Supervised machine learning

- Machine learning: methods that improve with access to more data.

Examples in genomics:

- Identify sequence elements: Genes, exons, splice sites, enhancers, positioned nucleosomes.
- Assign functional annotations to genes: GO terms, co-functionality.
- Dynamics: Gene expression, regulatory relationships, structure
- Input: sequence, functional genomics, gene expression, populations of sequences

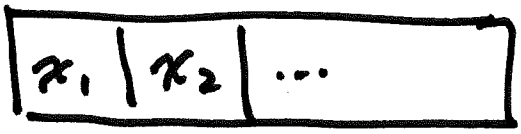
- Blurry line between ML and statistics. ML focuses on large, heterogeneous data sets.

- Supervised ML: predict "label". (distinct from unsupervised)
- Appropriate when prediction > interpretation
- Prediction is the real task: Diagnosis, drug side effects, DNA primer will bind
- Is it predictable? Gene expression, splice sites

Problem setup

- Example: Predict gene expression from functional genomics data.

- ML setup:



"Label" – prediction target (y)

"features" – anything used for prediction ($x_{1..M}$)

- Assume label-feature pairs are independent draws from the same distribution.

- Gene expression example:

  y: is gene expressed? (RNA-seq > cutoff)

  x_1: is H3K36me3 present at promoter? (reads > cutoff)

- Classification: 0/1 labels

  Regression: real-valued labels

Logistic regression

θ_j: Association of x_j with y

$x^{(i)}$:

x_1	x_2	...

θ:

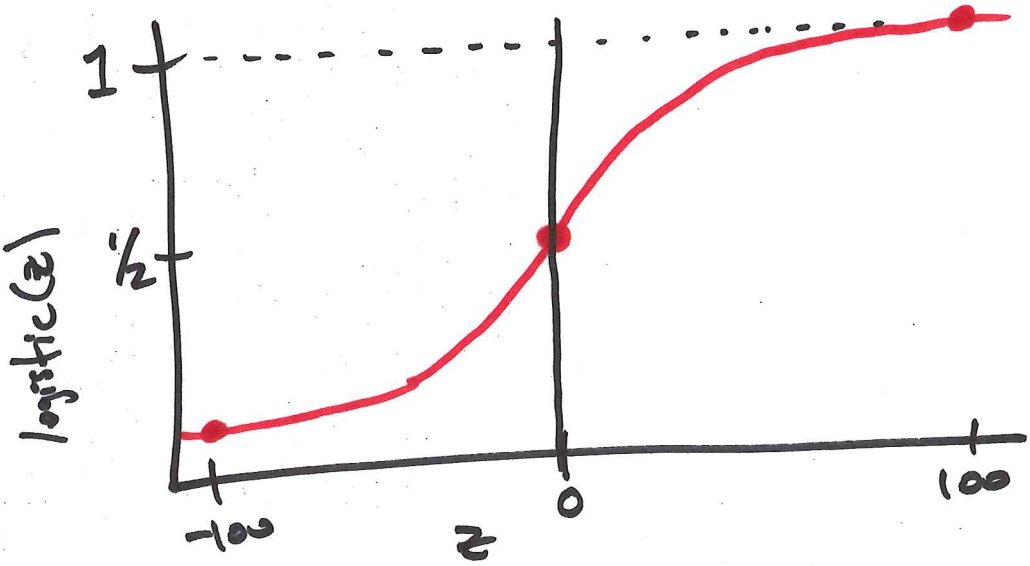
θ_1	θ_2	...

confidence that $y^{(i)} = 1$:

$$\sum_i x_i \cdot \theta_i$$

Map confidence $\longrightarrow$ probability: logistic function

$$\text{logistic}(z) = \frac{\exp(z)}{1 + \exp(z)}$$

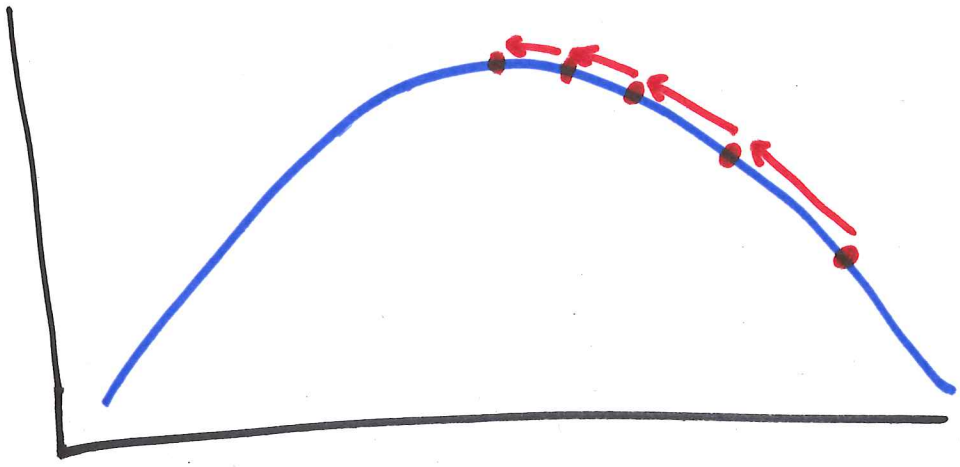


$$h_\theta(x) = \text{logistic}\left(\sum_j x_j \theta_j\right)$$

$$P_\theta(y \mid x) = \prod_i P_\theta(y_i \mid x_i)$$

Optimization

- General problem: Find the maximum of an "objective function".
- Gradient descent



- Derivative of logistic function

$$\text{logistic}(z) = \frac{\exp(z)}{1+\exp(z)} = \frac{1}{\exp(-z)+1}$$

$$\frac{d\,\text{logistic}(z)}{dz} = \left(\frac{1}{\exp(-z)+1}\right)^2 \exp(-z)$$

$$= \text{logistic}(z)\,\frac{\exp(-z)+1-1}{\exp(-z)+1}$$

$$= \text{logistic}(z)\left(1 - \text{logistic}(z)\right)$$

Optimization 2

$$h_\theta(x) = \text{logistic}\left(\sum_j \theta_j x_j\right)$$

$$P(y^{(i)} \mid x^{(i)}) = h_\theta(x^{(i)})^{y^{(i)}} \left(1 - h_\theta(x^{(i)})\right)^{(1-y^{(i)})}$$

$$\ell(\theta) = \log\left(P_\theta(y \mid x)\right) = \sum_i y \log h_\theta(x) + (1-y)\log\left(1 - h_\theta(x)\right)$$

$$\frac{\partial \ell(\theta)}{\partial \theta_j} = \sum_i \left[y^{(i)} \frac{1}{h_\theta(x^{(i)})} - \left(1 - y^{(i)}\right) \frac{1}{1 - h_\theta(x^{(i)})} \right] \left(\frac{\partial}{\partial \theta_j} h_\theta(x^{(i)}) \right)$$

$$= \sum_i \left[\quad \right] h_\theta(x)\left(1 - h_\theta(x)\right) \left(\frac{\partial}{\partial \theta_j} \sum_{j'} x_{j'} \theta_{j'} \right)$$

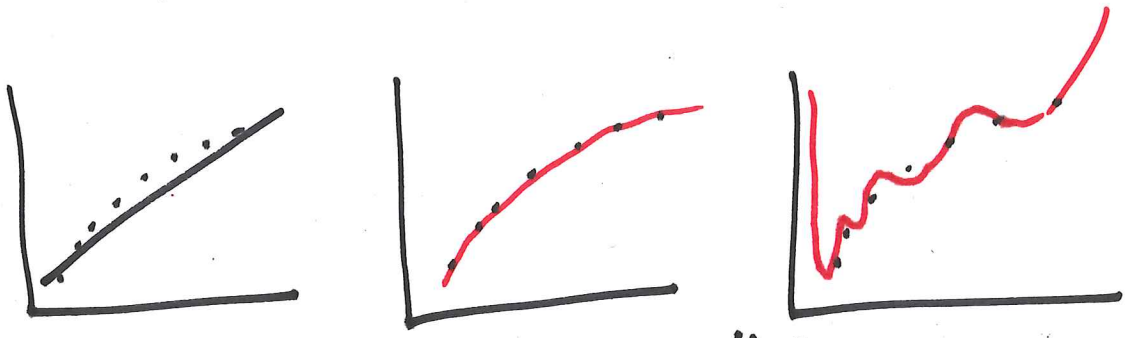
$$= \sum_i \left[y\left(1 - h_\theta(x)\right) - (1-y) h_\theta(x) \right] x_j$$

$$= \sum_i \left[y - h_\theta(x) \right] x_j$$

$$\theta_j := \theta_j + \alpha \sum_i \left(y^{(i)} - h_\theta(x^{(i)}) \right) x_j^{(i)}$$

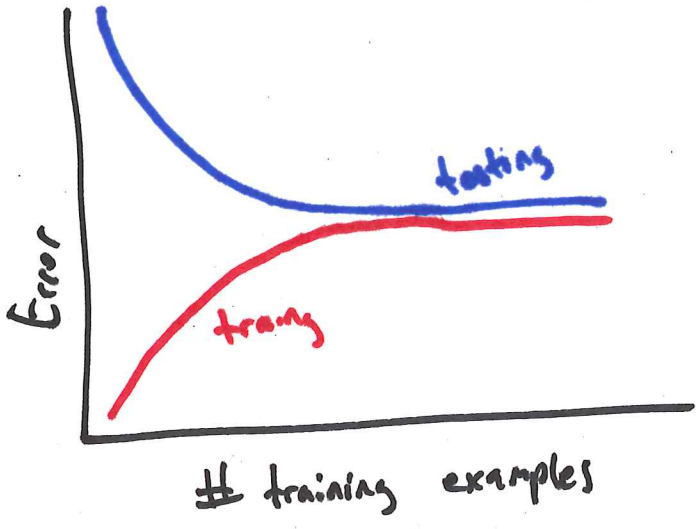
Overfitting and model complexity

- Example: fitting a polynomial



- How well does a model ~~generalize~~ generalize to data not in training set?

- Tradeoff: model complexity



- Use this behavior to guide your choice of model:
  - features; ML algorithm; regularization/prior