COS234:
Introduction To Machine Learning

Prof. Yoram Singer

PRINCETON UNIVERSITY

Topic: Ingredients of Machine Learning Problems

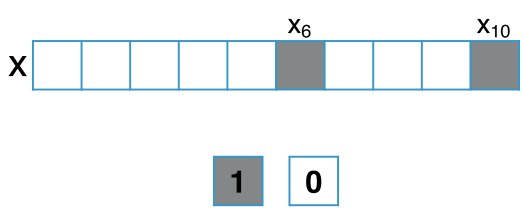
Bag of Words

- Need to classify an email as *spam* or *benign*
- Assume we are provided with a dictionary of "useful" words:
  D={clearance, order, singles, earn, money, cheap, cash, fast, dirt, diploma}
- Index the words in some lexicographic order:
  $\text{clearance} \mapsto 1 \quad \text{order} \mapsto 2 \;\ldots\; \text{diploma} \mapsto 10$
- Represent email as a bag-of-words vector:
  Entry **j** in b.o.w. vector (denoted **x**) is **1** iff word D[j] appears in email

$$x[j] = 1 \;\Leftrightarrow\; D[j] \in \text{Email}$$

Bag of Words (example I)

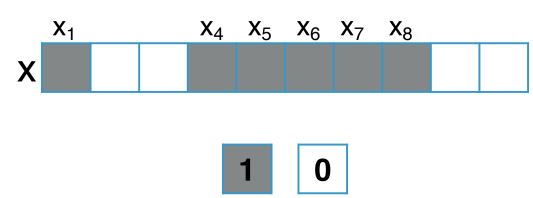
"Dear dad, I would to get a scuba-diving diploma. The course ain't cheap. Could I borrow $1,000 until Aug?"



Note:
$x[4]{=}{=}x_4$

Bag of Words (example II)

"Dear, wouldn't you like to earn some money fast? I got a box of clearance but genuine Rolexes which fell from the back of a driving truck. They are so cheap ($9.99+tax) that I would get one for your mom & dad. Order quickly as these gems are selling fast. I accept only cash though."



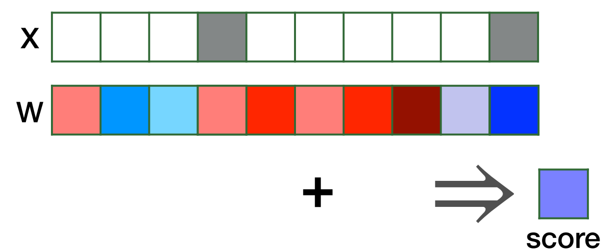
Spam/Benign Classification

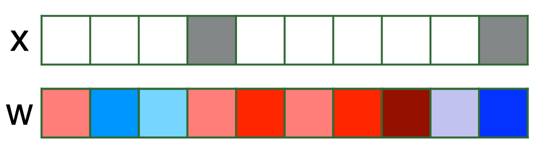
- Suppose each word j is associated with a weight w[j]
- weight w[j] represents relevance of j'th word to `spaminess" of email:
  - w[j] positive number ⇒ email containing D[j] more likely to be **spam**
  - w[j] negative number ⇒ email containing D[j] more likely to be **benign**
- single word (typically) does not provide sufficient evidence
- compounding evidence and weighing "pros" and "cons"

$$\sum_{j:D[j]\in \text{Email}} w[j] \;\Rightarrow\; \text{score}$$

Back to Example I

"Dear dad, I would to get a scuba-diving diploma. The course ain't cheap. Could I borrow $1,000 until Aug?"

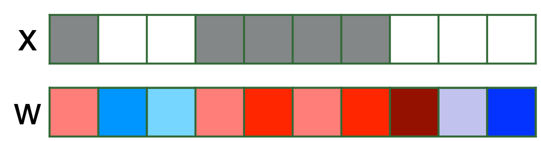




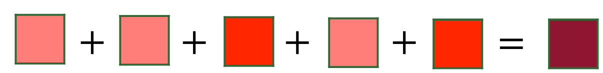
$$s = \sum_{j:D[j]\in \text{Email}} w[j]$$

$$s = \sum_{j:x[j]=1} w[j] = \sum_{j=1}^{|D|} x[j]\,w[j] = \mathbf{x}\cdot\mathbf{w}$$

Inner Product (Dot Product)

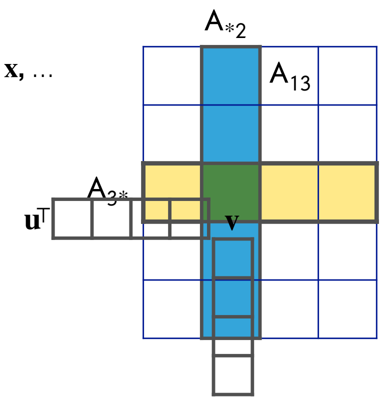


$$s = \sum_{j:x[j]=1} w[j] = \mathbf{x}\cdot\mathbf{w}$$

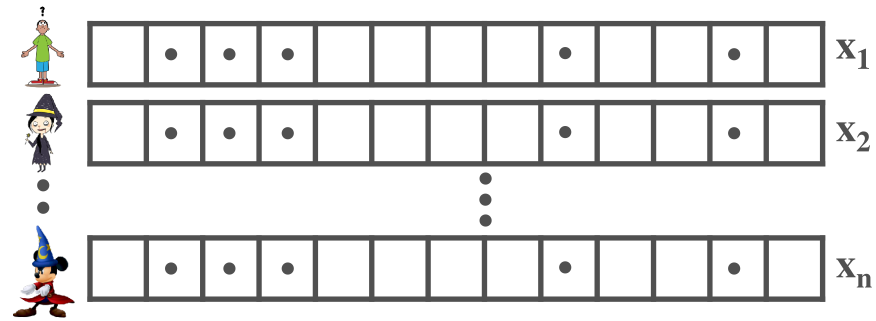


Note on Notation

- [column] vectors in boldface: $\mathbf{u}, \mathbf{v}, \mathbf{w}, \mathbf{x}, \ldots$
- i'th element of vector $\mathbf{v}$: v_i
- element of array $\mathbf{v}$: v[i] & v['abc']
- Matrices uppercase: A, W, X, …
- Element (i,j) of A: A_{ij}
- Row i of A: A_{i*} ; Column j of A: A_{*j}
- Inner (dot) product thus $\mathbf{u} \cdot \mathbf{v} = \mathbf{u}^\top \mathbf{v}$



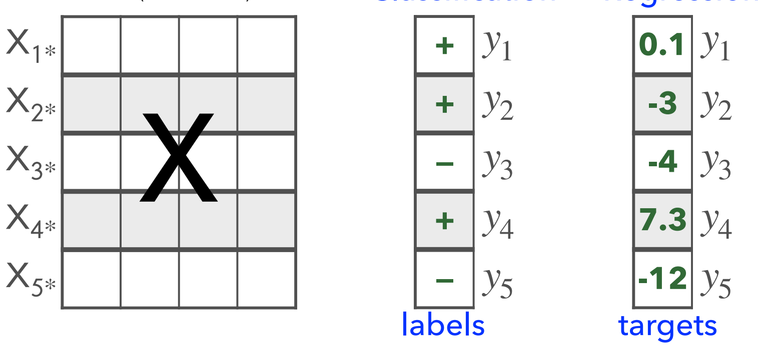
Training Data



Training Data as MATRIX

#examples n=5

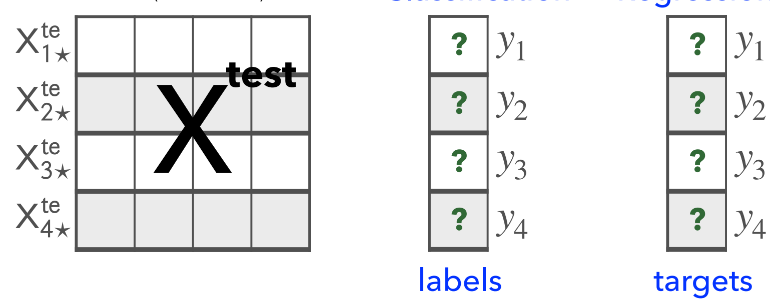
#features (dimension) d=4



Test Data: MATRIX With Unknown Outputs

#test examples n=5

#features (dimension) d=4



Goal of Learning (linear case)

- Given data matrix X and target vector $\mathbf{y}$
- Find weight vector $\mathbf{w}$ such that as often as possible ?

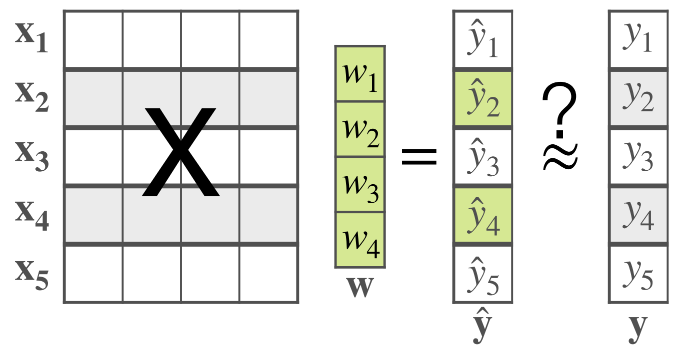
$$\mathbf{w} \cdot \mathbf{x_i} \approx y_i$$

- Goodness of fit between predicted outcome and observed for all data ?

$$\mathrm{X}\mathbf{w} \approx \mathbf{y} \quad ?$$

- And, importantly, also want good predictions on unseen (test) data X^{test}

Re-enter The Matrix

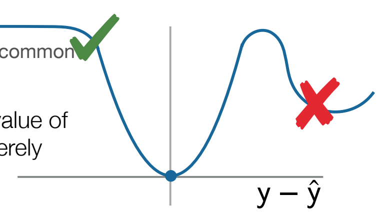


$\approx$

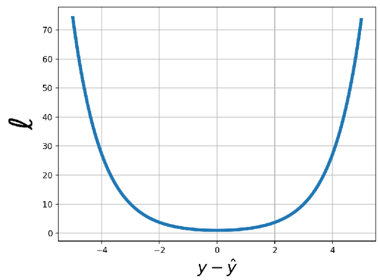
- Nobody is perfect:
  - input-output mapping is typically non-linear
  - noise in inputs & outputs
  - inherent ambiguity [in class experiment]
- Need to assess goodness of fit between y and $\hat{y}$
- Use a loss function: $\ell : \mathfrak{R} \times \mathfrak{R} \rightarrow \mathfrak{R}_+$
- For now, classification aside and focus on regression with $y, \hat{y} \in \mathfrak{R}$

Properties of ℓ

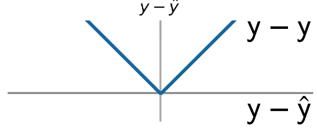
- When $y = \hat{y}$ loss should be 0
- When $y = \hat{y}$ loss should be ≥ 0
- If $|y_2 - \hat{y}_2| > |y_1 - \hat{y}_1|$ we typically want $\ell(y_2, \hat{y}_2) > \ell(y_1, \hat{y}_1)$
- May depend on the value of y and wouldn't be merely function of $|y - \hat{y}|$



Regression Losses



- Squared loss $\ell(y, \hat{y}) = (y - \hat{y})^2$
- Absolute loss $\ell(y, \hat{y}) = |y - \hat{y}|$
- Exponential loss $\ell(y, \hat{y}) = e^{y - \hat{y}} + e^{\hat{y} - y}$

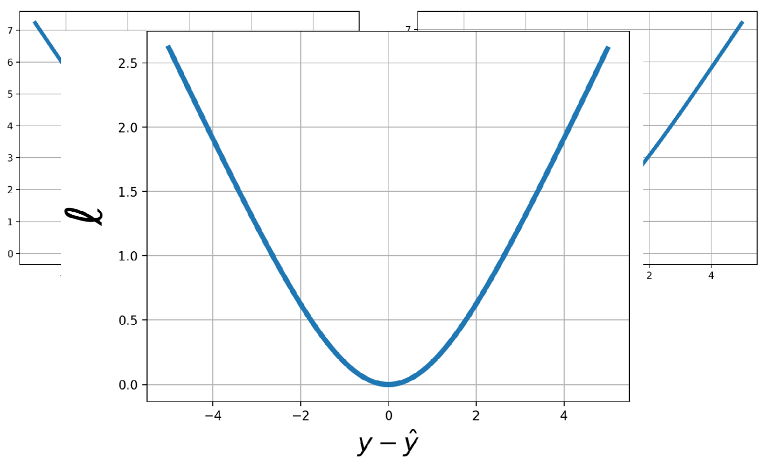


Symmetrization of Losses

- Given $f : \Re \to \Re$ bounded from below: $\exists c : \; f(z) > c$
- Symmetrization at 0:

$$\frac{1}{2}\big(f(z) + f(-z)\big) - f(0)$$

- Use $z = y - \hat{y}$
- Exp-Loss: $f(z) = e^z$
- Log-Loss: $f(z) = \log\big(1 + e^z\big)$



Training Loss

Average (why average?) loss over all examples

$$\frac{1}{n}\sum_{i=1}^{n} \ell\left(y_i, \hat{y}_i(\mathbf{w})\right)$$

For squared loss we can write

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{1}{n}\|\mathbf{y} - \hat{\mathbf{y}}\|^2$$