Machine Learning for Intelligent Systems

Lecture 7: Convergence of Perceptron

Reading: UML 9.1

Instructors: Nika Haghtalab (this time) and Thorsten Joachims

Homogenous vs. Non-homogenous

Any d-dimensional learning problem for non-homogenous linear classifiers has a homogenous form in (d+1) dimension.

<table>
<thead>
<tr>
<th>Non-Homogenous \ $H^{d} = {(\mathbf{w},b) \mid \mathbf{w} \in \mathbb{R}^{d}, b \in \mathbb{R}}$</th>
<th>Homogenous \ $H^{d+1}_{\text{homogenous}} = {(\mathbf{w}) \mid \mathbf{w} \in \mathbb{R}^{d+1}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{x}$ \ $\mathbf{w}, b$</td>
<td>$\mathbf{x}' = (x, +1)$ \ $\mathbf{w}' = (\mathbf{w}, b)$</td>
</tr>
<tr>
<td>$\mathbf{w} \cdot \hat{x} + b$</td>
<td>$\mathbf{w}' \cdot \mathbf{x}' = \mathbf{w} \cdot \mathbf{x} + b$</td>
</tr>
</tbody>
</table>

Without loss of generality, focus on homogenous linear classifiers.

Improve a linear classifier

If there is a homogeneous linear classifier that is consistent with \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)\}, how can we find it?

Last time: Do it with a linear program
This time: Start with a guess and improve it.

Move away from misclassified negative points: $\mathbf{w} - \mathbf{x}$

Improving a linear classifier

If there is a homogeneous linear classifier that is consistent with \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)\}, how can we find it?

Last time: Do it with a linear program
This time: Start with a guess and improve it.

Move towards misclassified positive points: $\mathbf{w} + \mathbf{x}$
Improving a linear classifier

If there is a homogeneous linear classifier that is consistent with \( (\tilde{x}_1, y_1), (\tilde{x}_2, y_2), \ldots, (\tilde{x}_m, y_m) \), how can we find it?

Last time: Do it with a linear program

This time: Start with a guess and improve it.

Move away from misclassified negative points: \( \hat{w} - \hat{w} \)
Move towards misclassified positive points: \( \hat{w} + \hat{w} \)

Example:

<table>
<thead>
<tr>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( y )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>y_1 = 1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>y_2 = 1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>y_3 = 1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>y_4 = 1</td>
</tr>
</tbody>
</table>

- \( \hat{w}^{(0)} = (0, 0) \) 
- \( y_1(\hat{w}^{(0)} \cdot \tilde{x}_1) = 1 \leq 0 \)
- \( y_2(\hat{w}^{(0)} \cdot \tilde{x}_2) = 5 > 0 \)
- \( y_3(\hat{w}^{(0)} \cdot \tilde{x}_3) = 4 > 0 \)
- \( y_4(\hat{w}^{(0)} \cdot \tilde{x}_4) = 3 > 0 \)
- \( \hat{w}^{(2)} = \hat{w}^{(1)} - x_1 = (2, 1) \)

Margin & Convergence

Given a data set \( S = \{ (\tilde{x}_1, y_1), (\tilde{x}_2, y_2), \ldots, (\tilde{x}_m, y_m) \} \) and a linear classifier \( h_{\hat{w}} \) that is consistent with \( S \), that is, \( y_i(\hat{w} \cdot \tilde{x}_i) > 0 \), the geometric margin of \( h_{\hat{w}} \) is defined as:

\[
\gamma = \min_{\hat{w}} \frac{y (\hat{w} \cdot \tilde{x})}{\|\hat{w}\|} 
\]

Margin \( \gamma \) is the distance of the closest instance to hyperplane \( \hat{w} \cdot \tilde{x} = 0 \).

Convergence of Perceptron

Theorem: Convergence of Perceptron

Given a data set \( S = \{ (\tilde{x}_1, y_1), (\tilde{x}_2, y_2), \ldots, (\tilde{x}_m, y_m) \} \) and radius \( R \) such that \( \|\tilde{x}_i\| \leq R \) for all \( i \in [m] \).

If \( h_{\hat{w}} \) is consistent with \( S \) with margin \( \gamma = \min_{\hat{w}} \frac{y (\hat{w} \cdot \tilde{x})}{\|\hat{w}\|} \), then Perceptron makes at most \( R^2 / \gamma^2 \) updates before predicting every label perfectly.

Idea: \( h_{\hat{w}} \) has \( \gamma \) margin \( \rightarrow \) there is wiggle room.

\[ \Rightarrow \text{Show that within } t = R^2 / \gamma^2, \] \[ \hat{w} \text{ is close to } \hat{w} \text{ in angle. } \]

Proof Ideas

Theorem: Convergence of Perceptron

Perceptron makes at most \( R^2 / \gamma^2 \) updates before predicting every label perfectly. Recall margin \( \gamma = \min_{\hat{w}} \frac{y (\hat{w} \cdot \tilde{x})}{\|\hat{w}\|} \).

Idea: Proof by Contradiction. Show that if within \( t > R^2 / \gamma^2 \),

\[
\cos(\theta (\hat{w}^t, \hat{w}^{t+1})) = \hat{w}^t \cdot \hat{w}^{t+1} \leq 1 \quad \text{Impossible}
\]

Plan:

1. Assume \( \hat{w}^t \) is normalized to be unit vector \( \Rightarrow \) margin doesn't change.
2. Show that \( \hat{w}^t \cdot \hat{w}^{t+1} \) is large (find a lower bound).
3. Show that \( \|\hat{w}^{t+1}\| \) is not too large (find an upper bound).

\[ \Rightarrow \text{So, if } t > R^2 / \gamma^2, \text{ the cosine will be larger than } 1 \rightarrow \text{Contradiction.} \]
Recall: Example

- Update on \((x_2, y_2)\)
  - \(\mathbf{w}^{(1)} = (2, 1)\) converges in 1 step.

- Update on \((x_3, y_3)\)
  - \(\mathbf{w}^{(1)} = (1, 1)\)
  - Update on \((x_4, y_4)\)
  - \(\mathbf{w}^{(1)} = (2, 0)\)

Online Perceptron

Example: Reuters Text Classification

**Theorem: Mistake Bound of Online Perceptron**

Given a sequence of data \((\hat{x}_1, \hat{y}_1), (\hat{x}_2, \hat{y}_2), \ldots, (\hat{x}_m, \hat{y}_m)\) one by one, with radius \(R\) and margin \(y = \min_{\mathbf{w} \in \mathbb{R}^d} \frac{|\langle \mathbf{w}, \hat{x}_i \rangle \rangle|}{||\mathbf{w}||^2}\) for some \(\mathbf{w}^*\).

**Online prediction:** At each time use the current \(\mathbf{w}\) to predict the label of incoming \((\hat{x}_i, \hat{y}_i)\), update if needed.

**Mistake Bound:** The number of mistakes that perceptron makes is at most \(\frac{R^2}{y^2}\).