The Perceptron

CMSC 422
SOHEIL FEIZI
sfeizi@cs.umd.edu

Slides adapted from MARINE CARPUAT
This week

• The perception: a new model/algorithm
  – its variants: voted, averaged
  – convergence proof

• Fundamental Machine Learning Concepts
  – Online vs. batch learning
  – Error-driven learning
  – Linear separability and margin of a dataset
Recap: Perceptron for binary classification

- Classifier = hyperplane that separates positive from negative examples
  \[ \hat{y} = \text{sign}(w^T x + b) \]

- Perceptron training
  - Finds such a hyperplane
  - Online & error-driven
Recap: Perceptron updates

Update for a misclassified positive example:

\[ w_{new} = w_{old} + x \]
Recap: Perceptron updates

Update for a misclassified negative example:

\[ \mathbf{w}_{\text{new}} = \mathbf{w}_{\text{old}} - \mathbf{x} \]
Standard Perceptron: predict based on final parameters

Algorithm 5 \textbf{PerceptronTrain}(D, \text{MaxIter})

\begin{algorithmic}[1]
\State $w_d \leftarrow 0$, for all $d = 1 \ldots D$ \Comment{initialize weights}
\State $b \leftarrow 0$ \Comment{initialize bias}
\For{$\text{iter} = 1 \ldots \text{MaxIter}$}
\ForAll{$(x,y) \in D$}
\State $a \leftarrow \sum_{d=1}^{D} w_d x_d + b$ \Comment{compute activation for this example}
\If{$ya \leq 0$}
\State $w_d \leftarrow w_d + yx_d$, for all $d = 1 \ldots D$ \Comment{update weights}
\State $b \leftarrow b + y$ \Comment{update bias}
\EndIf
\EndFor
\EndFor
\State \textbf{return} $w_0, w_1, \ldots, w_D, b$
\end{algorithmic}
Predict based on final + intermediate parameters

- The voted perceptron

\[ \hat{y} = \text{sign} \left( \sum_{k=1}^{K} c^{(k)} \text{sign} \left( w^{(k)} \cdot \hat{x} + b^{(k)} \right) \right) \]

- The averaged perceptron

\[ \hat{y} = \text{sign} \left( \sum_{k=1}^{K} c^{(k)} \left( w^{(k)} \cdot \hat{x} + b^{(k)} \right) \right) \]

- Require keeping track of “survival time” of weight vectors \( c^{(1)}, \ldots, c^{(K)} \)
Can the perceptron always find a hyperplane to separate positive from negative examples?
Convergence of Perceptron

• The perceptron has converged if it can classify every training example correctly
  – i.e. if it has found a hyperplane that correctly separates positive and negative examples

• Under which conditions does the perceptron converge and how long does it take?
Convergence of Perceptron

Theorem (Block & Novikoff, 1962)

If the training data $D = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ is **linearly separable** with margin $\gamma$ by a unit norm hyperplane $w_*$ ($||w_*|| = 1$) with $b = 0$,

Then **perceptron training converges after** $\frac{R^2}{\gamma^2}$ **errors** during training

(assuming ($||x|| < R$) for all $x$).
Margin of a data set $D$

\[
\text{margin}(D, w, b) = \begin{cases} \\
\min_{(x,y) \in D} y(w \cdot x + b) & \text{if } w \text{ separates } D \\
-\infty & \text{otherwise} \\
\end{cases}
\] (4.8)

Distance between the hyperplane $(w,b)$ and the nearest point in $D$

\[
\text{margin}(D) = \sup_{w,b} \text{margin}(D, w, b)
\] (4.9)

Largest attainable margin on $D$
Theorem (Block & Novikoff, 1962)

If the training data $D = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ is **linearly separable** with margin $\gamma$ by a unit norm hyperplane $w_*$ ($||w_*|| = 1$) with $b = 0$, then **perceptron training converges** after $\frac{R^2}{\gamma^2}$ errors during training (assuming ($||x|| < R$) for all $x$).

Proof:

- Margin of $w_*$ on any arbitrary example $(x_n, y_n)$: $\frac{y_n w_*^T x_n}{||w_*||} = y_n w_*^T x_n \geq \gamma$
- Consider the $(k + 1)^{th}$ mistake: $y_n w_k^T x_n \leq 0$, and update $w_{k+1} = w_k + y_n x_n$
- $w_{k+1}^T w_* = w_k^T w_* + y_n w_*^T x_n \geq w_k^T w_* + \gamma$ (why is this nice?)
- Repeating iteratively $k$ times, we get $w_{k+1}^T w_* > k \gamma$ \hspace{1cm} (1)
- $||w_{k+1}||^2 = ||w_k||^2 + 2y_n w_k^T x_n + ||x||^2 \leq ||w_k||^2 + R^2$ (since $y_n w_k^T x_n \leq 0$)
- Repeating iteratively $k$ times, we get $||w_{k+1}||^2 \leq kR^2$ \hspace{1cm} (2)
Theorem (Block & Novikoff, 1962)

If the training data $D = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ is \textbf{linearly separable} with margin $\gamma$ by a unit norm hyperplane $w_*(||w_*|| = 1)$ with $b = 0$, then \textbf{perceptron training converges after} $\frac{R^2}{\gamma^2}$ \textbf{errors} during training (assuming ($||x|| < R$) for all $x$).

What does this mean?

• Perceptron converges quickly when margin is large, slowly when it is small
• Bound does not depend on number of training examples $N$, nor on number of features
• Proof guarantees that perceptron converges, but not necessarily to the max margin separator
Practical Implications

• Sensitivity to noise
  – if the data is not linearly separable due to noise, no guarantee of convergence or accuracy

• Linear separability in practice
  – Data may be linearly separable in practice
  – Especially when number of features $>>$ number of examples

• Risk of overfitting mitigated by
  – Early stopping
  – Averaging
What you should know

• Perceptron concepts
  – training/prediction algorithms (standard, voting, averaged)
  – convergence theorem and what practical guarantees it gives us
  – how to draw/describe the decision boundary of a perceptron classifier

• Fundamental ML concepts
  – Determine whether a data set is linearly separable and define its margin
  – Error driven algorithms, online vs. batch algorithms