Perceptron

Gautam Kamath

Binary Classification

- Given: $(x_1, y_1), (x_2, y_2), \ldots$
  - x_i: "feature vector." Often $x_i \in \mathbf{R}^d$
  - y_i: "label." For binary classification, $y_i \in \{-1, +1\}$
    - You may also see $y_i \in \{0,1\}$
- Idea: "Learn" a function h such that $h(x) = y$
  - Given a feature vector, what is the label?

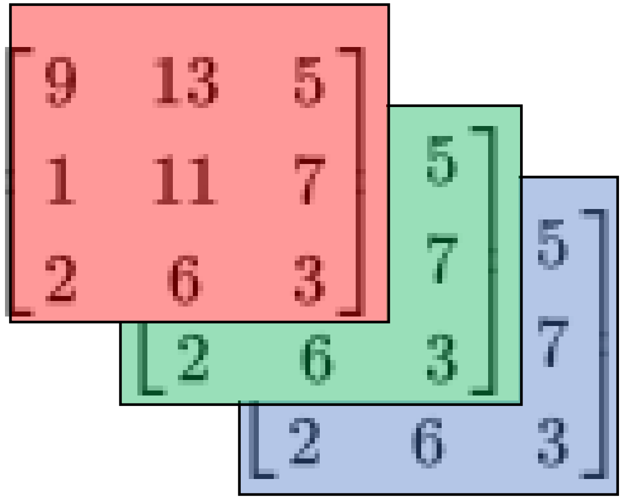
Pass class example

- Feature vector x_i: (homework mark, exam mark)
- Label y_i: Passed the class?
- Dataset (draw on board):
  - $\big((90, 80), +1\big), \big((40, 30), -1\big), \big((50, 40), -1\big)$
- Can always memorize training data
  - But we want to *generalize*!
  - $\big((50,60), ?\,\big)$

Image Classifier example

- Feature vector x_i:

=



- Label y_i: Is this a panda or not?

Statistical Learning

- Setup: Given $(x_1, y_1), \ldots, (x_n, y_n) \sim_{i.i.d.} P$
  - Independent and identically distributed – may be limiting, but common assn
- Goal: Learn $h : \mathbf{R}^d \rightarrow \{-1, +1\}$ such that $\Pr_{(x,y)\sim P}[h(x) = y]$ is large
  - Importantly, P is unknown (otherwise could use the "Bayes classifier")
  - What happens if we get something "out of distribution"?
    - (Draw two clusters on board, wrong label and unpredictable examples)

Online Learning

- Receive examples one by one and make predictions as we go
- At each time $t = 1,2, \ldots$
  - Receive feature vector x_i
  - Choose prediction function h_i, predict label $\widehat{y}_i = h_i(x_i)$
  - View true label y_i. Suffer mistake if $y_i \neq \widehat{y}_i$.

Intuition of Perceptron

- (Draw pass class example on board, add more points)
- Plausible grading scheme: if average of hw and exams > 0.5, pass.
- Equivalently: if $0.5 \cdot \text{homework} + 0.5 \cdot \text{exams} > 0.5$, pass.
  - Or: if $0.5 \cdot \text{homework} + 0.5 \cdot \text{exams} - 0.5 > 0$, pass.
- Rewrite as: $\text{sign}(\langle (0.5, 0.5), (\text{homework}, \text{exams}) \rangle - 0.5)$
  - Dot product notation: $\langle u, v \rangle = \sum_i u_i v_i$
  - $\text{sign}(a) = 1$ if $a > 0$, $= -1$ otherwise
- Let $x = (\text{homework}, \text{exams})$: $y = \text{sign}(\langle (0.5, 0.5), x \rangle - 0.5)$
- Implicit assumption in perceptron: there is some linear separator

Perceptron Algorithm

Algorithm: The Perceptron (Rosenblatt 1958)

Input: Dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i) \in \mathbb{R}^d \times \{\pm 1\} : i = 1, \ldots, n\}$, initialization $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$, threshold $\delta \geq 0$

Output: approximate solution $\mathbf{w}$ and b

for $t = 1, 2, \ldots$ **do**
  receive training example index $I_t \in \{1, \ldots, n\}$ `// the index` I_t `can be random`
  if $y_{I_t}(\mathbf{w}^\top \mathbf{x}_{I_t} + b) \leq \delta$ **then**
    $\mathbf{w} \leftarrow \mathbf{w} + y_{I_t} \mathbf{x}_{I_t}$ `// update only after making a "mistake"`
    $b \leftarrow b + y_{I_t}$

- Weight vector w, bias b
- Typically initialize $w = \vec{0}, b = 0$, set $\delta = 0$
- "Lazy" updates: only change if a prediction is wrong
- (Examples on board. $\big((1,1), 1\big)$ and $\big((-1,-1), -1\big)$, change to $\left(\left(-\frac{1}{4}, -\frac{1}{4}\right), -1\right)$)

Notation: Padding + Pre-Multiplication

- Goal: find w, b such that $y_i = \operatorname{sign}(\langle w, x_i \rangle + b)$ for all $i \in [n]$
  - $y_i = \operatorname{sign}(\langle (w, b), (x_i, 1) \rangle)$ ("padding trick")
  - $y_i = \operatorname{sign}(\langle z, (x_i, 1) \rangle)$ (Let $z = (w, b)$ to simplify notation)
  - $y_i \langle z, (x_i, 1) \rangle > 0$ (equivalent formulation)
  - $\langle z, y_i (x_i, 1) \rangle > 0$
  - $\langle z, a_i \rangle > 0$ (Let $a_i = y_i (x_i, 1)$ to simplify notation)
- Let A be the matrix with rows a_i (draw on board)
- Then goal is $Az > \vec{0}$ (entrywise)

Linear Separability

- (Draw picture of separable and non-separable datasets on board)
- There exists $z = (w, b)$ such that $\langle a_i, z \rangle \geq s > 0$ for all $i \in [n]$, for some constant s
- Equivalently: $Az \geq s\vec{1}$, where $s > 0$
- (Draw picture of what the s means)

Error Bound

- Theorem: Suppose there exists some weight vector and bias $z = (w, b)$ such that $Az \geq s\vec{1}$. Then perceptron will correctly classify the entire dataset after at most $R^2 \|z\|_2^2 / s^2$ mistakes, where $R = \max \|a_i\|_2$.
  - $\|x\|_2$ is the ℓ_2-norm of x: $\sqrt{\sum_i x_i^2}$, measures how "big" a vector is
- (Draw picture with intuition as to why R shows up: one with big R and one with small R)

Error Bound (continued)

- Theorem (informal): If $\exists z, s$ such that $Az \geq s\vec{1}$, perceptron makes at most $R^2\|z\|_2^2/s^2$ mistakes, where $R = \max\|a_i\|_2$.
  - But there may be many valid z, s. Scaling: if $Az \geq s\vec{1}$, then $A(2z) \geq (2s)\vec{1}$.
  - (Draw picture on board of non-uniqueness)
  - Pick the “best” one to minimize $\|z\|_2^2/s^2$ and thus the number of mistakes

Error Bound (continued)

- Theorem (informal): If $\exists z, s$ such that $Az \geq s\vec{1}$, perceptron makes at most $R^2\|z\|_2^2/s^2$ mistakes, where $R = \max\|a_i\|_2$.
  - Pick the "best" one to minimize $\|z\|_2^2/s^2$ and thus the number of mistakes

$$\min_{(z,s):Az\geq s\vec{1}} \frac{\|z\|_2^2}{s^2}$$

Error Bound (continued)

- Theorem (informal): If $\exists z, s$ such that $Az \geq s\vec{1}$, perceptron makes at most $R^2\|z\|_2^2/s^2$ mistakes, where $R = \max\|a_i\|_2$.
  - Pick the "best" one to minimize $\|z\|_2^2/s^2$ and thus the number of mistakes

$$\min_{(z,s):Az\geq s\vec{1}} \frac{\|z\|_2^2}{s^2} = \min_{(z,s):\|z\|_2=1,Az\geq s\vec{1}} \frac{1}{s^2}$$

$$= \frac{1}{\left(\max_{(z,s):\|z\|_2=1,Az\geq s\vec{1}} s\right)^2}$$

Error Bound (continued)

- Theorem (informal): If $\exists z, s$ such that $Az \geq s\vec{1}$, perceptron makes at most $R^2\|z\|_2^2/s^2$ mistakes, where $R = \max\|a_i\|_2$.
  - Pick the "best" one to minimize $\|z\|_2^2/s^2$ and thus the number of mistakes

$$\min_{(z,s):Az\geq s\vec{1}} \frac{\|z\|_2^2}{s^2} = \min_{(z,s):\|z\|_2=1, Az\geq s\vec{1}} \frac{1}{s^2}$$

$$= \frac{1}{\left(\max_{(z,s):\|z\|_2=1, Az\geq s\vec{1}} s\right)^2} = \frac{1}{\left(\max_{\|z\|_2=1} \min_i \langle a_i, z\rangle\right)^2}$$

Error Bound (continued)

- Theorem (informal): If $\exists z, s$ such that $Az \geq s\vec{1}$, perceptron makes at most $R^2\|z\|_2^2/s^2$ mistakes, where $R = \max\|a_i\|_2$.
  - Pick the "best" one to minimize $\|z\|_2^2/s^2$ and thus the number of mistakes

$$\min_{(z,s):Az\geq s\vec{1}} \frac{\|z\|_2^2}{s^2} = \min_{(z,s):\|z\|_2=1,Az\geq s\vec{1}} \frac{1}{s^2}$$

$$= \frac{1}{\left(\max_{(z,s):\|z\|_2=1,Az\geq s\vec{1}} s\right)^2} = \frac{1}{\left(\max_{\|z\|_2=1} \min_i \langle a_i, z\rangle\right)^2} = \frac{1}{\gamma^2}$$

$\gamma = \max_{\|z\|_2=1} \min_i \langle a_i, z\rangle$ is the *margin* of the solution wrt the dataset.

Large margin = easy, small margin = easy (draw small vs large margin)

Uniqueness?

- Perceptron only guarantees finding some solution (may be many)
- Certainly not the “best” solution (draw picture)
- Support Vector Machines (SVMs) in a few lectures

What if the data is non-separable?

- The algorithm will never halt, perceptron will “cycle”
- It is not the right algorithm for data which is not linearly separable

When to terminate?

- When all points are classified correctly
  - Training error stops decreasing
- “Validation error” stops decreasing
  - Validation dataset: another dataset that you don’t train on, use to measure quality of solution so far
- Some iteration or update budget is exhausted
- Weights aren’t changing much

Beyond Binary Classification: Multiclass

- Is a picture a dog, cat, bird, horse, frog?
- One-versus-all
  - Train k classifiers, one for dog vs. not dog, one for cat vs. not cat, etc.
  - Output prediction $\arg\max_i \langle z_i, x \rangle$ for the label of point x
- One-versus-one
  - Train $\binom{k}{2}$ classifiers, one for dog vs. cat, one for dog vs. bird, etc.
  - To classify a point, run all of these $\binom{k}{2}$ classifiers and output the majority vote