phi^T t.$$

which is the standard least squares solution.

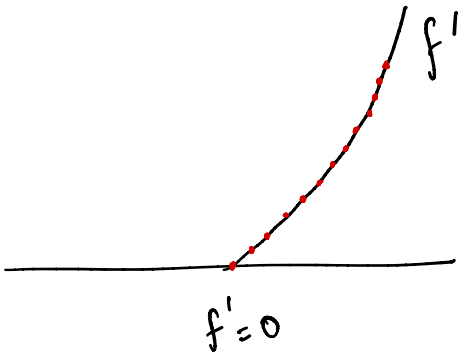
Gradient descent vs. Newton-Raphson : Geometric view



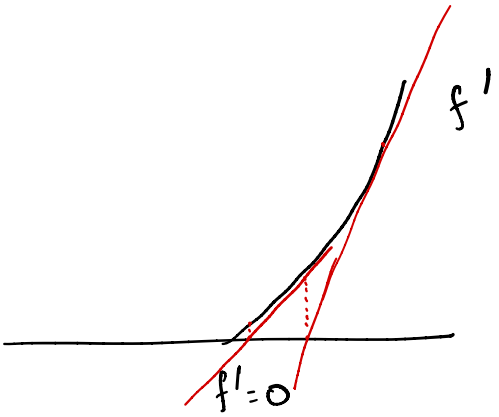
Gradient descent



Newton Raphson.



Gradient descent



Newton Raphson.

Newton-Raphson for logistic regression:-

$$E(w) = -\ln p(t|w) = -\sum_{n=1}^{N} \left\{ t^{(n)} \ln y^{(n)} + (1-t^{(n)}) \ln(1-y^{(n)}) \right\}$$

$$\nabla E(w) = \sum_{n=1}^{N} (y^{(n)} - t^{(n)}) x^{(n)} = X^T (y-t)$$

$$H = \nabla\nabla E(w) = \sum_{n=1}^{N} y^{(n)} (1-y^{(n)}) x^{(n)} x^{(n)T}$$

$$= X^T R X$$

where R is an $N \times N$ diagonal matrix with

$$R_{nn} = y^{(n)}(1 - y^{(n)})$$

Note 1: Hessian is no longer constant but depends on w through the weighting matrix R.

Note 2: $0 < y^{(n)} < 1$ [property of logistic sigmoid]

$\Rightarrow u^T H u > 0$ for arbitrary vector u.

$\Rightarrow H$ is positive definite.

$\Rightarrow$ Error function is a Convex fn. of w.

$\Rightarrow$ error function has unique minimum.

Newton-Raphson update formula:-

$$w^{(new)} = w^{(old)} - (X^T R X)^{-1} X^T (y - t)$$

$$= (X^T R X)^{-1} \left\{ (X^T R X) w^{(old)} - X^T (y - t) \right\}$$

$$w^{(new)} = (X^T R X)^{-1} X^T R z \quad \longrightarrow (1)$$

where z is an n-dimensional vector with elements

$$z = X w^{(old)} - R^{-1}(y - t).$$

Eqn ① looks like a weighted least-squares equation. However, the weight R is not constant but depends on w. So we must apply this update equation iteratively each time using the new 'w' to compute R.

- iterative reweighted least squares (IRLS).

Note: Maximum likelihood can exhibit severe overfitting for data sets that are linearly separable.

why?

max. likelihood solution occurs when the hyperplane corresponding to $\sigma = 0.5$, equivalent to $w^T x = 0$, separates two classes and the magnitude of w goes to infinity.

This problem will happen even if we have more data points than the number of parameters in the model.

How to solve this issue?

① Add regularization

② better estimators (More on this later).

Merits and demerits of discriminant-based, discriminative, and generative models for classification:-

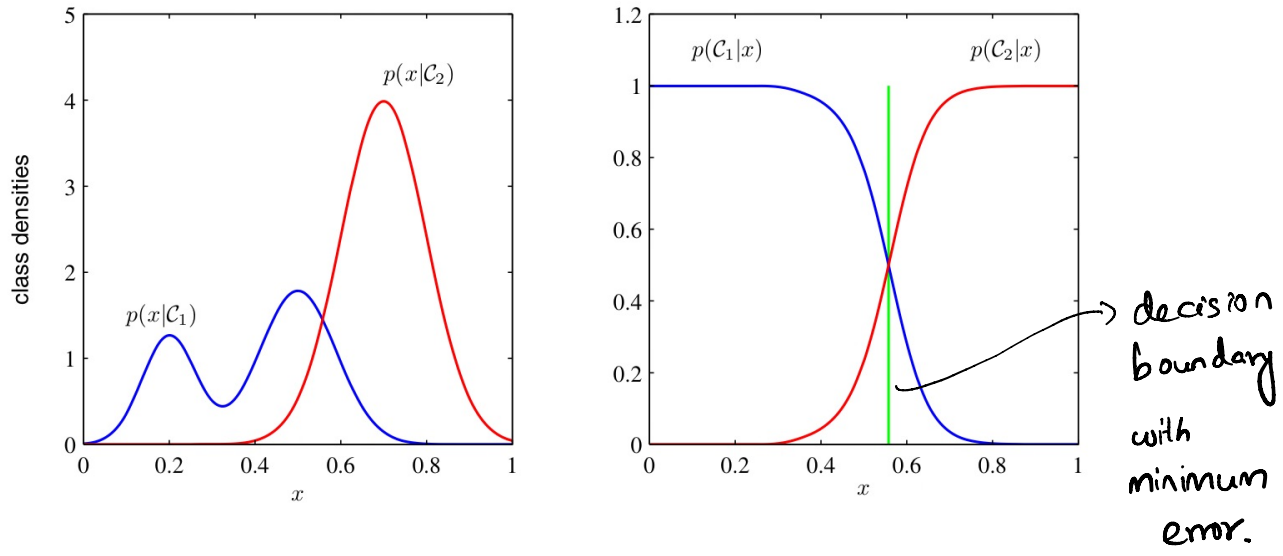
Generative model:

- Most demanding.
- If x is high-dimensional, we need a large training set in order to be able to determine the class conditional densities to reasonable accuracy.

+ we can compute $P(x)$ as $\sum_k P(x|C_k)\,P(C_k)$.
This can be useful for detecting new data points that have low probability under the model, for which prediction may be of low accuracy.
↳ Outlier detection.

Generative vs. discriminative:-

If we only wish to make classification decisions, we can just use discriminative models. Class conditional densities may have lot of structure that has little effect on the posterior probabilities.



Left hand mode of $p(x|C_1)$ has no effect on the posterior probabilities.

Discriminant-based vs discriminative:-

Discriminant based approach is even more simple. It combines the inference and decision stages into a single learning problem. However, in discriminant based approach, we no longer have access to $p(C_k|x)$.

Advantages of $P(C_k|x)$:-

① Minimizing risk: if the loss matrix is subject to change from time to time, if we know $P(C_k|x)$, we can trivially revise the minimum risk decision criterion by using decision theory. With discriminant based approach, we need to retrain the model.

② Compensating for class priors:

Consider cancer prediction. Dataset is heavily skewed with very few (1 in 1000) patients having cancer. Classifying every example as no-cancer gives 99.9% accuracy.

Soln: Steps: ① Sample the rare class more and balance the class distribution in the dataset.

② learn the posterior probabilities.

③ new posterior = $\frac{\text{old posterior}}{\text{class fraction in sampled data}} \times$ class fraction in original data.

We cant do this with discriminant-based models.

③ Combining models:-

$$P(C_k|x_1,x_2) \propto P(x_1,x_2|C_k)\,P(C_k)$$

$$\propto P(x_1|C_k)\,P(x_2|C_k)\,P(C_k)$$

$$\propto \frac{P(C_k|x_1)\,P(C_k|x_2)}{P(C_k)}$$

Here we combine $P(C_k|x_1)$ and $P(C_k|x_2)$ to predict $P(C_k|x_1,x_2)$.

④ Reject option: If the posterior probability of all classes are below a certain treshold θ, classifier can choose not to predict the class label. This is possible only when you have access to $P(C_k|x)$.

Summary of the Course so-far:

- Machine Learning
  - Supervised learning.
    - Regression
      - linear regression (LL)
      - Non-linear regression using basis fns. (LN)
    - Classification
      - discriminant
        - Indicator regression (LL)
        - LDA (LL)
      - discriminative
        - logistic regression (NL)
        - k-NN (NN)
      - Generative
        - GDA (NL)
        - QDA (NN)
        - GNB (NN)
        - DGDA (NL)
        - Discrete Naive Bayes. (NL)
  - Unsupervised learning
    - Dimensionality reduction.
      - PCA.

LL - Linear model, linear decision boundary

NL - Non-linear model, linear decision boundary.

NN - Non linear model, non-linear decision boundary.

LN - linear model, non-linear decision boundary.

You should know!

① Probabilistic discriminative models.

② logistic regression

③ Maximum likelihood and least squares.

④ Iterative reweighted least squares (IRLS)

⑤ Newton-Raphson method.

⑥ Discriminant vs. Discriminative vs. Generative models.
