Regression overview, least-squares and logistic regression

Classification: given pt. x, predict class

Regression: given pt. x, predict numerical value

- Choose regression fn $h(x; p)$ w/ parameters p (h = hypothesis)
- optimize a cost fn (based on lost fn)
  - ex. risk fn

fns:

① linear: $h(x; w, \alpha) = w^T x + \alpha$

② polynomial

③ logistic: $h(x; w, \alpha) = S(w^T x + \alpha)$; $S(\gamma) = \frac{1}{1+e^{-\gamma}}$ (logistic fn)
  - ↳ often used for estimating probabilities because its always between 0 and 1
  - ↳ natural way for LDA (can get w/o LDA "logistic regression")

loss fns: z is prediction $h(x)$; y is true label

Ⓐ $L(z, y) = (z-y)^2$ (squared error)
  - ↳ amplifies outliers

Ⓑ $L(z, y) = |z-y|$ (abs error)
  - ↳ hard to optimize

(c) $L(z, y) = -y \ln z - (1-y) \ln(1-z)$ (logistic loss)

↳ cross entropy; $y \in [0, 1]$, $z \in (0, 1)$

Cost fns:

(a) $J(h) = \boxed{\frac{1}{n}} \sum_{i=1}^{n} L(h(x_i), y_i)$ (mean loss)

↑ doesn't affect optimization

(b) $J(h) = \max_{i=1}^{n} L(h(x_i), y_i)$ (max loss)

↑ trustworthy data – no outliers to throw off

(c) $J(h) = \sum_{i=1}^{n} w_i \, L(h(x_i), y_i)$ (weighted sum)

↑ some data more important, so we weight more

(d) $J(h) = (a, b, c) + \lambda \|w\|_2^2$ → regularization term
→ l_2 penalized
→ dec. in multi sol cases

(e) " " " " $\lambda \|w\|_1$ → l_1 penalized
→ encourage sparsity

famous regression methods:

- ⇒ least squares linear regression: (1)-(A)-(a)

$$\frac{1}{n}\sum_{i=1}^{n}\|\vec{w}X_i+\alpha-y_i\|^2$$

- ⇒ " " but weighted: (1)-(A)-(c)

$$\sum_{i=1}^{n} w_i\|w^TX_i+\alpha-y_i\|^2$$

- ⇒ Ridge regression: (1)-(A)-(d)

- quadratic
- min w/ calc

- ⇒ LASSO: (1)-(A)-(e)] QP

- ⇒ logistic regression: (3)-(c)-(a)] convex; min w/ grad descent

- ⇒ least abs. deviations: (1)-(B)-(a)
- ⇒ Chebyshev criterion: (1)-(B)-(b)

] LP

least squares linear regression

$$\min_{w,\alpha}\frac{1}{n}\sum_{i=1}^{n}(\vec{w}X_i+\alpha-y_i)^2 \Rightarrow \underset{w,\alpha}{\operatorname{argmin}}\sum_{i=1}^{n}\|\vec{w}X_i+\alpha-y_i\|^2$$

X is nxd design matrix, y is n-vector of scalar labels

X_i^T is a point of dim d

$X_{:j}$ is a feature column

after fixing w dim trick:
X is nxd+1; w d+1 vector

$$[X_1\ X_2\ 1]\begin{bmatrix}w_1\\w_2\\\alpha\end{bmatrix} \rightarrow h(x)=w^TX$$

$\Rightarrow \arg\min_w \|Xw + y\|^2$ (new opt prob, same as before)

RSS(w), residual sum of squares

Solving: $w^T X^T X w + 2 w^T X^T y + y^T y$

$\nabla_w f = 2X^T Xw - 2X^T y = 0 \Rightarrow X^T X w = X^T y$

= normal equations

If $X^T X$ singular, problem underconstrained

If not, then $w^* = \underbrace{(X^T X)^{-1} X^T}_{\text{pseudo inverse } X^+} y$ linear transformation

$X^+ X = I$ (left inv.)

$\hat{y}_i = w^T X_i \Rightarrow \hat{y} = Xw = XX^+ y = Hy$ H := hat matrix

Advantages:
- easy to compute
- Unique, stable solution

disadvantages:
- sensitive to outliers
- $X^T X$ should be singular

Logistic Regression

- logistic regression fn, fits probs (0,1)
- usually used for classification
- discriminative model: jump to fitting, skip gaussians

$$\arg\min_w \sum_{i=1}^n l(X_i^T w, y_i) = -\sum_{i=1}^n [y_i \ln s(X_i^T w) + (1-y_i) \ln (1 - s(X_i^T w))]$$

$J(w)$ is convex! Solve by gradient descent

$$s'(\gamma) = \frac{e^{-\gamma}}{(1+e^{-\gamma})^2} = s(\gamma)(1 - s(\gamma))$$

$s_i = s(x_i^T w)$

$$\nabla_w J = -\sum \left(\frac{y_i}{s_i} \nabla s_i - \frac{1-y_i}{1-s_i} \nabla s_i \right)$$

$$= -\sum \left(\frac{y_i}{s_i} - \frac{1-y_i}{1-s_i} \right) s_i (1-s_i) x_i$$

$$= -\sum (y_i - s_i) x_i$$

$$= -X^T (y - s(Xw)) \qquad s(Xw) = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{bmatrix}$$

gradient descent rule:

$$w \leftarrow w + \varepsilon X^T (y - s(X^T w))$$

Stochastic:

$$w \leftarrow w + \varepsilon (y_i - s(x_i^T w)) x_i$$

Converges before we visit all pts for large n

start: $w = 0$ works well