Notes 4: Perceptron and Halving algorithms

1. Perceptron algorithm

Update weights **additively**
Learn well-separated (i.e. large margin) LTF with possibly negative weights
Let $c(x) = \mathbb{1}(v \cdot x \geqslant \theta)$ be the unknown LTF
Normalization: threshold $\theta = 0$ (halfspace through the origin)
Reason: Add extra coordinate $x_{n+1} = 1$ to every instance

$$v \cdot (x_1, \dots, x_n) \geqslant \theta \qquad \iff \qquad (v, -\theta) \cdot (x_1, \dots, x_{n+1}) \geqslant 0$$

Normalization: Every sample x has unit length, i.e. $\|x\| = 1$ (recall $\|x\| = \sqrt{x_1^2 + \cdots + x_n^2}$)
Reason: By previous assumption $\theta = 0$; rescaling x doesn't change the sign of $v \cdot x$
Normalization: weight vector v has unit length

Perceptron

Initialize: $w = 0$
On input x, output hypothesis $h(x) = \mathbb{1}(w \cdot x \geqslant 0)$ and get $c(x)$
False positive $(h(x) = 1, c(x) = 0)$: Update w as $w - x$
False negative $(h(x) = 0, c(x) = 1)$: Update w as $w + x$

On false positive, $w \cdot x$ is too big, so subtract x from w, so that $(w - x) \cdot x = w \cdot x - \|x\|^2 = w \cdot x - 1$
On false negative, $w \cdot x$ is too small, so add x to w, so that $(w + x) \cdot x = w \cdot x + \|x\|^2 = w \cdot x + 1$

Theorem 1. *(Perceptron convergence) Let $c(x) = \mathbb{1}(v \cdot x \geqslant 0)$ be centered LTF with $\|v\| = 1$. Suppose all samples x have unit length, let margin δ be* min $|v \cdot x|$ *over all samples x received by the algorithm. Then Perceptron Algorithm learns c with at most $1/\delta^2$ mistakes*

Claim 2. *After M mistakes, $w \cdot v \geqslant \delta M$*

Proof. True when $M = 0$ since $w = 0$
Will show that every mistake increases $w \cdot v$ by $\geqslant \delta$
On false positive, $w \cdot v$ becomes $(w - x) \cdot v = w \cdot v - x \cdot v \geqslant w \cdot v + \delta$
On false negative, $w \cdot v$ becomes $(w + w) \cdot v = w \cdot v + x \cdot v \geqslant w \cdot v + \delta$ □

Claim 3. *After M mistakes, $\|w\|^2 \leqslant M$*

Proof. True when $M = 0$ since $w = 0$
Will show that every mistake increases $\|w\|^2$ by $\leqslant 1$
On false positive, $\|w\|^2$ becomes $\|w - x\|^2 = (w - x) \cdot (w - x) = \|w\|^2 - 2\underbrace{w \cdot x}_{\geqslant 0} + \underbrace{\|x\|^2}_{=1}$

On false negative, $\|w\|^2$ becomes $\|w + x\|^2 = (w + x) \cdot (w + x) = \|w\|^2 + 2\underbrace{w \cdot x}_{<0} + \underbrace{\|x\|^2}_{=1}$ □

Proof of Perceptron Convergence. $\delta M \leqslant w \cdot v \underbrace{\leqslant}_{\text{Cauchy–Schwarz}} \|w\| \overbrace{\|v\|}^{=1} \leqslant \sqrt{M}$ □

The above bound is tight!

Claim 4. *When $X = \{x \in \mathbb{R}^d \mid \|x\| = 1\}$ and $d \geqslant \lfloor 1/\delta^2 \rfloor$, any deterministic algorithm for learning LTF makes $\lfloor 1/\delta^2 \rfloor$ mistakes on certain sample sequences and LTF with margin δ*

Proof. ith x^i sample is ith standard basis vector e_i (i.e. 1 at position i and 0 elsewhere)
Number of samples is $n \overset{\text{def}}{=} \lfloor 1/\delta^2 \rfloor$ (as most d by assumption)
All samples will be labeled as the opposite of algorithm's prediction

Will find $v \in \mathbb{R}^d$ with $\|v\| \leqslant 1$ that "correctly" classifies all e_i with margin δ, i.e.

$$\forall \text{ "correct label sequence" } y \in \{1,-1\}^n, \qquad y_i\delta = v \cdot e_i$$

This forces $v_i = \delta y_i$ for all $i \leqslant n$ (e.g. $v = \{+\delta, -\delta, -\delta, +\delta\}$)
Indeed $\|v\|^2 = \delta^2 \|y\|^2 = \delta^2 n \leqslant 1$ □

---

2. Dual perceptron

In Perceptron Algorithm w always ± 1-sum of samples, i.e. $\exists$ signs $\sigma_1, \ldots, \sigma_\ell \in \{1,-1\}$ s.t.

$$w = \sigma_1 x^{i_1} + \cdots + \sigma_\ell x^{i_\ell}$$

Initially $w = 0$; Every mistake adds a new term $\sigma_j x^{i_j}$ to w

Memorizing all mistakes, on sample x,

$$w \cdot x = \sum_{1 \leqslant j \leqslant \ell} \sigma_j (x^{i_j} \cdot x)$$

Computable given inner products $x^{i_j} \cdot x$ between samples
Now takes #mistakes time to compute w (slower)
Can replace inner product $\cdot$ with any **kernel function** $K(,)$

---

3. Halving algorithm

Given any finite concept class $\mathcal{C}$

Halving Algorithm

> K always contains all $c \in \mathcal{C}$ consistent with all the labeled samples so far (initially $K = \mathcal{C}$)
> Hypothesis h is the majority vote over concepts in K

Every mistake removes at least half of concepts from K
Claim: Halving Algorithm makes $\leqslant \log |\mathcal{C}|$ mistakes
Slow: $|K|$ **per round**
Hypothesis isn't from $\mathcal{C}$, but the majority over a subset of $\mathcal{C}$

---

4. Randomized halving algorithm

Randomized Halving Algorithm

> K always contains all $c \in \mathcal{C}$ consistent with all the labeled samples so far (initially $K = \mathcal{C}$)
> Randomly choose a concept $c \in K$ to be the hypothesis h

Claim 5. *On any sequence of samples $x^1, \ldots, x^m$ labeled by any $c \in \mathcal{C}$,*

$$\mathbb{E}[\textit{\#mistakes of the algorithm}] \leqslant \ln |\mathcal{C}| + O(1)$$

Proof. Fix $c \in \mathcal{C}$ and $x^1, \ldots, x^m$
Suppose at some point $|K| = r$
We will bound $\mathbb{E}[\text{\#future mistakes}] \leqslant M_r$ for some upper bound M_r defined below

Order concepts $c_1, \ldots, c_r$ in K according to when they are eliminated by the sequence
e.g. first eliminated batch $c_1, \ldots, c_3$, next c_4, c_5 etc, finally $c_r = c$ never eliminated
On first sample x^1, Algorithm randomly chooses one of $c_1, \ldots, c_r$
If c_r is chosen, no mistake ($1/r$ chance)
If chosen c_t makes mistake on x^1 ($1/r$ chance for each $t < r$)
  $c_1, \ldots, c_t$ (and possibly more) must be eliminated
  K shrinks to (at most) size $r - t$, expect M_{r-t} more mistakes

$$M_r \leqslant \sum_{1 \leqslant t < r} \frac{1}{r}(1 + M_{r-t}) \qquad \Longrightarrow \qquad rM_r \leqslant \sum_{1 \leqslant t < r} (1 + M_{r-t}) = r - 1 + M_1 + \cdots + M_{r-1}$$

Similarly for $r-1$: $\qquad (r-1)M_{r-1} = (r-2) + M_1 + \cdots + M_{r-2} \qquad (*)$
Subtracting, $\qquad r(M_r - M_{r-1}) \leqslant 1$

$$M_r \leqslant \frac{1}{r} + M_{r-1} \leqslant \frac{1}{r} + \frac{1}{r-1} + M_{r-2} \leqslant \ldots \leqslant \underbrace{\frac{1}{r} + \frac{1}{r-1} + \cdots + \frac{1}{1}}_{\text{Harmonic number}} = \ln r + O(1)$$

In the above, we defined M_r by $(*)$ $\square$

Constant factor improvement over deterministic halving: $\qquad \log |\mathcal{C}| / \ln |\mathcal{C}| = \log e = 1.44\ldots$