Machine learning: lecture 5

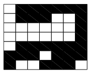
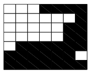
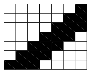
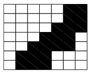
Tommi S. Jaakkola
MIT CSAIL
tommi@csail.mit.edu

Topics

- Classification and regression
  - regression approach to classification
  - Fisher linear discriminant
  - elementary decision theory
- Logistic regression
  - model, rationale
  - estimation, stochastic gradient
  - additive extension
  - generalization

Classification

Example: digit recognition (8x8 binary digits)

binary digit	actual label	target label in learning
	“2”	1
	“2”	1
	“1”	0
	“1”	0
. . .	. . .	

Classification via regression

- Suppose we ignore the fact that the target output y is binary (e.g., $0/1$) rather than a continuous variable
- So we will estimate a linear regression function

$$\begin{aligned} f(\mathbf{x};\mathbf{w}) &= w_0 + w_1 x_1 + \ldots + w_d x_d \\ &= w_0 + \mathbf{x}^T \mathbf{w}_1, \end{aligned}$$

  based on the available data as before.
- Assuming $y = f(\mathbf{x};\mathbf{w}) + \epsilon$, $\epsilon \sim N(0,\sigma^2)$, then the ML objective for the parameters $\mathbf{w}$ reduces to least squares fitting:

$$J_n(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} (y_i - f(\mathbf{x}_i;\mathbf{w}))^2$$

Classification via regression cont’d

- We can use the resulting regression function

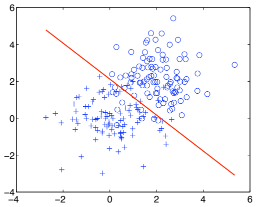
$$f(\mathbf{x};\hat{\mathbf{w}}) = w_0 + \mathbf{x}^T \hat{\mathbf{w}}_1,$$

  to classify any new (test) example $\mathbf{x}$ according to

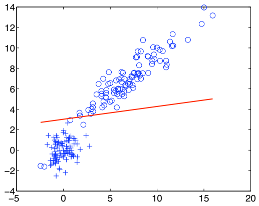
  label $= 1$ if $f(\mathbf{x};\mathbf{w}) > 0.5$, and label $= 0$ otherwise
- $f(\mathbf{x};\hat{\mathbf{w}}) = 0.5$ therefore defines a linear *decision boundary* that partitions the input space into two class specific regions (half spaces)

Classification via regression cont’d

- Given the dissociation between the objective (classification) and the estimation criterion (regression) it is not clear that this approach leads to sensible results



sometimes good



sometimes bad

Linear regression and projections

- A linear regression function (here in 2D)

$$f(\mathbf{x};\mathbf{w}) = w_0 + \mathbf{x}^T\mathbf{w}_1$$

projects each point $\mathbf{x} = [x_1\ x_2]^T$ to a line parallel to $\mathbf{w}_1$.

point in $\mathcal{R}^d$	projected point in $\mathcal{R}$
$\mathbf{x}_1$	$z_1 = \mathbf{x}_1^T\mathbf{w}_1$
$\mathbf{x}_2$	$z_2 = \mathbf{x}_2^T\mathbf{w}_1$
$\cdots$	$\cdots$
$\mathbf{x}_n$	$z_n = \mathbf{x}_n^T\mathbf{w}_1$

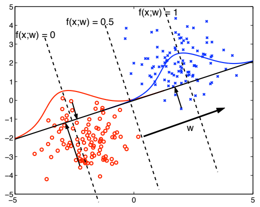
- We can study how well the projected points $\{z_1, \ldots, z_n\}$, viewed as functions of $\mathbf{w}_1$, are separated across the classes.

Linear regression and projections

- A linear regression function (here in 2D)

$$f(\mathbf{x};\mathbf{w}) = w_0 + \mathbf{x}^T\mathbf{w}_1$$

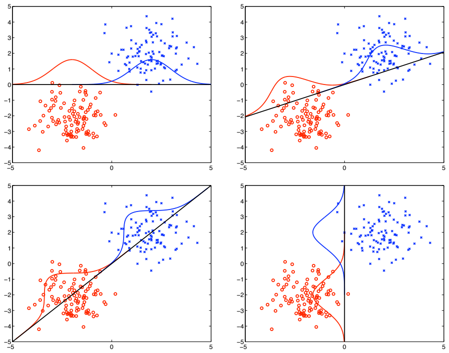
projects each point $\mathbf{x} = [x_1\ x_2]^T$ to a line parallel to $\mathbf{w}_1$.



- We can study how well the projected points $\{z_1, \ldots, z_n\}$, viewed as functions of $\mathbf{w}_1$, are separated across the classes.

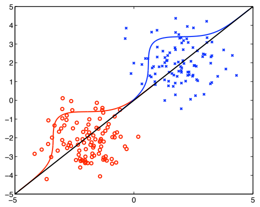
Projection and classification

- By varying $\mathbf{w}_1$ we get different levels of separation between the projected points



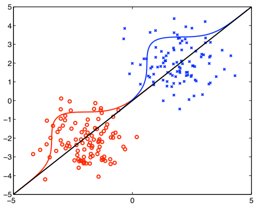
Optimizing the projection

- We would like to find $\mathbf{w}_1$ that somehow maximizes the separation of the projected points across classes



- We can quantify the separation (overlap) in terms of means and variances of the resulting 1-dimensional class distributions

Fisher linear discriminant: preliminaries



- Class descriptions in $\mathcal{R}^d$:
  class 0: n_0 samples, mean μ_0, covariance Σ_0
  class 1: n_1 samples, mean μ_1, covariance Σ_1
- Projected class descriptions in $\mathcal{R}$:
  class 0: n_0 samples, mean $\mu_0^T\mathbf{w}_1$, variance $\mathbf{w}_1^T\Sigma_0\mathbf{w}_1$
  class 1: n_1 samples, mean $\mu_1^T\mathbf{w}_1$, variance $\mathbf{w}_1^T\Sigma_1\mathbf{w}_1$

Fisher linear discriminant

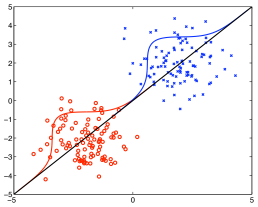
- Estimation criterion: we find $\mathbf{w}_1$ that maximizes

$$\begin{aligned} J_{Fisher}(\mathbf{w}) &= \frac{(\text{Separation of projected means})^2}{\text{Sum of within class variances}} \\ &= \frac{(\mu_1^T\mathbf{w}_1 - \mu_0^T\mathbf{w})^2}{n_1\mathbf{w}_1^T\Sigma_1\mathbf{w}_1 + n_0\mathbf{w}_1^T\Sigma_0\mathbf{w}_1} \end{aligned}$$

- The solution (class separation)

$$\hat{\mathbf{w}}_1 \propto (n_1\Sigma_1 + n_0\Sigma_0)^{-1}(\mu_1 - \mu_0)$$

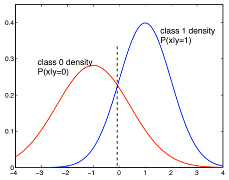
is decision theoretically optimal for two normal populations with equal covariances ($\Sigma_1 = \Sigma_0$)



Background: simple decision theory

- Suppose we know the class-conditional densities $p(\mathbf{x}|y)$ for $y = 0, 1$ as well as the overall class frequencies $P(y)$.

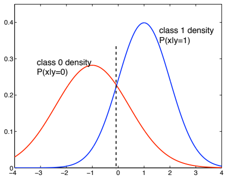
  How do we decide which class a new example $\mathbf{x}'$ belongs to so as to minimize the overall probability of error?



Background: simple decision theory

- Suppose we know the class-conditional densities $p(\mathbf{x}|y)$ for $y = 0, 1$ as well as the overall class frequencies $P(y)$.

  How do we decide which class a new example $\mathbf{x}'$ belongs to so as to minimize the overall probability of error?



The minimum probability of error decisions are given by

$$y' = \arg\max_{y=0,1} \{ p(\mathbf{x}'|y)P(y) \} = \arg\max_{y=0,1} \{ P(y|\mathbf{x}') \}$$

Logistic regression

- The optimal decisions are based on the posterior class probabilities $P(y|\mathbf{x})$. For binary classification problems, we can write these decisions as

$$y = 1 \text{ if } \quad \log \frac{P(y=1|\mathbf{x})}{P(y=0|\mathbf{x})} > 0$$

  and $y = 0$ otherwise.

Logistic regression

- The optimal decisions are based on the posterior class probabilities $P(y|\mathbf{x})$. For binary classification problems, we can write these decisions as

$$y = 1 \text{ if } \quad \log \frac{P(y=1|\mathbf{x})}{P(y=0|\mathbf{x})} > 0$$

  and $y = 0$ otherwise.

- We generally don't know $P(y|\mathbf{x})$ but we can parameterize the possible decisions according to

$$\log \frac{P(y=1|\mathbf{x})}{P(y=0|\mathbf{x})} = f(\mathbf{x};\mathbf{w}) = w_0 + \mathbf{x}^T\mathbf{w}_1$$

Logistic regression cont'd

- Our log-odds model

$$\log \frac{P(y=1|\mathbf{x})}{P(y=0|\mathbf{x})} = w_0 + \mathbf{x}^T\mathbf{w}_1$$

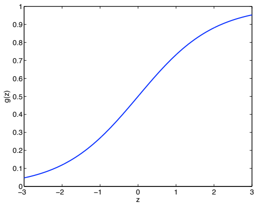
  gives rise to a specific form for the conditional probability over the labels (the logistic model):

$$P(y=1|\mathbf{x},\mathbf{w}) = g\left(\, w_0 + \mathbf{x}^T\mathbf{w}_1 \right)$$

  where

$$g(z) = (1 + \exp(-z))^{-1}$$

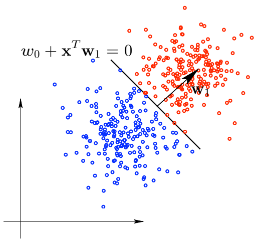
  is a logistic "squashing function" that turns linear predictions into probabilities



Logistic regression: decisions

- Logistic regression models imply a linear decision boundary

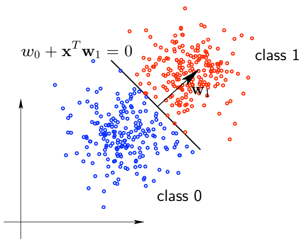
$$\log \frac{P(y=1|\mathbf{x})}{P(y=0|\mathbf{x})} = w_0 + \mathbf{x}^T\mathbf{w}_1 = 0$$



Logistic regression: decisions

- Logistic regression models imply a linear decision boundary

$$\log \frac{P(y=1|\mathbf{x})}{P(y=0|\mathbf{x})} = w_0 + \mathbf{x}^T\mathbf{w}_1 = 0$$



Fitting logistic regression models

- As with the linear regression models we can fit the logistic models using the maximum (conditional) log-likelihood criterion

$$l(D;\mathbf{w}) = \sum_{i=1}^{n} \log P(y_i|\mathbf{x}_i,\mathbf{w})$$

where

$$P(y=1|\mathbf{x},\mathbf{w}) = g\left(w_0 + \mathbf{x}^T\mathbf{w}_1\right)$$

- The log-likelihood function $l(D;\mathbf{w})$ is a *jointly concave* function of the parameters $\mathbf{w}$; a number of optimization techniques are available for finding the maximizing parameters

About the ML solution

- If we set the derivatives of the log-likelihood with respect to the parameters to zero

$$\begin{aligned}\frac{\partial}{\partial w_0} l(D;\mathbf{w}) &= \sum_{i=1}^{n} \big(y_i - P(y_i=1|\mathbf{x}_i,\mathbf{w})\big) = 0 \\ \frac{\partial}{\partial w_j} l(D;\mathbf{w}) &= \sum_{i=1}^{n} \big(y_i - P(y_i=1|\mathbf{x}_i,\mathbf{w})\big)x_{ij} = 0\end{aligned}$$

the optimality conditions again require that the prediction errors

$$\epsilon_i = \big(y_i - P(y_i=1|\mathbf{x}_i,\mathbf{w})\big), \quad i = 1,\ldots,n$$

corresponding to the optimal setting of the parameters are uncorrelated with any linear function of the inputs.

Stochastic gradient ascent

- We can try to maximize the log-likelihood in an *on-line* or incremental fashion.

Given each training input $\mathbf{x}_i$ and the binary (0/1) label y_i, we can change the parameters $\mathbf{w}$ slightly to increase the corresponding log-probability

$$\begin{aligned}\mathbf{w} &\leftarrow \mathbf{w} + \eta \frac{\partial}{\partial \mathbf{w}} \log P(y_i|\mathbf{x}_i,\mathbf{w}) \\ &= \mathbf{w} + \eta\, \underbrace{\big(y_i - P(y_i=1|\mathbf{x}_i,\mathbf{w})\big)}_{\text{prediction error}} \begin{bmatrix} 1 \\ \mathbf{x}_i \end{bmatrix}\end{aligned}$$

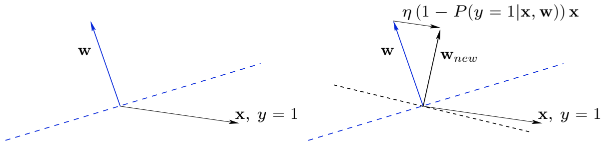
where η is the *learning rate*.

- The resulting update is similar to the mistake driven algorithm discussed earlier; examples that are already confidently classified do not lead to any significant updates

Stochastic gradient ascent cont'd

- To understand the procedure graphically we focus on a single example and omit the bias term w_0

$$\mathbf{w} \;\leftarrow\; \mathbf{w} + \eta\, \underbrace{\big(y_i - P(y_i=1|\mathbf{x}_i,\mathbf{w})\big)}_{\text{prediction error}}\, \mathbf{x}_i$$



Gradient ascent of the log-likelihood

- We can also perform gradient ascent steps on the log-likelihood of all the training labels given examples at the same time. In other words,

$$\begin{aligned}\mathbf{w} &\leftarrow \mathbf{w} + \eta \frac{\partial}{\partial \mathbf{w}} l(D;\mathbf{w}) \\ &= \mathbf{w} + \eta \sum_{i=1}^{n} \big(y_i - P(y_i=1|\mathbf{x}_i,\mathbf{w})\big) \begin{bmatrix} 1 \\ \mathbf{x}_i \end{bmatrix}\end{aligned}$$

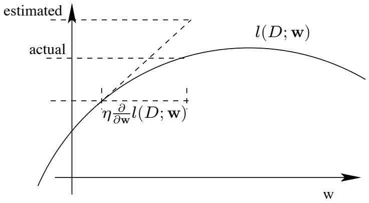
Still need to figure out a way to set the learning rate to guarantee convergence.

Setting the learning rate: Armijo rule

The learning rate in

$$\mathbf{w} \leftarrow \mathbf{w} + \eta \frac{\partial}{\partial \mathbf{w}} l(D;\mathbf{w})$$

"should" satisfy



$$l\Big(D;\, \overbrace{\mathbf{w} + \eta \frac{\partial}{\partial \mathbf{w}} l(D;\mathbf{w})}^{\mathbf{w}_{new}}\Big) - l(D;\mathbf{w}) \geq \eta \cdot \frac{1}{2} \| \frac{\partial}{\partial \mathbf{w}} l(D;\mathbf{w}) \|^2$$

The Armijo rule suggests finding the smallest integer m such that $\eta = \eta_0 q^m,\ q < 1$ is a valid choice in this sense.

- Armijo rule is guaranteed to converge to a (local) maximum under certain technical assumptions

Additive models and classification

- Similarly to linear regression models, we can extend the logistic regression models to additive (logistic) models

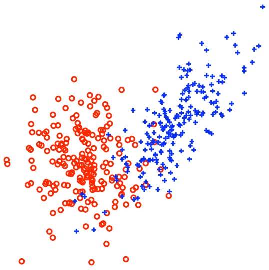
$$P(y = 1|\mathbf{x}, \mathbf{w}) = g\,(\,w_0 + w_1 \phi_1(\mathbf{x}) + \ldots w_m \phi_m(\mathbf{x})\,)$$

- As before we are free to choose the basis functions $\phi_i(\mathbf{x})$ to capture relevant properties of any specific classification problem
- Since we also over-fit easily, we can use leave-one-out cross-validation (in terms of log-likelihood or classification error) to estimate the generalization performance

$$\text{CV log-likelihood} = \frac{1}{n} \sum_{i=1}^{n} \log P(y_i|\mathbf{x}_i, \hat{\mathbf{w}}^{-i})$$

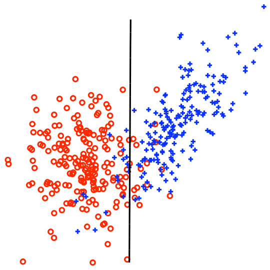
Logistic regression example

- Simple binary classification problem in $\mathcal{R}^2$



Logistic regression example

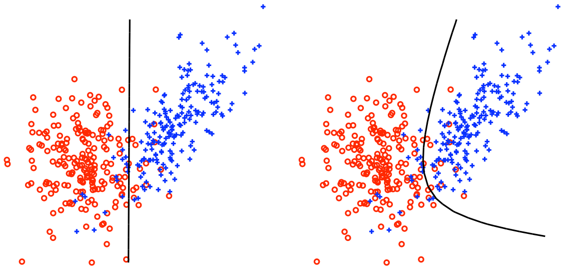
- Simple binary classification problem in $\mathcal{R}^2$



linear

Logistic regression example

- Simple binary classification problem in $\mathcal{R}^2$

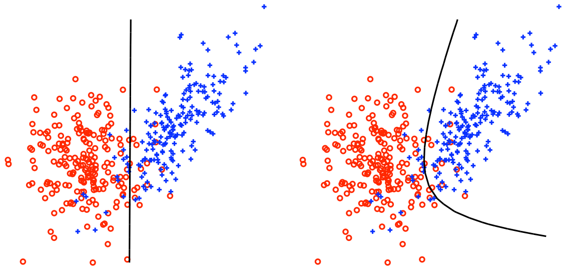


linear

quadratic

Logistic regression example

- Simple binary classification problem in $\mathcal{R}^2$



linear
CV $= -0.216$

quadratic
CV $= -0.202$