L26: Kernels + SVMs

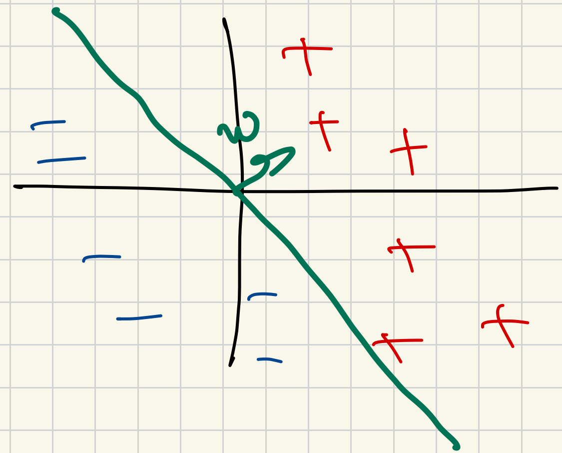
Apr 15, 2026

FoDA

Jeff M. Phillips

Input $X, y \in \mathbb{R}^d \times \{-1, +1\}$

(x_i, y_i)



Output $g(x) = \langle w, x \rangle$

predict $\text{sign}(g(x))$

Perceptron

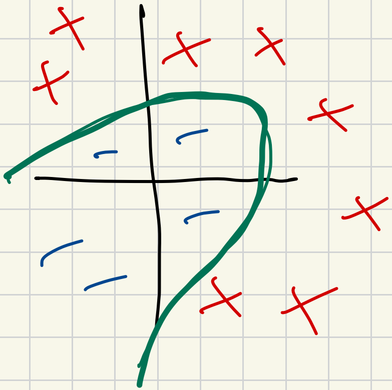
$w = y_i x_i$ for any $(x_i, y_i) \in (X, y)$

repeat

if any $(x_i y_i)$ so $\boxed{y_i \boxed{\langle X_i, w \rangle} < 0}$ misclassified

then update $w = w + y_i X_i$ linear

return $w / \|w\|$

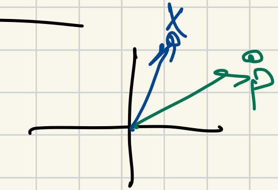


or

$$y_i = \begin{cases} +1 & \text{if student gets B} \\ -1 & \text{if student gets A, C, D, E} \end{cases}$$

Generalized Inner Product → Kernels

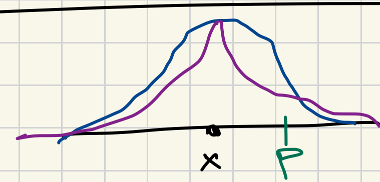
$p, x \in \mathbb{R}^d$

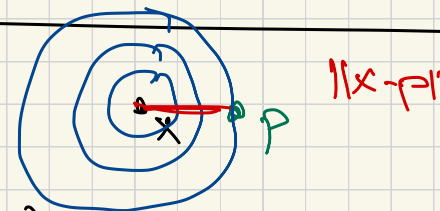


linear inner products
dot product

$$\boxed{\langle p, x \rangle} = \sum_{j=1}^{d} p_j x_j$$

Kernels





RBF

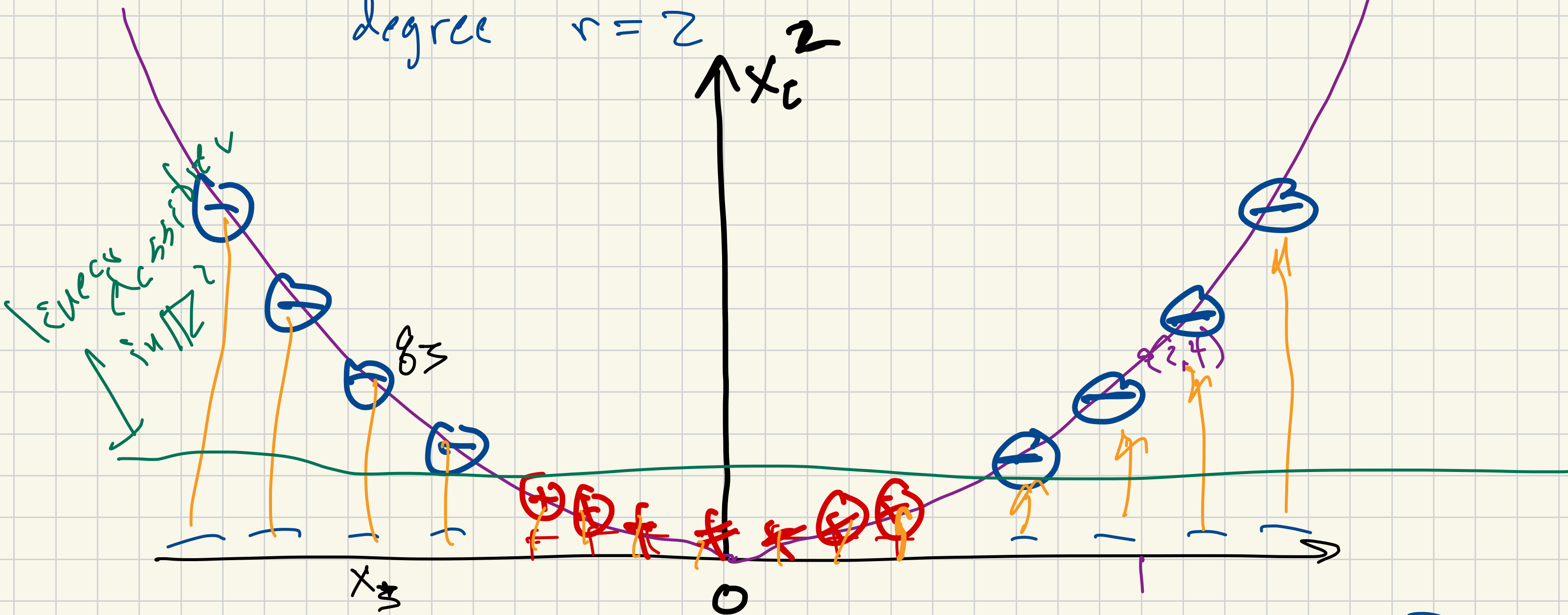
$\langle p, x \rangle_G = K(p,x) = \exp\left(-\frac{\|x-p\|^2}{\sigma^2}\right)$ Gaussian

$\langle p, x \rangle_L = K(p,x) = \exp\left(-\frac{\|x-p\|}{\sigma}\right)$ Laplacian

$\langle p, x \rangle_P = K(p,x) = \left(\langle p, x \rangle + c\right)^r$ polynomial

Polynomial classifier $x \in \mathbb{R}$

degree $r = 2$



$x_i \in \mathbb{R}$

$$x_i \to q_i = (x_i, x_i^2) \in \mathbb{R}^2$$

Option #1 Polynomial Expansion

linearize polynomial

example $p_i = (p_1, p_2) \in \mathbb{R}^2$ degree 2 expansion

$$q_i = (1, p_1, p_2, p_1^2, p_2^2, p_1 \cdot p_2) \in \mathbb{R}^6$$

new degree $\min(O(d^r, r^d))$

Can't expand for Gaussian kernel $e^{-\|x-p\|^2}$

↳ infinite # dimensions (if n pts → n-dimension

↳ approx w/ $\frac{1}{\varepsilon^2}$-dim error $\lesssim \varepsilon$

Option #2 Mistake counter → $\boxed{\text{Kernel Perceptron}}$

→ Kernel SVM

change classifier : Mistake Counter

$$w = w + x_i y_i = \sum_{i=1}^{n} \alpha_i y_i x_i$$

mistake counters

α_i = # times used mistake on (x_i, y_i)

sparse

modify α, except for $\left(\frac{1}{\gamma^*}\right)^2$ mistakes

$$\alpha = (\alpha_1, \alpha_2 \dots \alpha_n) \in \mathbb{R}^n$$

predictor function

$$g(p) = \langle w, p \rangle = \left\langle \sum_{i=1}^{n} \alpha_i y_i x_i , p \right\rangle$$

$$= \sum_{i=1}^{n} \alpha_i y_i \boxed{\langle x_i , p \rangle}$$

$$= \sum_{i=1}^{n} \alpha_i y_i K(x_i, p)$$

(nonlinear?) Kernel

Kernel Perception

0. Init: $\alpha = (0, 0, \dots 0, 0) \in \mathbb{R}^n$

choose some $\alpha_i = 1$ $\quad i = [1, \dots n]$

if Gaussian kernel
always separable?
↳ always works!

1. <u>repeat</u>
   if some $(x_i, y_i) \in (X, y)$
   $$\sum_{j=1}^{n} \alpha_j y_j \boxed{\langle x_j, x_i \rangle} < 0$$
   Replace → $K(x_j, x_i)$

   then $\alpha_i = \alpha_i + 1$

2. Return $g(\cdot) = \sum_{j=1}^{n} \alpha_j y_j \boxed{\langle x_j, \cdot \rangle}$

Which version to use?

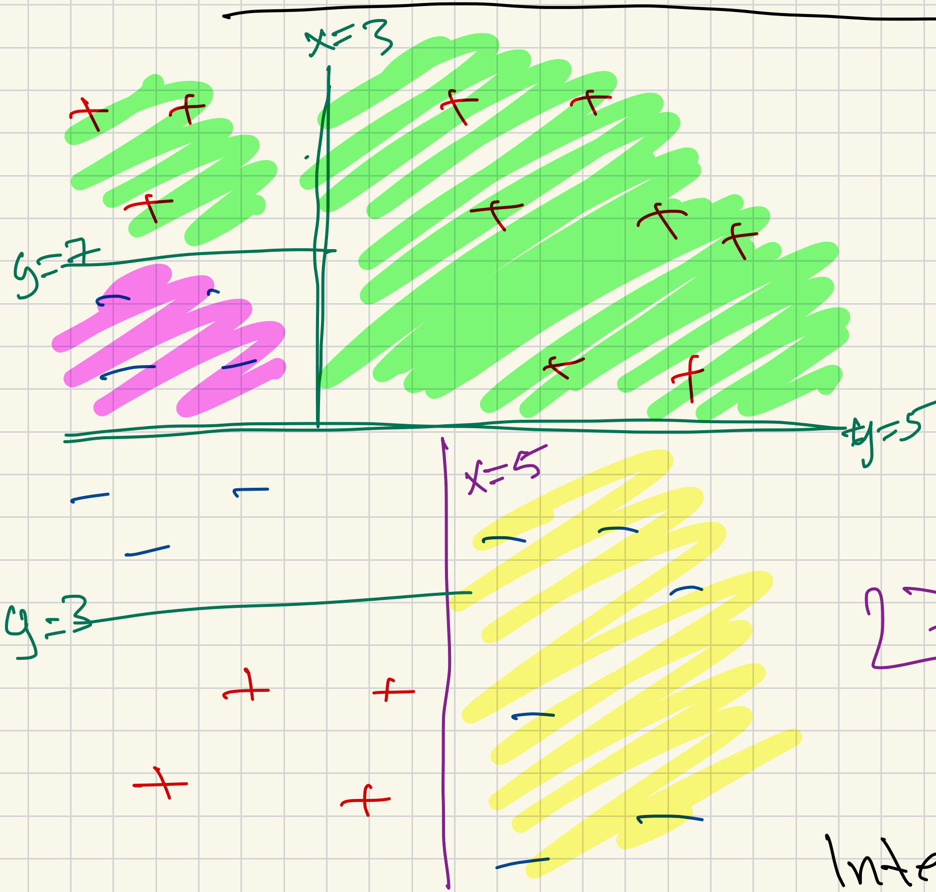
Input (x, y) $\xrightarrow{\text{split}}$ train $\Rightarrow$ built diff models

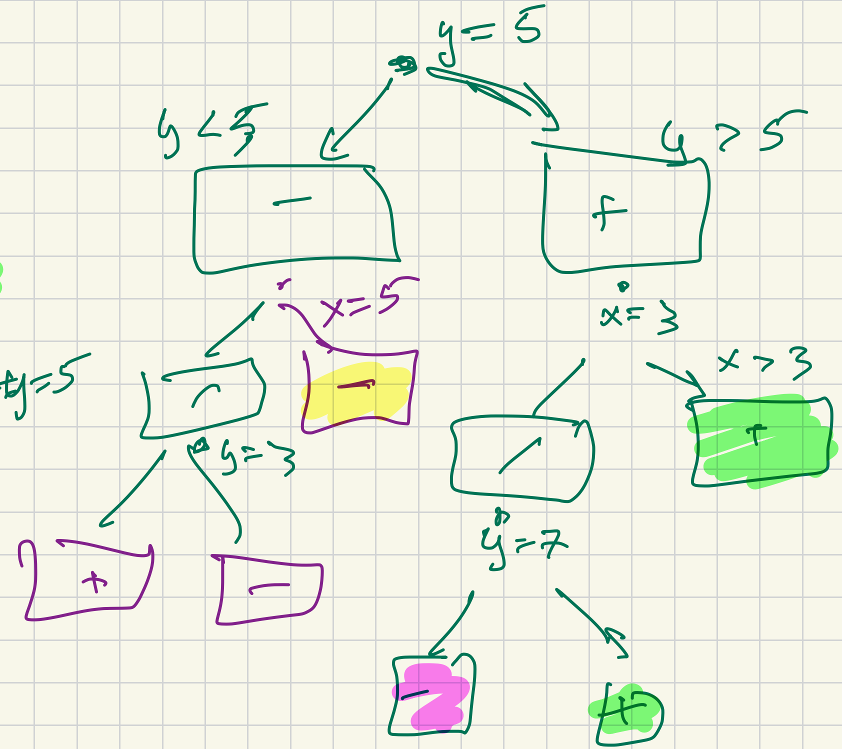
test $\Rightarrow$ evaluate models

cross-validation

count mistakes

Decision Trees $X \subset \mathbb{R}^{d=2}$



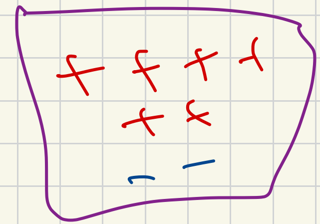


Interpretable?

How to choose split?

- Usually try many all splits, see which is best.

  $O(nd)$ time.



Gini Impurity Index

$$P_+ = \frac{\#(+\text{ pts})}{\text{total } \# \text{ pts}} = \text{e.g. } \frac{6}{8}$$

$$P_- = \frac{\#(-\text{ pts})}{\text{total pts}} = \frac{2}{8}$$

$$G = P_+(1-P_+) + (P_-(1-P_-))$$

Random Forests $\Rightarrow$ XGBoost

1. For s $(=100)$ steps
   subsample data $(X^j, y^j) \subset (X, y)$ 75%

2. For each j, build Decision Tree
   T_j on (X^j, y^j)

3. Build linear model

$$g(p) = \sum_{j=1}^{s} \alpha_j T_j(p)$$