Supervised Machine Learning Review

by Paul Hand
Northeastern University

Outline

Regression + Classification Problems
Statistical Framework for ML
Justification for square loss & cross entropy loss
Bias Variance Trade off, model selection, an unexpected twist

Common Problems in Supervised ML

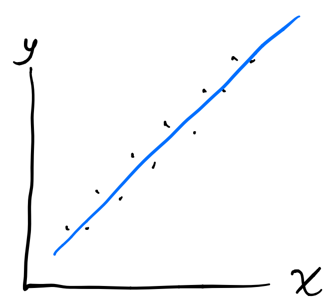
Regression: predict a continuous value

Let $f: \mathbb{R}^d \to \mathbb{R}$

$y = f(x) + \text{noise}$

Given: $\{(x_i, y_i)\}_{i=1\dots n}$

Find: f



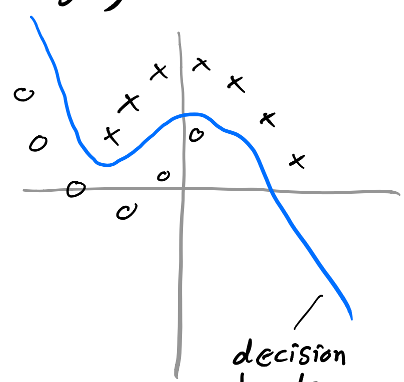
Classification: predict membership in a category

Let $f: \mathbb{R}^d \to \begin{Bmatrix} \text{cat } 1 \\ \vdots \\ \text{cat } m \end{Bmatrix}$

$y = f(x) + \text{noise}$

Given: $\{(x_i, y_i)\}_{i=1\dots n}$

Find: f



Terminology:

x – input variables, predictors, independent vars, features
y – response, dependent variable, output variable
f – model, predictor, hypothesis

Statistical Framework for ML (supervised)

Assume:

- (x, y) are sampled from a joint probability distribution
- Training data $D = \{(x_i, y_i)\}_{i=1 \cdots n}$ are iid samples
- Test data are also iid samples

Can estimate the model/predictor by maximum likelihood estimation

Results (usually) in an optimization problem

$$\hat{f} = \underset{f \in \mathcal{H}}{\operatorname{argmin}} \sum_{i=1}^{n} \ell(f(x_i), y_i)$$

where

ℓ ~ loss function eg $\ell(\hat{y}, y) = |\hat{y} - y|^2$

$\mathcal{H}$ - hypothesis class eg degree d polynomial

Q's: What loss do you choose and why?

What hypotheses should you search over?

Linear Regression and Square Loss

Let $a \in \mathbb{R}^d$, $x \in \mathbb{R}^d$

Model: $y_i = x_i^t a + \varepsilon_i$ w/ $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$

Data: $D = \{(x_i, y_i)\}_{i=1\cdots n}$

Estimate a by maximum likelihood

pdf of ε_i is $\frac{1}{\sqrt{2\pi}\,\sigma} e^{-\frac{z^2}{2\sigma^2}}$ over $z \in \mathbb{R}$

likelihood of data (using $\varepsilon_i = y_i - x_i^t a$)

$$L(a) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\,\sigma} e^{-(y_i - x_i^t a)^2 / 2\sigma^2}$$

$$\log L(a) = -\sum_{i=1}^{n} \frac{(y_i - x_i^t a)^2}{2\sigma^2} + \text{terms constant in } a$$

maximizing data likelihood $\Leftrightarrow$ minimizing square loss

$$\max_a L(a) \Leftrightarrow \min_a \sum_{i=1}^{n} \underbrace{(x_i^t a - y_i)^2}_{\text{square loss } \ell(\hat{y}, y) = |\hat{y} - y|^2}$$

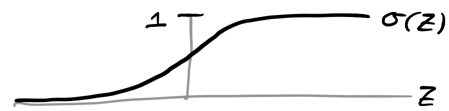
Logistic Regression and Cross Entropy Loss

Model:

Let $a \in \mathbb{R}^d$

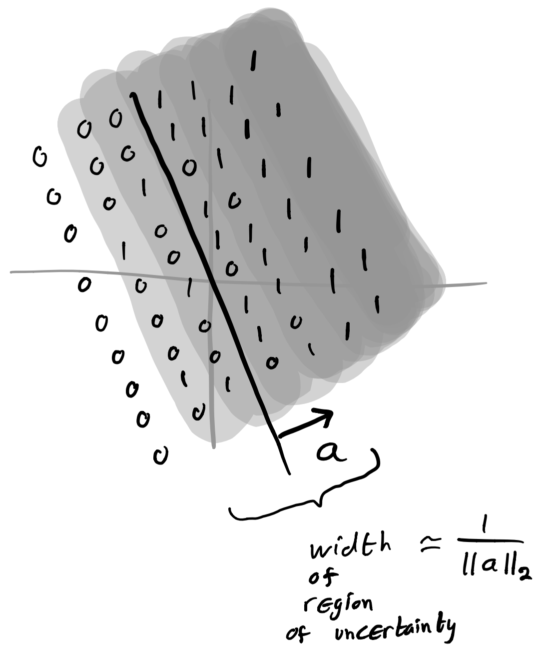
$$P(y=1 \mid x) = \sigma(x^t a)$$
$$P(y=0 \mid x) = 1 - \sigma(x^t a)$$

w/ $\sigma(z) = \dfrac{e^z}{e^z + 1} = \dfrac{1}{1 + e^{-z}}$



Data: $\{(x_i, y_i)\}$

Visually:



Estimate a by maximum likelihood

$$L(a) = \prod_{i=1}^{n} P(y_i = 0 \mid x_i)^{1-y_i} \, P(y_i = 1 \mid x_i)^{y_i}$$

$$\log L(a) = \sum_{i=1}^{n} (1-y_i) \log P(y_i = 0 \mid x_i) + y_i \log P(y_i = 1 \mid x_i)$$

Cross entropy loss

$$\ell_{CE}(\underbrace{p, q}_{\text{discrete r.v.s over } Z}) = -\sum_{z \in Z} p(z) \log q(z) = -\mathbb{E}_p(\log q)$$

Maximizing data likelihood $\Leftrightarrow$ minimizing cross entropy loss

$$\max_a L(a) \Leftrightarrow \min_a -\sum_{i=1}^{n} \underbrace{\left(y_i \log(\sigma(x_i^t a)) + (1-y_i)\log(1-\sigma(x_i^t a))\right)}_{\ell_{CE}\left(\begin{pmatrix} y_i \\ 1-y_i \end{pmatrix}, \begin{pmatrix} \sigma(x_i^t a) \\ 1-\sigma(x_i^t a) \end{pmatrix}\right)}$$

Note:

Cross entropy loss penalizes data points of a observed category to which the model assigns a very low probability.

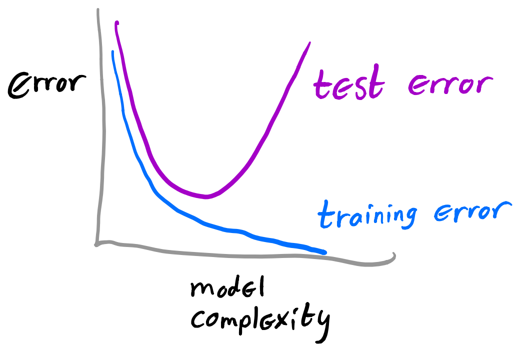
Question to ponder:

Is minimizing Cross Entropy loss all that different from minimizing a square loss in the case of regression?

Bias-Variance Tradeoff

What class of hypotheses should you search over?

Standard Statistical ML story:



higher complexity models have lower bias but higher variance

If complexity is too high, it overfits data, variance term dominates test error

after a certain threshold, "larger models are worse"

Bias-Variance Decomposition

Consider regression model

$$y = f(x) + \varepsilon \qquad w/ \quad \mathbb{E}[\varepsilon \mid x] = 0$$

Let $D = \{(x_i, y_i)\}_{i=1 \cdots n}$ be iid samples

Estimate f by an algorithm producing $\hat{f}_D$

Evaluate $\hat{f}_D$ by expected loss on a new sample

$$\underbrace{R(\hat{f}_D)}_{\text{risk}} = \underbrace{\mathbb{E}_{x,y}}_{\text{test sample}} \underbrace{(\hat{f}_D(x) - y)^2}_{\text{square loss}}$$

Performance will vary based on D. Take expectation over D.

$$\mathbb{E}_D \, R(\hat{f}_D) = \mathbb{E}_{x,y,D} (\hat{f}_D(x) - y)^2$$

We will decompose into 3 effects: bias, variance, irreducible error

$$\mathbb{E}_D R(\hat f_D) = \mathbb{E}_{x,y,D}\left[(\hat f_D(x) - f(x) - \varepsilon)^2\right]$$

$$= \mathbb{E}_{x,y,D}(\hat f_D(x) - f(x))^2 - 2\,\mathbb{E}\left[\cancelto{0}{(\hat f_D(x) - f(x))\varepsilon}\right] + \underbrace{\mathbb{E}[\varepsilon^2]}_{Var(\varepsilon)}$$

$$= \mathbb{E}_{x,y,D}(\hat f_D(x) - f(x))^2 + Var(\varepsilon)$$

Evaluating the first term, conditioning on x,

$$\mathbb{E}_D(\hat f_D(x) - f(x))^2 = \mathbb{E}_D\left[\left((\hat f_D(x) - \mathbb{E}_D \hat f_D(x)) + (\mathbb{E}_D \hat f_D(x) - f(x))\right)^2\right]$$

$$= \mathbb{E}_D(\hat f_D(x) - \mathbb{E}_D \hat f_D(x))^2 + 2\,\mathbb{E}_D \underbrace{\cancelto{0}{(\hat f_D(x) - \mathbb{E}_D \hat f_D(x))}}_{0 \text{ in expectation in } D} \underbrace{(\mathbb{E}_D \hat f_D(x) - f(x))}_{\text{does not depend on } D} + \mathbb{E}_D(\mathbb{E}_D \hat f_D(x) - f(x))^2$$

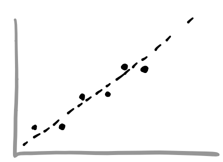
$$= \underbrace{\mathbb{E}_D(\hat f_D(x) - \mathbb{E}_D \hat f_D(x))^2}_{\text{Variance of } \hat f_D(x)} + \underbrace{(\mathbb{E}_D(\hat f_D(x) - f(x)))^2}_{\text{squared bias}}$$

So,

$$\mathbb{E}_D R(\hat f) = \underbrace{\mathbb{E}_x (f(x) - \mathbb{E}_D \hat f_D(x))^2}_{\text{expected squared bias of estimate}} + \underbrace{\mathbb{E}_x Var_D \hat f_D(x)}_{\text{expected variance of estimate}} + \underbrace{Var(\varepsilon)}_{\text{irreducible error}}$$

Illustration of bias variance tradeoff

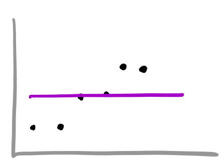
Suppose $y = x + \varepsilon$

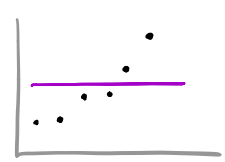


Low complexity model: $y = c$

$\mathbb{E}_x(f(x) - \mathbb{E}_D \hat f_D)^2$ is high

$\mathbb{E}_x Var_D \hat f_D(x)$ is low

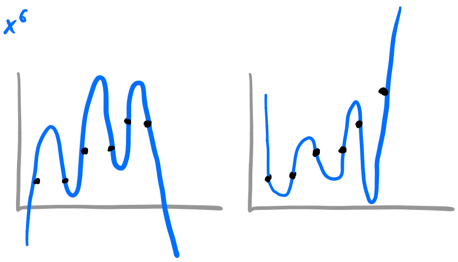




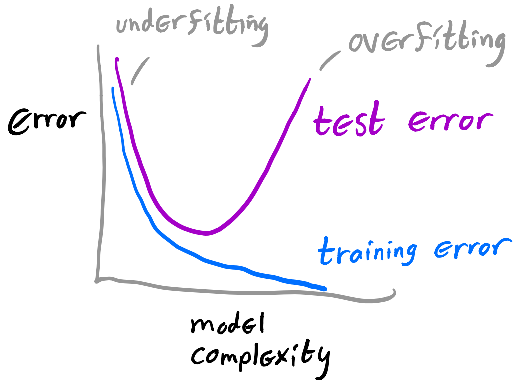
High complexity model: $y = c_0 + c_1 x + c_2 x^2 + \cdots c_6 x^6$

$\mathbb{E}_x \left(f(x) - \mathbb{E}_D \hat{f}_D \right)^2$ is low

$\mathbb{E}_x \, \mathrm{Var}_D \hat{f}_D(x)$ is high



Standard Statistical ML story:

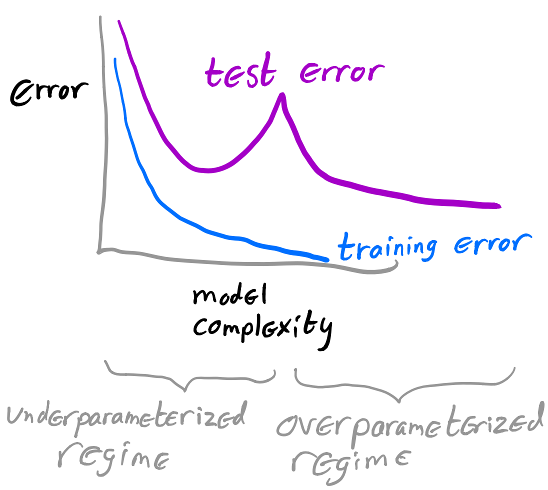


higher complexity models have lower bias but higher variance

If complexity is too high, it overfits data, variance term dominates test error

after a certain threshold, "larger models are worse"

Modern Story based on Neural Nets:



Test error can decrease as model complexity continues increasing.

And it can be lower than in underparameterized regime

Phenomenon: double descent

"larger models are better"