# Probabilistic classification: discriminative models

Course of Machine Learning
Master Degree in Computer Science

University of Rome "Tor Vergata"

a.a. 2020-2021

Giorgio Gambosi

# Generalized linear models

In the cases considered above, the posterior class distributions $p(C_k|\mathbf{x})$ are sigmoidal or softmax with argument given by a linear combination of features in $\mathbf{x}$, i.e., they are a instances of generalized linear models

A generalized linear model (GLM) is a function

$$y(\mathbf{x}) = f(\mathbf{w}^T\mathbf{x} + w_0)$$

where $f$ (usually called the *response function*) is in general a non linear function.

Each iso-surface of $y(\mathbf{x})$ , such that by definition $y(\mathbf{x}) = c$ (for some constant $c$), is such that

$$f(\mathbf{w}^T\mathbf{x} + w_0) = c$$

and

$$\mathbf{w}^T\mathbf{x} + w_0 = f^{-1}(y) = c'$$

($c'$ constant).

Hence, iso-surfaces of a GLM are hyper-planes, thus implying that boundaries are hyperplanes themselves.

# GLM and normal distribution

1. Let $y \in \mathbb{R}$, and
$$p(y|\mathbf{x}) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y-\mu(\mathbf{x}))^2}{2\sigma^2}}$$
is a normal distribution with mean $\mu(\mathbf{x})$ and constant variance $\sigma^2$
2. Then, a linear regression results
$$y(\mathbf{x}) = \mathbf{w}^T \overline{\mathbf{x}}$$

# GLM and Bernoulli distribution

1. Let $y \in \{0, 1\}$, and

$$p(y|\mathbf{x}) = \pi(\mathbf{x})^y (1 - \pi(\mathbf{x}))^{1-y}$$

   is a Bernoulli distribution with parameter $\pi(\mathbf{x})$

2. then, a logistic regression derives

$$y(\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w}^T \overline{\mathbf{x}}}}$$

# GLM and categorical distribution

**1.** Let $y \in \{1, \ldots, K\}$, and

$$p(y|\mathbf{x}) = \prod_{i=1}^{K} \pi_i(\mathbf{x})^{y_i}$$

(where $y_i = 1$ if $y = i$ and $y = 0$ otherwise) is a categorical distribution with probabilities $\pi_1(\mathbf{x}), \ldots, \pi_K(\mathbf{x})$

**2.** then, a softmax regression results, with

$$\begin{aligned} y_i(\mathbf{x}) &= C\; e^{\mathbf{w}_i^T \overline{\mathbf{x}}} && \text{if } i \neq K \\ y_K(\mathbf{x}) &= C \end{aligned}$$

where $C = \dfrac{1}{\sum_{j=1}^{K} e^{\mathbf{w}_j^T \overline{\mathbf{x}}}}$

# GLM and additional regressions

Other regression types can be defined by considering different models for $p(y|\mathbf{x})$. For example,

1. Assume $y \in \{0, \dots, \}$ is a non negative integer (for example we are interested to count data), and $p(y|\mathbf{x}) = \frac{\lambda(\mathbf{x})^y}{y!} e^{-\lambda(\mathbf{x})}$ is a Poisson distribution with parameter $\lambda(\mathbf{x})$
2. then, a Poisson regression derives

$$y(\mathbf{x}) = e^{\mathbf{w}^T \overline{\mathbf{x}}}$$

# GLM and additional regressions

1. Assume $y \in [0, \infty)$ is a non negative real (for example we are interested to time intervals), and $p(y|\mathbf{x}) = \lambda(\mathbf{x})e^{-\lambda(\mathbf{x})y}$ is an exponential distribution with parameter $\lambda(\mathbf{x})$
2. then, an exponential regression derives

$$y(\mathbf{x}) = -\frac{1}{\mathbf{w}^T\overline{\mathbf{x}}}$$

# Discriminative approach

We could directly assume that $p(C_k|\mathbf{x})$ is a GLM and derive its coefficients (for example through ML estimation).

Comparison wrt the generative approach:

- Less information derived (we do not know $p(\mathbf{x}|C_k)$, thus we are not able to generate new data)
- Simpler method, usually a smaller set of parameters to be derived
- Better predictions, if the assumptions done with respect to $p(\mathbf{x}|C_k)$ are poor.

# Logistic regression

Logistic regression is a GLM deriving from the hypothesis of a Bernoulli distribution of $y$, which results into

$$p(C_1|\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w}^T\overline{\mathbf{x}}}}$$

where base functions could also be applied.

The model is equivalent, for the binary classification case, to linear regression for the regression case.

# Degrees of freedom

- In the case of $d$ features, logistic regression requires $d+1$ coefficients $w_0, \ldots, w_d$ to be derived from a training set
- A generative approach with gaussian distributions requires:
  - $2d$ coefficients for the means $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$,
  - for each covariance matrix

$$\sum_{i=1}^{d} i = d(d+1)/2 \qquad \text{coefficients}$$

  - one prior cla probability $p(C_1)$
- As a total, it results into $d(d+1) + 2d + 1 = d(d+3) + 1$ coefficients (if a unique covariance matrix is assumed $d(d+1)/2 + 2d + 1 = d(d+5)/2 + 1$ coefficients)

# Maximum likelihood estimation

Let us assume that targets of elements of the training set can be conditionally (with respect to model coefficients) modeled through a Bernoulli distribution. That is, assume

$$p(t_i|\mathbf{x}_i, \mathbf{w}) = p_i^{t_i}(1 - p_i)^{1-t_i}$$

where $p_i = p(C_1|\mathbf{x}_i) = \sigma(\mathbf{w}^T\mathbf{x}_i)$.
Then, the likelihood of the training set targets $\mathbf{t}$ given $\mathbf{X}$ is

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}) = L(\mathbf{w}|\mathbf{X}, \mathbf{t}) = \prod_{i=1}^{n} p(t_i|\mathbf{x}_i, \mathbf{w}) = \prod_{i=1}^{n} p_i^{t_i}(1 - p_i)^{1-t_i}$$

and the log-likelihood is

$$l(\mathbf{w}|\mathbf{X}, \mathbf{t}) = \log L(\mathbf{w}|\mathbf{X}, \mathbf{t}) = \sum_{i=1}^{n} (t_i \log p_i + (1 - t_i)\log(1 - p_i))$$

# Maximum likelihood estimation

To maximize the log likelihood, we could apply a gradient ascent algorithm, where at each iteration the following update of the currently estimated $\mathbf{w}$ is performed

$$\mathbf{w}^{(j+1)} = \mathbf{w}^{(j)} + \alpha \frac{\partial l(\mathbf{w}|\mathbf{X}, \mathbf{t})}{\partial \mathbf{w}}\Big|_{\mathbf{w}^{(j)}}$$

Since

$$\frac{\partial l(\mathbf{w}|\mathbf{X}, \mathbf{t})}{\partial \mathbf{w}} = \sum_{i=1}^{n}(t_i - p_i)\overline{\mathbf{x}}_i = \sum_{i=1}^{n}(t_i - \sigma(\mathbf{w}^T\overline{\mathbf{x}}_i))\overline{\mathbf{x}}_i$$

The solution update at each step is

$$\begin{aligned}
\mathbf{w}^{(j+1)} &= \mathbf{w}^{(j)} + \alpha \sum_{i=1}^{n}(t_i - \sigma((\mathbf{w}^{(j)})^T\overline{\mathbf{x}}_i))\overline{\mathbf{x}}_i \\
&= \mathbf{w}^{(j)} + \alpha \sum_{i=1}^{n}(t_i - y(\mathbf{x}_i))\overline{\mathbf{x}}_i
\end{aligned}$$

# Maximum likelihood estimation

As a possible alternative, at each iteration only one coefficient in $\mathbf{w}$ is updated

$$
\begin{aligned}
w_k^{(j+1)} &= w_k^{(j)} + \alpha \frac{\partial l(\mathbf{w}|\mathbf{X}, \mathbf{t})}{\partial w_k}\Big|_{\mathbf{w}^{(j)}} \\
&= w_k^{(j+1)} + \alpha \sum_{i=1}^{n} (t_i - \sigma((\mathbf{w}^{(j)})^T \overline{\mathbf{x}}_i)) x_{ik} \\
&= w_k^{(j+1)} + \alpha \sum_{i=1}^{n} (t_i - y(\mathbf{x}_i)) x_{ik}
\end{aligned}
$$

# Newton-Raphson method

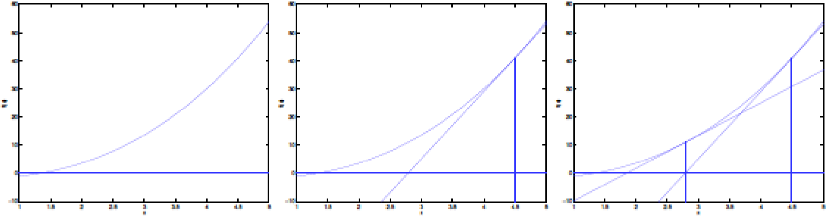
- Maximization of $l(\mathbf{w}|\mathbf{X}, \mathbf{t})$ through the well-known Newton-Raphson algorithm to compute the roots of a given function
- Given $f : \mathbb{R} \mapsto \mathbb{R}$, the algorithm finds $z \in \mathbb{R}$ such that $f(z) = 0$ through a sequence of iterations, starting from an initial value $z_0$ and performing the following update

$$z_{i+1} = z_i - \frac{f(z_i)}{f'(z_i)}$$

- At each iteration, the algorithm approximates $f$ by a line tangent to $f$ in $(z_i, f(z_i))$, and defines $z_{i+1}$ as the value where the line intersects the $x$ axis

# Newton-Raphson method

- Example of application of the method



- Newton-Raphson method can be also applied to compute maximum and minimum points for a function by finding zeros of the first derivative: this corresponds to applying the following update

$$z_{i+1} = z_i - \frac{f'(z_i)}{f''(z_i)}$$

# Newton-Raphson and multivariate functions

- To apply Newton-Raphson to logistic regression we have to extend it to the case of a vector variable, since the maximization has to be performed with respect to the vector $\mathbf{w}$ of coefficients
- In a multivariate framework, the first derivative is substituted by the gradient $\frac{\partial}{\partial \mathbf{w}}$, while the second derivative corresponds to the Hessian matrix $\mathbf{H}$, defined as follows

$$\mathbf{H}_{ij}(f) = \frac{\partial^2 f}{\partial w_i \partial w_j}$$

- The update operation turns out to be

$$\mathbf{w}^{(i+1)} = \mathbf{w}^{(i)} - (\mathbf{H}(f)|_{\mathbf{w}^{(i)}})^{-1} \frac{\partial f}{\partial \mathbf{w}}|_{\mathbf{w}_{(i)}}$$

# Newton-Raphson and linear regression

- In the case of linear regression, the error function to be minimized is

$$E(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{n}(y_i - t_i)^2 = \frac{1}{2}\sum_{i=1}^{n}(\mathbf{w}^T\overline{\mathbf{x}}_i - t_i)^2$$

- Then, it is possible to shoaw that

$$\frac{\partial E}{\partial \mathbf{w}} = \overline{\mathbf{X}}^T\overline{\mathbf{X}}\mathbf{w} - \overline{\mathbf{X}}^T\mathbf{t}$$

and

$$\mathbf{H} = \frac{\partial}{\partial \mathbf{w}}\frac{\partial E}{\partial \mathbf{w}} = \overline{\mathbf{X}}^T\overline{\mathbf{X}}$$

- At each iteration, the update is

$$\mathbf{w}^{(i+1)} = \mathbf{w}^{(i)} - (\overline{\mathbf{X}}^T\overline{\mathbf{X}})^{-1}(\overline{\mathbf{X}}^T\overline{\mathbf{X}}\mathbf{w}^{(i)} - \overline{\mathbf{X}}^T\mathbf{t}) = (\overline{\mathbf{X}}^T\overline{\mathbf{X}})^{-1}\overline{\mathbf{X}}^T\mathbf{t}$$

- We get the well-known solution, which is obtained in a single iteration.

# Newton-Raphson and logistic regression

Here, we have the cross-entropy loss function

$$E(\mathbf{w}) = -l(\mathbf{w}|\mathbf{X}, \mathbf{t}) = -\sum_{i=1}^{n} (t_i \log y_i + (1 - t_i) \log(1 - y_i))$$

with $y_i = \sigma(a_i)$ and $a_i = \mathbf{w}^T \overline{\mathbf{x}}_i$. It results,

$$\begin{aligned} \frac{\partial E}{\partial \mathbf{w}} &= -\frac{\partial l(\mathbf{w}|\mathbf{X}, \mathbf{t})}{\partial \mathbf{w}} = \overline{\mathbf{X}}^T (\mathbf{y} - \mathbf{t}) \\ \mathbf{H} = \frac{\partial}{\partial \mathbf{w}} \frac{\partial E}{\partial \mathbf{w}} &= \overline{\mathbf{X}}^T \mathbf{Y} \overline{\mathbf{X}} \end{aligned}$$

where

- $\mathbf{y}$ is the vector of predictions $y_i = \sigma(a_i) = \sigma(\mathbf{w}^T \overline{\mathbf{x}}_i)$ for $i = 1, \ldots, n$
- $\mathbf{Y}$ is a $n \times n$ diagonal matrix such that

$$Y_{ii} = y_i(1 - y_i)$$

# Newton-Raphson and logistic regression

- In the case of logistic regression, the update is then

$$\mathbf{w}^{(i+1)} = \mathbf{w}^{(i)} - (\overline{\mathbf{X}}^T \mathbf{Y}^{(i)} \overline{\mathbf{X}})^{-1} \overline{\mathbf{X}}^T (\mathbf{y}^{(i)} - \mathbf{t})$$

where both $\mathbf{y}$ and $\mathbf{Y}$ are dependent from $\mathbf{w}^{(i)}$, hence from $i$. Then,

$$\mathbf{w}^{(i+1)} = (\overline{\mathbf{X}}^T \mathbf{Y}^{(i)} \overline{\mathbf{X}})^{-1} \overline{\mathbf{X}}^T \mathbf{Y}^{(i)} \mathbf{z}^{(i)}$$

with

$$\mathbf{z}^{(i)} = \mathbf{a}^{(i)} - {\mathbf{Y}^{(i)}}^{-1} (\mathbf{y}^{(i)} - \mathbf{t})$$

Clearly, $\mathbf{z}^{(i)}$ is a function of $\mathbf{w}^{(i)}$, hence of the step $i$.

# Iterated reweighted least squares

- Let us consider the weighted extension of the least squares cost function, denoted as weighted least squares cost function, defined as

$$\sum_{i=1}^{n} \psi_i(\mathbf{y}_i - t_i)^2 = \sum_{i=1}^{n} \psi_i(\mathbf{w}^T \overline{\mathbf{x}}_i - t_i)^2$$

  for given weights $\psi_1, \ldots, \psi_n$. Clearly, the least squares problems corresponds to the case $\psi_i = 1$ for $i = 1, \ldots, n$
- It can be proved that, for this problem, the optimum is

$$\mathbf{w} = (\overline{\mathbf{X}}^T \mathbf{\Psi} \overline{\mathbf{X}})^{-1} \overline{\mathbf{X}}^T \mathbf{\Psi} \mathbf{t}$$

  where the weight matrix $\mathbf{\Psi}$ is a diagonal matrix with $\mathbf{\Psi}_{ii} = \psi_i$

# Iterated reweighted least squares

- Let us remind that, at each step of NR algorithm applied to logistic regression, the following update is performed

$$\mathbf{w}^{(i+1)} = (\overline{\mathbf{X}}^T \mathbf{Y}^{(i)} \overline{\mathbf{X}})^{-1} \overline{\mathbf{X}}^T \mathbf{Y}^{(i)} \mathbf{z}^{(i)}$$

- This corresponds to optimizing the weighted least squares cost function for feature matrix $\mathbf{X}$, target vector $\tilde{\mathbf{t}} = \mathbf{z}^{(i)}$, and weights $\psi_k = y_k^{(i)}(1 - y_k^{(i)})$
- The update of $\mathbf{w}^{(i)}$ performed at each iteration can then be computed by solving a new instance of the weighted least square problem, setting $\mathbf{w}^{(i+1)}$ to the solution obtained, and deriving the new values of $\mathbf{\Psi} = \mathbf{Y}^{(i+1)}$ and $\tilde{\mathbf{t}} = \mathbf{z}^{(i+1)}$.

# Logistic regression and GDA

- Observe that assuming $p(\mathbf{x}|C_1)$ are $p(\mathbf{x}|C_2)$ as multivariate normal distributions with same covariance matrix $\mathbf{\Sigma}$ results into a logistic $p(C_1|\mathbf{x})$.
- The opposite, however, is not true in general: in fact, GDA relies on stronger assumptions than logistic regression.
- The more the normality hypothesis of class conditional distributions with same covariance is verified, the more GDA will tend to provide the best models for $p(C_1|\mathbf{x})$

# Logistic regression and GDA

- Logistic regression relies on weaker assumptions than GDA: it is then less sensible from a limited correctness of such assumptions, thus resulting in a more robust technique
- Since $p(C_i|\mathbf{x})$ is logistic under a wide set of hypotheses about $p(\mathbf{x}|C_i)$, it will usually provide better solutions (models) in all such cases, while GDA will provide poorer models as far as the normality hypotheses is less verified.

# Softmax regression

- In order to extend the logistic regression approach to the case $K > 2$, let us consider the matrix $\mathbf{W} = (\mathbf{w}_1, \ldots, \mathbf{w}_K)$ of model coefficients, of size $(d+1) \times K$, where $\mathbf{w}_j$ is the $d+1$-dimensional vector of coefficients for class $C_j$.
- In this case, the likelihood is defined as

$$p(\mathbf{T}|\mathbf{X}, \mathbf{W}) = \prod_{i=1}^{n} \prod_{k=1}^{K} p(C_k|\mathbf{x}_i)^{t_{ik}} = \prod_{i=1}^{n} \prod_{k=1}^{K} \left( \frac{e^{\mathbf{w}_k^T \overline{\mathbf{x}}_i}}{\sum_{r=1}^{K} e^{\mathbf{w}_r^T \overline{\mathbf{x}}_i}} \right)^{t_{ik}}$$

  where $\mathbf{X}$ is the usual matrix of features and $\mathbf{T}$ is the $n \times K$ matrix where row $i$ is the 1-to-$K$ coding of $t_i$. That is, if $\mathbf{x}_i \in C_k$ then $t_{ik} = 1$ and $t_{ir} = 0$ for $r \neq k$.

# ML and softmax regression

The log-likelihood is then defined as

$$l(\mathbf{W}) = \sum_{i=1}^{n} \sum_{k=1}^{K} t_{ik} \log \left( \frac{e^{\mathbf{w}_k^T \overline{\mathbf{x}}_i}}{\sum_{r=1}^{K} e^{\mathbf{w}_r^T \overline{\mathbf{x}}_i}} \right)$$

And the gradient is defined as

$$\frac{\partial l(\mathbf{W})}{\partial \mathbf{W}} = \left( \frac{\partial l(\mathbf{W})}{\partial \mathbf{w}_1}, \ldots, \frac{\partial l(\mathbf{W})}{\partial \mathbf{w}_K} \right)$$

# ML and softmax regression

It results

$$\frac{\partial l(\mathbf{W})}{\partial \mathbf{w}_j} = \sum_{i=1}^{n} (t_{ij} - y_{ij})\overline{\mathbf{x}}_i$$

Observe that the gradient has the same structure than in the case of linear regression and logistic regression.

# Bayesian logistic regression

- Used to overcome the overfitting problem by assuming a prior distribution
- The aim is to estimate the posterior class (predictive) distribution, that is the expectation of the model prediction wrt to the distribution of model coefficients,

$$
\begin{aligned}
p(\mathcal{C}_1|\mathbf{x}, \mathbf{X}, \mathbf{t}) &= \int p(\mathcal{C}_1|\mathbf{x}, \mathbf{w})p(\mathbf{w}|\mathbf{X}, \mathbf{t})d\mathbf{w} \\
&= \int \sigma(\mathbf{w}^T\overline{\mathbf{x}})p(\mathbf{w}|\mathbf{X}, \mathbf{t})d\mathbf{w}
\end{aligned}
$$

- we need some way to evaluate the posterior distribution of coefficients $p(\mathbf{w}|\mathbf{X}, \mathbf{t})$ for any $\mathbf{w}$

# Posterior distribution of coefficients

By Bayes' rule,

$$p(\mathbf{w}|\mathbf{X}, \mathbf{t}) = \frac{p(\mathbf{t}|\mathbf{X}, \mathbf{w})p(\mathbf{w})}{p(\mathbf{t}|\mathbf{X})} = \frac{p(\mathbf{t}|\mathbf{X}, \mathbf{w})p(\mathbf{w})}{\int p(\mathbf{t}|\mathbf{X}, \mathbf{w}')p(\mathbf{w}')d\mathbf{w}'}$$

where the likelihood is $p(\mathbf{t}|\mathbf{X}, \mathbf{w}) = \prod_{i=1}^{n} p(t_i|\mathbf{x}_i, \mathbf{w})$, with

$$p(t_i|\mathbf{x}_i, \mathbf{w}) = \begin{cases} \sigma(\mathbf{w}^T\overline{\mathbf{x}}) & \text{if } t_i = 1 \\ 1 - \sigma(\mathbf{w}^T\overline{\mathbf{x}}) & \text{if } t_i = 0 \end{cases}$$

# Posterior distribution of coefficients

That is,

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}) = \prod_{i=1}^{n} \sigma(\mathbf{w}^T \overline{\mathbf{x}})^{t_i} \left(1 - \sigma(\mathbf{w}^T \overline{\mathbf{x}})\right)^{1-t_i}$$

and

$$p(\mathbf{w}|\mathbf{X}, \mathbf{t}) = \frac{p(\mathbf{w}) \prod_{i=1}^{n} \sigma(\mathbf{w}^T \overline{\mathbf{x}})^{t_i} \left(1 - \sigma(\mathbf{w}^T \overline{\mathbf{x}})\right)^{1-t_i}}{Z}$$

with the normalization factor

$$Z = \int p(\mathbf{w}) \prod_{i=1}^{n} \sigma(\mathbf{w}^T \overline{\mathbf{x}})^{t_i} \left(1 - \sigma(\mathbf{w}^T \overline{\mathbf{x}})\right)^{1-t_i} d\mathbf{w}$$

# Predictive distribution intractability

$Z$ is hard to compute: we are only able to evaluate the numerator

$$g(\mathbf{w}; \mathbf{X}, \mathbf{t}) = p(\mathbf{w}) \prod_{i=1}^{n} \sigma(\mathbf{w}^T \overline{\mathbf{x}})^{t_i} \left(1 - \sigma(\mathbf{w}^T \overline{\mathbf{x}})\right)^{1-t_i}$$

which is proportional to $p(\mathbf{w}|\mathbf{X}, \mathbf{t})$ through an unknown proportionality coefficient.

# Predictive distribution intractability

Possible options:

1. find a single value of $\mathbf{w}$ which maximizes $p(\mathbf{w}|\mathbf{X}, \mathbf{t})$: this corresponds to the value which maximizes $g(\mathbf{w}; \mathbf{X}, \mathbf{t})$ (this is the usual MAP approach)
2. approximate $p(\mathbf{w}|\mathbf{X}, \mathbf{t})$ with some other probability density which can be treated analytically (*variational* approach)
3. sample from $p(\mathbf{w}|\mathbf{X}, \mathbf{t})$, knowing only $g(\mathbf{w}; \mathbf{X}, \mathbf{t})$ (*Montecarlo* approach)