Overview

W_01: Introduction

Coding_1

W_02: Linear Regression
W_03: Variable Selection & Regularization
W_04: Regression Trees & Ensemble
W_05: Nonlinear Regression

Coding_2
Project_1

Supervised Leaning
Regression

W_06: Clustering Analysis
W_07: Latent Structure Models

Coding_3
Coding_4

Unsupervised Leaning

W_09: Discriminant Analysis
W_10: Logistic Regression
W_11: Support Vector Machine
W_12: Classification Trees & Boosting

Coding_5
Project_3

Supervised Leaning
Classification

W_13: Recommender System
~~W_15: Introduction to Deep Learning~~

Project_4

Week 1: Introduction

- Supervised vs unsupervised
- Training vs test/generalization errors
- Why learning is difficult
  - Overfitting
- Bias-variance trade off
  - kNN
  - Linear regression
- A glimpse of statistical learning theory
  - Bayes error, Bayes classifier

How does Supervised Learning Work?

Data: $\{x\_i, y\_i\}$
i=1, 2, …, n

Goal: learn w
y <—- $f(x; w)$

Minimize an objective function

$$F(w) = \sum_{i=1}^{n} \text{loss}(y_i, f(x_i; w))$$

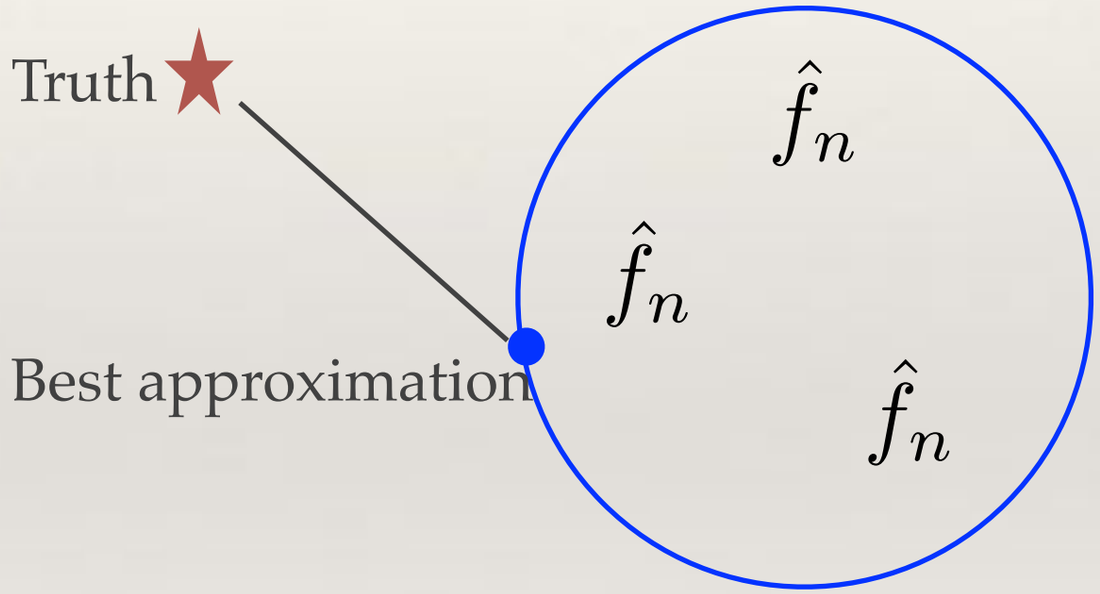
1. The minimizer w^* may be in closed form.
2. Try optimization algorithms that can guarantee to converge to the global minimizer.
3. In the worst case, try *gradient descent*.

Bias Variance Tradeoff

Goal of ML: Minimize *generalization error* (i.e., error on unseen future datasets), not training error.

Source of errors:

- Bias
- Variance



Week 1: Introduction

- Supervised vs unsupervised
- Training vs test/generalization errors
- Why learning is difficult
  - Overfitting
- Bias-variance trade off
  - kNN
  - linear regression
- A glimpse of statistical learning theory
  - Bayes error, Bayes classifier

Coding_1

- Mixture dist
- Implement kNN
  - voting ties
  - distance ties
- Implement CV for k
  - Non-uniqueness of modes
- Bayes classifier

Week 2: Linear Regression

- Least squares estimates
- Geometric interpretation: projection

- Influences of affine transformations of X
- Residual vector orthogonal to col(X) and y-hat
- Influence of co-linearity (rank deficiency)
- Partial regression coefficients
  - Frisch-Waugh-Lovell theorem
  - Double machine learning (causal inference)

- How to handle categorical predictors
  - How to code them
  - How to interpret their coefficients
  - How to interpret interactions

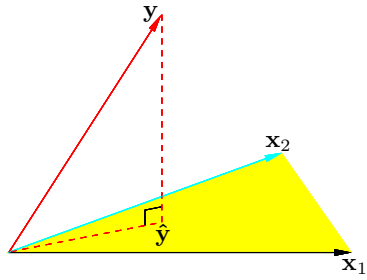
The Geometric Interpretation of LS

Recall that the LS optimization

$$\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2,$$

which is equivalent to finding a vector $\mathbf{v}$ from the subspace $C(\mathbf{X})$ that minimizes $\|\mathbf{y} - \mathbf{v}\|^2$.

Intuitively we know what the optimal $\mathbf{v}$ is: it's the projection of $\mathbf{y}$ onto the space $C(\mathbf{X})$.



The essence of LS: decompose the data vector $\mathbf{y}$ into two orthogonal components,

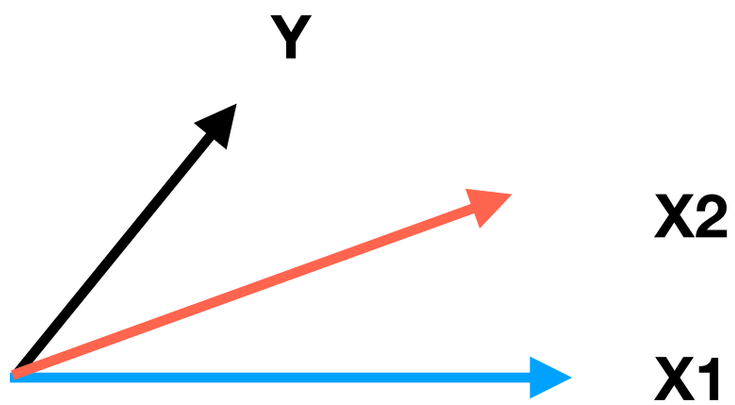
$$\mathbf{y}_{n\times 1} = \hat{\mathbf{y}}_{n\times 1} + \mathbf{r}_{n\times 1}.$$

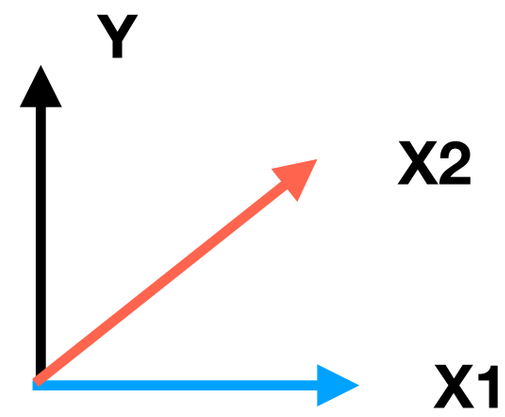
Rank Deficiency

- Rank deficiency is not a serious issue: the linear subspace $C(\mathbf{X})$, spanned by the columns of $\mathbf{X}$, is well-defined and therefore $\hat{\mathbf{y}}$ is well-defined and can be computed.
- Due to rank deficiency, $\hat{\boldsymbol{\beta}}$ is not unique.
- In `R`, LS coefficients $=$ `NA` means rank deficiency. You can still use the returned model to do prediction.

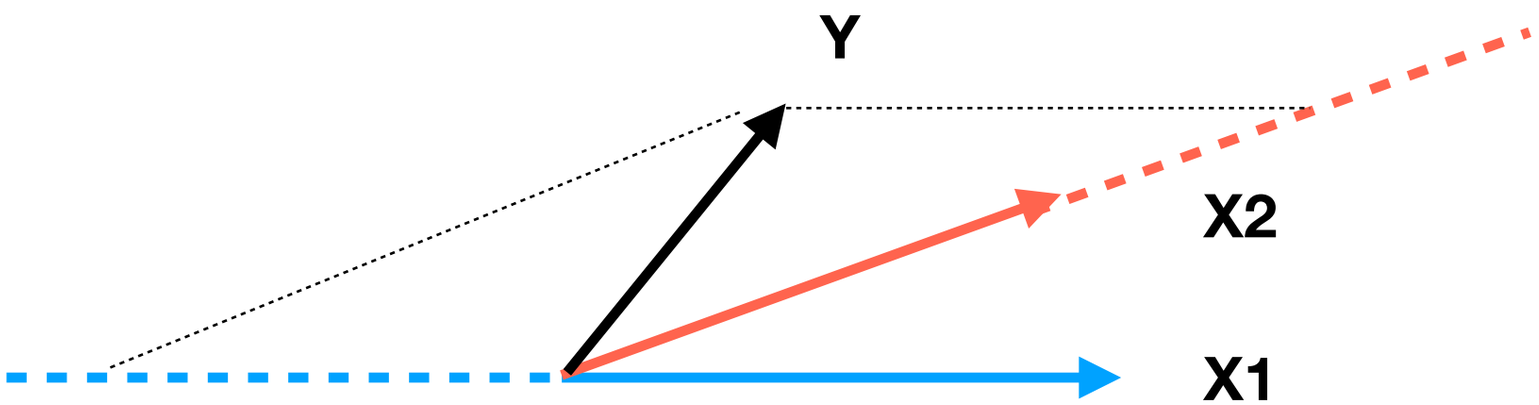
$$\mathbf{X}_{n\times 2} = \begin{pmatrix} 1 & 2 \\ 1 & 2 \\ . & . \\ 1 & 2 \end{pmatrix}$$

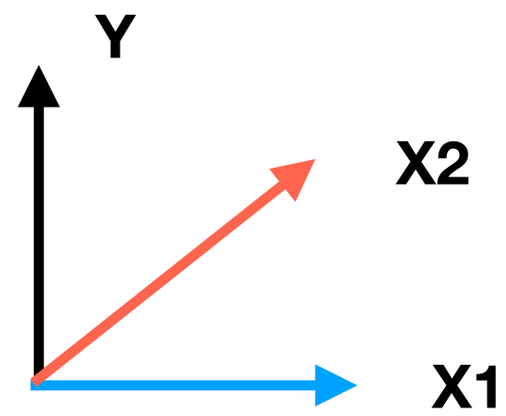
Week 2: Linear Regression





Week 2: Linear Regression





Week 2: Linear Regression

- Least squares estimates
- Geometric interpretation: projection

- Influences of affine transformations of X
- Residual vector orthogonal to col(X) and y-hat
- Influence of co-linearity (rank deficiency)
- Partial regression coefficients
  - Frisch-Waugh-Lovell theorem
  - Double machine learning (causal inference)

- How to handle categorical predictors
  - How to code them
  - How to interpret their coefficients
  - How to interpret interactions

Partial Regression Coefficients

Consider a multiple linear regression model

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k + \cdots + \beta_p X_p + \text{err}.$$

The LS estimate $\hat{\beta}_k$ describes the partial correlation between Y and X_k **adjusted for the other predictors**.

The LS estimate $\hat{\beta}_k$ can be obtained as follows (see Algorithm 3.1 from ESL):

1. Y^*: residual from regressing Y onto all other predictors except X_k
2. X_k^*: residual from regressing X_k onto all other predictors except X_k
3. Regress Y^* onto X_k^*

Week 3: Variable Selection & Regularization

- AIC/BIC
- Lasso
  - Exact solution for one-dim Lasso
  - Coordinate descent algorithm
- Ridge
  - Exact solution
- ElasticNet
- glmnet / scikit-learn
- PCA and PCR

Coding_2

- Implement coordinate descent for Lasso
- Compare Lasso/Ridge/PCR/variants on two datasets

Week 4: Regression Trees & Ensemble

- How to build a tree
  - Choose a split criterion
  - Grow a big tree and then prune
- How to build a forest
  - Random forest via bagging
  - XGBoost via boosting
- Invariant to monotonic transformation of features
- How to code categorical features

Project_1

- Pre-processing
- Try two algorithms
- Handle unseen new levels

Week 5: Nonlinear Regression

- Polynomial regression
- Splines
  - Regression splines
  - Smoothing splines
- Local regression
  - Kernel smoothing
  - Loess
- GAM

Week 5: Nonlinear Regression

- Polynomial regression
- Splines
  - Regression splines
  - Smoothing splines
- Local regression
  - Kernel smoothing
  - Loess
- GAM

Coding_3

- Leave-one-out CV
- Use cubic splines to represent functional data

Project_2

- Simple LR + efficient implementation
- `prophet, forecast`

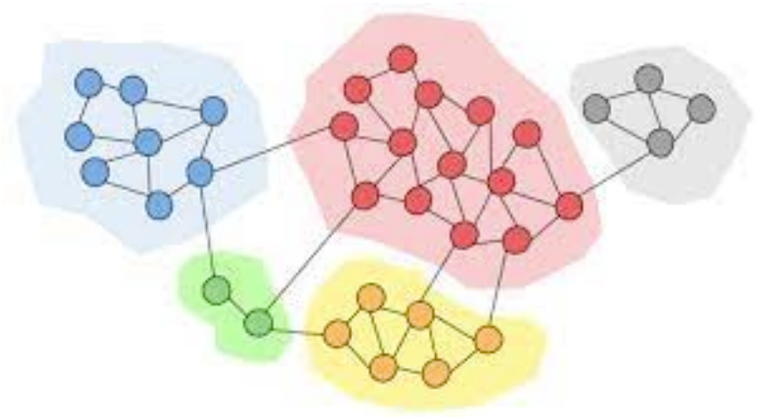
Week 6: Clustering Analysis

- K-means / K-mediods
- MDS
- How to select K
  - Gap statistics
  - Silhouette statistics
  - Prediction strength

Week 6: Clustering Analysis

- K-means / K-mediods
- MDS
- How to select K
  - Gap statistics
  - Silhouette statistics
  - Prediction strength

- Spectral/Graph clustering
- Community detection

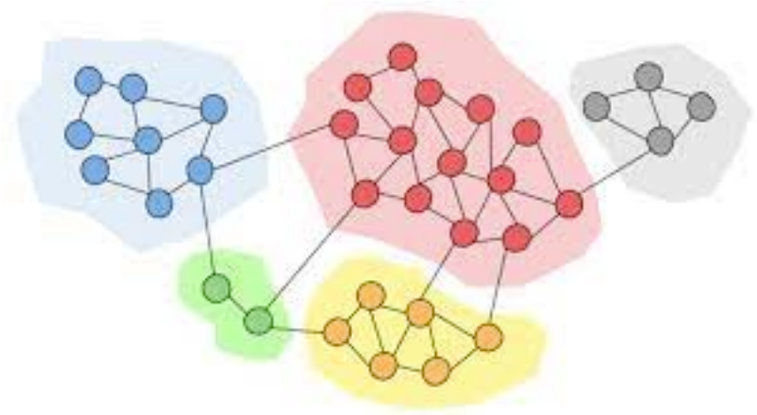


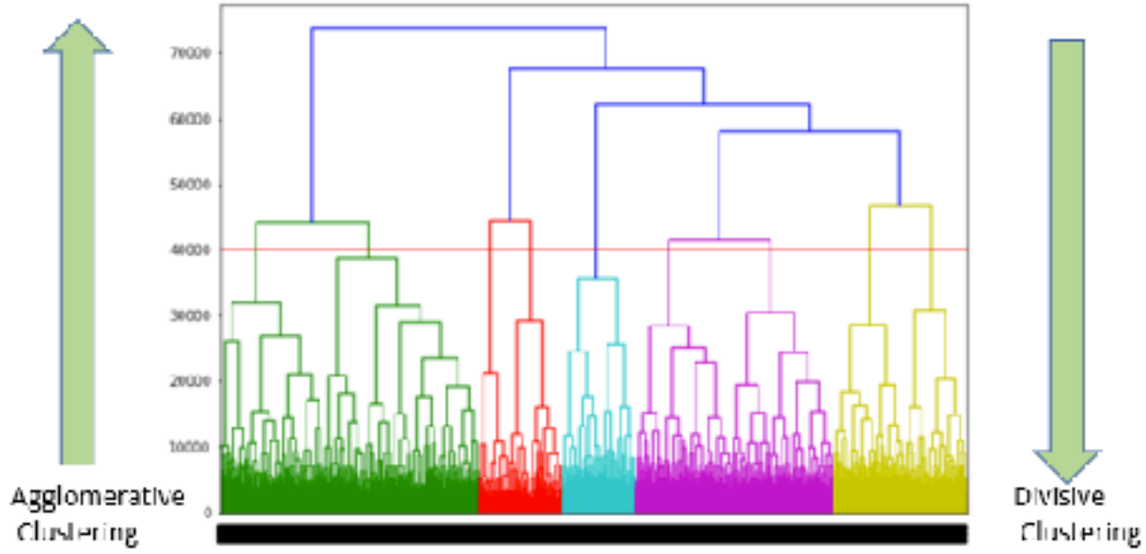
Week 6: Clustering Analysis

- K-means / K-mediods
- MDS
- How to select K
  - Gap statistics
  - Silhouette statistics
  - Prediction strength

- Hierarchical clustering
  - Bottom-up; top-down
  - Single-link, Complete-link & Average-link

- Spectral/Graph clustering
- Community detection





Week 7: Latent Structure Models

$$p_\theta(x) = \sum_k p_\theta(x, z = k) = \sum_k p_\theta(x|z = k) p_\theta(z = k)$$

- Gaussian mixture model
- Hidden Markov model (HMM)
- EM algorithm

Coding_4

- EM algorithm for Gaussian mixture model
- EM (Baum-Welch) and Viterbi algorithms for HMM

Week 9: Discriminant Analysis

1. The Bayes rule is based on **P(Y=y I X=x)**
2. In DA, we **estimate** the joint and then **obtain** P(Y=y | X=x)

$$P(x, y) = P(Y=y \mid X=x)\, P(X=x)$$
$$= P(X=x \mid Y=y)\, P(Y=y)$$

Joint dist — **Conditions of XIY** — **Marginal of Y**

Dist of p-dim X given Y=k: **QDA, LDA (FDA), NB**

Week 10: Logistic Regression

$$\eta(x) = P(Y = 1|X = x)$$
$$1 - \eta(x) = P(Y = 0|X = x)$$

$$\log \frac{\eta(x)}{1 - \eta(x)} = x^t \boldsymbol{\beta}$$

- Maximum likelihood & Newton-Raphson
- Variable selection & shrinkage
  - AIC/BIC: stepwise/forward/backward
  - Lasso/Ridge/Elastic-net: glmnet / scikit-learn

- Convergence issue with separable data
- How to handle retrospect sampling
- Multinomial logistic regression

Evaluate Classification Accuracy

Confusion Matrix and ROC Curve

		Predicted Class	
		No	Yes
Observed Class	No	TN	FP
	Yes	FN	TP

TN	True Negative
FP	False Positive
FN	False Negative
TP	True Positive

Model Performance

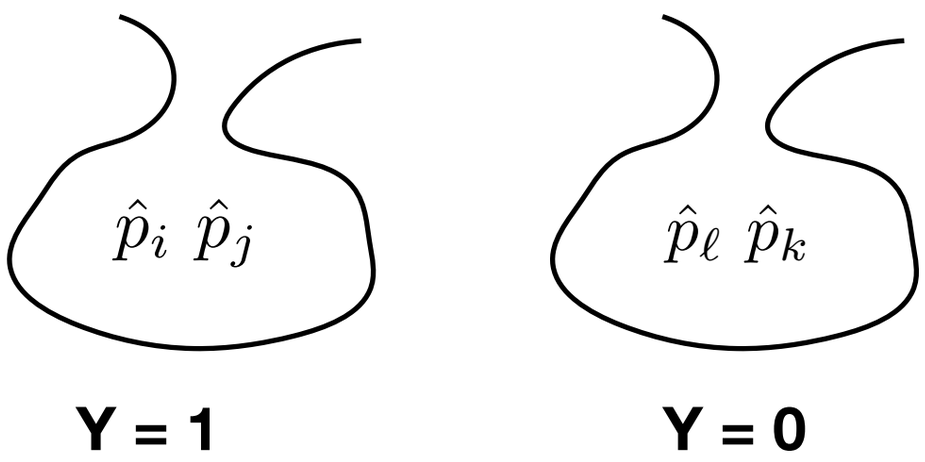
Accuracy	= (TN+TP)/(TN+FP+FI
Precision	= TP/(FP+TP)
Sensitivity	= TP/(TP+FN)
Specificity	= TN/(TN+FP)

Week 10: Logistic Regression

$$\eta(x) = P(Y = 1|X = x)$$
$$1 - \eta(x) = P(Y = 0|X = x)$$

$$\log \frac{\eta(x)}{1 - \eta(x)} = x^t \boldsymbol{\beta}$$

- Maximum likelihood & Newton-Raphson
- Variable selection & shrinkage
  - AIC/BIC
  - glmnet (Lasso/Ridge/Elastic-net)

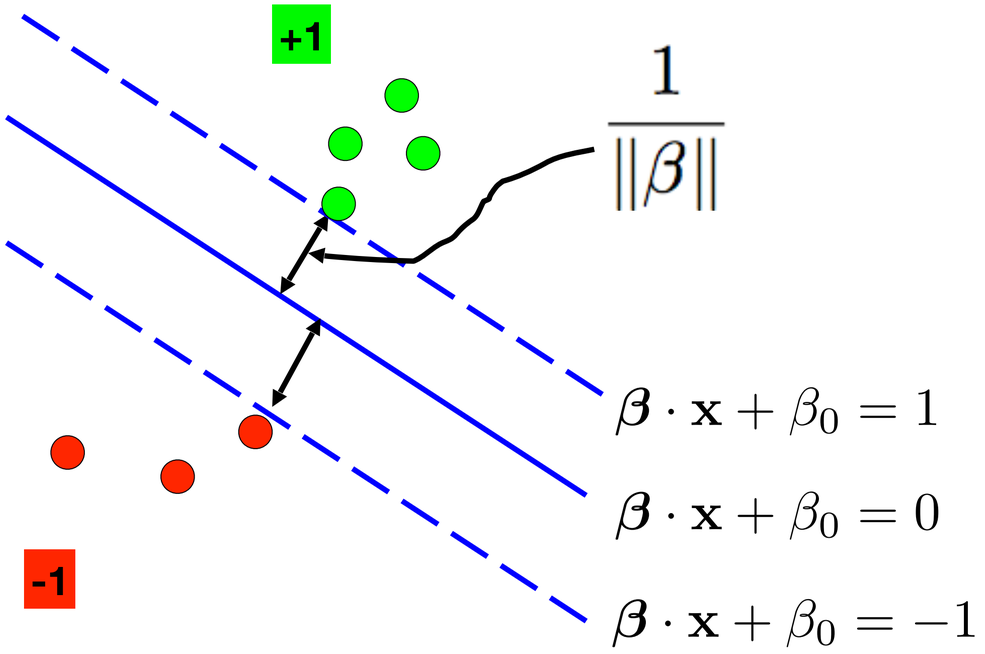


Project_3
- AUC

Week 11: Support Vector Machine

- Optimization with inequality constraints
- Convex optimization
- RKHS & Representer Theorem

- Choice of kernels
- Computation cost
  - NIPS 2017 test-of-time award: *Random Features for Large-Scale Kernel Machines* (NIPS 2007) by Ali Rahimi, Benjamen Recht.
- Connection to Gaussian Process (GP)



Max-Margin Problem

$$\begin{aligned} \min_{\boldsymbol{\beta},\beta_0} \quad & \frac{1}{2}\|\boldsymbol{\beta}\|^2 \\ \text{subject to} \quad & y_i(\boldsymbol{\beta} \cdot \mathbf{x}_i + \beta_0) - 1 \geq 0, \end{aligned} \tag{1}$$

where $\boldsymbol{\beta} \cdot \mathbf{x}_i = \boldsymbol{\beta}^t \mathbf{x}_i$ denotes the (Euclidian) inner product between two vectors. The constraints are imposed to make sure that the points are on the correct side of the dashed lines, i.e.,

$$\begin{aligned} \boldsymbol{\beta} \cdot \mathbf{x}_i + \beta_0 \geq +1 \quad & \text{for } y_i = +1, \\ \boldsymbol{\beta} \cdot \mathbf{x}_i + \beta_0 \leq -1 \quad & \text{for } y_i = -1. \end{aligned}$$

Primal

$$\min_{\boldsymbol{\beta},\beta_0} \frac{1}{2}\|\boldsymbol{\beta}\|^2$$

$$\text{subj to } y_i(\mathbf{x}_i \cdot \boldsymbol{\beta} + \beta_0) - 1 \geq 0,$$

$$i = 1, \ldots, n$$

Dual

$$\max_{\lambda_{1:n}} \sum \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j)$$

$$\text{subj to } \sum \lambda_i y_i = 0,$$

$$\lambda_i \geq 0$$

KKT conditions

$$\sum_i \lambda_i y_i \mathbf{x}_i = \boldsymbol{\beta}$$

$$\sum_i \lambda_i y_i = 0$$

$$\lambda_i \geq 0$$

$$y_i(\mathbf{x}_i \cdot \boldsymbol{\beta} + \beta_0) - 1 \geq 0$$

$$\lambda_i \Big[y_i(\mathbf{x}_i \cdot \boldsymbol{\beta} + \beta_0) - 1 \Big] = 0$$

Why work with Dual?

1. Easier to solve
2. Many lambda_i's are zero
3. Leads to kernel trick

Lagrange function

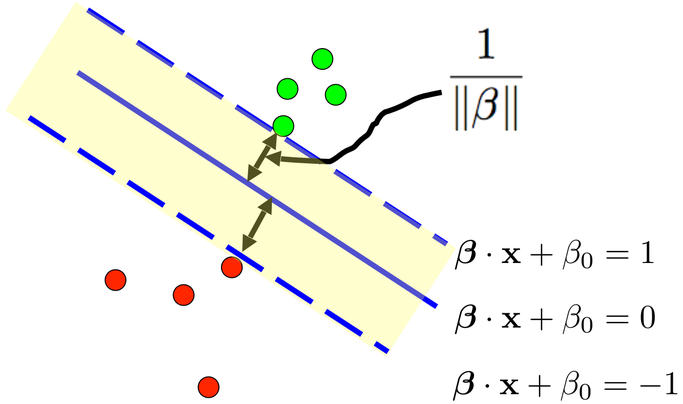
$$L(\boldsymbol{\beta}, \beta_0, \lambda_{1:n})$$

$$= \frac{1}{2}\|\boldsymbol{\beta}\|^2 - \sum_i \lambda_i \Big[y_i(\mathbf{x}_i^t \boldsymbol{\beta} + \beta_0) - 1 \Big]$$

$$= \frac{1}{2}\|\boldsymbol{\beta}\|^2 - \sum_i \lambda_i y_i (\mathbf{x}_i^t \boldsymbol{\beta} + \beta_0) + \sum_i \lambda_i$$

Hard Margin

Linear SVM for Separable Data

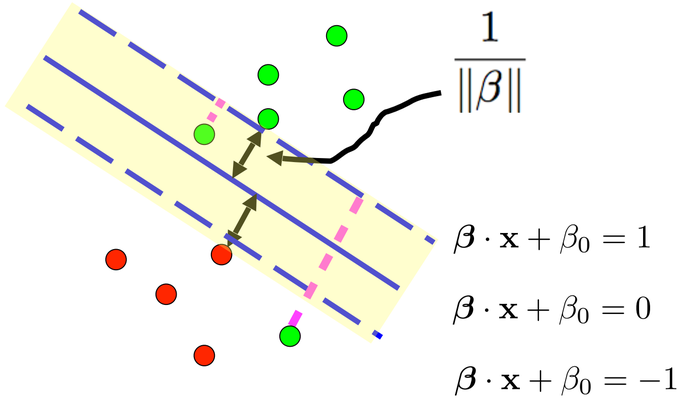


Kernel Machine

Nonlinear SVM for Separable/Non-separable Data

Soft Margin

Linear SVM for Non-separable/Separable Data



1. Formulate the **Primal** Problem (dim = p+1)
2. Solve the **Dual** Problem (dim = n)
3. **KKT Conditions** link the two sets of solutions
4. **SV**: data points on the dashed lines or on the wrong side of the datelines

Some Practical Issues

1. Binary decision to probability
2. Multiclass SVM

Primal

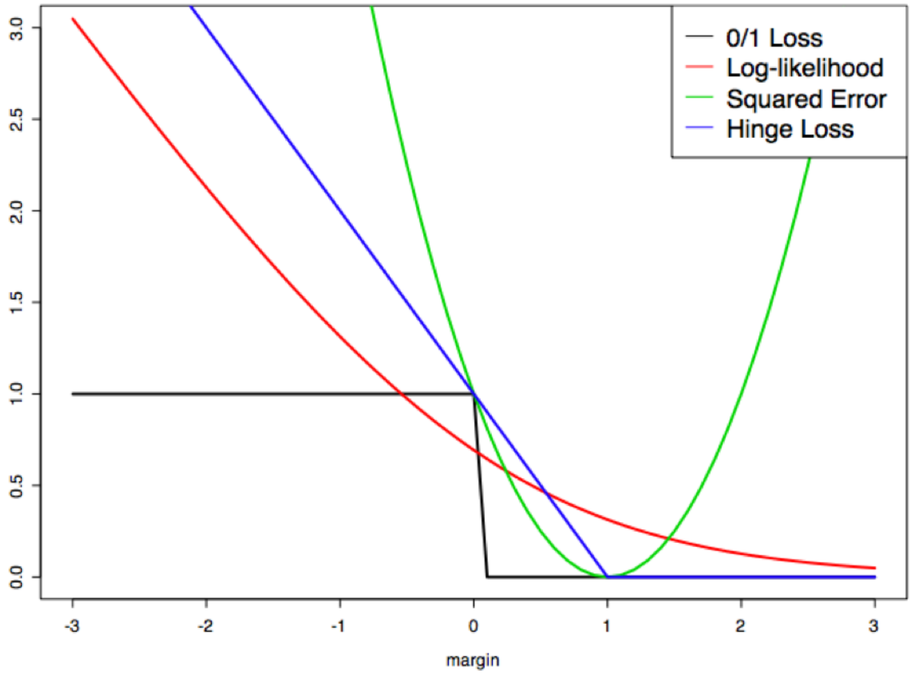
$$\min_{\boldsymbol{\beta},\beta_0,\xi_{1:n}} \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \gamma \sum \xi_i$$

$$\text{subj to } y_i(\mathbf{x}_i \cdot \boldsymbol{\beta} + \beta_0) \geq 1 - \xi_i,$$

$$\xi_i \geq 0$$

$$1 < y_i f(\mathbf{x}_i) \Longrightarrow 1 - y_i f(\mathbf{x}_i) < 0, \quad \xi_i = 0$$

$$1 \geq y_i f(\mathbf{x}_i), \Longrightarrow 1 - y_i f(\mathbf{x}_i) = \xi_i$$



SVM as a penalization method

Let $f(x) = \mathbf{x} \cdot \boldsymbol{\beta} + \beta_0$ and $y_i \in \{-1, 1\}$. Then

Hinge Loss

$$\min_{\boldsymbol{\beta},\beta_0} \sum_{i=1}^{n} [1 - y_i f(\mathbf{x}_i)]_+ + \nu\|\boldsymbol{\beta}\|^2 \tag{6}$$

has the same solution as the linear SVM (3), when the tuning parameter ν is properly chose (which will depend on γ in (3)). So SVM is a special case of the following Loss + Penalty framework

Reciprocally related

Week 11: Support Vector Machine

- Optimization with inequality constraints
- Convex optimization
- RKHS & Representer Theorem

- Choice of kernels
- Computation cost
  - NIPS 2017 test-of-time award: *Random Features for Large-Scale Kernel Machines* (NIPS 2007) by Ali Rahimi, Benjamen Recht.
- Connection to Gaussian Process (GP)