Discriminative models

Machine Learning

Hamid R Rabiee – Zahra Dehghanian

Spring 2025

Sharif University of Technology

Probabilistic classifiers

- How can we find the probabilities required in the Bayes decision rule?

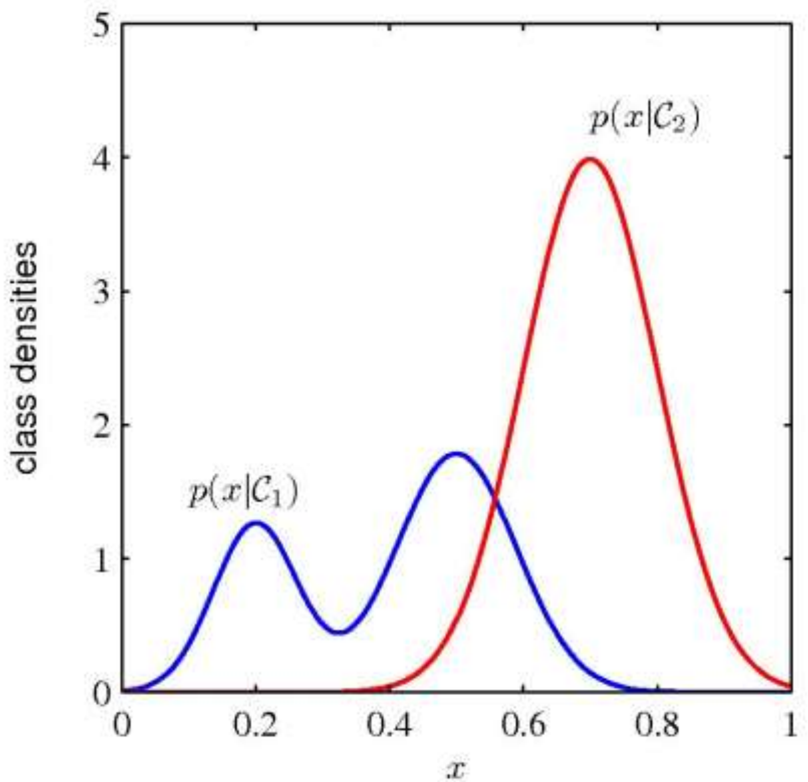
- Probabilistic classification approaches can be divided in two main categories:
  - Generative
    - Estimate pdf $p(\boldsymbol{x}, \mathcal{C}_k)$ for each class $\mathcal{C}_k$ and then use it to find $p(\mathcal{C}_k|\boldsymbol{x})$
      - or alternatively estimate both pdf $p(\boldsymbol{x}|\mathcal{C}_k)$ and $p(\mathcal{C}_k)$ to find $p(\mathcal{C}_k|\boldsymbol{x})$
  - Discriminative
    - Directly estimate $p(\mathcal{C}_k|\boldsymbol{x})$ for each class $\mathcal{C}_k$

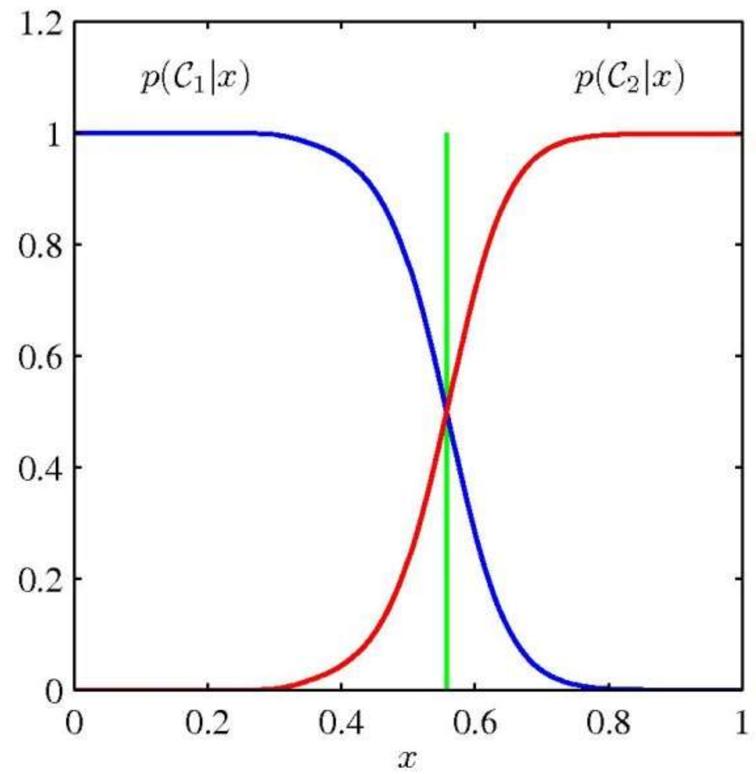
Generative approach

- <u>Inference stage</u>
  - Determine class conditional densities $p(\boldsymbol{x}|\mathcal{C}_k)$ and priors $p(\mathcal{C}_k)$
  - Use the Bayes theorem to find $p(\mathcal{C}_k|\boldsymbol{x})$

- <u>Decision stage:</u> After learning the model (inference stage), make optimal class assignment for new input
  - if $p(\mathcal{C}_i|\boldsymbol{x}) > p(\mathcal{C}_j|\boldsymbol{x}) \;\; \forall j \neq i$ then decide $\mathcal{C}_i$

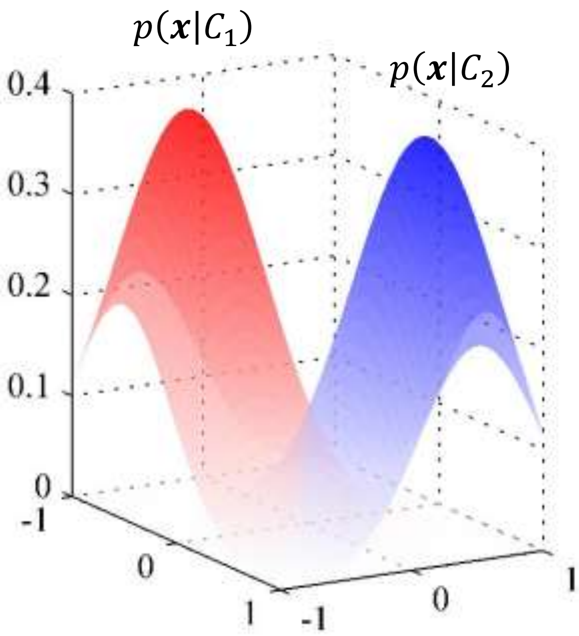
Discriminative vs. generative approach

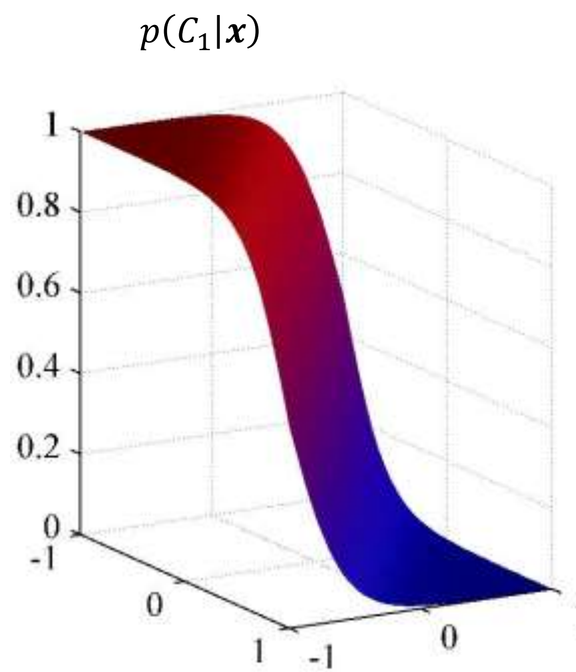




[Bishop]

Class conditional densities vs. posterior





[Bishop]

$$p(\mathcal{C}_1|\boldsymbol{x}) = \sigma(\boldsymbol{w}^T\boldsymbol{x} + w_0)$$

$$\boldsymbol{w} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

$$w_0 = -\frac{1}{2}\boldsymbol{\mu}_1^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2 + \ln\frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}$$

$$\sigma(z) = \frac{1}{1+\exp(z)}$$

Probabilistic discriminant functions

- **Discriminant functions**: A popular way of representing a classifier
  - A discriminant function $f_i(\boldsymbol{x})$ for each class $\mathcal{C}_i$ $(i = 1, \dots, K)$:
    - $\boldsymbol{x}$ is assigned to class $\mathcal{C}_i$ if:

$$f_i(\boldsymbol{x}) > f_j(\boldsymbol{x}) \quad \forall j \neq i$$

- Representing Bayesian classifier using discriminant functions:
  - Classifier minimizing error rate: $f_i(\boldsymbol{x}) = P(\mathcal{C}_i|\boldsymbol{x})$
  - Classifier minimizing risk: $f_i(\boldsymbol{x}) = -\sum_{j=1}^{K} L_{ij} p(\mathcal{C}_j|\boldsymbol{x})$

Discriminative approach

- <u>Inference stage</u>
  - Determine the posterior class probabilities $P(\mathcal{C}_k|\boldsymbol{x})$ directly

- <u>Decision stage:</u> After learning the model (inference stage), make optimal class assignment for new input
  - if $P(\mathcal{C}_i|\boldsymbol{x}) > P(\mathcal{C}_j|\boldsymbol{x}) \quad \forall j \neq i$ then decide $\mathcal{C}_i$

Discriminative approach: logistic regression

$K = 2$

- More general than discriminant functions:
  - $f(\boldsymbol{x}; \boldsymbol{w})$ predicts posterior probabilities $P(y = 1|\boldsymbol{x})$

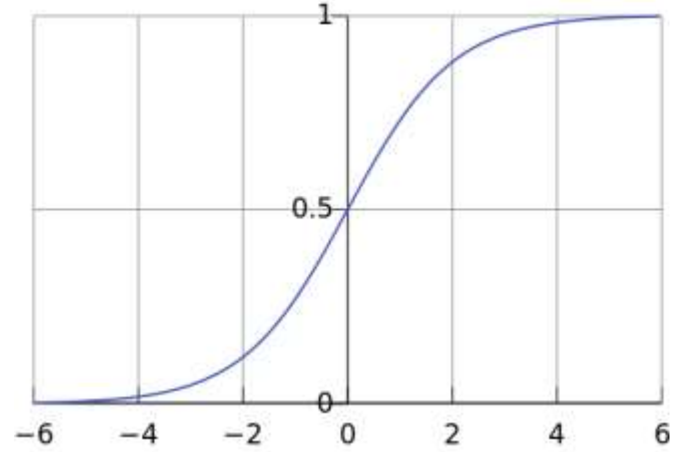
$$f(\boldsymbol{x}; \boldsymbol{w}) = \sigma(\boldsymbol{w}^T \boldsymbol{x})$$

$\boldsymbol{x} = [1, x_1, \dots, x_d]$
$\boldsymbol{w} = [w_0, w_1, \dots, w_d]$

$\sigma(.)$ is an activation function

- Sigmoid (logistic) function

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$



Logistic regression

- $f(\boldsymbol{x};\boldsymbol{w})$: probability that $y=1$ given $\boldsymbol{x}$ (parameterized by $\boldsymbol{w}$)

$$P(y=1|\boldsymbol{x};\boldsymbol{w}) = f(\boldsymbol{x};\boldsymbol{w})$$

$K=2$
$y \in \{0,1\}$

$$P(y=0|\boldsymbol{x};\boldsymbol{w}) = 1 - f(\boldsymbol{x};\boldsymbol{w})$$

$f(\boldsymbol{x};\boldsymbol{w}) = \sigma(\boldsymbol{w}^T\boldsymbol{x})$
$0 \le f(\boldsymbol{x};\boldsymbol{w}) \le 1$
estimated probability of $y=1$ on input x

- Example: Cancer (Malignant, Benign)
  - $f(\boldsymbol{x};\boldsymbol{w}) = 0.7$
  - 70% chance of tumor being malignant

Logistic regression: Decision surface

- Decision surface $f(\boldsymbol{x}; \boldsymbol{w}) = \text{constant}$
  - $f(\boldsymbol{x}; \boldsymbol{w}) = \sigma(\boldsymbol{w}^T\boldsymbol{x}) = \frac{1}{1+e^{-(\boldsymbol{w}^T\boldsymbol{x})}} = 0.5$

- Decision surfaces are linear functions of $\boldsymbol{x}$

if $f(\boldsymbol{x}; \boldsymbol{w}) \geq 0.5$ then $y = 1$
else $y = 0$

Equivalent to

if $\boldsymbol{w}^T\boldsymbol{x} + w_0 \geq 0$ then $y = 1$
else $y = 0$

Logistic regression: ML estimation

- Maximum (conditional) log likelihood:

$$\widehat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmax}} \ \log \prod_{i=1}^{n} p\left(y^{(i)} \middle| \boldsymbol{w}, \boldsymbol{x}^{(i)}\right)$$

$$p\left(y^{(i)} \middle| \boldsymbol{w}, \boldsymbol{x}^{(i)}\right) = f\left(\boldsymbol{x}^{(i)}; \boldsymbol{w}\right)^{y^{(i)}} \left(1 - f\left(\boldsymbol{x}^{(i)}; \boldsymbol{w}\right)\right)^{\left(1-y^{(i)}\right)}$$

$$\log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w}) = \sum_{i=1}^{n} \left[y^{(i)} \log\left(f\left(\boldsymbol{x}^{(i)}; \boldsymbol{w}\right)\right) + (1 - y^{(i)}) \log\left(1 - f\left(\boldsymbol{x}^{(i)}; \boldsymbol{w}\right)\right)\right]$$

Logistic regression: cost function

- $$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmin}}\, J(\boldsymbol{w})$$

$$J(\boldsymbol{w}) = -\sum_{i=1}^{n} \log p\left(y^{(i)} \middle| \boldsymbol{w}, \boldsymbol{x}^{(i)}\right)$$
$$= \sum_{i=1}^{n} -y^{(i)} \log\left(f\left(\boldsymbol{x}^{(i)}; \boldsymbol{w}\right)\right) - (1 - y^{(i)}) \log\left(1 - f\left(\boldsymbol{x}^{(i)}; \boldsymbol{w}\right)\right)$$

- No closed form solution for

$$\nabla_{\boldsymbol{w}} J(\boldsymbol{w}) = 0$$

- However $J(\boldsymbol{w})$ is convex.

Logistic regression: Gradient descent

- $$\boldsymbol{w}^{t+1} = \boldsymbol{w}^t - \eta \nabla_{\boldsymbol{w}} J(\boldsymbol{w}^t)$$

$$\nabla_{\boldsymbol{w}} J(\boldsymbol{w}) = \sum_{i=1}^{n} \left(f\left(\boldsymbol{x}^{(i)}; \boldsymbol{w}\right) - y^{(i)} \right) \boldsymbol{x}^{(i)}$$

- Is it similar to gradient of SSE for linear regression?

$$\nabla_{\boldsymbol{w}} J(\boldsymbol{w}) = \sum_{i=1}^{n} \left(\boldsymbol{w}^T \boldsymbol{x}^{(i)} - y^{(i)} \right) \boldsymbol{x}^{(i)}$$

Logistic regression: loss function

- $$\mathrm{Loss}\big(y, f(\boldsymbol{x};\boldsymbol{w})\big) = -y \times \log\big(f(\boldsymbol{x};\boldsymbol{w})\big) - (1-y) \times \log(1 - f(\boldsymbol{x};\boldsymbol{w}))$$

Since $y = 1$ or $y = 0$ $\Rightarrow$ $\mathrm{Loss}\big(y, f(\boldsymbol{x};\boldsymbol{w})\big) = \begin{cases} -\log(f(\boldsymbol{x};\boldsymbol{w})) & \text{if } y = 1 \\ -\log(1 - f(\boldsymbol{x};\boldsymbol{w})) & \text{if } y = 0 \end{cases}$

How is it related to zero-one loss?

$$\mathrm{Loss}(y, \hat{y}) = \begin{cases} 1 & y \neq \hat{y} \\ 0 & y = \hat{y} \end{cases}$$

$$f(\boldsymbol{x};\boldsymbol{w}) = \frac{1}{1 + exp(-\boldsymbol{w}^T\boldsymbol{x})}$$

Logistic regression: cost function (summary)

- Logistic Regression (LR) has a more proper cost function for classification than SSE and Perceptron

- Why is the cost function of LR also more suitable than?

$$J(\boldsymbol{w}) = \frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)} - f\left(\boldsymbol{x}^{(i)};\boldsymbol{w}\right)\right)^2$$

  where $f(\boldsymbol{x};\boldsymbol{w}) = \sigma(\boldsymbol{w}^T\boldsymbol{x})$

  - The conditional distribution $p(y|\boldsymbol{x};\boldsymbol{w})$ in the classification problem is not Gaussian (it is Bernoulli)
  - The cost function of LR is convex

Posterior probabilities

- Two-class: $p(\mathcal{C}_k|\boldsymbol{x})$ can be written as a logistic sigmoid for a wide choice of $p(\boldsymbol{x}|\mathcal{C}_k)$ distributions

$$p(\mathcal{C}_1|\boldsymbol{x}) = \sigma(a(\boldsymbol{x})) = \frac{1}{1 + \exp(-a(\boldsymbol{x}))}$$

- Multi-class: $p(\mathcal{C}_k|\boldsymbol{x})$ can be written as a soft-max for a wide choice of $p(\boldsymbol{x}|\mathcal{C}_k)$

$$p(\mathcal{C}_k|\boldsymbol{x}) = \frac{\exp(a_k(\boldsymbol{x}))}{\sum_{j=1}^{K} \exp(a_j(\boldsymbol{x}))}$$

Multi-class logistic regression

- For each class k, $f_k(\boldsymbol{x}; \boldsymbol{W})$ predicts the probability of $y = k$
  - i.e., $P(y = k|\boldsymbol{x}, \boldsymbol{W})$

- On a new input $\boldsymbol{x}$, to make a prediction, pick the class that maximizes $f_k(\boldsymbol{x}; \boldsymbol{W})$:

$$\alpha(\boldsymbol{x}) = \underset{k=1,\ldots,K}{\operatorname{argmax}} f_k(\boldsymbol{x})$$

if $f_k(\boldsymbol{x}) > f_j(\boldsymbol{x}) \quad \forall j \neq k$ then
decide C_k

Multi-class logistic regression

$K > 2$
$y \in \{1,2, \dots, K\}$

$$f_k(\boldsymbol{x}; \boldsymbol{W}) = p(y = k|\boldsymbol{x}) = \frac{\exp(\boldsymbol{w}_k^T \boldsymbol{x})}{\sum_{j=1}^{K} \exp(\boldsymbol{w}_j^T \boldsymbol{x})}$$

- Normalized exponential (aka softmax)
  - If $\boldsymbol{w}_k^T \boldsymbol{x} \gg \boldsymbol{w}_j^T \boldsymbol{x}$ for all $j \neq k$ then $p(C_k|\boldsymbol{x}) \simeq 1, p(C_j|\boldsymbol{x}) \simeq 0$

$$p(C_k|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|C_k)p(C_k)}{\sum_{j=1}^{K} p(\boldsymbol{x}|C_j)p(C_j)}$$

Logistic regression: multi-class

-

$$\widehat{\boldsymbol{W}} = \underset{\boldsymbol{W}}{\operatorname{argmin}}\, J(\boldsymbol{W})$$

$$J(\boldsymbol{W}) = -\log \prod_{i=1}^{n} p(\boldsymbol{y}^{(i)} | \boldsymbol{x}^{(i)}, \boldsymbol{W})$$

$$= -\log \prod_{i=1}^{n} \prod_{k=1}^{K} f_k(\boldsymbol{x}^{(i)}; \boldsymbol{W})^{y_k^{(i)}}$$

$$= -\sum_{i=1}^{n} \sum_{k=1}^{K} y_k^{(i)} \log\left(f_k(\boldsymbol{x}^{(i)}; \boldsymbol{W})\right)$$

$$\boldsymbol{W} = [\boldsymbol{w}_1 \quad \cdots \quad \boldsymbol{w}_K]$$

$\boldsymbol{y}$ is a vector of length K (1-of-K coding)
e.g., $\boldsymbol{y} = [0,0,1,0]^T$ when the target class is C_3

$$\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y}^{(1)} \\ \vdots \\ \boldsymbol{y}^{(n)} \end{bmatrix} = \begin{bmatrix} y_1^{(1)} & \cdots & y_K^{(1)} \\ \vdots & \ddots & \vdots \\ y_1^{(n)} & \cdots & y_K^{(n)} \end{bmatrix}$$

Logistic regression: multi-class

-

$$\boldsymbol{w}_j^{t+1} = \boldsymbol{w}_j^t - \eta \nabla_{\boldsymbol{W}} J(\boldsymbol{W}^t)$$

$$\nabla_{\boldsymbol{w}_j} J(\boldsymbol{W}) = \sum_{i=1}^{n} \left(f_j(\boldsymbol{x}^{(i)}; \boldsymbol{W}) - y_j^{(i)} \right) \boldsymbol{x}^{(i)}$$

- We usually consider also a regularization term and the gradient will be

$$\nabla_{\boldsymbol{w}_j} J(\boldsymbol{W}) = \lambda \boldsymbol{W} + \sum_{i=1}^{n} \left(f_j(\boldsymbol{x}^{(i)}; \boldsymbol{W}) - y_j^{(i)} \right) \boldsymbol{x}^{(i)}$$

Log-odds Ratio

- Optimal rule $y = arg\max_c p(t = c|x)$ is equivalent to

$$y = c \quad \Leftrightarrow \quad \frac{p(t = c|x)}{p(t = j|x)} \geq 1 \quad \forall j \neq c$$
$$\Leftrightarrow \quad \log \frac{p(t = c|x)}{p(t = j|x)} \geq 0 \quad \forall j \neq c$$

- For the binary case

$$y = 1 \quad \Leftrightarrow \quad \log \frac{p(t = 1|x)}{p(t = 0|x)} \geq 0$$

Logistic Regression (LR): summary

- LR is a linear classifier
- LR optimization problem is obtained by maximum likelihood
  - when assuming Bernoulli distribution for conditional probabilities whose mean is $\frac{1}{1+e^{-(w^T x)}}$
- No closed-form solution for its optimization problem
  - But convex cost function and global optimum can be found by gradient ascent

Discriminative vs. generative: number of parameters

- d-dimensional feature space
- Logistic regression: $d + 1$ parameters
  - $\boldsymbol{w} = (w_0, w_1, \ldots, w_d)$
- Generative approach:
  - Gaussian class-conditionals with shared covariance matrix
    - $2d$ parameters for means
    - $d(d + 1)/2$ parameters for shared covariance matrix
    - one parameter for class prior $p(C_1)$.
- But LR is more robust, less sensitive to incorrect modeling assumptions

Summary of alternatives

- Generative
  - Most demanding, because it finds the joint distribution $p(\boldsymbol{x}, \mathcal{C}_k)$
  - Usually needs a large training set to find $p(\boldsymbol{x}|\mathcal{C}_k)$
  - Can find $p(\boldsymbol{x})$ $\Rightarrow$ Outlier or novelty detection

- Discriminative
  - Specifies what is really needed (i.e., $p(\mathcal{C}_k|\boldsymbol{x})$)
  - More computationally efficient

Feed back

? https://forms.gle/vKRbyVVsWRKcZuqr8

Resources

- C. Bishop, "Pattern Recognition and Machine Learning", Chapter 4.2-4.3.
- Course CE-717, Dr. M.Soleymani