Perceptrons

CSci 5525: Machine Learning

Instructor: Nicholas Johnson

October 6, 2020

Announcements

- HW2 is posted (due Oct 15)
- Exam 1 will be posted on Oct 20 (due 48 hours later)

Perceptrons

- Linear classifiers with weight vector $\mathbf{w}$
- Labels are encoded as $y_i \in \{-1, 1\}$
- Prediction on $\mathbf{x}_i$ is incorrect if $y_i \mathbf{w}^\top \mathbf{x}_i < 0$
- Let $\mathcal{M}(\mathbf{w})$ be the set of points on which prediction is incorrect
- The objective function to be minimized

$$E(\mathbf{w}) = -\sum_{i \in \mathcal{M}(\mathbf{w})} y_i \mathbf{w}^\top \mathbf{x}_i$$

- For any point $i \in \mathcal{M}(\mathbf{w})$, gradient $\nabla E_i(\mathbf{w}) = -y_i \mathbf{x}_i$
- The gradient based update

$$\mathbf{w}_{(new)} = \mathbf{w}_{(old)} - \eta \nabla E_i(\mathbf{w}) = \mathbf{w}_{(old)} + \eta y_i \mathbf{x}_i$$

- The learning rate parameter η can be set to 1
- No update corresponding to the correctly predicted points

Analysis of Perceptron Training

$w^T x_i > 0 \quad C_1 \quad y_i = +1$

$< 0 \quad C_2 \quad y_i = -1$

- Algorithm goes through all the points sequentially
  - If prediction is correct, do not change anything
  - If prediction is wrong, and $y_i = +1$ then $\mathbf{w}_{(new)} = \mathbf{w}_{(old)} + \mathbf{x}_i$
  - If prediction is wrong, and $y_i = -1$ then $\mathbf{w}_{(new)} = \mathbf{w}_{(old)} - \mathbf{x}_i$
- Each update reduces the error contribution for that point

$$-\mathbf{w}_{(new)}^\top \mathbf{x}_i y_i = -\mathbf{w}_{(old)}^\top \mathbf{x}_i y_i - (\mathbf{x}_i y_i)^\top \mathbf{x}_i y_i < -\mathbf{w}_{(old)}^\top \mathbf{x}_i y_i$$

$w_{t+1} = w_t + y_i x_i$

- But the update can increase the contribution of other terms
- Will this ever converge?

Perceptrons: Loss Function, SGD

- The objective function to be minimized

$$E(\mathbf{w}) = -\sum_{i \in \mathcal{M}(\mathbf{w})} y_i \mathbf{w}^\top \mathbf{x}_i$$

- For any point $i \in \mathcal{M}(\mathbf{w})$, gradient $\nabla E_i(\mathbf{w}) = -y_i \mathbf{x}_i$
- Perceptron objective is hinge loss, with hinge at 0

$$E(\mathbf{w}) = \sum_{i=1}^{n} \max(0, -y_i \mathbf{w}^\top \mathbf{x}_i)$$

- Recall SVM objective, hinge at 1 plus regularization

$$E(\mathbf{w}) = \sum_{i=1}^{n} \max(0, 1 - y_i \mathbf{w}^\top \mathbf{x}_i) + \frac{\lambda}{2} \|\mathbf{w}\|^2$$

- Stochastic sub-gradient descent (SGD), fixed step-size
- Improvements: Use mini-batch?

Illustration of Perceptron Algorithm: Step 1

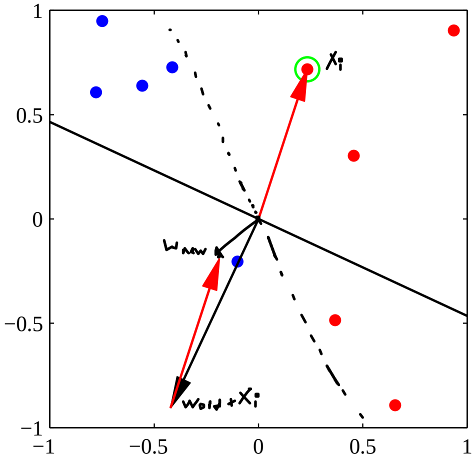


Illustration of Perceptron Algorithm: Step 2

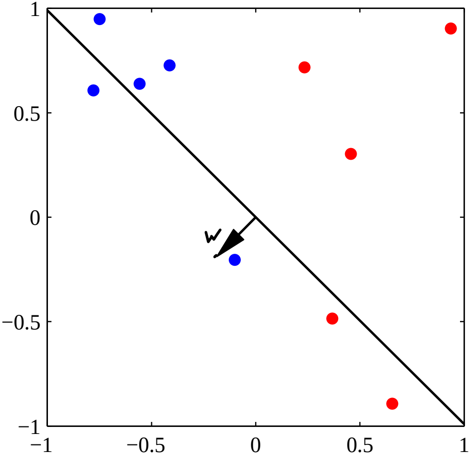


Illustration of Perceptron Algorithm: Step 3

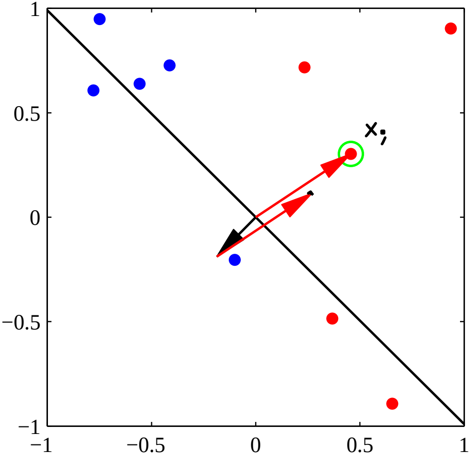
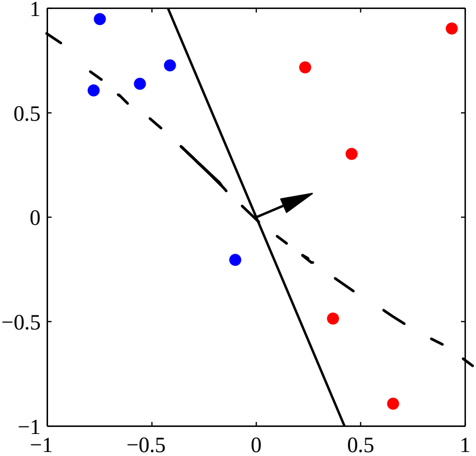


Illustration of Perceptron Algorithm: Step 4



Perceptron Convergence Theorem

- If the data is linearly separable
  - Training converges in finite number of iterations
- Consider a separable dataset $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$
- Let $\mathbf{u}$, $\|\mathbf{u}\| = 1$ be a separator so that $\forall i, y_i \mathbf{u}^\top \mathbf{x}_i \geq \gamma > 0$
- Let $R = \max_i \|\mathbf{x}_i\|$
- Theorem: Training converges after at most $\left(\frac{R}{\gamma}\right)^2$ mistakes

Proof

- Let $\mathbf{v}_1 = \mathbf{0}$
- Let $\mathbf{v}_{k+1}$ be the vector making the k^{th} mistake

$$\mathbf{v}_{k+1} = \mathbf{v}_k + y_i \mathbf{x}_i$$

- Then,

$$\begin{aligned} \mathbf{v}_{k+1}^\top \mathbf{u} &= \mathbf{v}_k^\top \mathbf{u} + y_i \mathbf{u}^\top \mathbf{x}_i \\ &\geq \mathbf{v}_k^\top \mathbf{u} + \gamma \\ &\geq k\gamma \end{aligned}$$

- Also,

$$\begin{aligned} \|\mathbf{v}_{k+1}\|^2 &= \|\mathbf{v}_k\|^2 + 2y_i \mathbf{v}_k^\top \mathbf{x}_i + \|\mathbf{x}_i\|^2 \\ &\leq \|\mathbf{v}_k\|^2 + R^2 \\ &\leq kR^2 \end{aligned}$$

Proof (cont.)

- Since $\mathbf{u}$ is a unit vector

$$
\begin{aligned}
\mathbf{v}_{k+1}^\top \mathbf{u} &\leq \|\mathbf{v}_{k+1}\| \\
k\gamma \leq \mathbf{v}_{k+1}^\top \mathbf{u} &\leq \|\mathbf{v}_{k+1}\| \leq \sqrt{k}R \\
\sqrt{k} &\leq \frac{R}{\gamma} \\
k &\leq \left(\frac{R}{\gamma}\right)^2
\end{aligned}
$$

Perceptrons: Ideas

- Loss function is non-shifted hinge loss

$$E(\mathbf{w}) = -\sum_{i \in \mathcal{M}} y_i \mathbf{w}^\top \mathbf{x}_i = \sum_{i=1}^{n} \max(0, -y_i \mathbf{w}^\top \mathbf{x}_i)$$

- Perceptron training is stochastic gradient descent (SGD)
  - Gradient based updated considering one data point at random

$$\mathbf{w}_{(new)} = \mathbf{w}_{(old)} - \eta \nabla E_i(\mathbf{w}) = \mathbf{w}_{(old)} + \eta y_i \mathbf{x}_i$$

- The convergence depends on the margin γ as $\left(\frac{R}{\gamma}\right)^2$
  - Larger margin leads to faster convergence (in the worst case)