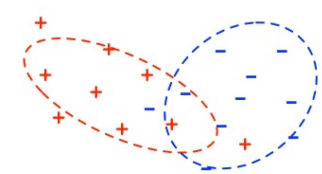
Classification with discriminative models

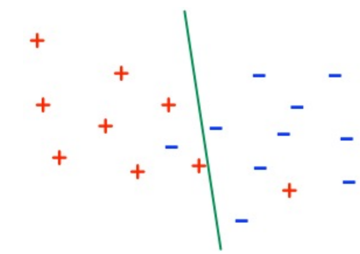
DSE 210

Classification with parametrized models

Classifiers with a fixed number of parameters can represent a limited set of functions. Learning a model is about picking a good approximation.

Typically the x's are points in p-dimensional Euclidean space, $\mathbb{R}^p$.





Two ways to classify:

- **Generative**: model the individual classes.
- **Discriminative**: model the decision boundary between the classes.

Generative models: pros and cons

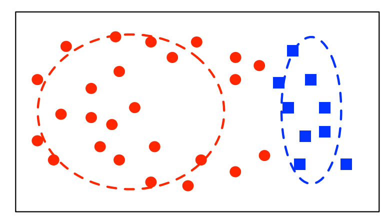
Advantages:

- Multiclass is a breeze
- Special density models (such as Bayes nets or hidden Markov models) can model temporal and other dependencies
- Returns not just a classification but also a confidence $\Pr(y|x)$
- For many common models: converges fast

Disadvantages:

- Formula for $\Pr(y|x)$ assumes the class-specific density models are perfect, but this is never true
- If we only care about classification, shouldn't we focus on the decision boundary rather than trying to model other aspects of the distribution of x?

Generative versus discriminative



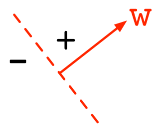
The generative way:

- Fit: π_0, π_1, P_0, P_1
- This determines a full joint distribution $\Pr(x, y)$
- Use Bayes' rule to obtain $\Pr(y|x)$

Discriminative: model $\Pr(y|x)$ directly

What model to use for $\Pr(y|x)$?

- Say $\mathcal{X} = \mathbb{R}^p$ and $\mathcal{Y} = \{-1, 1\}$
- Start with a linear function: $w \cdot x$.



Classification rule: predict sign$(w \cdot x)$

- $\Pr(y|x)$ should depend on how far x is from the decision boundary.

$$\Pr(y = 1|x) = \text{function of } w \cdot x$$

where the probability is 1/2 when $w \cdot x = 0$ and

$$\Pr(y = 1|x) \;\to\; \begin{cases} 1 & \text{as } w \cdot x \to \infty \\ 0 & \text{as } w \cdot x \to -\infty \end{cases}$$

Logistic regression model parametrized by w:

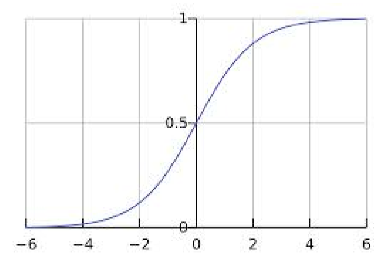
$$\Pr_w(y \mid x) = \frac{1}{1 + e^{-y(w \cdot x)}}$$

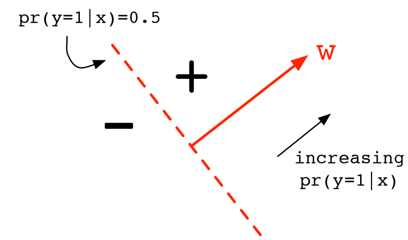
The squashing function

Take $\mathcal{X} = \mathbb{R}^p$ and $\mathcal{Y} = \{-1, 1\}$. The model specified by $w \in \mathbb{R}^p$ is

$$\Pr_w(y \mid x) \;=\; \frac{1}{1 + e^{-y(w \cdot x)}} \;=\; g(y(w \cdot x)),$$

where $g(z) = 1/(1 + e^{-z})$ is the *squashing function*.





Fitting w

The maximum-likelihood principle: given a data set

$$(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)}) \in \mathbb{R}^p \times \{-1, 1\},$$

pick the $w \in \mathbb{R}^p$ that maximizes

$$\prod_{i=1}^{n} \Pr_w(y^{(i)} \mid x^{(i)}).$$

Easier to work with sums, so take log to get **loss function**

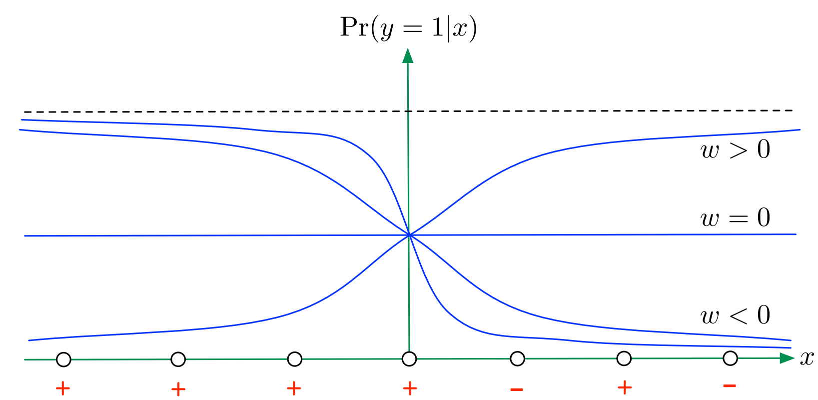
$$L(w) \;=\; -\sum_{i=1}^{n} \ln \Pr_w(y^{(i)} \mid x^{(i)}) \;=\; \sum_{i=1}^{n} \ln(1 + e^{-y^{(i)}(w \cdot x^{(i)})})$$

Our goal is to minimize $L(w)$.

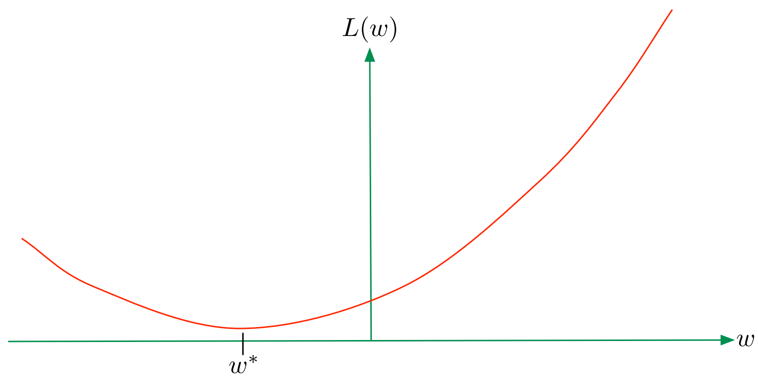
The good news: $L(w)$ is **convex** in w.

One dimensional example

$$\Pr_w(y \mid x) = \frac{1}{1 + e^{-ywx}}, \qquad w \in \mathbb{R}$$



:xample, cont'd



How to find the minimum of this convex function? A variety of options:

- Gradient descent
- Newton-Raphson

and many others.

Gradient descent procedure for logistic regression

Given $(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)}) \in \mathbb{R}^p \times \{-1, 1\}$, find

$$\arg\min_{w \in \mathbb{R}^p} L(w) \;=\; \sum_{i=1}^{n} \ln(1 + e^{-y^{(i)}(w \cdot x^{(i)})})$$

- Set $w_0 = 0$
- For $t = 0, 1, 2, \ldots$, until convergence:

$$w_{t+1} \;=\; w_t \;+\; \eta_t \sum_{i=1}^{n} y^{(i)} x^{(i)} \underbrace{\Pr_{w_t}(-y^{(i)}|x^{(i)})}_{\text{doubt}_t(x^{(i)}, y^{(i)})},$$

where η_t is a step size chosen by line search to minimize $L(w_{t+1})$.

Newton-Raphson procedure for logistic regression

- Set $w_0 = 0$
- For $t = 0, 1, 2, \ldots$, until convergence:

$$w_{t+1} \;=\; w_t \;+\; \eta_t (X^T D_t X)^{-1} \sum_{i=1}^{n} y^{(i)} x^{(i)} \Pr_{w_t}(-y^{(i)}|x^{(i)}),$$

where

- X is the $n \times p$ data matrix with one point per row
- D_t is an $n \times n$ diagonal matrix with (i, i) entry

$$D_{t,ii} \;=\; \Pr_{w_t}(1|x^{(i)}) \Pr_{w_t}(-1|x^{(i)})$$

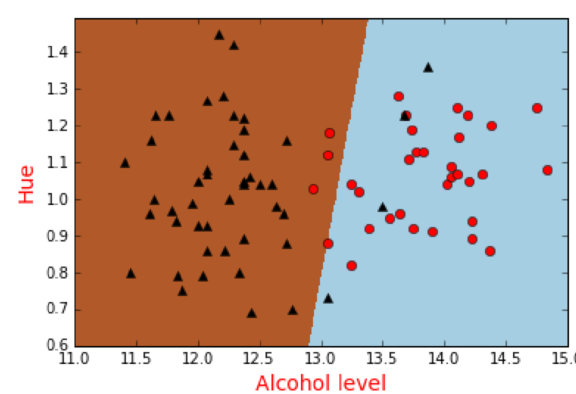
- η_t is a step size that is either fixed to 1 ("iterative reweighted least squares") or chosen by line search to minimize $L(w_{t+1})$.

Example: "wine" data set

Recall: data from three wineries from the same region of Italy.

- 13 attributes: hue, color intensity, flavanoids, ash content, ...
- 178 instances in all: split into 118 train, 60 test

Pick two classes and just two attributes (hue, alcohol content).



Test error using logistic regression: 10%.