CSC411
LEC 3 20 09 2017

KEY CONCEPTS:
Dec bound
Loss fxn
S.G.D.

CATEGORICAL y: $\vec{x} \in \mathbb{R}^d \rightarrow \{1..K\}$ or $\{0,1\}, \{-1,1\}$. (d = dimension $X_1 .. X_d$). ← feature vector.

Remember: $\hat{y} = f(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T\mathbf{x}$ (we still have a BIAS).

Binary decision:

$$\hat{y} = f(\mathbf{x}, \mathbf{w}) = \text{sign}(\mathbf{w}^T\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{w}^T\mathbf{x} \geq 0 \\ -1 & \text{if } \mathbf{w}^T\mathbf{x} < 0 \end{cases} \} \text{ threshold.}$$

X_1 X_2 (features).
1 1 ...

$$\mathbf{X} = \begin{bmatrix} \cdot \\ \\ \\ \\ \end{bmatrix}$$ - data pt, - data pt, ⋮

$\mathbf{w}^T\mathbf{x} = 0$ is HYPERPLANE.
ORTHOG TO $\mathbf{w}$.
$\mathbf{w}^T\mathbf{x} + w_0 = 0$ shifts by w_0 (bias).

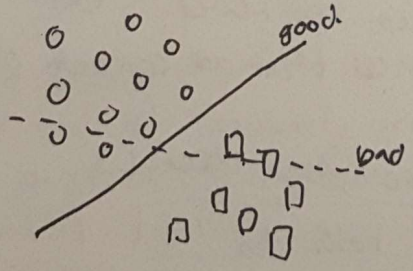
LINEARLY SEPARABLE: separate classes by hyperplane, ($d-1$ plane).

Some issues that make it not linsep:
- model too simple (polar not as polar rep)
- bad feats.
- noise

Should we have perfect separation in training data? Just b/c we have perfect sep. in training doesn't mean we have it in test.

Need to find $\mathbf{w}$ (direction), w_0 (location) of boundary.

What is a "good" boundary? → we need a LOSS FXN.



Two options:

- NATURAL ZERO-ONE LOSS

$$\ell_{0-1}(\hat{y}, y) = \begin{cases} 1 & \text{if } y \neq \hat{y} \\ 0 & \text{if } y = y \end{cases}$$

NP-H to optimize

But HARD TO MINIMIZE --
nonconvex, noncontinuous piecewise constant.

- ASYMMETRIC BINARY LOSS:
"two types of mistakes"

$$\ell_{ABL}(\hat{y}, y) = \begin{cases} \alpha & \text{if } y=0 \wedge \hat{y}=1 \\ \beta & \text{if } y=1 \wedge \hat{y}=0 \\ 0 & \text{if } y=\hat{y} \end{cases}$$

} if repercussions for 2 types of mistakes are different.

- USE A SURROGATE LOSS INSTEAD, $\tilde{\ell}$.

Good surrog. loss
- Easy to optimize
- Representative:
low s. loss means low orig. loss

$\ell_2(y, \hat{y}) = (y - \hat{y})^2$ is easy to optimize, but meh on representation.
(very harsh on inliers).

We use METRICS to evaluate loss functions

- METRICS:
- ACCURACY: % of correct predictions
noninformative when considering RARE OUTCOMES.

TP = true pos
FN = False neg
FP = False pos

Recall: relevant instances retrieved (R)
Precision: retrival instances that are correct. (P)

F_1 SCORE: HARMONIC MEAN OF PRECISION AND RECALL

$$F_1 = 2\frac{PR}{P+R}$$

} need both to be "good" to work.

PRECISION-RECALL CURVE.
tradeoff btwn precision, recall via decision threshold.

OPTIMIZATION.

How do we find $\mathbf{w}$ once we have a loss fcn ℓ?
We use GRADIENT DESCENT not b/c it is THE BEST, but b/c it is EASY TO SCALE. It is best for SOLVING BG PROBLEMS.

$$\mathbf{w} = \frac{1}{N}\sum_{i=1}^{N} \ell(y_i, f(x_i, \mathbf{w})).$$

GRADIENT DESCENT.
- Initialize w_0.
- Compute gradient (tangent dir).
- Take a step.
- Keep updating

$$w_{t+1} = w_t - \underbrace{\lambda_t}_{\text{LEARNING RATE}} \nabla_w L(w_t).$$

λ:
LARGE λ → overshoot
SMALL λ → long time.
line search (searching on along grad dir line) too slow.
standard: decay λ as learning progresses

THERE IS A LOT OF THEORY ON CONVEX OPTIMIZATION. (ask Fetaya for the literature).

Comp. cost of computing $\nabla_w L(w_t)$?
$L(w) = \frac{1}{N}\sum_{i=1}^{N} \ell(y_i, f(x_i, w))$ = grows LINEARLY in N.
HUGE DATASET → large cost for small update

SOLN TO ABOVE:
Pick one pt randomly instead of gradient over all.
Find gradient at that
THIS IS STOCHASTIC GRADIENT DESCENT

Pick datum j
Compute $\hat{g}_t = \nabla \ell(y_j, f(x_j, w))$.

Theoretically, works, but practically VERY NOISY.
INSTEAD OF ONE PT (S.G.D) vs ALL PTS (G.D).
pick a "small subset": MINIBATCH.
TRADEOFF: big batch accuracy $\propto$ small batch runtime.

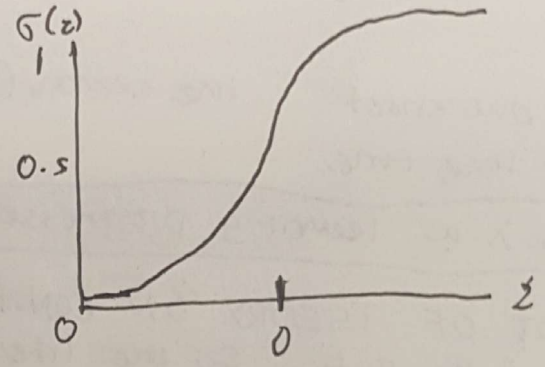
Instead of mapping X to $\{0, 1\}$,
we now map to $[0, 1]$ and we find $P(y=1|x)$.
(STILL CLASSIFICATION)

Need to SQUASH $W^T x$ into $[0,1]$ ≈ $P(y=1|x)$.
What about $P(y=-1|x)$? (prob of other class).
$$P(y=-1|x) = 1 - P(y=1|x).$$

HOW TO CHOOSE THE LABEL?
Pick the MOST PROBABLE.

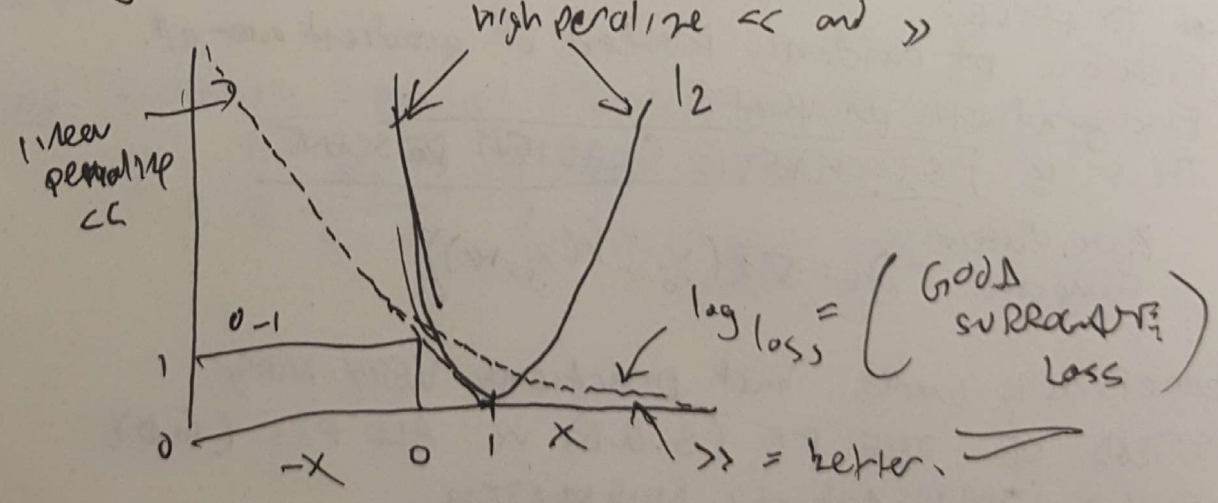
BENEFITS:
- MODELS UNCERTAINTY (in LIMITED WAY)
- Can use Pr for decision making
- Can use Pr for prob/stat opt. methods.

SQUASHING FXN:
$$\sigma(z) = \frac{1}{1+e^{-z}}$$



Modifying w and w_0 changes the shape!
w = "flatness/width" → SLOPE / sharpness of 0→1 convergence.
w_0 = "location".

High w ≈ step fxn.
Low w ≈ gentle increase.



logistic regression: optimize surrogate loss.