Linear Predictors Part 1

COMPSCI 371D — Machine Learning

Outline

Definitions

- A linear *regressor* fits an affine function to the data
  $y \approx h(\mathbf{x}) = b + \mathbf{w}^T\mathbf{x} \quad \text{for} \quad \mathbf{x} \in \mathbb{R}^d$ and $y \in \mathbb{R}$
- A linear, binary *classifier* separates the data in $X \subseteq \mathbb{R}^d$ corresponding to the two classes in $Y = \{c_0, c_1\}$ with a hyperplane
- The actual data can be separated only if it is linearly separable (!)
- Multi-class linear classifiers separate any two classes with a hyperplane
- The resulting decision regions are convex and simply connected (polyhedra)

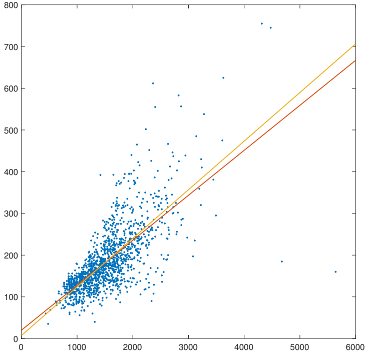
Properties of Linear Predictors

- Linear Predictors...
  - ...have a very small $\mathcal{H}$ with $d + 1$ parameters
    (resist overfitting)
  - ... are trained by solving a convex optimization problem
    (global optimum)
  - ... are fast at inference time
    (and training is not too slow)
  - ... work well if the data is close to linearly separable

The Least-Squares Linear Regressor

- *Déjà vu*: Polynomial regression with $k = 1$
  $y \approx h_{\mathbf{v}}(\mathbf{x}) = b + \mathbf{w}^T\mathbf{x}$ for $\mathbf{x} \in \mathbb{R}^d$
- Parameter vector $\mathbf{v} = \begin{bmatrix} b \\ \mathbf{w} \end{bmatrix} \in \mathbb{R}^{d+1}$
  $\mathcal{H}$ isomorphic to $\mathbb{R}^m$ with $m = d + 1$
- "Least Squares:" $\ell(y, \hat{y}) = (y - \hat{y})^2$
- $\hat{\mathbf{v}} = \arg\min_{\mathbf{v}\in\mathbb{R}^m} L_T(\mathbf{v})$
- Risk $L_T(\mathbf{v}) = \frac{1}{N}\sum_{n=1}^{N} \ell(y_n, h_{\mathbf{v}}(\mathbf{x}_n))$
- We know how to solve this

Linear Regression Example



- Left: All of Ames. Residual $\sqrt{\text{Risk}}$: \$55,800
- Right: One Neighborhood. Residual $\sqrt{\text{Risk}}$: \$23,600
- Left, yellow: Ignore two largest homes

Binary Classification by Logistic Regression

$$Y = \{c_0, c_1\}$$

- Multi-class case later
- The *logistic-regression classifier* is a classifier!
- A *linear* classifier implemented through regression
- The *logistic* is a particular function

Score-Based Classifiers

$Y = \{c_0, c_1\}$

- Think of c_0, c_1 as *numbers*: $Y = \{0, 1\}$
- We saw the idea of level sets:

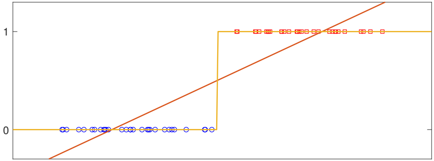
  Regress a *score* function $s(\mathbf{x})$ such that

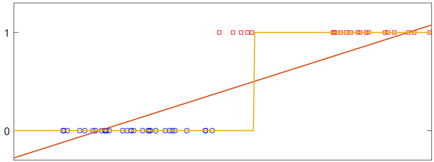
  $s(\mathbf{x})$ is large where $y = 1$, small where $y = 0$
- Threshold s to obtain a classifier:

  $h(\mathbf{x}) = \begin{cases} c_0 & \text{if } s(\mathbf{x}) \leq \text{threshold} \\ c_1 & \text{otherwise.} \end{cases}$
- A linear classifier implemented through regression

Idea 1

- $s(\mathbf{x}) = b + \mathbf{w}^T\mathbf{x}$ and quadratic loss
- Red: score function. Orange: $h(\mathbf{x})$

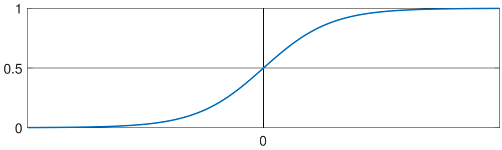




- Not so good!
- The quadratic loss is not what we care about
- A line does not approximate a step well
- Why not fit a step function and use the 0-1 loss?
- No gradient. NP-hard unless the data is separable or $d = 1$

Idea 2

- How about a "soft step?" The *logistic function* $f(x) \stackrel{\text{def}}{=} \frac{1}{1+e^{-x}}$



- Distant points are no longer a big problem
- If a true step moves, the risk does not change until a data point flips label
- If the logistic function moves ($f(x) \to f(x-s)$), the risk changes gradually
- We have a nonzero gradient almost everywhere!
- The optimization problem is no longer combinatorial

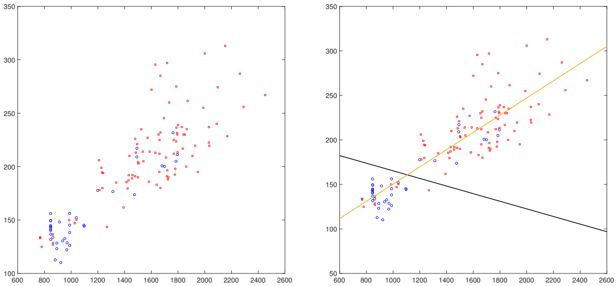
What is a Logistic Function in d Dimensions?

- We want a *linear* classifier
- The level crossing must be a hyperplane
- Level crossing: Solution to $s(\mathbf{x}) = 1/2$
- Shape of the crossing depends on s
- Compose an affine $a(\mathbf{x}) = c + \mathbf{u}^T\mathbf{x}$ $\quad (a : \mathbb{R}^d \rightarrow \mathbb{R})$

  ...with a monotonic $f(a)$ that crosses $1/2$ $\quad (f : \mathbb{R} \rightarrow \mathbb{R})$

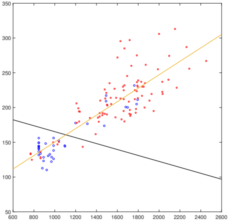
  $s(\mathbf{x}) = f(a(\mathbf{x})) = f(c + \mathbf{u}^T\mathbf{x})$
- Then, if $f(\alpha) = 1/2$, the equation $s(\mathbf{x}) = 1/2$ is the same as $c + \mathbf{u}^T\mathbf{x} = \alpha$
- A hyperplane!
- Let f be the logistic function

Example



- Gold line: Regression problem $\mathbb{R} \rightarrow \mathbb{R}$
- Black line: Classification problem $\mathbb{R}^2 \rightarrow \mathbb{R}$
  (result of running a logistic-regression classifier)
- Labels: Good (red squares, $y = 1$) or poor quality (blue circles, $y = 0$) homes
- **All that matters is how far a point is from the black line**

A Probabilistic Interpretation



- **All that matters is how far a point is from the black line**
- Convert activation $a(\mathbf{x})$ to a signed distance $\Delta(\mathbf{x})$
- $s(\mathbf{x}) = f(\Delta(\mathbf{x}))$ where Δ is a *signed* distance
- We could interpret the score $s(\mathbf{x})$ as "the probability that $y = 1$:" $\quad f(\Delta(\mathbf{x})) = \mathbb{P}[y = 1]$
- (...or as "$1-$ the probability that $y = 0$")
  $\lim_{\Delta \to -\infty} \mathbb{P}[y = 1] = 0 \qquad \lim_{\Delta \to \infty} \mathbb{P}[y = 1] = 1$
  $\Delta = 0 \Rightarrow \mathbb{P}[y = 1] = 1/2$ (just like the logistic function)

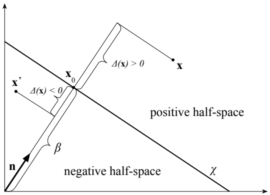
Ingredients for the Regression Part

- Determine the distance Δ of a point $\mathbf{x} \in X$ from a hyperplane χ, and the side of χ on which the point is on (Geometry: *affine functions* as unscaled, signed distances)
- Specify a monotonically increasing function$f$ that turns $\Delta(\mathbf{x})$ into a probability $p = f(\Delta(\mathbf{x}))$ (Choice based on convenience: the *logistic function*)
- Define a loss function $\ell(y, p)$ that measures how good p is given the true label y (Convenience again: choose ℓ so that $\ell(y, f(\Delta(\mathbf{x})))$ is a *convex* risk: The *cross-entropy loss*)

Normal to a Hyperplane

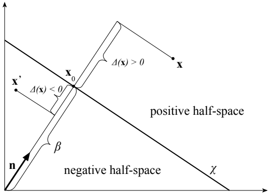
- Hyperplane χ: $\quad b + \mathbf{w}^T\mathbf{x} = 0 \quad$ (w.l.o.g. $b \leq 0$)
  $\mathbf{a}_1, \mathbf{a}_2 \in \chi \Rightarrow \mathbf{c} = \mathbf{a}_1 - \mathbf{a}_2$ parallel to χ
- Subtract $b + \mathbf{w}^T\mathbf{a}_1 = 0$ from $b + \mathbf{w}^T\mathbf{a}_2 = 0$
- Obtain $\mathbf{w}^T\mathbf{c} = 0$ for *any* $\mathbf{a}_1, \mathbf{a}_2 \in \chi$
- **w is perpendicular to** χ

Distance of a Hyperplane from the Origin



- Unit-norm version of **w**: $\mathbf{n} = \frac{\mathbf{w}}{\|\mathbf{w}\|}$
- Rewrite χ: $b + \mathbf{w}^T\mathbf{x} = 0$ (w.l.o.g. $b \leq 0$) as $\mathbf{n}^T\mathbf{x} = \beta$ where $\beta = -\frac{b}{\|\mathbf{w}\|} \geq 0$
- Line along **n**: $\mathbf{x} = \alpha\mathbf{n}$ for $\alpha \in \mathbb{R}$ (parametric form) α is the signed distance from the origin
- Replace into eq. for χ: $\alpha\mathbf{n}^T\mathbf{n} = \beta$ that is, $\alpha = \beta \geq 0$
- In particular, $\mathbf{x}_0 = \beta\mathbf{n}$
- **β is the distance of χ from the origin**

Signed Distance of a Point from a Hyperplane



$\mathbf{n}^T\mathbf{x} = \beta$ where $\beta = -\frac{b}{\|\mathbf{w}\|} \geq 0$ and $\mathbf{n} = \frac{\mathbf{w}}{\|\mathbf{w}\|}$

$\mathbf{x}_0 = \beta\mathbf{n}$

- In one half-space, $\mathbf{n}^T\mathbf{x} \geq \beta$
- Distance of $\mathbf{x}$ from χ is $\mathbf{n}^T\mathbf{x} - \beta \geq 0$
- In other half-space, $\mathbf{n}^T\mathbf{x}' \leq \beta$
- Distance of $\mathbf{x}'$ from χ is $\beta - \mathbf{n}^T\mathbf{x}' \geq 0$
- On decision boundary, $\mathbf{n}^T\mathbf{x} = \beta$
- $\Delta(\mathbf{x}) \stackrel{\text{def}}{=} \mathbf{n}^T\mathbf{x} - \beta$ is the **signed distance of x from the hyperplane**

Summary

If $\mathbf{w}$ is nonzero (which it has to be), the distance from the origin of the hyperplane χ with equation $b + \mathbf{w}^T\mathbf{x} = 0$ is

$$\beta \stackrel{\text{def}}{=} \frac{|b|}{\|\mathbf{w}\|}$$

(a nonnegative number) and the quantity

$$\Delta(\mathbf{x}) \stackrel{\text{def}}{=} \frac{b + \mathbf{w}^T\mathbf{x}}{\|\mathbf{w}\|}$$

is the *signed distance* of point $\mathbf{x} \in X$ from hyperplane χ. Specifically, the distance of $\mathbf{x}$ from χ is $|\Delta(\mathbf{x})|$, and $\Delta(\mathbf{x})$ is nonnegative if and only if $\mathbf{x}$ is on the side of χ pointed to by $\mathbf{w}$. Let us call that side the *positive half-space* of χ.