Logistic regression

CS 446 / ECE 449

2022-01-25 09:57:46 -0600 (4c2f15b)
Plan for today

- Linear classifiers.
- ERM for classification.
- Solving the ERM problem.
1. Start from training data \( ((x_i, y_i))_{i=1}^n \), with \( x_i \in \mathbb{R}^d \) and \( y_i \in \mathbb{R} \).

   **Last lecture:** \( y_i \in \mathbb{R} \); **this lecture:** \( y_i \in \{+1, -1\} \).

2. Model is a linear predictor: pick \( w \in \mathbb{R}^d \) with

   \[ x_i \mapsto w^\top x_i =: \hat{y}_i \approx y_i. \]

3. Choose \( w \) by minimizing empirical risk (average loss over training set):

   \[ \hat{R}(w) = \frac{1}{n} \sum_{i=1}^n \ell(w^\top x_i, y_i). \]

   **Last lecture:** squared loss \( \ell_{sq} \); **this lecture:** logistic loss \( \ell_{\text{logistic}} \).

4. Basic method: gradient descent. Set \( w_0 = 0 \), and thereafter

   \[ w_{i+1} := w_i - \eta \nabla \hat{R}(w_i), \]

   where \( \eta \) is a learning rate (step size).

   Same in both lectures, however least squares also had SVD solution.
Last lecture, we studied *regression*; the output/label space was $\mathbb{R}$. 
Today, the goal is a classification; the output/label space is discrete.
Binary classification means output space $\mathcal{Y} = \{-1, +1\}$.
A linear predictor $\mathbf{w} \in \mathbb{R}^d$ classifies according to $\text{sign}(\mathbf{w}^\top \mathbf{x}) \in \{-1, +1\}$.

Given $((\mathbf{x}_i, y_i))_{i=1}^n$ and a weight vector $\mathbf{w} \in \mathbb{R}^d$,
we want $\hat{y}_i := \text{sign}(\mathbf{w}^\top \mathbf{x}_i) \in \{-1, +1\}$ and $y_i$ to agree.
Geometry of linear classifiers

Given $w \in \mathbb{R}^d$, predict with

$$x \mapsto \text{sign}(w^T x) \in \{\pm 1\}.$$ 

Let $H$ be the hyperplane orthogonal to $w$:

$$H := \left\{ x \in \mathbb{R}^d : x^T w = 0 \right\}.$$ 

$H$ splits $\mathbb{R}^d$ into points we label positive,

$$\{ x \in \mathbb{R}^d : w^T x > 0 \},$$

and points we label negative,

$$\{ x \in \mathbb{R}^d : w^T x < 0 \}.$$ 

The decision boundary is $H$. 
Linear separability

Is it always possible to find $w$ with $\text{sign}(w^T x_i) = y_i$?
I.e., is it always possible to find a (homogeneous) hyperplane which separates the data?
Is it always possible to find $w$ with $\text{sign}(w^T x_i) = y_i$? 
I.e., is it always possible to find a (homogeneous) hyperplane which separates the data?

Lecture 4: adding features can make things separable.
Decision boundary with quadratic feature expansion

With old faithful data, we appended 1 to $x$.
In general, we can map $x$ to more exotic things, and separate more data.

We’ll discuss this next lecture.
Why not try "pytorch meta-algorithm", with empirical risk
minimum
\arg \min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}[\text{sign}(w^T x_i) \neq y_i]?

Discrete/combinatorial search; NP-hard in general; awkward for continuous optimization algorithms.
Why not try “pytorch meta-algorithm”, with empirical risk

\[
\arg \min_{\mathbf{w} \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} 1[\text{sign}(\mathbf{w}^T \mathbf{x}_i) \neq y_i]
\]

- Discrete/combinatorial search;
  NP-hard in general; awkward for continuous optimization algorithms.
Relaxing the ERM problem

▶ **Step 1**: remove $\text{sign}(\cdot)$. Let's remove one source of discreteness:

$$\frac{1}{n} \sum_{i=1}^{n} 1[\text{sign}(w^T x_i) \neq y_i] \quad \rightarrow \quad \frac{1}{n} \sum_{i=1}^{n} 1\left[y_i(w^T x_i) \leq 0\right].$$

Are these equivalent?

▶ **Step 2**: remove $1[\cdot]$. Rewrite the preceding as

$$\hat{R}_{zo}(w) = \frac{1}{n} \sum_{i=1}^{n} \ell_{zo}(y_i w^T x_i) \quad \text{where} \quad \ell_{zo}(z) = 1[z \leq 0].$$

Here, $y_i(w^T x_i)$ is the (unnormalized) margin of $w$ on example $i$.

Next let's replace $\ell_{zo}$ with something continuous!

*(Side note: squared loss had two arguments, today we'll use one. We'll discuss this point next lecture.)*
Logistic loss

We want to choose a nice loss $\ell$ in

$$\hat{R}(w) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i w^T x_i).$$

Key desired properties:

- $\ell$ is continuous (for sake of optimization);
- $\ell$ prefers correct classifications:
  if $y_i w^T x_i > 0$ (correct), then $\ell(y_i w^T x_i) < \ell(-y_i w^T x_i)$. 

Examples.

- Squared loss, written in margin form:
  $$\ell_{\text{ls}}(z) := \frac{1}{2} (1 - z)^2;$$
  note $\ell_{\text{ls}}(y \hat{y}) = (1 - y \hat{y})^2 = (y - \hat{y})^2$.

- Logistic loss:
  $$\ell_{\text{log}}(z) = \ln(1 + \exp(-z)).$$
  This one doesn't care so long as $z = y \hat{y} > 0$. 
Logistic loss

We want to choose a nice loss $\ell$ in

$$\widehat{\mathcal{R}}(w) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i w^T x_i).$$

Key desired properties:

- $\ell$ is continuous (for sake of optimization);
- $\ell$ prefers correct classifications:
  if $y_i w^T x_i > 0$ (correct), then $\ell(y_i w^T x_i) < \ell(-y_i w^T x_i)$.

Examples.

- **Squared loss**, written in margin form: $\ell_{ls}(z) := \frac{1}{2} (1 - z)^2$; note
  $$2\ell_{ls}(y \hat{y}) = (1 - y \hat{y})^2 = y^2 (1 - y \hat{y})^2 = (y - \hat{y})^2.$$
  Squared loss doesn’t just push $y_i \hat{y}_i$ positive; it wants $y_i \hat{y}_i = 1$!

- **Logistic loss**: $\ell_{\log}(z) = \ln(1 + \exp(-z))$.
  This one doesn’t care so long as $z = y \hat{y} > 0$. 
Squared and logistic losses on linearly separable data

Logistic loss.  
Squared loss.
Squared and logistic losses on linearly separable data II

Logistic loss.  Squared loss.
Squared and logistic losses on linearly separable data II

Logistic loss.  
Squared loss.  

(Math note: it’s easy to prove this.)
Least squares and logistic ERM

Least squares:

$\text{Take gradient of } \ell_{\text{LS}}(w) = \|Xw - y\|^2$; set to 0; obtain normal equations $X^TXw = X^Ty$.

Suffice to take OLS solution $\hat{w}_{\text{ols}} = (X^TX)^{-1}X^Ty$.

Alternatively, gradient descent.

Logistic loss:

$\text{Unclear how to solve } \nabla_w R_{\text{log}}(w) = \nabla_w \sum_{i=1}^n \ln(1 + \exp(-y_i w^T x_i)) = 0$.

$\text{Gradient descent still fine!}$
Least squares and logistic ERM

Least squares:

- Take gradient of $2n\hat{R}_{ls}(w) = \|Xw - y\|^2$, set to 0;
  
  - obtain normal equations $X^TXw = X^Ty$.
  
  Suffice to take OLS solution $\hat{w}_{ols} = X^+y$.

- Alternatively, gradient descent.

Logistic loss:

- Unclear how to solve

$$\nabla_w \hat{R}_{log}(w) = \nabla_w \frac{1}{n} \sum_{i=1}^n \ln(1 + \exp(-y_i w^T x_i)) = 0.$$ 

- Gradient descent still fine!
Given an empirical risk $\hat{R} : \mathbb{R}^P \rightarrow \mathbb{R}$, gradient descent is the iteration

$$w_{i+1} := w_i - \eta_i \nabla_w \hat{R}(w_i),$$

where $w_0$ is given, and $\eta_i$ is a learning rate (step size).

Gradient descent goes down the contours of $\hat{R}$:
Given an empirical risk $\hat{R} : \mathbb{R}^P \rightarrow \mathbb{R}$, gradient descent is the iteration

$$w_{i+1} := w_i - \eta_i \nabla_w \hat{R}(w_i),$$

where $w_0$ is given, and $\eta_i$ is a learning rate (step size).

Gradient descent goes down the contours of $\hat{R}$:

![Graph showing contours of $\hat{R}$]

**Remarks.**

- In the convexity lecture, we'll show this works for linear and logistic regression.
- For deep networks, $w_0$ is random, and $\eta_i$ is highly tuned/varying.
For logistic loss and linear predictors, typically $w_0 = 0$, and

```python
def GD(X, y, loss, step = 0.1, n_iters = 10000):
    w = torch.zeros(X.shape[1], requires_grad = True)
    for i in range(n_iters):
        l = loss(X, y, w).mean()
        l.backward()
        with torch.no_grad():
            w = step * w.grad
            w.grad.zero()
    return w
```
For logistic loss and linear predictors, typically $w_0 = 0$, and

$$w_{i+1} := w_i - \eta_i \nabla_{w} \hat{R}_\text{log}(w_i) = w_i - \frac{\eta_i}{n} \sum_{j=1}^{n} \ell'_{\text{log}}(y_j x_j^T w_i) y_j x_j,$$

where $\ell'_{\text{log}}(z) = \frac{-1}{1 + \exp(z)}$. 

```python
def GD(X, y, loss, step = 0.1, n_iters = 10000):
    w = torch.zeros(X.shape[1], requires_grad = True)
    for i in range(n_iters):
        l = loss(X, y, w).mean()
        l.backward()
        with torch.no_grad():
            w = step * w.grad
            w.grad.zero()
    return w
```
For logistic loss and linear predictors, typically $w_0 = 0$, and

$$w_{i+1} := w_i - \eta_i \nabla_w \hat{R}_{\text{log}}(w_i) = w_i - \frac{\eta_i}{n} \sum_{j=1}^{n} \ell'_{\text{log}}(y_j x_j^T w_i) y_j x_j,$$

where $\ell'_{\text{log}}(z) = \frac{-1}{1+\exp(z)}$.

**But who cares**, pytorch does it for us:

```python
def GD(X, y, loss, step = 0.1, n_iters = 10000):
    w = torch.zeros(X.shape[1], requires_grad = True)
    for i in range(n_iters):
        l = loss(X, y, w).mean()
        l.backward()
        with torch.no_grad():
            w -= step * w.grad
            w.grad.zero_()
    return w
```
The (negative) derivative $-\ell'_\text{log}(z) = \frac{1}{1+e^z}$ is the logistic function.

It can suggest a probability / confidence of the output label. We'll revisit this next lecture.

**Warning:** many treat these explicitly as confidences, but they are not.
Summary for today

- Linear classifiers.
- ERM for classification.
- Solving the ERM problem.
(Appendix.)
If there exists a perfect linear separator, empirical logistic risk minimization should find it.

**Theorem.**

If there exists a \( \bar{w} \) with \( y_i \bar{w}^T x_i > 0 \) for all \( i \), then every \( w \) with \( b_R \log(w) < \frac{\ln(2)}{2n} + \inf_v b_R \log(v) \) also satisfies \( y_i w^T x_i > 0 \).

**Proof.**

**Step 1:** low risk implies few mistakes. For any \( w \) with \( y_j w^T x_j \leq 0 \) for some \( j \), \( b_R \log(w) \geq \frac{1}{n} \ln(1 + \exp(-y_j w^T x_j)) \geq \frac{\ln(2)}{n} \). By contrapositive, any \( w \) with \( b_R \log(w) < \frac{\ln(2)}{n} \) makes no mistakes.

**Step 2:** \( \inf_v b_R \log(v) = 0 \). Note: \( 0 \leq \inf_v b_R \log(v) \leq \inf_{r > 0} \frac{1}{n} \sum_{i=1}^{n} \ln(1 + \exp(-ry_i \bar{w}^T x_i)) = 0 \).

**Remark.** We didn't prove that gradient descent finds such a predictor, but that in turn is an easy consequence of the above and one bound from the upcoming convexity lecture.
Logistic risk and separation

If there exists a perfect linear separator, empirical logistic risk minimization should find it.

**Theorem.** If there exists \( \bar{w} \) with \( y_i \bar{w}^T x_i > 0 \) for all \( i \), then every \( w \) with \( \hat{R}_{\text{log}}(w) < \frac{\ln(2)}{2n} + \inf_v \hat{R}_{\text{log}}(v) \), also satisfies \( y_i w^T x_i > 0 \).
If there exists a perfect linear separator, empirical logistic risk minimization should find it.

**Theorem.** If there exists $\bar{w}$ with $y_i \bar{w}^T x_i > 0$ for all $i$, then every $w$ with $\hat{R}_{\log}(w) < \ln(2)/2n + \inf_v \hat{R}_{\log}(v)$, also satisfies $y_i w^T x_i > 0$.

**Proof.**

**Step 1:** low risk implies few mistakes. For any $w$ with $y_j w^T x_j \leq 0$ for some $j$,

$$\hat{R}_{\log}(w) \geq \frac{1}{n} \ln(1 + \exp(-y_j w^T x_j)) \geq \frac{\ln(2)}{n}.$$

By contrapositive, any $w$ with $\hat{R}_{\log}(w) < \ln(2)/n$ makes no mistakes.

**Step 2:** $\inf_v \hat{R}_{\log}(v) = 0$. Note:

$$0 \leq \inf_v \hat{R}_{\log}(v) \leq \inf_{r > 0} \frac{1}{n} \sum_{i=1}^n \ln(1 + \exp(-r y_i \bar{w}^T x_i)) = 0.$$

**Remark.** We didn’t prove that gradient descent finds such a predictor, but that in turn is an easy consequence of the above and one bound from the upcoming convexity lecture.
The Perceptron algorithm (streamed data!)

Start from $w_0 = 0$, and thereafter rotate towards $y_i x_i$ if wrong:

$$w_{i+1} := w_i + y_{i+1} x_{i+1} \mathbb{1}[y_{i+1} w_i^T x_{i+1} \leq 0].$$

Remark: this is a specific subgradient of $\partial_w \max\{0, -y_{i+1} x_{i+1}^T w\}$.

Theorem (perceptron convergence, Novikoff '62). Suppose there exists $u$ with $\|u\|_2 = 1$ and $u^T x_i y_i \geq \gamma > 0$, and $\|x_i\| \leq 1$. Then perceptron makes at most $1/\gamma^2$ mistakes:

$$\sum_{i < t} \mathbb{1}[w_i^T x_{i+1} y_{i+1} \leq 0].$$

Proof. Define mistake set $\mathcal{M} := \{i < t : \mathbb{1}[w_i^T x_{i+1} y_{i+1}]\}$. Note

$$\|w_t\| \geq w_t^T u = \sum_{i \in \mathcal{M}} y_i x_i^T u \geq \gamma |\mathcal{M}|.$$

On the other hand, by induction,

$$\|w_t\|^2 = \|w_{t-1}\|^2 + y_t x_t^T w_{t-1} \mathbb{1}[t - 1 \in \mathcal{M}] + \|y_t x_i \mathbb{1}[t - 1 \in \mathcal{M}]\|^2 \leq \|w_{t-1}\|^2 + \mathbb{1}[t - 1 \in \mathcal{M}] \leq \cdots \leq |\mathcal{M}|.$$

Together, $\gamma |\mathcal{M}| \leq \|w_t\| \leq \sqrt{|\mathcal{M}|}$, and $|\mathcal{M}| \leq 1/\gamma^2$. \qed