Machine Learning (CS 181):
6. Linear Classification and the Perceptron

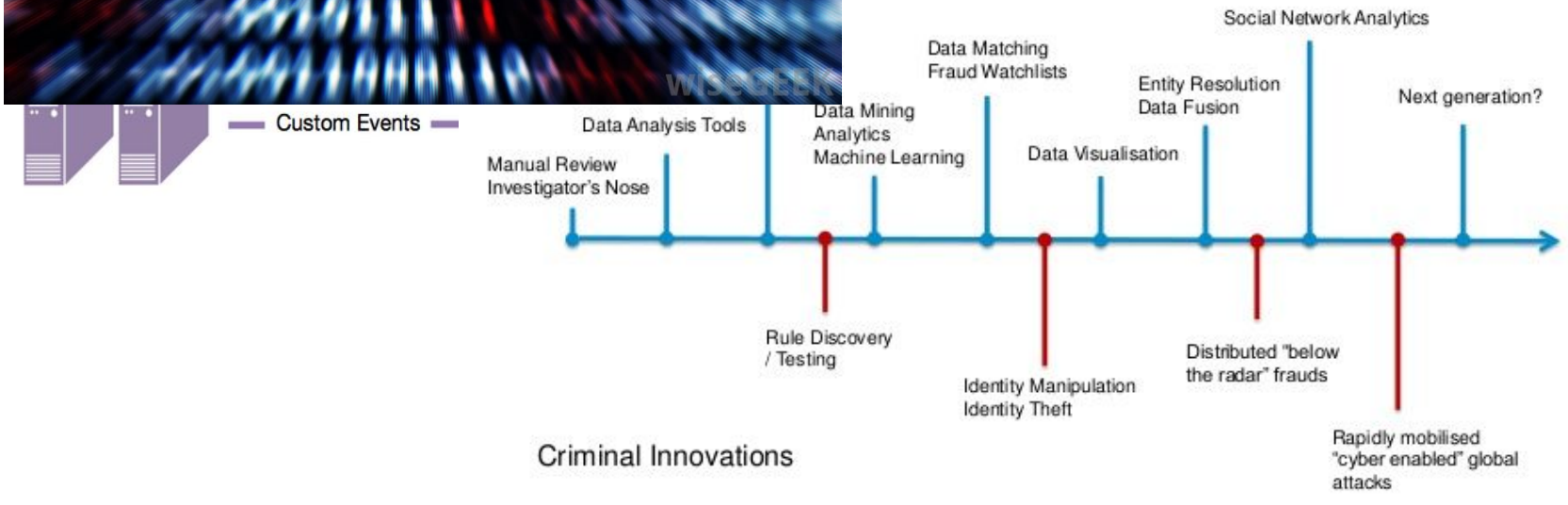
Contents

Linear Classification

Fraud Scores

BAE SYSTEMS

ection Technologies

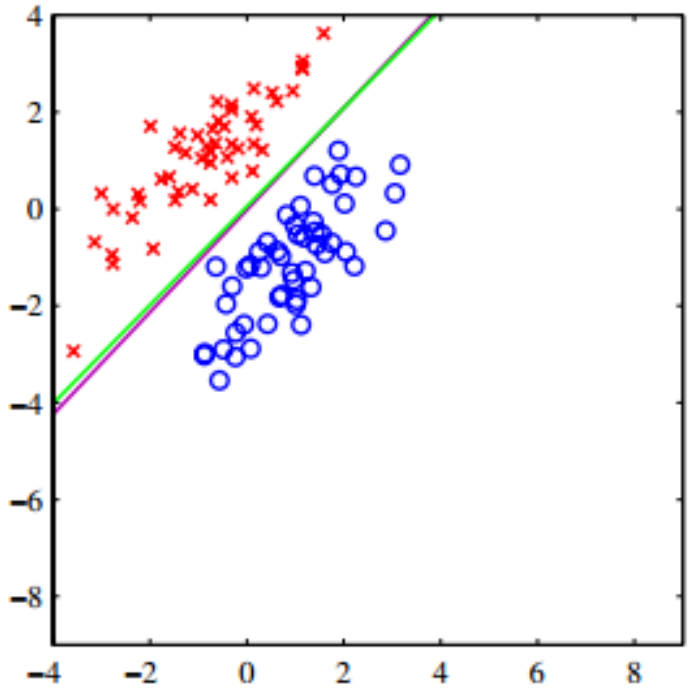


thenextweb.com

Contents

Binary Classification

- Output space $\mathcal{Y}$ is a fixed set of classes.
- Simplest case $\mathcal{Y} = \{-1, 1\}$ (red/blue)



Binary Classification

Multiclass Classification

- Output space $\mathcal{Y}$ is a fixed set of classes.
- Made up of c discrete classes $\{C_1, \ldots C_c\}$

e.g.

- Image recognition
- Disease diagnostics
- Product recommendation

Example: Digit Classification

- Data: Handwritten US zip codes

Contents

(Binary) Linear Classification

$$h(\mathbf{x}; \mathbf{w}) = \mathbf{w}^\top \mathbf{x} + w_0$$

- w_0; threshold or bias
- $h(\mathbf{x}; \mathbf{w}) > 0$; prediction

Decision Boundary

Point where model score is balanced.

$$h(\mathbf{x}; \mathbf{w}) = 0 \tag{1}$$

In the case of linear model, hyperplane defined by:

$$\mathbf{w}^{\mathsf{T}}\mathbf{x} + w_0 = 0$$

Decision Boundary Orientation

Orientation is determined by $\mathbf{w}$

Let $\mathbf{x}_1$ and $\mathbf{x}_2$ be on decision boundary:

$$\begin{aligned}\mathbf{w}^\top(\mathbf{x}_1 - \mathbf{x}_2) &= \mathbf{w}^\top\mathbf{x}_1 - \mathbf{w}^\top\mathbf{x}_2 \\ &= -w_0 + w_0 \\ &= 0\end{aligned}$$

Therefore hyperplane orthogonal to $\mathbf{w}$.

Decision Boundary Location

Consider point in plane closest to origin

- Orthogonal to $\mathbf{w}$.
- $c\frac{\mathbf{w}}{||\mathbf{w}||}$ for some unknown constant c

$$\mathbf{w}^\top \left(c\frac{\mathbf{w}}{||\mathbf{w}||} \right) + w_0 = 0$$

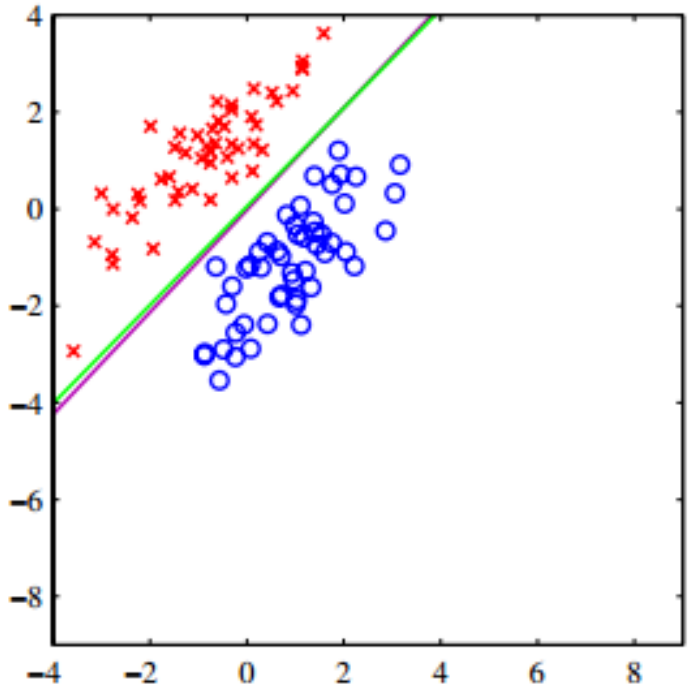
$$c||\mathbf{w}|| + w_0 = 0$$

$$c = -\frac{w_0}{||\mathbf{w}||}$$

Distance from origin determined by $w_0/||\mathbf{w}||$;

Separating Hyperplane

$$\mathbf{w}^{\mathsf{T}}\mathbf{x} + w_0 = 0$$



Binary Classification

Linear Separability

(Requirement on data).

Bases

TODO:

Contents

Least Squares?

Preliminary Idea: treat using least squares loss

$$\mathcal{L}(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^{n} \big(y_i - h(\mathbf{x}_i; \mathbf{w})\big)^2$$

- Target y_i always $\{-1, 1\}$.
- What goes wrong?

Least Squares Regression Demo

- Least Squares is sensitive to easy points.

Show Demo

Perceptron

Perceptron History

- Classic classification idea due to Rosenblatt in late 1950's.
- Built into hardware.
- Minsky and Papert wrote a book that showed that single layer perceptrons couldn't do more than linear separability.
- They incorrectly conjectured that this would also be true for multi-layer systems.

0/1 Activation Function

$$g_{0/1}(a) = \begin{cases} +1 & a \geq 0 \\ -1 & o.w. \end{cases}$$

Classifier

$$\hat{y} = f(h(\mathbf{x}, \mathbf{w}))$$

.

Loss Function

Ideal loss function: penalty for misclassification

$$\mathcal{L}(\mathbf{w}) = -\sum_{i=1}^{n} 1(y_i == \hat{y}_i) y_i g_{0/1}(\mathbf{w}^\top \mathbf{x})$$

0/1 Loss Function

ReLU Activatino

$$g_{0/1}(a) = \begin{cases} +1 & a \geq 0 \\ -1 & o.w. \end{cases}$$

Perceptron Loss

Ideal loss function: penalty for misclassification

$$\mathcal{L}(\mathbf{w}) = -\sum_{i=1}^{n} 1(y_i == \hat{y}_i) y_i \mathbf{w}^\top \mathbf{x}$$

Penalty scales linearly to mistakes.

Gradient Descent

Recall:

Stochastic Gradient Descent

Gradient of Perceptron Loss

$$\frac{\partial}{\partial \mathbf{w}} \mathcal{L}(\mathbf{w}) = \frac{\partial}{\partial \mathbf{w}} - \sum_{i=1}^{n} y_i \mathbf{w}^\top \mathbf{x}$$

$$= \frac{\partial}{\partial \mathbf{w}} - \sum_{i=1}^{n} y_i \mathbf{x}$$

$$\frac{\partial}{\partial \mathbf{w}} \mathcal{L}(\mathbf{w}) = -y_i \mathbf{x}$$

Stochastic version, just one incorrect datum and learning rate η:

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} + \eta y_i \mathbf{x}_i$$

Because $\mathbf{w}$ doesn't change with scale, we can take $\eta = 1$ wlog.

Perceptron Algorithm

Loop: Iterate over the data: If correct, do nothing. If incorrect, add $y_i \mathbf{x}_i$ to weights.

Guarantee

Guaranteed to converge if data are linearly separable.

Perceptron Demo

Contents

Generative Classification View

$$p(y = 1, |\mathbf{x}, \mathbf{w}) \propto p(y)p(\mathbf{x}|y)$$

- Prior probability $p(y)$
- Likelihood $p(\mathbf{x}|y)$

Prior

Choice of prior can depend on problem format,

- In binary case, we will use Bernoulli prior,

$$p(y = 1; \pi) = \pi \quad p(y = 0; \pi) = 1 - \pi$$

  or more compactly

$$p(y; \pi) = \pi^y (1 - \pi)^{1-y}$$

- In multiclass (HW), can use categorical prior,

$$p(y = \boldsymbol{\delta}_k; \boldsymbol{\pi}) = \pi_k$$

Likelihood Model

Choice of likelihood depends on data and modeling assumptions, Select parameteric model for likelihood

- For discrete or indicator data, simplest to use Naive Bayes (next class).
- Use continuous data, use multivariate Gaussian (HW)

$$p(\mathbf{x}|y=0) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$$

$$p(\mathbf{x}|y=1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$$

Maximum Likelihood

Model is fit using the same approach as for regression.

1. Decide on generating process
2. Fix the parameterization of the model
3. Maximize likelihood of the data.

$$\min_{\pi,\mathbf{w}} \mathcal{L}(\pi, \mathbf{w}) = \min_{\pi,\mathbf{w}} -\sum_{i=1}^{n} \ln p(y_i; \pi) + \ln p(\mathbf{x}_i | y_i; \mathbf{w})$$

Contents

F-Score

Area Under Curve

Cost Functions