Supervised learning: linear regression and SVM

Shikui Tu

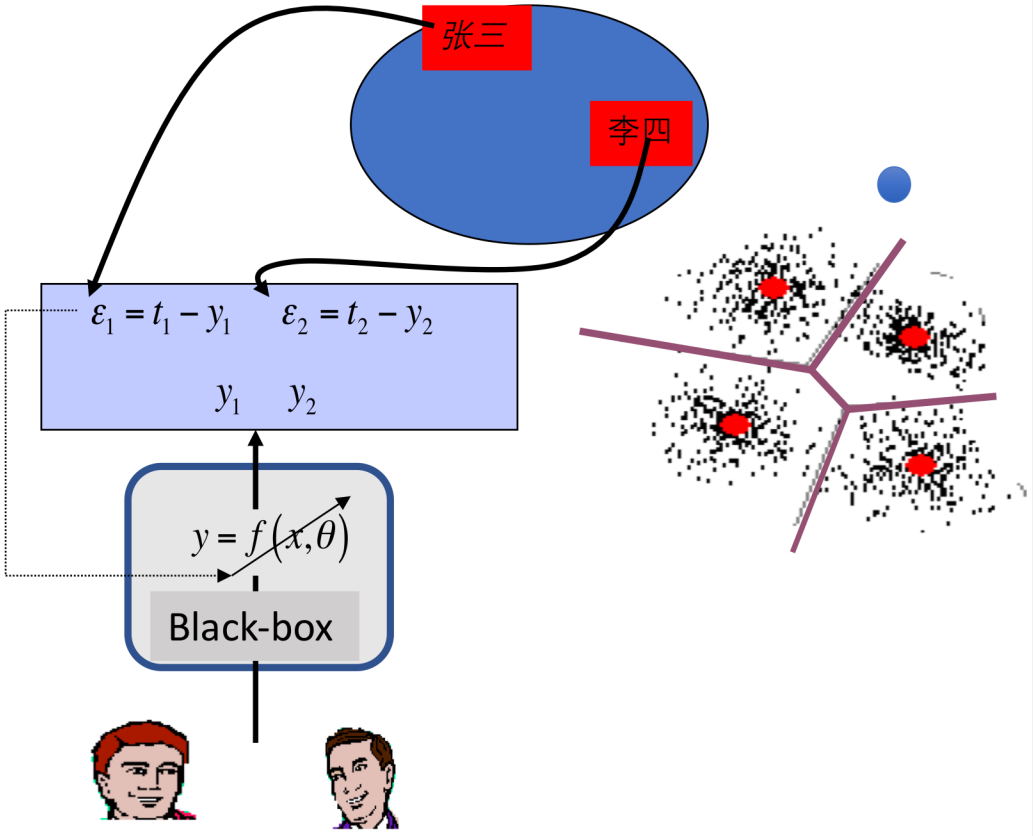
Department of Computer Science and Engineering, Shanghai Jiao Tong University

2018-04-26

Outline

- **Supervised learning**
- Linear regression
- Logistic regression
- Support Vector Machine (SVM)

Supervised learning



Models and algorithms

- linear regression
- logistic regression
- Support Vector Machines
- naive Bayes
- linear discriminant analysis
- decision trees
- k-nearest neighbor algorithm
- Neural Networks (Multilayer perceptron)
- …

Outline

- Supervised learning

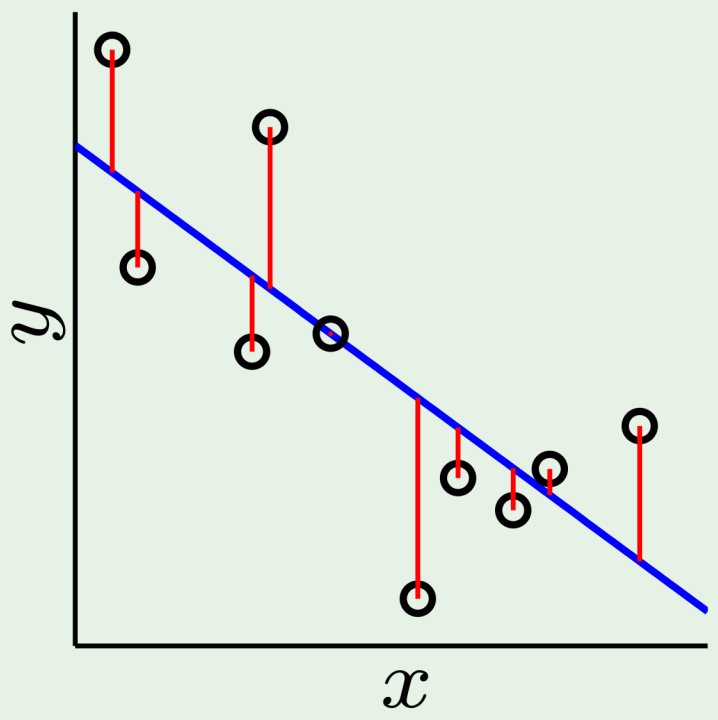
- **Linear regression**

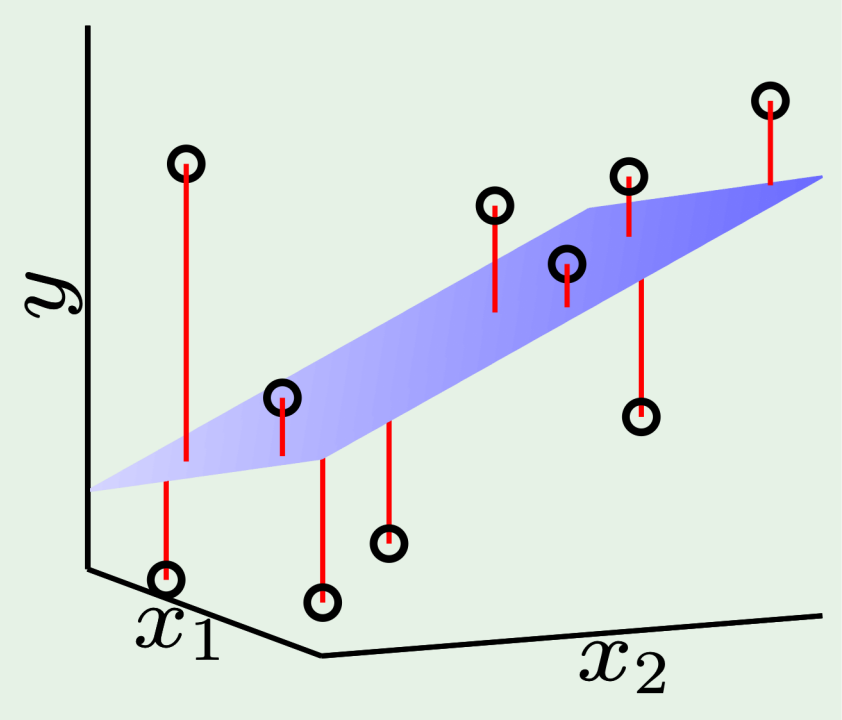
- Logistic regression

- Support Vector Machine (SVM)

Linear regression

"Regression" usually means "real-value output".

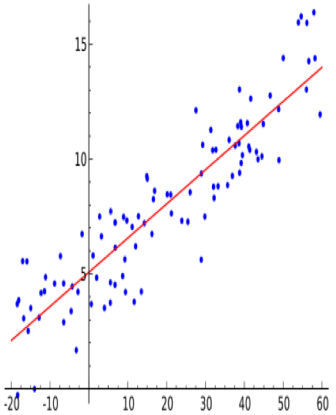




Basic concepts of linear regression

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

the *regressand*, *endogenous variable*, *response variable*, *measured variable*, *criterion variable*, or *dependent variable*



$$y_i = \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta} + \varepsilon_i, \qquad i = 1, \ldots, n,$$

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} \mathbf{x}_1^{\mathrm{T}} \\ \mathbf{x}_2^{\mathrm{T}} \\ \vdots \\ \mathbf{x}_n^{\mathrm{T}} \end{pmatrix} = \begin{pmatrix} x_{11} & \cdots & x_{1p} \\ x_{21} & \cdots & x_{2p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{pmatrix}, \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix}, \quad \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}.$$

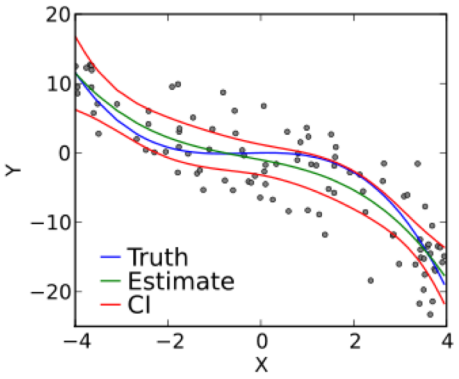
regressors, *exogenous variables*, *explanatory variables*, *covariates*, *input variables*, *predictor variables*, or *independent variables*

The matrix ***X*** is sometimes called the design matrix.

$$h_i = \beta_1 t_i + \beta_2 t_i^2 + \varepsilon_i,$$

$\boldsymbol{x}_i = (x_{i1}, x_{i2}) = (t_i, t_i^2)$

$$h_i = \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta} + \varepsilon_i.$$



Least Square Estimation

Matrix form: $Y = X\beta + \epsilon$

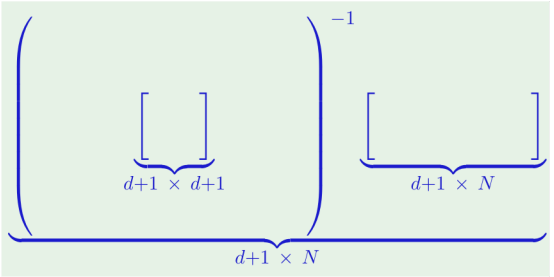
Minimize the sum of squared residuals (errors):

$$\min_{\beta} \varepsilon^T \varepsilon = (Y - X\beta)^T (Y - X\beta) \quad \Leftrightarrow \quad \min_{W} \sum_{t} \left\| e_t \left(x_t, w \right) \right\|^2$$

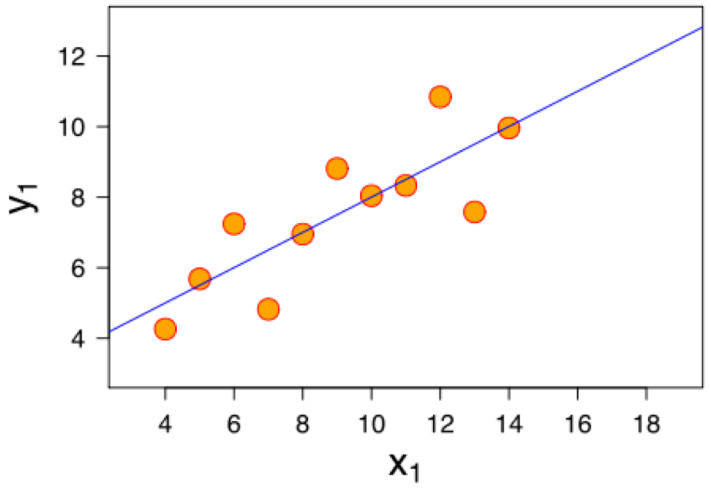
By taking the derivative to zero, we have:

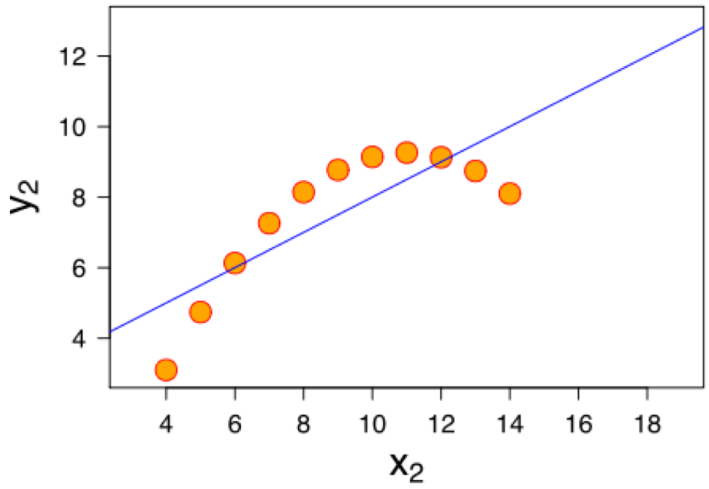
$$\beta = (X^T X)^{-1} X^T Y$$

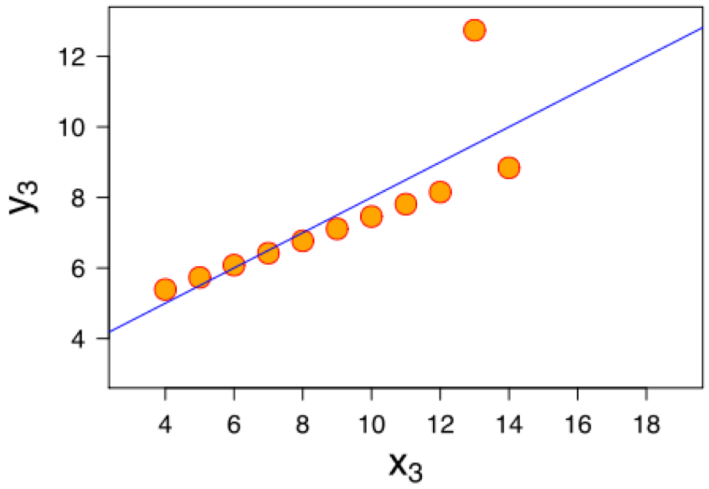
The pseudo-inverse $X^+ = (X^T X)^{-1} X^T$

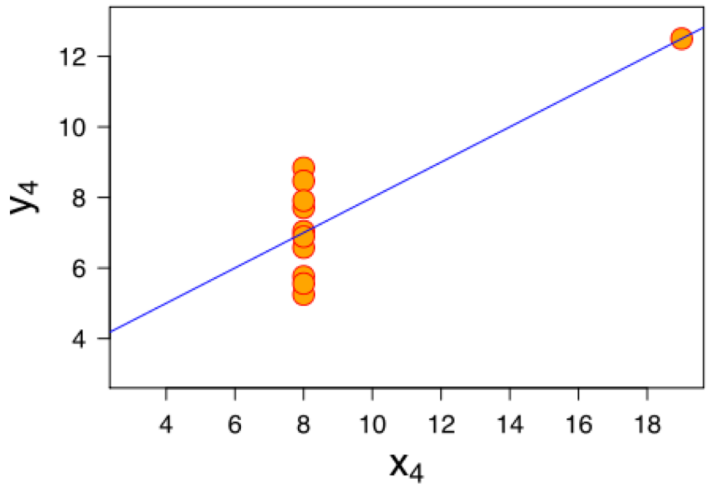


Limitations of linear regression





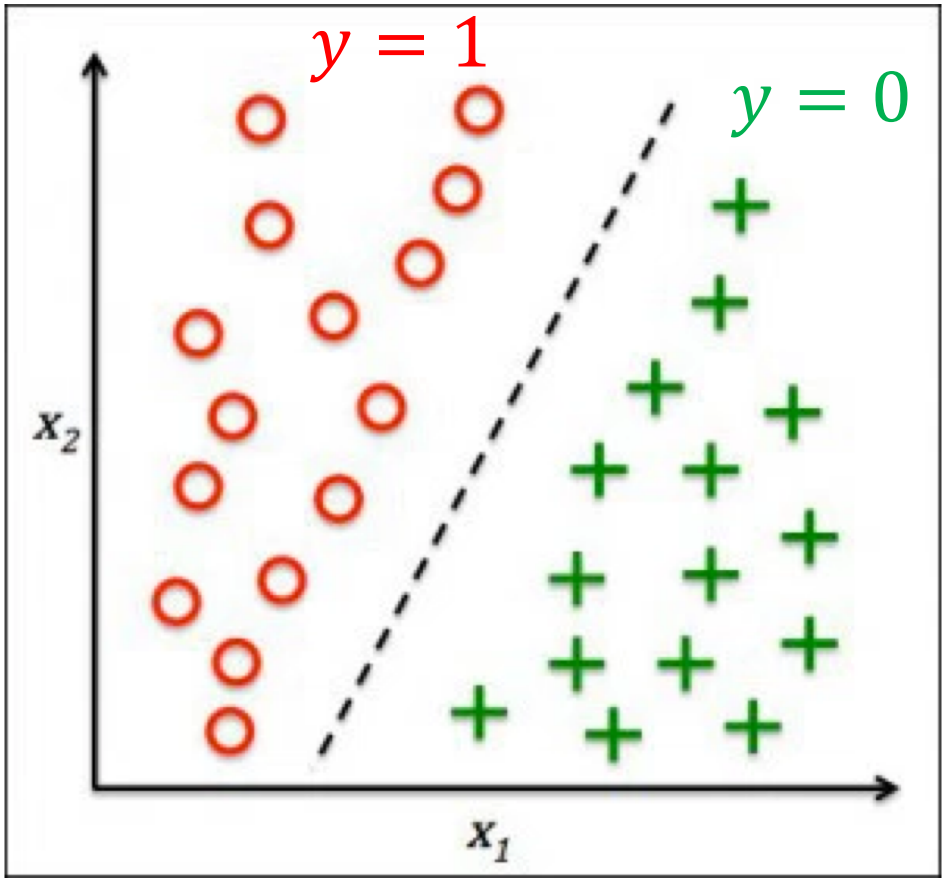




Outline

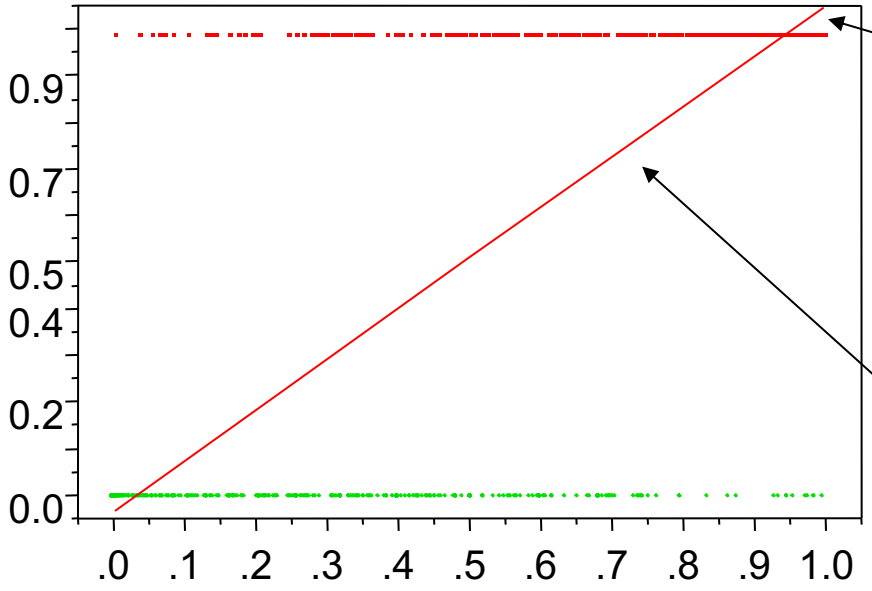
- Supervised learning
- Linear regression
- **Logistic regression**
- Support Vector Machine (SVM)

Classification



Why not Linear Regression?

- For classification, Y only takes on values of 0 and 1



How do we interpret values greater than 1 and smaller than 1?

How do we interpret values of Y between 0 and 1?

Problems

- The regression line $\beta_0+\beta_1 X$ can take on any value between negative infinity and positive infinity.

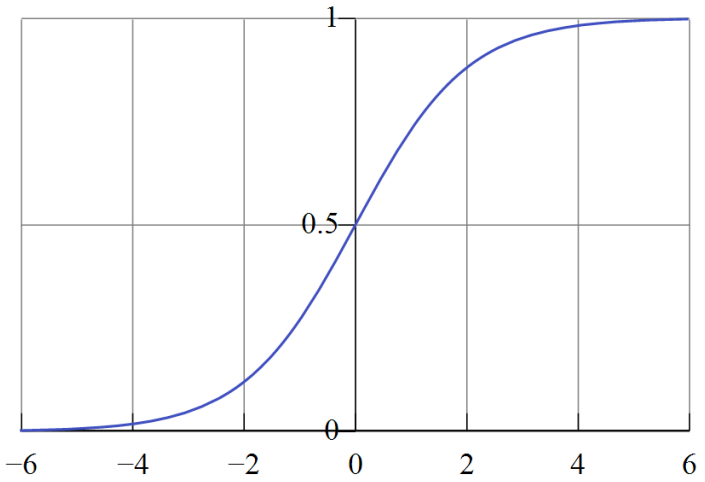
- In the classification problem, output Y can only take on two possible values: 0 or 1.

- Therefore the regression line almost always predicts the wrong value for output Y in classification problems

Solution: Use Logistic Function

- Instead of trying to predict Y, let's try to predict P(Y = 1)
- Model P(Y = 1) using a function that gives outputs between 0 and 1 → logistic function!

$$Sigmoid(x) = Logistic(x) = \frac{1}{1 + e^{-x}}$$



Wikipedia

Logistic Regression

$$p(x) = P(Y = 1 \mid x) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}} = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}}$$

- Euler's number e = 2.71828…
- no matter what values 0, 1 or X take, p(X) will have values between 0 and 1

$$\log\left(\frac{p(x)}{1 - p(x)}\right) = \beta_0 + \beta_1 X$$

Log-odds or logit transformation of p(X) gives a linear model.

Interpreting β_1

- Not very easy with logistic regression →we are predicting P(Y), not Y directly
  - If β_1 =0 → no relationship between Y and X
  - If β_1 > 0 → when X gets larger so does the probability that Y = 1
  - If β_1 < 0 → when X gets larger, the probability that Y = 1 gets smaller
  - How much bigger or smaller depends on where we are on the slope

Logistic Regression with Multiple Variables

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}.$$

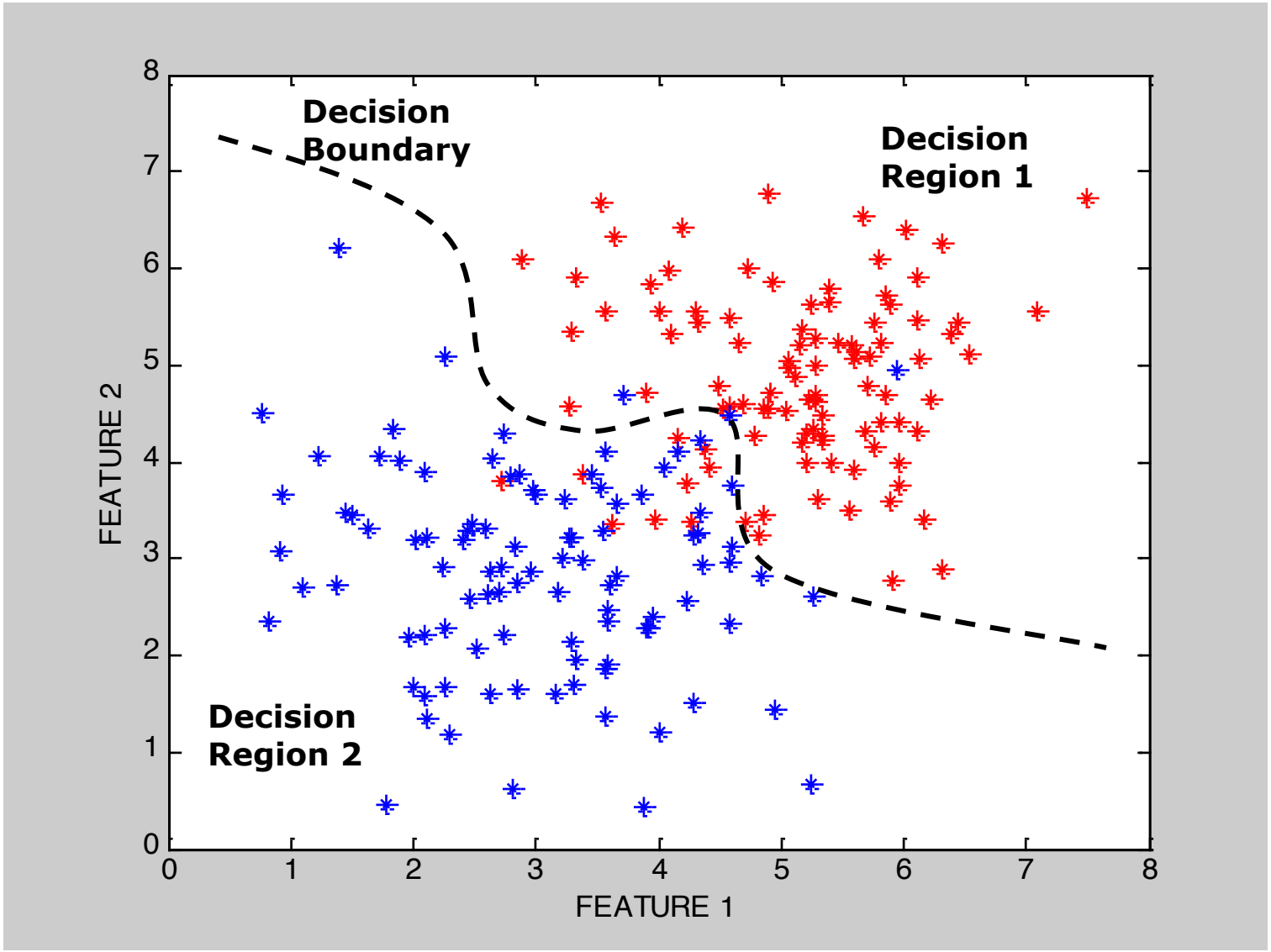
Logistic regression with more than two classes

- Logistic regression easily generalized to more than two classes

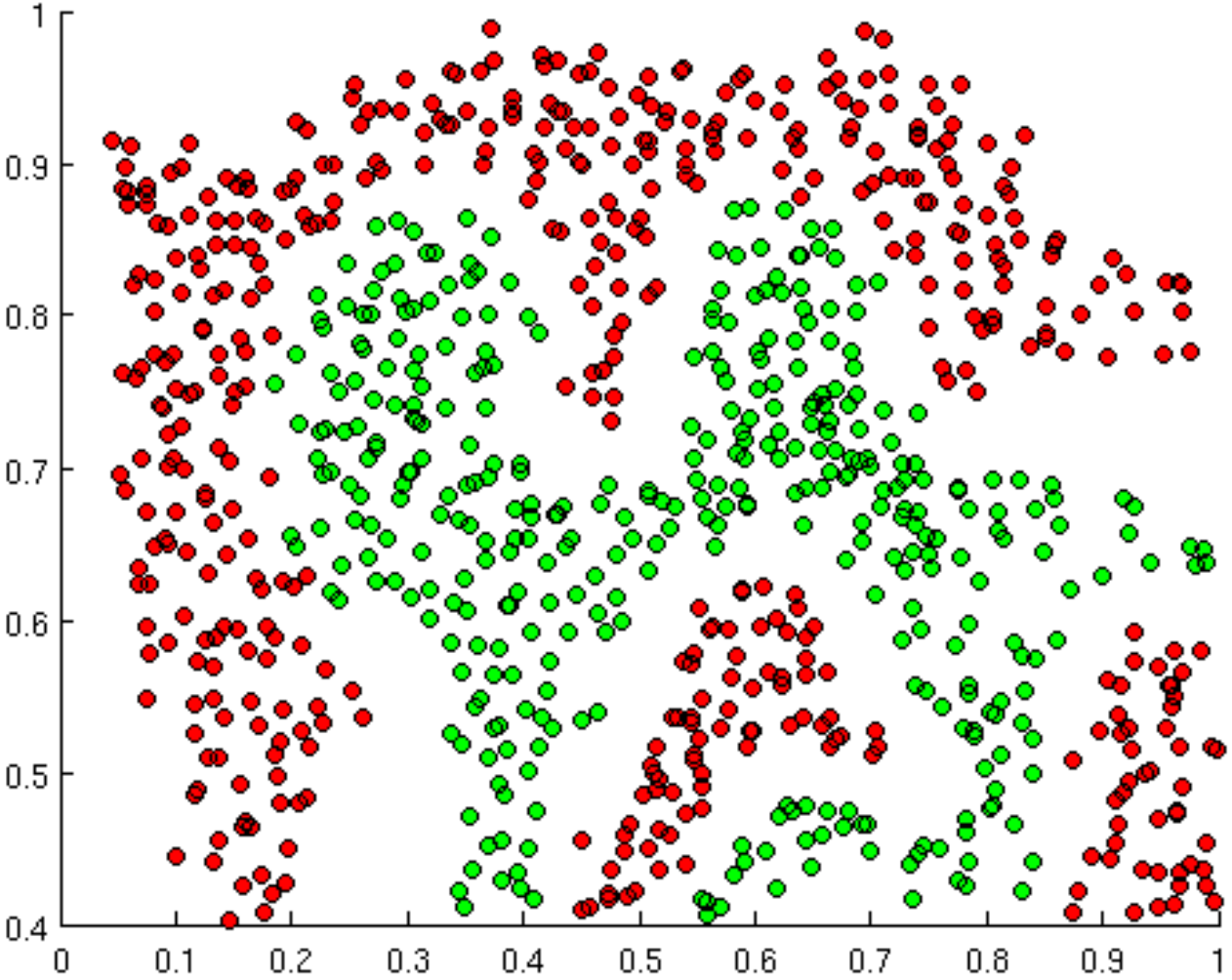
$$P(Y = k \mid \mathrm{X}) = \frac{e^{\beta_{0k} + \beta_{1k} X + \ldots + \beta_{pk} X}}{1 + e^{\beta_{0k} + \beta_{1k} X + \ldots + \beta_{pk} X}}$$

- Linear function for each class
- Multiclass logistic regression is also referred to as *multinomial regression*

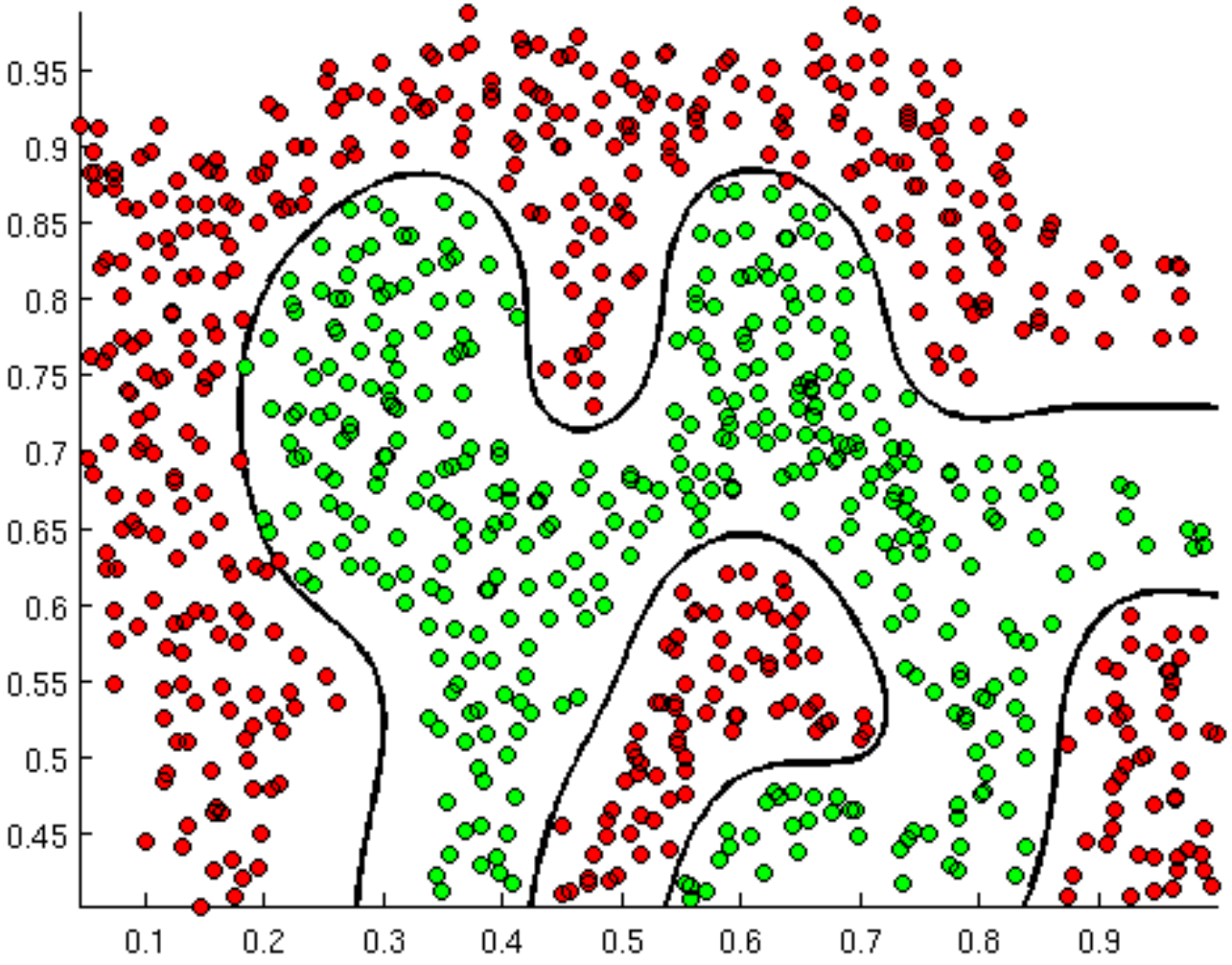
Decision Boundaries



Non-linear Decision Boundaries

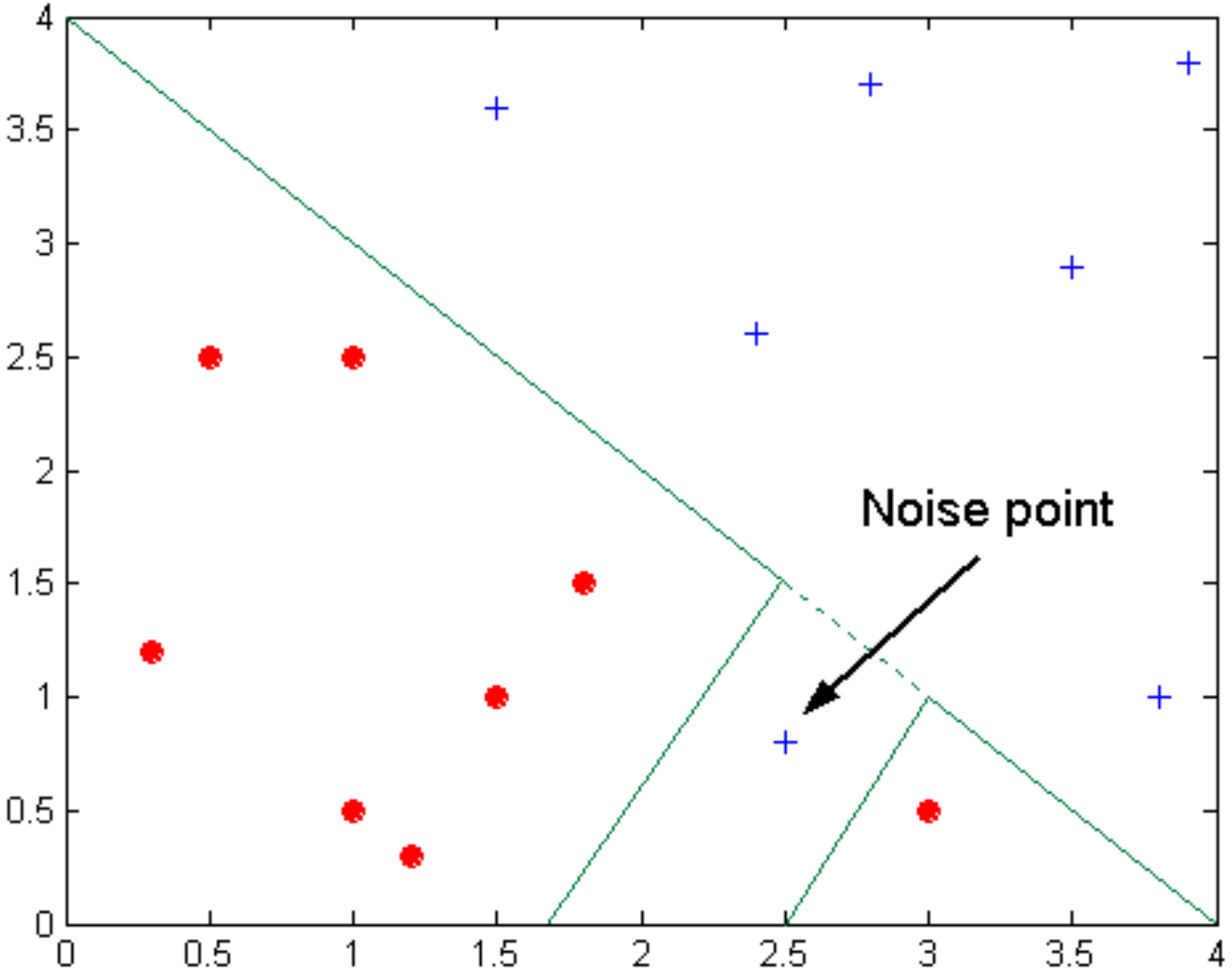


Non-linear Decision Boundaries



Stanford OpenClassroom Machine Learning

Overfitting Decision Boundary



Logistic Regression Regularization

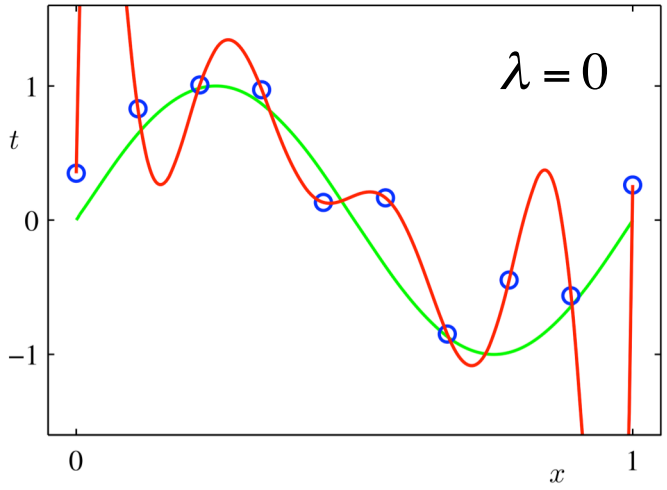
- Regularization (shrinking) is a method that can be used to prevent overfitting
- It works by adding a term to the cost function that penalizes for extreme parameter values

Regularization parameter

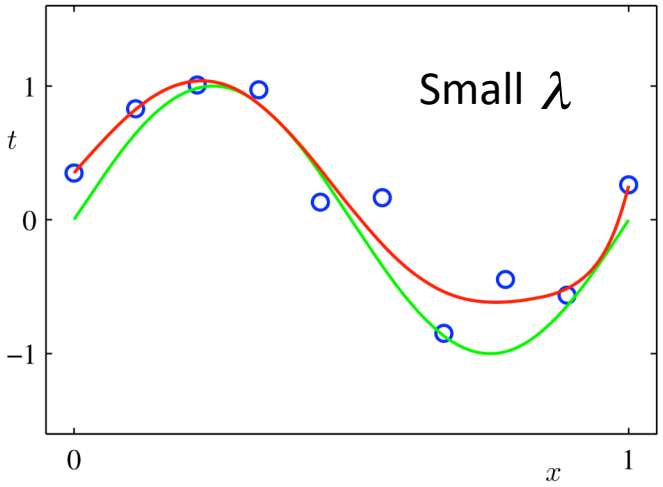
$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda\sum_{j=1}^{n}\theta_j^2\right]$$

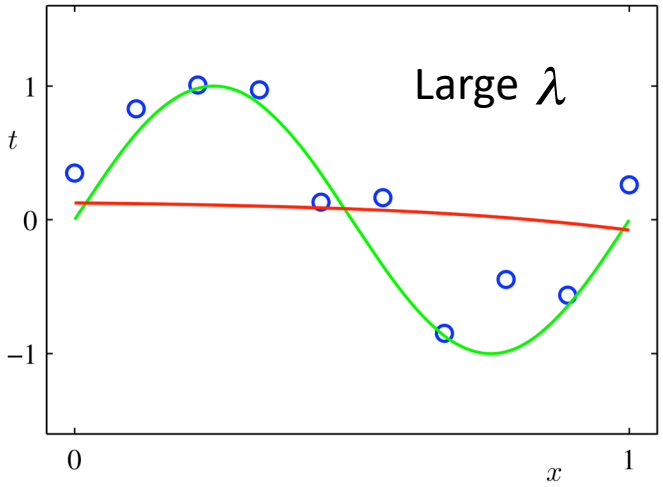
squared L2-norm
(≈magnitude of θ)

The effect of regularization



$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda\sum_{j=1}^{n}\theta_j^2\right]$$





Logistic Regression Regularization

- Several types
  - L1 (LASSO) (L1-norm)
    - somewhat indifferent to very correlated predictors
      - will tend to pick one and ignore the rest
    - expects many coefficients to be close to zero, and a small subset to be larger and nonzero
    - tends to sparse models → coefficients for irrelevant features set to 0
  - L2 (Ridge) (squared L2-norm)
    - tends to shrink the coefficients of correlated predictors toward each other → allows coefficients to borrow strength from each other
    - extreme case of k identical predictors, they each get identical coefficients with $1/k^{th}$ the size that any single one would get if fit alone

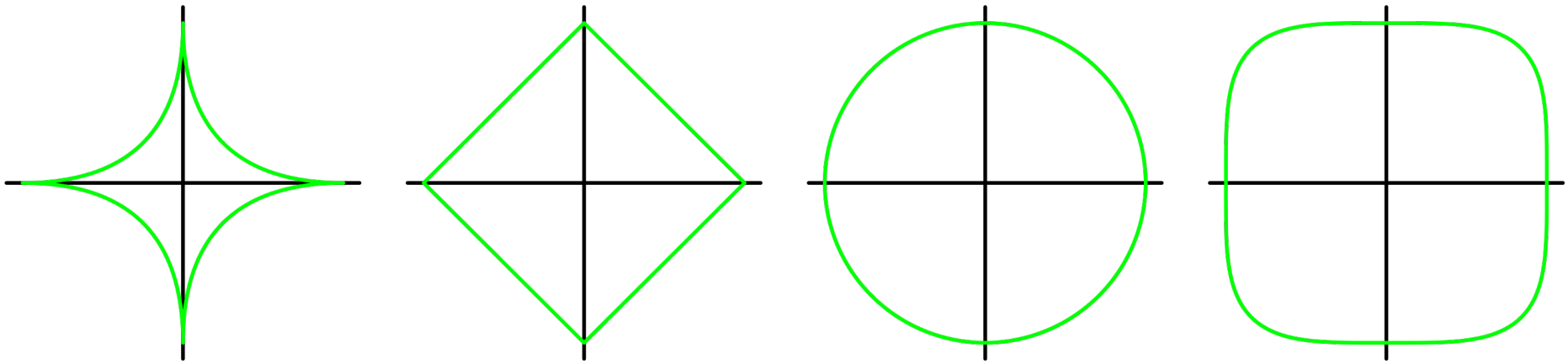
Friedman, J., Hastie, T., & Tibshirani, R. (2010)

Logistic Regression Regularization

- *Elastic net* regularization uses a linear combination of L1 and L2 norms
  - two regularization parameters, λ_1 and λ_2
  - performs much like the lasso
  - removes any degeneracies and wild behavior caused by extreme correlations.
  - creates a useful compromise between ridge and lasso

Friedman, J., Hastie, T., & Tibshirani, R. (2010)

Contours of regularization term $|\theta|^q$

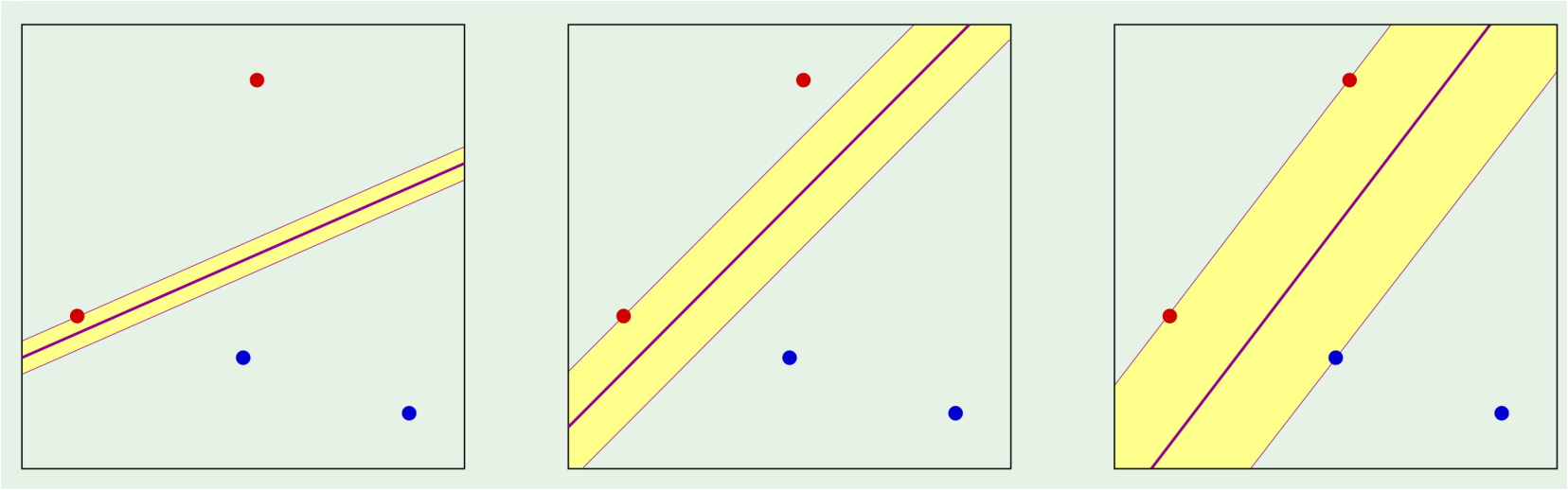


$q = 0.5$ $q = 1$ $q = 2$ $q = 4$

Outline

- Supervised learning
- Linear regression
- Logistic regression
- Support Vector Machine (SVM)

Which is a better linear separation



Two questions:

1. Why is bigger margin better?

2. Which $\mathbf{w}$ maximizes the margin?

Finding w with large margin

Let $\mathbf{x}_n$ be the nearest data point to the plane $\mathbf{w}^\intercal \mathbf{x} = 0$. How far is it?

2 preliminary technicalities:

1. **Normalize $\mathbf{w}$:**

   $$|\mathbf{w}^\intercal \mathbf{x}_n| = 1$$

2. **Pull out w_0:**

   $\mathbf{w} = (w_1, \cdots, w_d)$ apart from b

   The plane is now $\boxed{\mathbf{w}^\intercal \mathbf{x} + b = 0}$ (no x_0)

Computing the distance

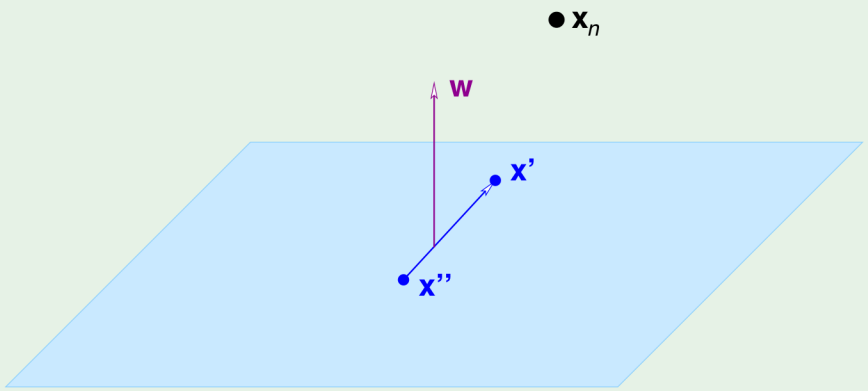
The distance between $\mathbf{x}_n$ and the plane $\mathbf{w}^\mathsf{T}\mathbf{x} + b = 0$ where $|\mathbf{w}^\mathsf{T}\mathbf{x}_n + b| = 1$

The vector $\mathbf{w}$ is $\perp$ to the plane in the $\mathcal{X}$ space:

Take $\mathbf{x}'$ and $\mathbf{x}''$ on the plane

$$\mathbf{w}^\mathsf{T}\mathbf{x}' + b = 0 \quad \text{and} \quad \mathbf{w}^\mathsf{T}\mathbf{x}'' + b = 0$$

$$\implies \mathbf{w}^\mathsf{T}(\mathbf{x}' - \mathbf{x}'') = 0$$

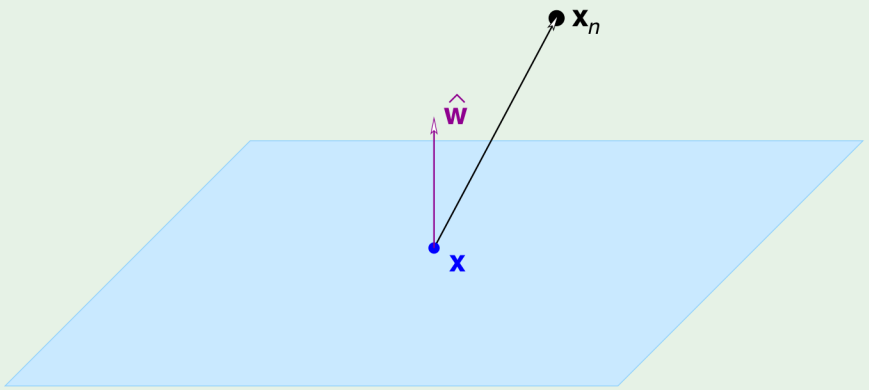


The distance from x_n to the plane

Distance between $\mathbf{x}_n$ and the plane:

Take any point $\mathbf{x}$ on the plane

Projection of $\mathbf{x}_n - \mathbf{x}$ on $\mathbf{w}$



$$\hat{\mathbf{w}} = \frac{\mathbf{w}}{\|\mathbf{w}\|} \implies \text{distance} = \left|\hat{\mathbf{w}}^\intercal(\mathbf{x}_n - \mathbf{x})\right|$$

$$\text{distance} = \frac{1}{\|\mathbf{w}\|}\left|\mathbf{w}^\intercal\mathbf{x}_n - \mathbf{w}^\intercal\mathbf{x}\right| = \frac{1}{\|\mathbf{w}\|}\left|\mathbf{w}^\intercal\mathbf{x}_n + b - \mathbf{w}^\intercal\mathbf{x} - b\right| = \frac{1}{\|\mathbf{w}\|}$$

The SVM optimization problem

Maximize $\dfrac{1}{\|\mathbf{w}\|}$

subject to $\min_{n=1,2,\ldots,N} |\mathbf{w}^{\mathsf{T}}\mathbf{x}_n + b| = 1$

Notice: $|\mathbf{w}^{\mathsf{T}}\mathbf{x}_n + b| = y_n\left(\mathbf{w}^{\mathsf{T}}\mathbf{x}_n + b\right)$

Minimize $\dfrac{1}{2}\,\mathbf{w}^{\mathsf{T}}\mathbf{w}$

subject to $y_n\left(\mathbf{w}^{\mathsf{T}}\mathbf{x}_n + b\right) \geq 1$ for $n = 1, 2, \ldots, N$

Constrained optimization

Minimize $\frac{1}{2}\,\mathbf{w}^{\mathsf{T}}\mathbf{w}$

subject to $y_n\left(\mathbf{w}^{\mathsf{T}}\mathbf{x}_n + b\right) \geq 1$ for $n = 1, 2, \ldots, N$

$\mathbf{w} \in \mathbb{R}^d,\ b \in \mathbb{R}$

Lagrange? inequality constraints $\Longrightarrow$ KKT

Lagrange formulation

Minimize $\mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\mathbf{w}^{\mathsf{T}}\mathbf{w} - \sum_{n=1}^{N} \alpha_n (y_n (\mathbf{w}^{\mathsf{T}}\mathbf{x}_n + b) - 1)$

w.r.t. $\mathbf{w}$ and b and maximize w.r.t. each $\alpha_n \geq 0$

$$\nabla_{\mathbf{w}}\mathcal{L} = \mathbf{w} - \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n = \mathbf{0}$$

$$\frac{\partial \mathcal{L}}{\partial b} = - \sum_{n=1}^{N} \alpha_n y_n = 0$$

Dual formulation

$$\mathbf{w} = \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n \qquad \text{and} \qquad \sum_{n=1}^{N} \alpha_n y_n = 0$$

in the Lagrangian $$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\mathbf{w}^\intercal\mathbf{w} - \sum_{n=1}^{N} \alpha_n \left(y_n \left(\mathbf{w}^\intercal\mathbf{x}_n + b\right) - 1\right)$$

we get $$\mathcal{L}(\boldsymbol{\alpha}) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N}\sum_{m=1}^{N} y_n y_m \, \alpha_n \alpha_m \, \mathbf{x}_n^\intercal \mathbf{x}_m$$

Maximize w.r.t. to $\boldsymbol{\alpha}$ subject to $\alpha_n \geq 0$ for $n = 1, \cdots, N$ **and** $\sum_{n=1}^{N} \alpha_n y_n = 0$

Quadratic programming (QP)

$$\min_{\boldsymbol{\alpha}} \quad \frac{1}{2}\,\boldsymbol{\alpha}^{\mathsf{T}} \underbrace{\begin{bmatrix} y_1 y_1\, \mathbf{x}_1^{\mathsf{T}}\mathbf{x}_1 & y_1 y_2\, \mathbf{x}_1^{\mathsf{T}}\mathbf{x}_2 & \dots & y_1 y_N\, \mathbf{x}_1^{\mathsf{T}}\mathbf{x}_N \\ y_2 y_1\, \mathbf{x}_2^{\mathsf{T}}\mathbf{x}_1 & y_2 y_2\, \mathbf{x}_2^{\mathsf{T}}\mathbf{x}_2 & \dots & y_2 y_N\, \mathbf{x}_2^{\mathsf{T}}\mathbf{x}_N \\ \dots & \dots & \dots & \dots \\ y_N y_1\, \mathbf{x}_N^{\mathsf{T}}\mathbf{x}_1 & y_N y_2\, \mathbf{x}_N^{\mathsf{T}}\mathbf{x}_2 & \dots & y_N y_N\, \mathbf{x}_N^{\mathsf{T}}\mathbf{x}_N \end{bmatrix}}_{\text{quadratic coefficients}} \boldsymbol{\alpha} + \underbrace{(-\mathbf{1}^{\mathsf{T}})}_{\text{linear}} \boldsymbol{\alpha}$$

subject to $\underbrace{\mathbf{y}^{\mathsf{T}}\boldsymbol{\alpha} = 0}_{\text{linear constraint}}$

$$\underbrace{\mathbf{0}}_{\text{lower bounds}} \leq \boldsymbol{\alpha} \leq \underbrace{\infty}_{\text{upper bounds}}$$

Solution

Solution: $\boldsymbol{\alpha} = \alpha_1, \cdots, \alpha_N$

$$\Longrightarrow \quad \mathbf{w} \;=\; \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n$$

KKT condition: For $n = 1, \cdots, N$

$$\alpha_n \left(y_n \left(\mathbf{w}^\intercal \mathbf{x}_n + b \right) - 1 \right) \;=\; 0$$

$\alpha_n > 0 \implies \mathbf{x}_n$ is a **support vector**

Support vectors

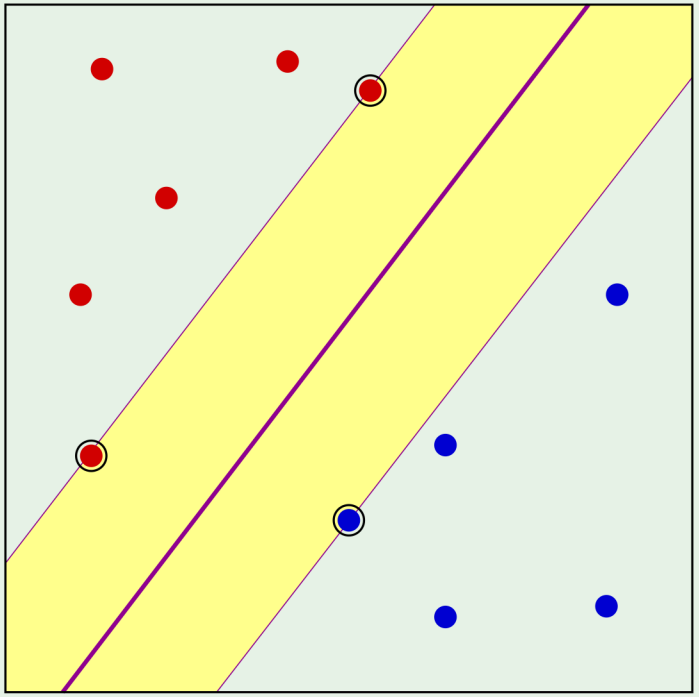
Closest $\mathbf{x}_n$'s to the plane: achieve the margin

$$\implies \quad y_n \left(\mathbf{w}^\mathsf{T} \mathbf{x}_n + b \right) = 1$$

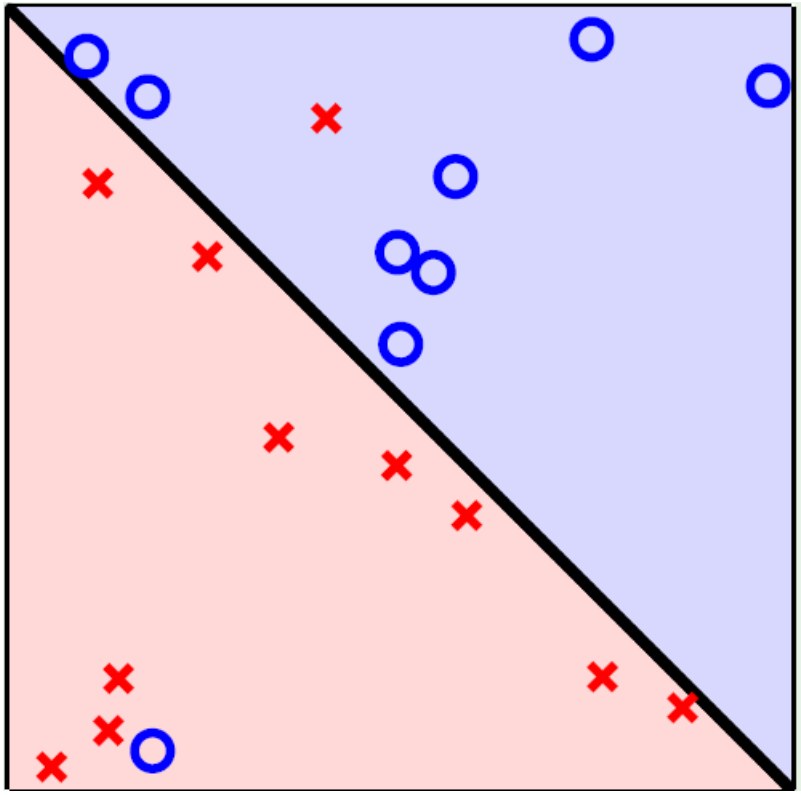
$$\mathbf{w} \;=\; \sum_{\mathbf{x}_n \text{ is SV}} \alpha_n y_n \mathbf{x}_n$$

Solve for b using any SV:

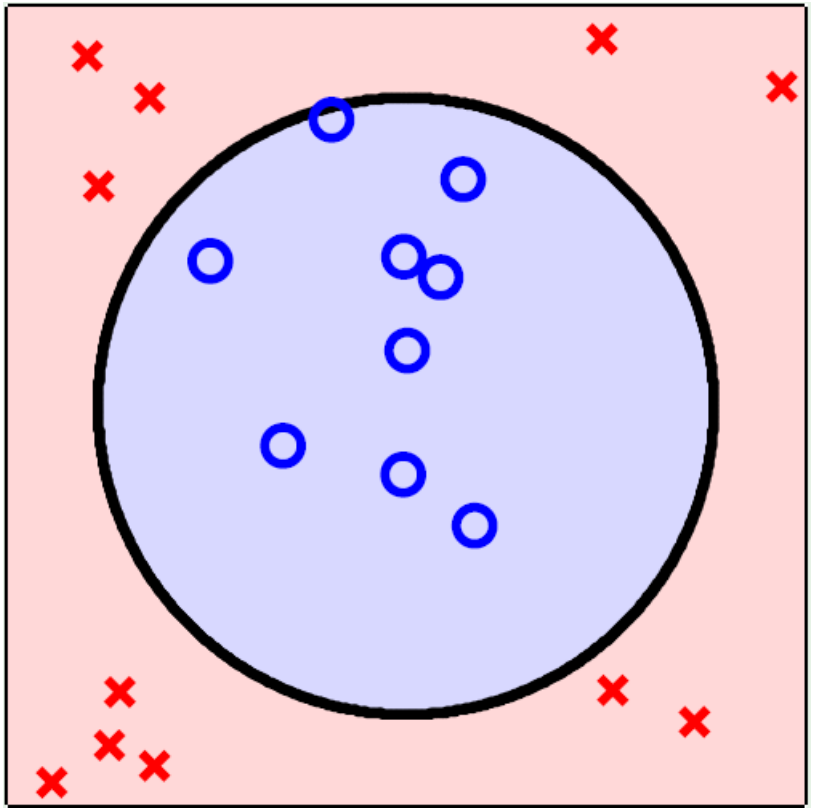
$$y_n \left(\mathbf{w}^\mathsf{T} \mathbf{x}_n + b \right) = 1$$



Non-separable Data



Slightly
non-separable

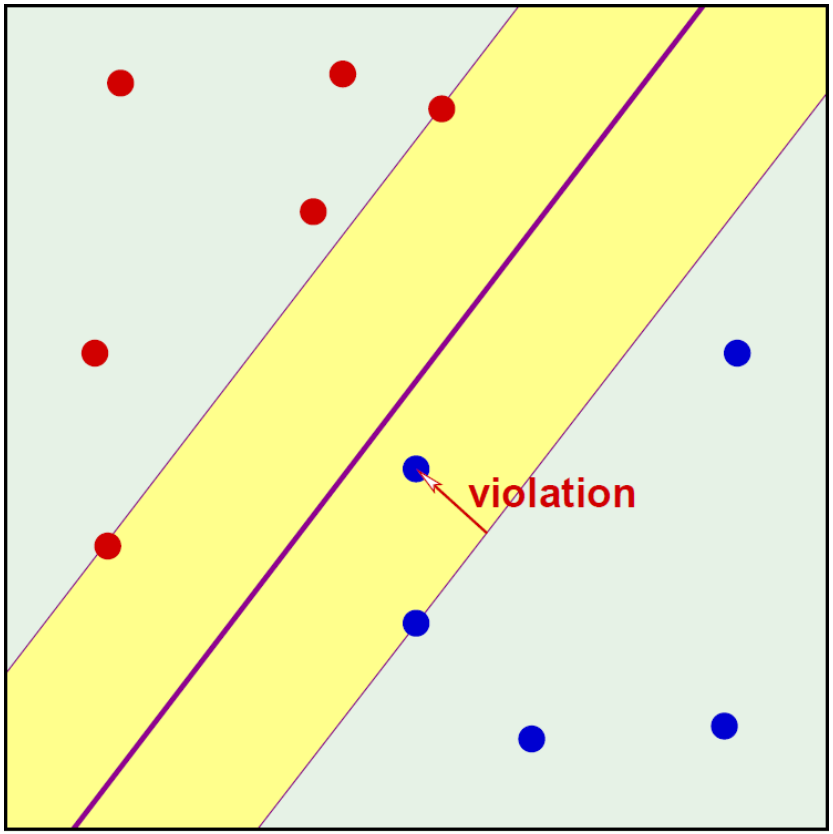


Seriously
non-separable

Soft-margin SVM

Margin violation: $y_n \left(\mathbf{w}^\intercal \mathbf{x}_n + b\right) \geq 1$ fails

Quantify: $y_n \left(\mathbf{w}^\intercal \mathbf{x}_n + b\right) \geq 1 - \xi_n$



$$\text{Total violation} = \sum_{n=1}^{N} \xi_n$$

ξ_n = ksi = "amount" of margin violation

$$\boxed{\xi_n \geq 0}$$

Soft-margin SVM optimization

- Similar to hard-margin optimization, but get compromise between maximizing the margin and allowing for violations

Still get large margin after term minimization

Allow for small violations by minimizing

$$\text{Minimize} \quad \frac{1}{2}\mathbf{w}^{\mathsf{T}}\mathbf{w} + C\sum_{n=1}^{N}\xi_n$$

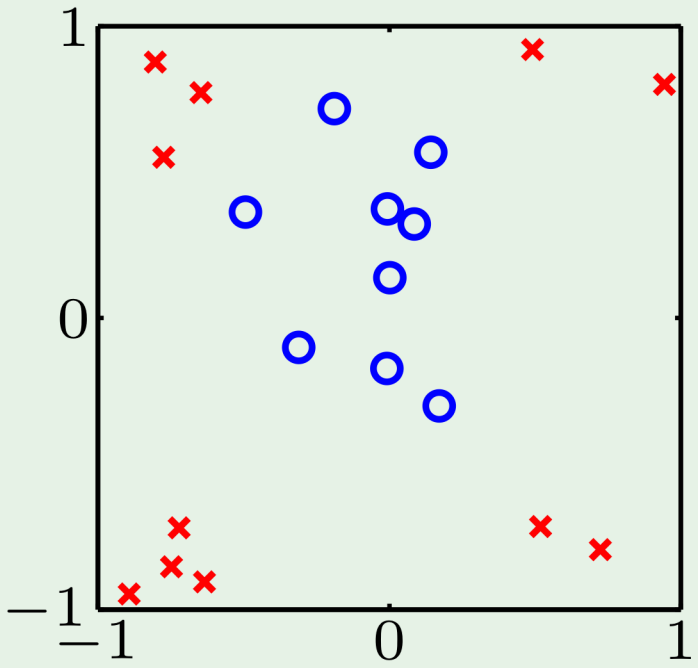
$$\text{subject to} \quad y_n\left(\mathbf{w}^{\mathsf{T}}\mathbf{x}_n + b\right) \geq 1 - \xi_n$$

$$\text{for} \quad n = 1, \ldots, N$$

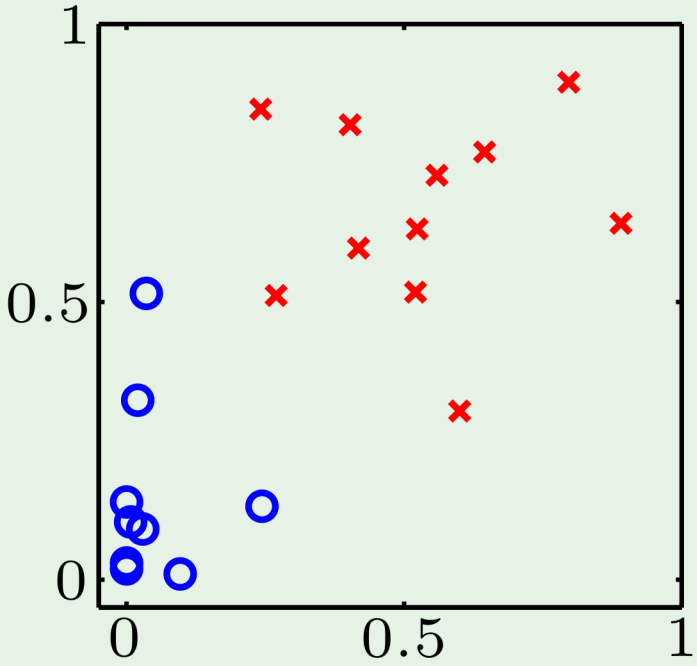
$$\text{and} \quad \xi_n \geq 0 \quad \text{for} \quad n = 1, \ldots, N$$

Kernel transform

$$\mathcal{L}(\boldsymbol{\alpha}) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} y_n y_m \, \alpha_n \alpha_m \, \mathbf{z}_n^{\mathsf{T}} \mathbf{z}_m$$



$\mathcal{X} \longrightarrow \mathcal{Z}$



Thank you!