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

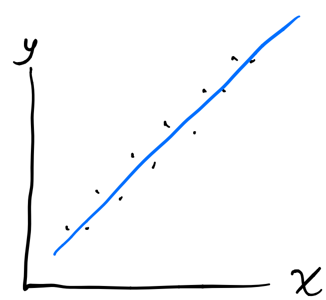
Regression: predict a continuous value

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

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

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

Find: f



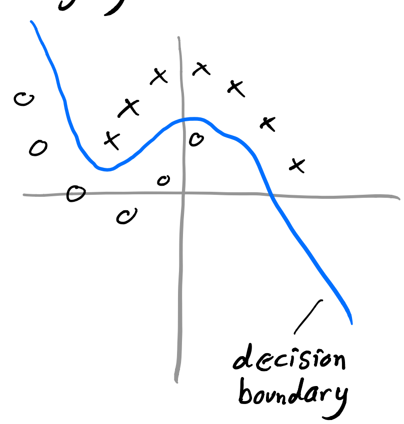
Classification: predict membership in a category

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

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

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 $D = \{(x_i, y_i)\}_{i=1 \ldots n}$ are iid samples
- Test data are also iid samples

Can estimate the model/predictor by maximum likelihood estimation

Results (usually) in an optimization problem

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

"empirical risk minimization"

separable

where

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

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

What is MLE?

Estimate parameters of a model by maximizing likelihood of the observed data

What is MLE in contrast to?

MAP - maximum a posteriori estimation - parameters have some prior distribution, data is collected, that changes the posterior distribution via Bayes Rule. Seek mode of that posterior

I just choose to minimize square loss for a binary classification problem

Is ERM guaranteed to give you a "good" predictor?

Perhaps you get a local minimum instead of the global minimum

No, You may be doing well on training data but not on test data - overfitting

What property is desired in the learned predictor?

Good performance on test data (future i.i.d. Samples of the distribution)

Want: Minimize the expected loss under the test distribution

What is risk?

Risk is expected loss

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

$$\approx \mathbb{E}_{x,y \sim D} \, \ell(f(x), y)$$

this is empirical because it uses empirical data to estimate the expectation of loss over the training distribution

Is risk minimization biased (in a cultural sense) when applied to real problems where $\{x_i\}$ correspond to people?

Biased toward the training data - If a group is underrepresented in training data, then performance on that group may be worse

The data itself could have historical biases baked in

Just because a group has a larger fraction of the data might not mean that we want improvements in performance on that group to balance decreases in performance of other smaller groups

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

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

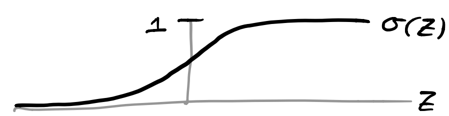
Logistic Regression and Cross Entropy Loss

Model:

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

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

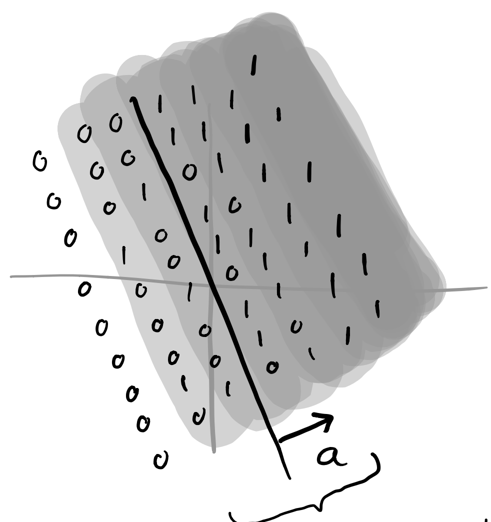
w/ $\sigma(z) = \frac{e^z}{e^z+1} = \frac{1}{1+e^{-z}}$



$x^t a$ is a logit

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

Visually:



width of region of uncertainty $\simeq \frac{1}{\|a\|_2}$

Estimate a by maximum likelihood

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

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

Cross entropy loss

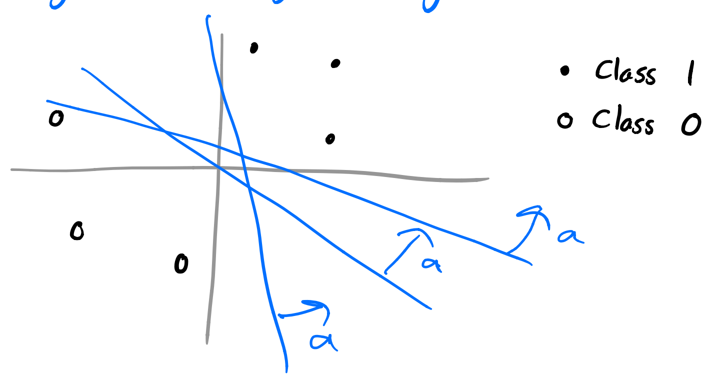
$$\ell_{CE}(p, q) = -\sum_{z \in \mathcal{Z}} p(z) \log q(z) = -\mathbb{E}_p(\log q)$$

discrete r.v.s over $\mathcal{Z}$

Maximizing data likelihood $\Leftrightarrow$ minimizing cross entropy loss

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

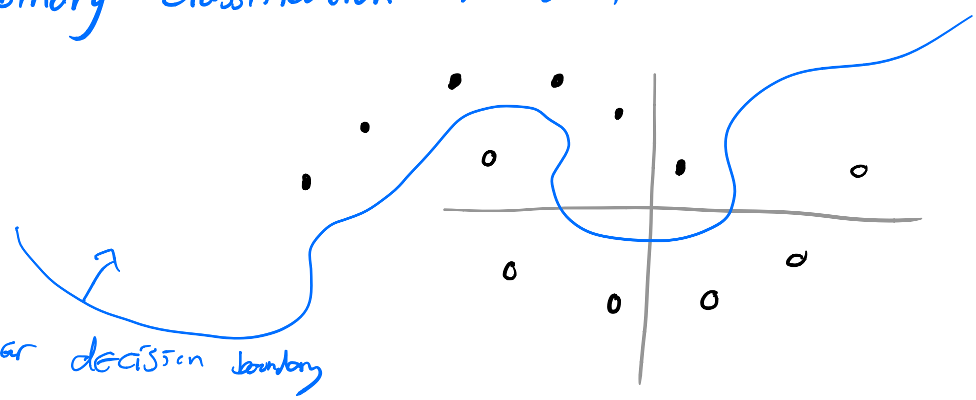
What happens if you do logistic regression on the following data by solving the above optimization problem?



What magnitude of a will result from solving this problem —- infinity - because that will increase the likelihood of the day

Cross-entropy is an asymmetric measure of the distance between two distributions.

Logistic Regression is like a simple version of binary classification w/ neural nets



nonlinear decision boundary

Note:

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

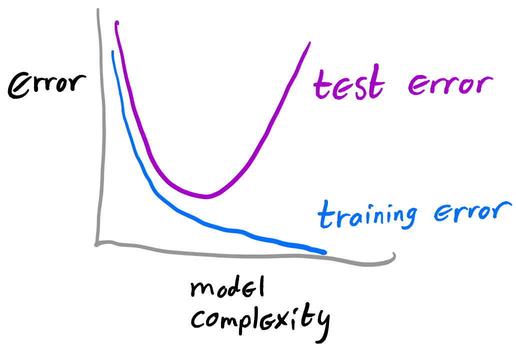
Question to ponder:

Is minimizing Cross Entropy loss all that different from minimizing a square loss in the case of logistic 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"

Why is training error monotonically decreasing?

The search space of larger complexity models is larger

Why is test error initially decreasing?

If its too low, it underfits the data (can not represent the "true model")

If you have 10^3 data samples, how complex of a data model would you consider?

< 10^3.   ..... so choose something like like 30 or 100

Why does understanding this tradeoff matter?

Help select the right level of complexity

Say to look for evidence of overfitting

Bias - Variance Decomposition

Consider regression model

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

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

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

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

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

Evaluating the first term, Conditioning on x,

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

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

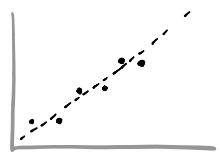
$$= \underbrace{\mathbb{E}_D\left(\hat{f}_D(x) - \mathbb{E}_D \hat{f}_D(x)\right)^2}_{\text{Variance of } \hat{f}_D(x)} + \underbrace{\left(\mathbb{E}_D\left(\hat{f}_D(x) - f(x)\right)\right)^2}_{\text{squared bias}}$$

So,

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

Illustration of bias variance tradeoff

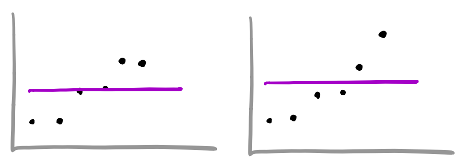
Suppose $y = x + \varepsilon$



Low complexity model: $y = c$

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

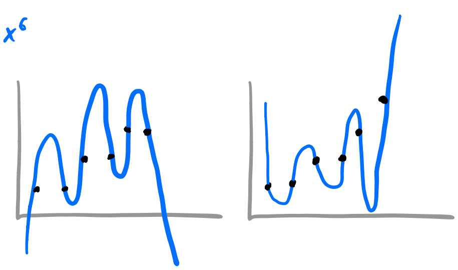
$\mathbb{E}_x \, Var_D \hat{f}_D(x)$ is low



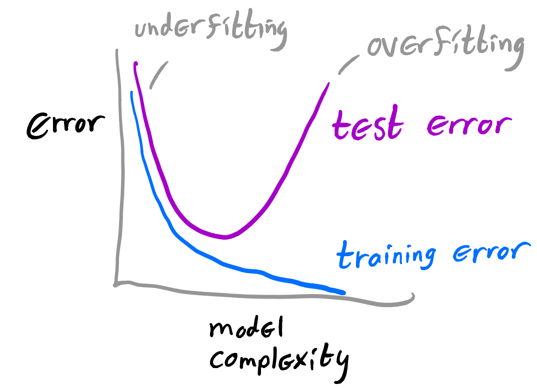
High complexity model: $y = c_0 + c_1 x + c_2 x^2 + \cdots c_6 x^6$

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

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



Standard Statistical ML story:

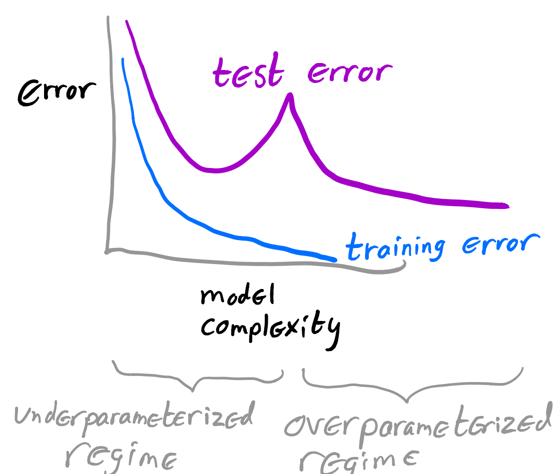


higher complexity models have lower bias but higher variance

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"

Q: Are larger models better b/c we have so much data that it captures the entire problem domain and is actually overfitting?

If you have 10^3 data samples, how complex of a data model would you consider?

Choose a neural network with 10000 or 100000 parameters

Why is being critically parametrized bad for generalization?

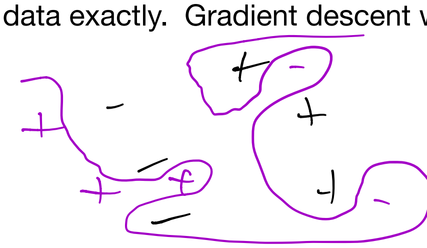
Critically parameterized: # parameters = # data points

How many values of parameters would fit data exactly? 1. Neural net must contort itself to fit the exact data. No expectation for generalization.

In the overparametrized regime, do all models with 0 training error generalize well?

there is an infinity of model parameters that fit data exactly. Gradient descent will find one of them. Would all solutions generalize well?

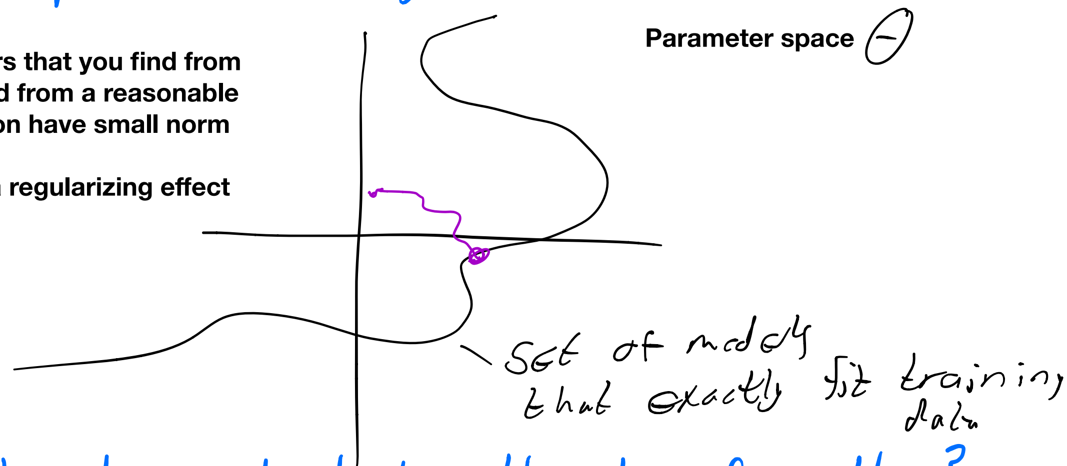
There are solutions that don't generalize well.
Build them by adding poison training data



How is good generalization possible in the overparameterized regime?

Parameters that you find from running Gd from a reasonable initialization have small norm

That has a regularizing effect



Why does understanding this tradeoff matter?

Expect near perfect fitting of your training data