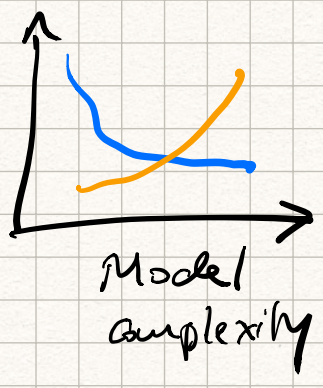
ECE-GY 6143
Intro to ML

- [x] Recap: Model selection
- [x] Regularization
- [x] Logistic Regression.

- Bias vs Variance
- Thumb rule
  - [ ] Low bias
    ("higher" complexity)
  - [ ] low variance
    ("lower" complexity).



Regularization

- Method to control complexity of an ML model.

Components
- Representation ✓
- Loss function ?
- Optimization ✓

Instead of minimizing MSE, define a new loss function

$$L(w) = MSE(w) + \alpha \phi(w)$$

α: scalar ≥ 0

"Regularizer"

Promote simpler w.

Penalize "unlikely" w

Examples.

1) $\phi(w) = \|w\|_2^2$ squared L2-norm.

2) $\phi(w) = \|w\|_1$ L1 norm.

3) $\phi(w) = \frac{1}{2}\|w\|_2^2 + \frac{1}{2}\|w\|_1$ "Elastic net".

4) $\phi(w) = \sum_{i=1}^{d} \left(w_i - \frac{1}{d} \sum_{j=1}^{d} w_j \right)^2$

$$\phi(w) = \sum_{i=1}^{d-1} (w_i - w_{i+1})^2$$

Interpretation in terms of linear regression

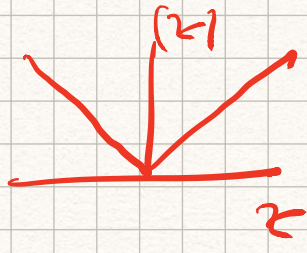
① $L(w) = \frac{1}{2}\|y - Xw\|_2^2 + \alpha \|w\|_2^2$

Set $\nabla L(w) = 0$ and solve for w.

$$w^* = \left(X^T X + \alpha I\right)^{-1} X^T y.$$

"Ridge regression"

② $$L(w) = \frac{1}{2} \|y - Xw\|_2^2 + \alpha \|w\|_1$$



$\|w\|_1 = \sum_{i=1}^{d} |w_i|$

Cannot optimize using gradient descent but can use other algorithms

[sub-gradient descent, LARS, fixed point continuation, ...]

"LASSO regression"

$\phi(w) = \|w\|_2^2 \longrightarrow$ Shrinks values of w.

$\phi(w) = \|w\|_1 \longrightarrow$ Sparsifies w.

Logistic regression

Thus far: regression

Training Dataset: (x, y)

Data ↑ ↑ label (Continuous)

Q: How to model classification.

$y = 0 \longrightarrow$ NO

$y = 1 \longrightarrow$ YES.

Pros: Can use linear regression directly.

Cons:

$$y = \langle w, x \rangle$$

- Not scale invariant.
  [100x gives 100y].
- Labelling not unique.

Solution: Instead of finding model f

s.t. $y_i = f(x_i)$

instead we predict

$$\text{Prob}(y_i = 1 \mid x_i) = f(x_i).$$

↳ Conditional probability of predicting label YES given data point x_i.

$\Rightarrow$ $\text{Prob}(y_i = 0 \mid x_i) = 1 - f(x_i)$.

[assuming 2 classes].

Combine into one equation.

$$\text{Prob}(y_i \mid x_i) = f(x_i)^{y_i} \left(1 - f(x_i)\right)^{1-y_i}$$

Dataset (x_1, y_1), (x_2, y_2), $\vdots$, (x_n, y_n) with independent samples.

$$X = [x_1, \dots x_n] \quad y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

$$\text{Prob}(y \mid X) = p(y_1 \mid x_1)\, p(y_2 \mid x_2) \dots p(y_n \mid x_n)$$

$$= \prod_{i=1}^{n} \text{prob}(y_i \mid x_i)$$

$$= \prod_{i=1}^{n} f(x_i)^{y_i} \left(1 - f(x_i)\right)^{1-y_i}$$

"Likelihood" of training dataset.

More convenient to take log.

$$-\log \text{Prob}(y|x) = -\left[\sum_{i=1}^{n} y_i \log f(x_i) + (1-y_i)\log(1-f(x_i))\right]$$

Negative log likelihood.

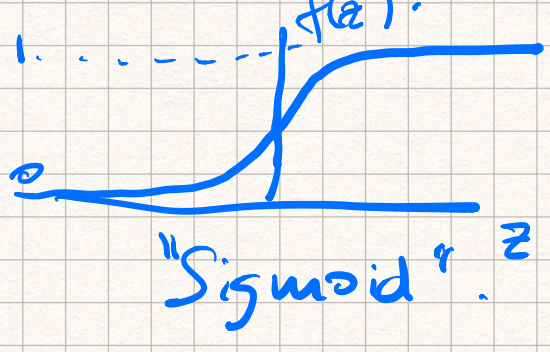
$$l_i(f) = -\left[y_i \log f(x_i) + (1-y_i)\log(1-f(x_i))\right]$$

"Cross-entropy".

$$\rightarrow L(f) = \sum_{i=1}^{n} l_i(f).$$

$$f(x_i) = \frac{1}{1+e^{-\langle w, x_i\rangle}}$$

$$f(z) = \frac{1}{1+e^{-z}}$$



"Sigmoid".

$$L(w) = \sum_{i=1}^{n} l_i(w)$$

$$1 - \frac{1}{1+e^{-z}} = \frac{e^{-z}}{1+e^{-z}}$$

$$l_i(w) = -\left(y_i \log \frac{1}{1+e^{-\langle w, x_i\rangle}} + \right.$$

$$(1-y_i)\log\frac{e^{-\langle w,x_i\rangle}}{1+e^{-\langle w,x_i\rangle}}\Big)$$

Algorithm to minimize $L(w)$

: gradient descent.

Fact: $\frac{df(z)}{dz} = f(z)(1-f(z))$

GD

$$w_{k+1} = w_k + \alpha_k \sum_{i=1}^{n} \left(y_i - \frac{1}{1+e^{-\langle w,x_i\rangle}}\right) x_i$$

Multiclass classification.

k classes $1, \ldots, k$.

$$\text{Prob}(y_i = k \mid x_i) = \frac{\exp(\langle w, x_i\rangle)}{Z}$$

Z: normalization "partition function"

$$Z = \sum_{k=1}^{K} \exp(\langle w_k, x_i\rangle).$$

$$L(w) = \sum_{i=1}^{n} l_i(w).$$

$$l_i(w) = \sum_{k=1}^{K} \mathbb{1}(y_i = k) \log \frac{\exp(\langle w_k, x_i \rangle)}{Z}$$

"Soft-max"