

Randomization and Learning

Nicolò Cesa-Bianchi

Università degli Studi di Milano

Joint work with Claudio Gentile (Università dell'Insubria)



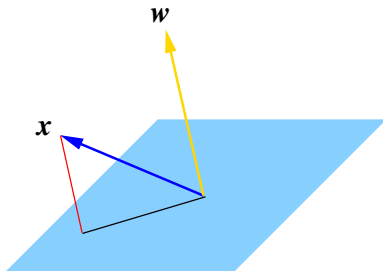
When randomization makes the analysis of learning algorithms easier

- 1 Preliminaries: Linear pattern classification
- 2 Memory bounded learning
- 3 Risk of classifiers in an ensemble



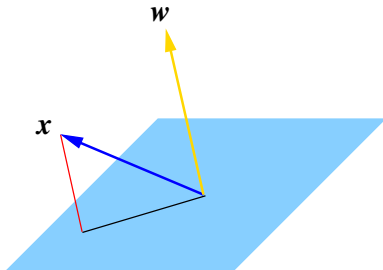
INCREMENTAL LINEAR CLASSIFIERS

- **Stream** x_1, x_2, \dots of data instances encoded as vectors $x_t \in \mathbb{R}^d$ with $\|x_t\| = 1$



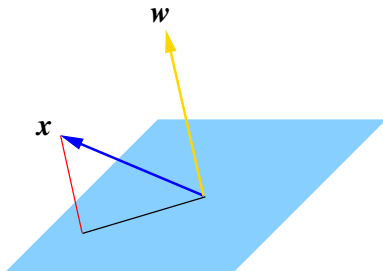
INCREMENTAL LINEAR CLASSIFIERS

- **Stream** x_1, x_2, \dots of data instances encoded as vectors $x_t \in \mathbb{R}^d$ with $\|x_t\| = 1$
- A **binary label** $y_t \in \{-1, 1\}$ associated to each x_t



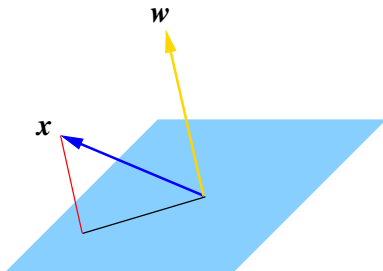
INCREMENTAL LINEAR CLASSIFIERS

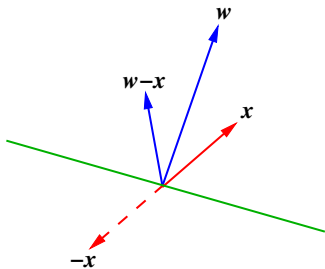
- **Stream** $\mathbf{x}_1, \mathbf{x}_2, \dots$ of data instances encoded as vectors $\mathbf{x}_t \in \mathbb{R}^d$ with $\|\mathbf{x}_t\| = 1$
- A **binary label** $y_t \in \{-1, 1\}$ associated to each \mathbf{x}_t
- A **linear classifier** $\mathbf{w} \in \mathbb{R}^d$ predicts label y_t of \mathbf{x}_t with $\hat{y}_t = \text{SGN}(\mathbf{w}^\top \mathbf{x}_t)$ $\mathbf{w} \in \mathbb{R}^d$



INCREMENTAL LINEAR CLASSIFIERS

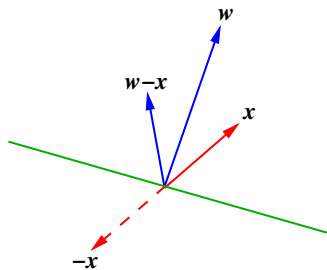
- **Stream** $\mathbf{x}_1, \mathbf{x}_2, \dots$ of data instances encoded as vectors $\mathbf{x}_t \in \mathbb{R}^d$ with $\|\mathbf{x}_t\| = 1$
- A **binary label** $y_t \in \{-1, 1\}$ associated to each \mathbf{x}_t
- A **linear classifier** $\mathbf{w} \in \mathbb{R}^d$ predicts label y_t of \mathbf{x}_t with $\hat{y}_t = \text{SGN}(\mathbf{w}^\top \mathbf{x}_t)$ $\mathbf{w} \in \mathbb{R}^d$
- The true label y_t is observed after each prediction





- The next instance \mathbf{x} is observed
- The current \mathbf{w} classifies \mathbf{x} with $\hat{y} = \text{SGN}(\mathbf{w}^\top \mathbf{x})$
- If $\hat{y} \neq y$ then \mathbf{w} is adjusted:
 $\mathbf{w} \leftarrow \mathbf{w} + y \mathbf{x}$





- The next instance \mathbf{x} is observed
- The current \mathbf{w} classifies \mathbf{x} with $\hat{y} = \text{SGN}(\mathbf{w}^\top \mathbf{x})$
- If $\hat{y} \neq y$ then \mathbf{w} is adjusted:
 $\mathbf{w} \leftarrow \mathbf{w} + y \mathbf{x}$

- Note that $\mathbf{w} = \sum_k y_k \mathbf{x}_k$ where k ranges over the list of misclassified examples (\mathbf{x}_k, y_k)
- The **dual Perceptron** represents \mathbf{w} **implicitly** through the list of misclassified examples



Start with empty list \mathcal{L}

Loop:

- 1 Read next instance \mathbf{x}_t in stream
- 2 Predict y_t with $\hat{y}_t = \text{SGN}\left(\sum_{(\mathbf{x}_k, y_k) \in \mathcal{L}} y_k \mathbf{x}_k^\top \mathbf{x}_t\right)$
- 3 Obtain true label y_t
- 4 If $\hat{y}_t \neq y_t$ (mistake) then add (\mathbf{x}_t, y_t) to \mathcal{L}



Start with empty list \mathcal{L}

Loop:

- 1 Read next instance \mathbf{x}_t in stream
- 2 Predict y_t with $\hat{y}_t = \text{SGN}\left(\sum_{(\mathbf{x}_k, y_k) \in \mathcal{L}} y_k \mathbf{x}_k^\top \mathbf{x}_t\right)$
- 3 Obtain true label y_t
- 4 If $\hat{y}_t \neq y_t$ (**mistake**) then add (\mathbf{x}_t, y_t) to \mathcal{L}

- The examples in the list \mathcal{L} are called **supports**
- A support is added for each mistake
- Can we bound the number of supports/mistakes?



- Let \mathbf{w}_{t-1} the hyperplane used by the Perceptron to classify (\mathbf{x}_t, y_t)
- On any example sequence, Perceptron competes against the **best linear classifier**

$$\underbrace{\sum_t \mathbb{I}_{\{\text{SGN}(\mathbf{w}_{t-1}^\top \mathbf{x}_t) \neq y_t\}}}_{\text{Perceptron's mistakes}} \quad \text{vs.} \quad \inf_{\mathbf{u} \in \mathbb{R}^d} \underbrace{\sum_t \mathbb{I}_{\{\text{SGN}(\mathbf{u}^\top \mathbf{x}_t) \neq y_t\}}}_{\text{mistake of best lin. class.}}$$



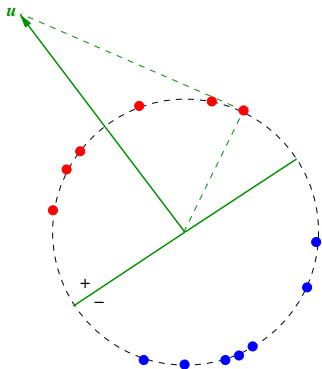
- Let \mathbf{w}_{t-1} the hyperplane used by the Perceptron to classify (\mathbf{x}_t, y_t)
- On any example sequence, Perceptron competes against the **best linear classifier**

$$\underbrace{\sum_t \mathbb{I}_{\{\text{SGN}(\mathbf{w}_{t-1}^\top \mathbf{x}_t) \neq y_t\}}}_{\text{Perceptron's mistakes}} \quad \text{vs.} \quad \inf_{\mathbf{u} \in \mathbb{R}^d} \underbrace{\sum_t \mathbb{I}_{\{\text{SGN}(\mathbf{u}^\top \mathbf{x}_t) \neq y_t\}}}_{\text{mistake of best lin. class.}}$$

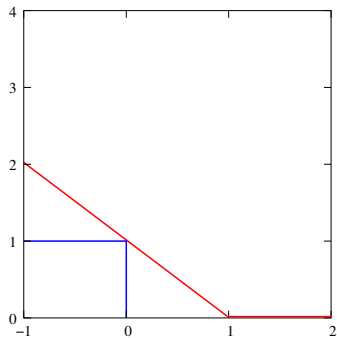
- Problem:** finding \mathbf{u} minimizing number of misclassified examples is NP-hard
- Relaxed goal:** compete against best linear classifier scored with the **hinge loss**



THE HINGE LOSS



Stretch u in order to achieve
margin $y_t u^\top x_t \geq 1$



$$\underbrace{\mathbb{I}_{\{\text{SGN}(z) \neq y\}}}_{\text{mistake ind.}} \leq \underbrace{(1 - yz)_+}_{\text{hinge loss}}$$



PERCEPTRON'S MISTAKE BOUND WITH HINGE LOSS

[Gentile and Warmuth (1999), Freund and Schapire (1999)]

Theorem

Fix any sequence of examples and let

$$M = \sum_t \mathbb{I}_{\{\text{SGN}(\mathbf{w}_{t-1}^\top \mathbf{x}_t) \neq y_t\}} \quad D_{\mathbf{u}} = \sum_t (1 - y_t \mathbf{u}^\top \mathbf{x}_t)_+$$

Then
$$M \leq \inf_{\mathbf{u} \in \mathbb{R}^d} \left(D_{\mathbf{u}} + \|\mathbf{u}\|^2 + \|\mathbf{u}\| \sqrt{D_{\mathbf{u}}} \right)$$



PERCEPTRON'S MISTAKE BOUND WITH HINGE LOSS

[Gentile and Warmuth (1999), Freund and Schapire (1999)]

Theorem

Fix any sequence of examples and let

$$M = \sum_t \mathbb{I}_{\{\text{SGN}(\mathbf{w}_{t-1}^\top \mathbf{x}_t) \neq y_t\}} \quad D_{\mathbf{u}} = \sum_t (1 - y_t \mathbf{u}^\top \mathbf{x}_t)_+$$

Then $M \leq \inf_{\mathbf{u} \in \mathbb{R}^d} \left(D_{\mathbf{u}} + \|\mathbf{u}\|^2 + \|\mathbf{u}\| \sqrt{D_{\mathbf{u}}} \right)$

Corollary (Block, 1962)

In the *linearly separable* case, $M \leq \|\mathbf{u}^*\|^2$ where \mathbf{u}^* is the shortest vector $\mathbf{u} \in \mathbb{R}^d$ s.t.

$$(\forall t) y_t \mathbf{u}^\top \mathbf{x}_t \geq 1$$

\mathbf{u}^* is the *SVM hyperplane*





- Perceptron in dual variables uses the supports in \mathcal{L} to represent a classifier
- On linearly separable data sequences, $\|\mathbf{u}^*\|^2$ supports are sufficient
- On arbitrary data sequences, the number of supports/mistakes grows at rate bounded by

$$\underbrace{\|\mathbf{u}\|^2}_{\text{flat cost}} + \underbrace{D_{\mathbf{u}} + \|\mathbf{u}\| \sqrt{D_{\mathbf{u}}}}_{\text{cost for nonlinearity}} \quad \text{for any } \mathbf{u} \in \mathbb{R}^d$$



- Perceptron in dual variables uses the supports in \mathcal{L} to represent a classifier
- On linearly separable data sequences, $\|\mathbf{u}^*\|^2$ supports are sufficient
- On arbitrary data sequences, the number of supports/mistakes grows at rate bounded by

$$\underbrace{\|\mathbf{u}\|^2}_{\text{flat cost}} + \underbrace{D_{\mathbf{u}} + \|\mathbf{u}\| \sqrt{D_{\mathbf{u}}}}_{\text{cost for nonlinearity}} \quad \text{for any } \mathbf{u} \in \mathbb{R}^d$$

Can we control the rate of mistakes when at most $B < \infty$ supports are used?



Recall

Bound on mistakes depends both on $D_{\mathbf{u}}$ and $\|\mathbf{u}\|$
These are competing terms since $D_{\mathbf{u}}$ grows as $\|\mathbf{u}\| \rightarrow 0$



Recall

Bound on mistakes depends both on $D_{\mathbf{u}}$ and $\|\mathbf{u}\|$
These are competing terms since $D_{\mathbf{u}}$ grows as $\|\mathbf{u}\| \rightarrow 0$

Fact (Dekel, Shalev-Shwartz and Singer, 2006)

Using at most B supports, any learner makes an unbounded number of mistakes on some sequence that is perfectly classified by some $\mathbf{u} \in \mathbb{R}^d$ with $\|\mathbf{u}\| = \sqrt{B+1}$



Recall

Bound on mistakes depends both on $D_{\mathbf{u}}$ and $\|\mathbf{u}\|$
These are competing terms since $D_{\mathbf{u}}$ grows as $\|\mathbf{u}\| \rightarrow 0$

Fact (Dekel, Shalev-Shwartz and Singer, 2006)

Using at most B supports, any learner makes an unbounded number of mistakes on some sequence that is perfectly classified by some $\mathbf{u} \in \mathbb{R}^d$ with $\|\mathbf{u}\| = \sqrt{B+1}$

Main issues

- Can we compete against \mathbf{u} using $(1 + \varepsilon) \|\mathbf{u}\|^2$ supports?
- How does ε enter in the mistake bound?
- Can we do this with a simple policy to manage our supports?



- [Crammer, Kandola and Singer, 2005]
State the problem and propose heuristic to manage supports, partial analysis
- [Weston, Bordes and Bottou, 2005]
Propose more heuristics
- [Dekel, Shalev-Shwartz and Singer, 2006]
Full analysis, but they need $\|\mathbf{u}\| = O(\sqrt{B/(\ln B)})$



Randomized Budget Perceptron

Parameter: size B of cache for supports

Start with empty list \mathcal{L}

Loop:

- 1 Read next instance \mathbf{x}_t in stream
- 2 Predict \mathbf{y}_t with $\hat{\mathbf{y}}_t = \text{SGN}\left(\sum_{(\mathbf{x}_k, \mathbf{y}_k) \in \mathcal{L}} \mathbf{y}_k \mathbf{x}_k^\top \mathbf{x}_t\right)$
- 3 Obtain true label \mathbf{y}_t
- 4 If $\hat{\mathbf{y}}_t \neq \mathbf{y}_t$ then:
 - 1 If $|\mathcal{L}| = B$, then **throw away a random support** from \mathcal{L}
 - 2 Add $(\mathbf{x}_t, \mathbf{y}_t)$ to \mathcal{L}



Theorem (Cesa-Bianchi and Gentile, 2006)

For any cache size B and for any $\mathbf{u} \in \mathbb{R}^d$ such that $\sqrt{B} = (1 + \varepsilon) \|\mathbf{u}\|$, the expected number of mistakes is at most

$$\left(1 + \frac{2}{\varepsilon}\right) \left(D_{\mathbf{u}} + (1 + \varepsilon)^2 \|\mathbf{u}\|^3 + 2(1 + \varepsilon) \|\mathbf{u}\|^2 \ln \left(\|\mathbf{u}\| + \frac{\|\mathbf{u}\|}{\varepsilon}\right)\right)$$



Theorem (Cesa-Bianchi and Gentile, 2006)

For any cache size B and for any $\mathbf{u} \in \mathbb{R}^d$ such that $\sqrt{B} = (1 + \varepsilon) \|\mathbf{u}\|$, the expected number of mistakes is at most

$$\left(1 + \frac{2}{\varepsilon}\right) \left(D_{\mathbf{u}} + (1 + \varepsilon)^2 \|\mathbf{u}\|^3 + 2(1 + \varepsilon) \|\mathbf{u}\|^2 \ln \left(\|\mathbf{u}\| + \frac{\|\mathbf{u}\|}{\varepsilon}\right)\right)$$

Tight budget (ε close to 0)

$$\left(1 + \frac{2}{\varepsilon}\right) \left(D_{\mathbf{u}} + \|\mathbf{u}\|^3 + 2 \|\mathbf{u}\|^2 \ln \frac{\|\mathbf{u}\|}{\varepsilon}\right)$$



Theorem (Cesa-Bianchi and Gentile, 2006)

For any cache size B and for any $\mathbf{u} \in \mathbb{R}^d$ such that $\sqrt{B} = (1 + \varepsilon) \|\mathbf{u}\|$, the expected number of mistakes is at most

$$\left(1 + \frac{2}{\varepsilon}\right) \left(D_{\mathbf{u}} + (1 + \varepsilon)^2 \|\mathbf{u}\|^3 + 2(1 + \varepsilon) \|\mathbf{u}\|^2 \ln \left(\|\mathbf{u}\| + \frac{\|\mathbf{u}\|}{\varepsilon}\right)\right)$$

Tight budget (ε close to 0)

$$\left(1 + \frac{2}{\varepsilon}\right) \left(D_{\mathbf{u}} + \|\mathbf{u}\|^3 + 2 \|\mathbf{u}\|^2 \ln \frac{\|\mathbf{u}\|}{\varepsilon}\right)$$

Large budget ($\varepsilon \gg 1$)

$$D_{\mathbf{u}} + (1 + \varepsilon)^2 \|\mathbf{u}\|^3 + 2(1 + \varepsilon) \|\mathbf{u}\|^2 \ln \|\mathbf{u}\|$$



- Let \mathbf{w}_k be the Perceptron's weight after k mistakes
- Let Q_k be the random support evicted when $k + 1$ st mistake is made: $\mathbf{w}_{k+1} = \mathbf{w}_k - Q_k + y_t \mathbf{x}_t$



- Let \mathbf{w}_k be the Perceptron's weight after k mistakes
- Let Q_k be the random support evicted when $k + 1$ st mistake is made: $\mathbf{w}_{k+1} = \mathbf{w}_k - Q_k + y_t \mathbf{x}_t$
- For any $\mathbf{u} \in \mathbb{R}^d$ and (\mathbf{x}_t, y_t) ,

$$y_t \mathbf{u}^\top \mathbf{x}_t \geq 1 - \underbrace{(1 - y_t \mathbf{u}^\top \mathbf{x}_t)_+}_{\text{hinge loss}}$$



- Let \mathbf{w}_k be the Perceptron's weight after k mistakes
- Let Q_k be the random support evicted when $k + 1$ st mistake is made: $\mathbf{w}_{k+1} = \mathbf{w}_k - Q_k + y_t \mathbf{x}_t$
- For any $\mathbf{u} \in \mathbb{R}^d$ and (\mathbf{x}_t, y_t) ,

$$y_t \mathbf{u}^\top \mathbf{x}_t \geq 1 - \underbrace{(1 - y_t \mathbf{u}^\top \mathbf{x}_t)_+}_{\text{hinge loss}}$$

- Start from recurrence

$$\begin{aligned} \mathbf{u}^\top \mathbf{w}_{k+1} &= \mathbf{u}^\top (\mathbf{w}_k - Q_k + y_t \mathbf{x}_t) \\ &\geq \mathbf{u}^\top \mathbf{w}_k - \mathbf{u}^\top Q_k + 1 - (1 - y_t \mathbf{u}^\top \mathbf{x}_t)_+ \end{aligned}$$



- Let \mathbf{w}_k be the Perceptron's weight after k mistakes
- Let Q_k be the random support evicted when $k + 1$ st mistake is made: $\mathbf{w}_{k+1} = \mathbf{w}_k - Q_k + y_t \mathbf{x}_t$
- For any $\mathbf{u} \in \mathbb{R}^d$ and (\mathbf{x}_t, y_t) ,

$$y_t \mathbf{u}^\top \mathbf{x}_t \geq 1 - \underbrace{(1 - y_t \mathbf{u}^\top \mathbf{x}_t)_+}_{\text{hinge loss}}$$

- Start from recurrence

$$\begin{aligned} \mathbf{u}^\top \mathbf{w}_{k+1} &= \mathbf{u}^\top (\mathbf{w}_k - Q_k + y_t \mathbf{x}_t) \\ &\geq \mathbf{u}^\top \mathbf{w}_k - \mathbf{u}^\top Q_k + 1 - (1 - y_t \mathbf{u}^\top \mathbf{x}_t)_+ \end{aligned}$$

- Since $\mathbf{u}^\top \mathbf{w}_0 = 0$,

$$\mathbf{u}^\top \mathbf{w}_M \geq - \sum_{k=B}^{M-1} \mathbf{u}^\top Q_k + M - D_{\mathbf{u}}$$



- Rearranging and taking expectations

$$\begin{aligned}\mathbb{E} M &\leq D_{\mathbf{u}} + \mathbb{E}[\mathbf{u}^\top \mathbf{w}_M] + \mathbb{E} \left[\sum_{k=B}^{M-1} \mathbf{u}^\top Q_k \right] \\ &\leq D_{\mathbf{u}} + \|\mathbf{u}\| B + \mathbb{E} \left[\sum_{k=B}^{M-1} \mathbf{u}^\top Q_k \right]\end{aligned}$$

since \mathbf{w}_M is the sum of at most B unit norm vectors $\mathbf{y}_t \mathbf{x}_t$,



- Rearranging and taking expectations

$$\begin{aligned} \mathbb{E} M &\leq D_{\mathbf{u}} + \mathbb{E}[\mathbf{u}^T \mathbf{w}_M] + \mathbb{E} \left[\sum_{k=B}^{M-1} \mathbf{u}^T Q_k \right] \\ &\leq D_{\mathbf{u}} + \|\mathbf{u}\| B + \mathbb{E} \left[\sum_{k=B}^{M-1} \mathbf{u}^T Q_k \right] \end{aligned}$$

since \mathbf{w}_M is the sum of at most B unit norm vectors $\mathbf{y}_t \mathbf{x}_t$,

- Now we need to show

$$\mathbb{E} \left[\sum_{k=B}^{M-1} \mathbf{u}^T Q_k \right] \leq \|\mathbf{u}\| \frac{\mathbb{E} M}{\sqrt{B}}$$

so to get $\mathbb{E} M \leq D_{\mathbf{u}} + \|\mathbf{u}\| B + \|\mathbf{u}\| \frac{\mathbb{E} M}{\sqrt{B}}$



Since $\mathbb{E}_k Q_k = \frac{\mathbf{w}_k}{B}$, we have

$$\mathbb{E} \left[\sum_{k=B}^{M-1} \mathbf{u}^\top Q_k \right] = \mathbb{E} \left[\sum_{k=B}^{M-1} \mathbf{u}^\top \frac{\mathbf{w}_k}{B} \right] \leq \frac{\|\mathbf{u}\|}{B} \mathbb{E} \left[\sum_{k=B}^{M-1} \|\mathbf{w}_k\| \right]$$



Since $\mathbb{E}_k Q_k = \frac{\mathbf{w}_k}{B}$, we have

$$\mathbb{E} \left[\sum_{k=B}^{M-1} \mathbf{u}^\top Q_k \right] = \mathbb{E} \left[\sum_{k=B}^{M-1} \mathbf{u}^\top \frac{\mathbf{w}_k}{B} \right] \leq \frac{\|\mathbf{u}\|}{B} \mathbb{E} \left[\sum_{k=B}^{M-1} \|\mathbf{w}_k\| \right]$$

Lemma (contraction)

$$\mathbb{E}_k \|\mathbf{w}_{k+1}\|^2 \leq \left(1 - \frac{2}{B}\right) \|\mathbf{w}_k\|^2 + 2 \quad (k \geq B)$$



Since $\mathbb{E}_k Q_k = \frac{\mathbf{w}_k}{B}$, we have

$$\mathbb{E} \left[\sum_{k=B}^{M-1} \mathbf{u}^\top Q_k \right] = \mathbb{E} \left[\sum_{k=B}^{M-1} \mathbf{u}^\top \frac{\mathbf{w}_k}{B} \right] \leq \frac{\|\mathbf{u}\|}{B} \mathbb{E} \left[\sum_{k=B}^{M-1} \|\mathbf{w}_k\| \right]$$

Lemma (contraction)

$$\mathbb{E}_k \|\mathbf{w}_{k+1}\|^2 \leq \left(1 - \frac{2}{B}\right) \|\mathbf{w}_k\|^2 + 2 \quad (k \geq B)$$

Taking expectations on both sides, we get

$$\mathbb{E} \|\mathbf{w}_{k+1}\|^2 \leq \left(1 - \frac{2}{B}\right) \mathbb{E} \|\mathbf{w}_k\|^2 + 2$$

that is $\mathbb{E} \|\mathbf{w}_k\|^2 \leq B$ for all k



- We have $\mathbb{E} \|\mathbf{w}_k\| \leq \sqrt{B}$ and we want to show

$$\mathbb{E} \left[\sum_{k=B}^{M-1} \|\mathbf{w}_k\| \right] \leq \sqrt{B} \mathbb{E} M$$

- If the \mathbf{w}_k were independent, we could use **Wald's Lemma**
- But the \mathbf{w}_k are independent, thus we need a more general result



Theorem

Let the process X_1, X_2, \dots be such that $0 \leq X_k \leq B$ and assume

$$\mathbb{E}_k X_{k+1}^2 \leq \begin{cases} B & \text{for } k = 0, \dots, B-1 \\ (1 - \frac{2}{B}) X_k^2 + 2 & \text{for } k \geq B \end{cases}$$

If M is a stopping time for X_1, X_2, \dots , then for all $\varepsilon > 0$

$$\mathbb{E} \left[\sum_{k=1}^M X_k \right] \leq (1 + \varepsilon) \sqrt{B} \mathbb{E} M + \frac{B^{3/2}}{2} \ln \frac{B^2}{2\varepsilon}$$



Applying the Lemma to the process $\|\mathbf{w}_1\|, \|\mathbf{w}_2\|, \dots$ we get

$$\begin{aligned} \mathbb{E} M &\leq D_{\mathbf{u}} + \|\mathbf{u}\| B + \|\mathbf{u}\| \mathbb{E} \left[\sum_{k=1}^M \|\mathbf{w}_k\| \right] \\ &\leq D_{\mathbf{u}} + \|\mathbf{u}\| B + \|\mathbf{u}\| (1 + \varepsilon) \sqrt{B} \mathbb{E} M + \|\mathbf{u}\| \frac{B^{3/2}}{2} \ln \frac{B^2}{2\varepsilon} \end{aligned}$$

This gives the desired bound for $\sqrt{B} = \Theta(\|\mathbf{u}\|)$



THE PERCEPTRON'S ENSEMBLE OF CLASSIFIERS



- Assume examples (\mathbf{x}_t, y_t) are drawn i.i.d. from a fixed and unknown distribution



- Assume examples (\mathbf{x}_t, y_t) are drawn i.i.d. from a fixed and unknown distribution
- Is there a low-risk classifier in the ensemble $\mathbf{w}_0, \mathbf{w}_1, \dots, \mathbf{w}_{n-1}$ of hyperplanes used by Perceptron in the first n steps?



- Assume examples (\mathbf{x}_t, y_t) are drawn i.i.d. from a fixed and unknown distribution
- Is there a low-risk classifier in the ensemble $\mathbf{w}_0, \mathbf{w}_1, \dots, \mathbf{w}_{n-1}$ of hyperplanes used by Perceptron in the first n steps?
- \mathbf{w}_{n-1} is a natural candidate, but no good risk bounds are available



- Assume examples (\mathbf{x}_t, y_t) are drawn i.i.d. from a fixed and unknown distribution
- Is there a low-risk classifier in the ensemble $\mathbf{w}_0, \mathbf{w}_1, \dots, \mathbf{w}_{n-1}$ of hyperplanes used by Perceptron in the first n steps?
- \mathbf{w}_{n-1} is a natural candidate, but no good risk bounds are available
- On the other hand, it is easy to bound the risk of a **random** classifier in the ensemble



- Run Perceptron on i.i.d. sample $(\mathbf{X}_1, Y_1), \dots, (\mathbf{X}_n, Y_n)$ and obtain ensemble $\mathbf{w}_0, \dots, \mathbf{w}_{n-1}$



THE RANDOM ENSEMBLE CLASSIFIER

- Run Perceptron on i.i.d. sample $(\mathbf{X}_1, Y_1), \dots, (\mathbf{X}_n, Y_n)$ and obtain ensemble $\mathbf{w}_0, \dots, \mathbf{w}_{n-1}$
- For $t = 0, \dots, n - 1$ let H_t be the linear classifier using hyperplane \mathbf{w}_t



THE RANDOM ENSEMBLE CLASSIFIER

- Run Perceptron on i.i.d. sample $(\mathbf{X}_1, Y_1), \dots, (\mathbf{X}_n, Y_n)$ and obtain ensemble $\mathbf{w}_0, \dots, \mathbf{w}_{n-1}$
- For $t = 0, \dots, n - 1$ let H_t be the linear classifier using hyperplane \mathbf{w}_t
- Let $\text{RISK}(H_t) = \mathbb{P}(H_t(\mathbf{X}) \neq Y) = \text{prob. } H_t \text{ misclassifies a random example}$



- Run Perceptron on i.i.d. sample $(\mathbf{X}_1, Y_1), \dots, (\mathbf{X}_n, Y_n)$ and obtain ensemble $\mathbf{w}_0, \dots, \mathbf{w}_{n-1}$
- For $t = 0, \dots, n - 1$ let H_t be the linear classifier using hyperplane \mathbf{w}_t
- Let $\text{RISK}(H_t) = \mathbb{P}(H_t(\mathbf{X}) \neq Y) = \text{prob. } H_t \text{ misclassifies a random example}$
- $\text{RISK}(H_{t-1}) - \mathbb{I}_{\{H_{t-1}(\mathbf{X}_t) \neq Y_t\}}$ is a **martingale difference sequence**

$$\mathbb{E} \left[\text{RISK}(H_{t-1}) - \mathbb{I}_{\{H_{t-1}(\mathbf{X}_t) \neq Y_t\}} \mid (\mathbf{X}_1, Y_1), \dots, (\mathbf{X}_{t-1}, Y_{t-1}) \right] = 0$$

since H_{t-1} is determined by $(\mathbf{X}_1, Y_1), \dots, (\mathbf{X}_{t-1}, Y_{t-1})$



The associated martingale is

$$\sum_{t=1}^n \left(\text{RISK}(H_{t-1}) - \mathbb{I}_{\{H_{t-1}(X_t) \neq Y_t\}} \right)$$

$$\iff \underbrace{\frac{1}{n} \sum_{t=1}^n \text{RISK}(H_{t-1})}_{\text{risk of random classifier}} - \underbrace{\frac{1}{n} \sum_{t=1}^n \mathbb{I}_{\{H_{t-1}(X_t) \neq Y_t\}}}_{\text{rate of Perceptron mistakes}}$$



Theorem (Bernstein inequality)

If Z_1, Z_2, \dots is a *martingale difference sequence* with increments bounded by 1 and

$$V_n = \sum_{t=1}^n \mathbb{E} [Z_t^2 \mid Z_1, \dots, Z_{t-1}]$$

then for all $S, K > 0$

$$\mathbb{P} \left(\sum_{t=1}^n Z_t \geq S, \quad V_n \leq K \right) \leq \exp \left(-\frac{S^2}{2(S/3 + K)} \right)$$



Applying Bernstein, we find out that the risk of the random ensemble classifier is close to M/n (the Perceptron's mistake rate)

$$\frac{1}{n} \sum_{t=1}^n \text{RISK}(H_{t-1}) \leq \frac{M}{n} + \frac{c}{n} \left(\sqrt{M \ln M} + \ln M \right) \quad \text{w.h.p.}$$



- This trick applies to any learning algorithm run incrementally on the training data
- A more involved analysis allows to derandomize and get, for a sample size of n ,

$$\text{RISK}(\hat{H}) \leq \frac{M}{n} + \frac{c}{n} \left(\sqrt{M \ln n} + (\ln n)^2 \right) \quad \text{w.h.p.}$$

- For regression with convex loss, Jensen's inequality allows us to get the randomized bound for the **average** linear regression function

