

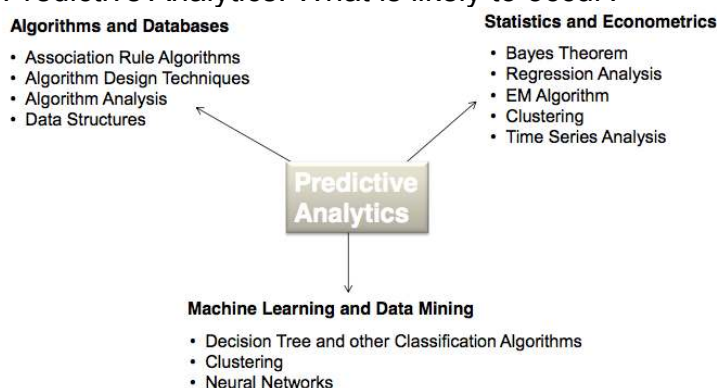


Business Analytics - Wintersemester

Business Analytics (IN2028) (Technische Universität München)

1. INTRODUCTION

- Business analytics makes extensive use of statistical analysis, including explanatory and predictive modeling, and fact-based management to drive decision making. It is therefore closely related to management science. Analytics may be used as input for human decisions or may drive fully automated decisions.
- Descriptive Analytics: What has occurred?
- Predictive Analytics: What is likely to occur?



- Prescriptive Analytics: What is likely to occur, if...? What impact does an action have on future?
- Classification: From data with known labels, create a classifier that determines which label to apply to a new observation
- Clustering: Identify natural groupings in data; Unsupervised learning, no predefined groups
- Association Rule Analysis: Identify relationships in data from co-occurring terms or items

Statistics

- Descriptive: Summarize data (numerically or graphically) to describe sample
- Inferential: Model patterns in data, accounting for randomness and drawing inferences about larger population
→ Estimation, hypothesis testing, forecasting, correlation, regression

Standard Normal

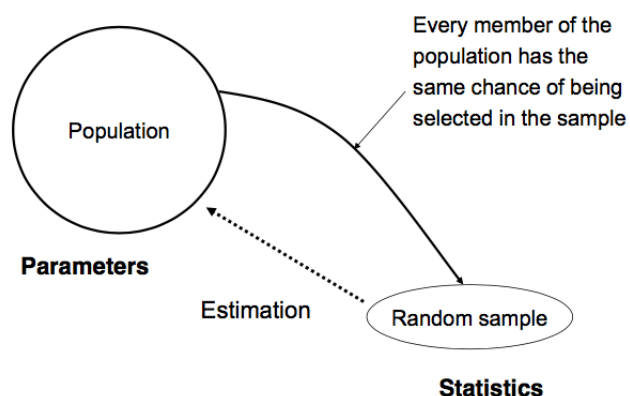
- Standardize random variable by subtracting the mean, dividing by standard deviation
- $N(0,1)$ has probability density function (pdf)

$$\varphi(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}}$$

- For a pdf, $f(x)$, where $f(x)$ is $P(X = x)$, the cumulative distribution function (cdf), $F(x)$, is $P(X \leq x)$; $P(X > x) = 1 - F(x)$
- For the standard normal, $\varphi(z)$, the cdf is

$$\Phi(z) = P(Z \leq z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^z e^{-\frac{t^2}{2}} dt$$

Statistical Estimation



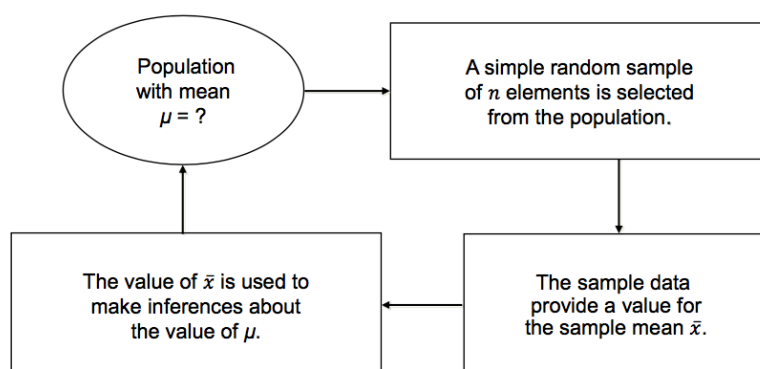
Standard Error of the Mean

$$\begin{aligned} \text{Var}(\bar{X}) &= \text{Var}\left(\frac{1}{n} \sum_{i=1}^n (X_i)\right) = \frac{1}{n^2} \sum_{i=1}^n \text{Var}(X_i) \\ &= \frac{1}{n^2} \sum_{i=1}^n \sigma^2 = \frac{1}{n^2} (n\sigma^2) = \frac{\sigma^2}{n} \\ \frac{\sigma}{\sqrt{n}} &= SD(\bar{X}) = \sqrt{\text{Var}(\bar{X})} = \frac{\sigma}{\sqrt{n}} \end{aligned}$$

$$\text{Rule: } \text{Var}[aX + b] = a^2 \text{Var}[X]$$

2. INFERENCE STATISTICS

Statistical Estimation



Confidence Interval (CI)

- Range of values that we believe, with given level of confidence, contains a population parameter CI for the population means
 $\Pr(\bar{X} - 1.96 SD \leq \mu \leq \bar{X} + 1.96 SD) = 0.95$
 \rightarrow With 95% chance the interval contains the mean
- Larger sample size \rightarrow Smaller interval
- Lower confidence level \rightarrow Smaller interval
- More variation \rightarrow Larger interval

Estimation for Population Mean

Point estimate:	$\bar{X} = \frac{\sum X}{n}$
Estimate of variability in population (if σ is unknown, use s)	$s = \sqrt{\frac{1}{n-1} \sum (X_i - \bar{X})^2}$
True standard deviation of sample mean	$SD = \sigma / \sqrt{n}$
Standard error of sample mean	$SE = s / \sqrt{n}$
95% Confidence Interval , or	$\bar{X} \pm 1.96 SD$ $\bar{X} \pm 1.96 SE$

Hypothesis Testing

- State null and alternative hypothesis (H_0 and H_1)
- Choose α level (related to confidence level)
- Calculate test statistic, find p-value (Measures how far data from null hypothesis)
- State conclusion
 $p \leq \alpha$, reject H_0
 $p > \alpha$, insufficient evidence to reject H_0
- Possible results

		What we decide	
		Reject null	Fail to reject null
Reality	Null true	Type I Error (α) (false positive)	Correct
	Null false	Correct	Type II Error (β) (false negative)

- Power of statistical test: Probability that it correctly rejects H_0 when it is false
 \rightarrow Sample size matters

Z-Test

- Interest: Population mean of normal distribution
- Known σ

Z-confidence interval: $\bar{x} \pm z_{\alpha/2} \frac{\sigma}{\sqrt{n}}$

Z-test: $z = \frac{\bar{x} - \mu_0}{\sigma / \sqrt{n}}$

H_a	Rejection region
$\mu \neq \mu_0$	$ z \geq z_{\alpha/2}$
$\mu > \mu_0$	$z \geq z_{\alpha}$
$\mu < \mu_0$	$z \leq -z_{\alpha}$

Test Statistics for Normal Mean with unknown σ

$$z = \frac{\bar{X} - \mu}{\sigma / \sqrt{n}} \quad t = \frac{\bar{X} - \mu}{s / \sqrt{n}}$$

- Sample standard deviation s used to estimate σ
- t has Student t-distribution with $n-1$ degrees of freedom (DF)

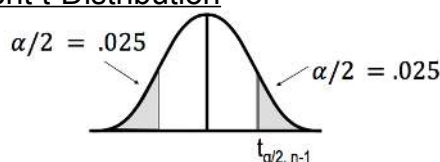
CI and 2-Sided Tests

- A level α 2-sided test rejects $H_0: \mu = \mu_0$ exactly when the value μ_0 falls outside a level $1 - \alpha$ confidence interval for μ .
- Calculate $1 - \alpha$ level confidence interval, then
 - if μ_0 within the interval, do not reject the null hypothesis,

$$|t| < t_{\alpha/2}$$

– otherwise, $|t| \geq t_{\alpha/2} \Rightarrow$ reject the null hypothesis.

Student t-Distribution



Degrees of Freedom	$t_{.100}$	$t_{.05}$	$t_{.025}$	$t_{.01}$	$t_{.005}$
1	3.078	6.314	12.706	31.821	63.657
2	1.886	2.92	4.303	6.965	9.925
...
24	...	1.711	2.064	2.492	...
...
200	1.286	1.653	1.972	2.345	2.601
∞	1.282	1.645	1.96	2.326	2.576

t-distribution critical values

- If population normally distributed, statistic t is Student t distributed

$$t(df = n - 1) = \frac{\bar{X} - \mu}{s/\sqrt{n}}$$

- DF determine how spread the distribution is (compared to normal distr.)

t-Tests

- Single sample: Is sample mean significantly different from pre-existing value?
- Paired samples: Relationship between 2 linked samples (eg means obtained in 2 conditions by a single group of participants)
- Independent samples: Relationship between 2 independent populations

Paired t-Test with 2 Paired Samples

Null hypothesis: $H_0: \mu_d = \mu_1 - \mu_2 = \Delta_0$

Test statistic: $t = \frac{\bar{d} - \Delta_0}{s/\sqrt{n}}$

H_1

$$\mu_d \neq \Delta_0$$

$$\mu_d > \Delta_0$$

$$\mu_d < \Delta_0$$

Rejection region

$$|t| \geq t_{\alpha/2, n-1}$$

$$t \geq t_{\alpha, n-1}$$

$$t \leq -t_{\alpha, n-1}$$

Observations are dependent, e.g., pre and post test,
left and right eyes, brother-sister pairs

p-Value

- Describes probability of having certain t -value (or larger), given the null hypothesis
- The smaller p , the more unlikely the null hypothesis seems
- Same as significance level

Two-Sample t-Test

$$t = \frac{\bar{x}_1 - \bar{x}_2}{S_{\bar{x}_1 - \bar{x}_2}} \quad S_{\bar{x}_1 - \bar{x}_2} = \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}$$

Two-sample unpaired t -test with unequal sample sizes, assuming unequal variance

Under H_0 t follows a t -distribution with $\frac{(s_1^2/n_1 + s_2^2/n_2)^2}{(s_1^2/n_1)^2/(n_1-1) + (s_2^2/n_2)^2/(n_2-1)}$ degrees of freedom (df)

Selected Statistical Tests

- Parametric Tests
 - F-Test (Compares equivalence of variances of 2 samples)
 - Family of t -tests (Compares 2 sample means or tests single sample mean)
- Non-parametric Tests
 - Wilcoxon signed-rank test (Independence of 2 means for 2 paired i.i.d. samples, when normality cannot be assumed; Mann-Whitney-U test used for 2 independent samples)
 - Kruskal-Wallis-Test (Equivalence of multiple means in case of several i.i.d. non-normally distributed samples)
- Tests of the Probability Distribution
 - Kolmogorov-Smirnov and Chi-square test (Determine whether 2 underlying probability distribution differ, or whether underlying prob. distribution differs from hypothesized distribution)

Linear Regression

Regressions identify relations between dependent and independent variables

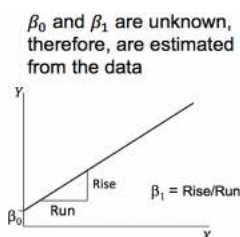
- Association between the 2 variables?
- Estimation of impact of an independent variable
- Formulation of relation in a functional form
- Used for numerical prediction and time series forecasting

Simple Linear Regression Model

- First order linear model

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

Y = respond variable
 X = predictor variable
 β_0 = y-axis intercept
 β_1 = slope of the line
 ε = random error term (residual)



Estimating the Coefficients

- Coefficients are random variables
- (Ordinary Least Squares) estimates determined by
 - Drawing sample from population
 - Calculating sample statistics
 - Producing straight line that cuts into data

- OLS approach

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

$$\min \sum e^2 = \min \sum (y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i))^2$$

$$\hat{\beta}_1 = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2} = \frac{Cov(x, y)}{Var(x)}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

Residual Sum of Squares (RSS)

- Sum of squared difference between points and regression line → How well line fits data
- Unbiased estimator of RSS of the population:

$$RSS = \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Coefficient of Determination

- R^2 measures proportion of variation in y that is explained by variation in x

$$ESS = \sum_{i=1}^n (\hat{y}_i - \bar{y})^2, TSS = \sum_{i=1}^n (y_i - \bar{y})^2 = ESS + RSS$$

$$R^2 = \frac{TSS - RSS}{TSS} = 1 - \frac{RSS}{TSS} = \frac{ESS}{TSS}$$

- $R^2 = 1$: Perfect match between line and data points
- $R^2 = 0$: No linear relation between x and y

Testing the Coefficients

- Test the significance of the linear relationship

$$H_0: \beta_1 = 0$$

$$H_1: \beta_1 \neq 0$$

- The test statistic is

$$t = \frac{\hat{\beta}_1}{SE(\hat{\beta}_1)} = \frac{\hat{\beta}_1}{\sqrt{\frac{RSS}{\sum_{i=1}^n (x_i - \bar{x})^2} \cdot \frac{1}{n-2}}}$$

← Variance of $\hat{\beta}_1$

- If $SE(\hat{\beta}_1)$ is large, then $\hat{\beta}_1$ must be large to reject H_0
- $SE(\hat{\beta}_1)$ is smaller, if the x_i are more spread out
- If the error variable is normally distributed, the statistic is a Student t distribution with $n - 2$ degrees of freedom (if n is large, draw on the CLT)
- Reject H_0 , if: $t < -t_{\alpha/2}$ or $t > t_{\alpha/2}$

Multiple Linear Regression Model

- p-variable regression model expressed as series of equations
- Equations condensed into matrix form, give general linear model
- β coefficients as partial regression coefficients
- Example

X_1, X_2 , for example,

– X_1 = 'years of experience'

– X_2 = 'age'

– y = 'salary'

Estimated equation:

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2 = \mathbf{X}\hat{\beta}$$

- Matrix notation

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \\ 1 & x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{np} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}$$

$y =$	\mathbf{X}	β	$+ \varepsilon$
$(n \times 1)$	$(n \times (p+1))$	$((p+1) \times 1)$	$(n \times 1)$

OLS Estimation

- Sample-based counter part to population regression model:

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

$$y = \mathbf{X}\hat{\beta} + e$$

- OLS requires choosing values of estimated coefficients, such that RSS is as small as possible for the sample

$$RSS = e^T e = (y - \mathbf{X}\hat{\beta})(y - \mathbf{X}\hat{\beta})$$

- Need to differentiate with respect to the unknown coefficients

Gauss-Markov Theorem

- In linear regression model in which errors have expectation zero and are uncorrelated and have equal variances: Best linear unbiased estimator (BLUE) of the coefficients is given by OLS estimator
- Unbiased means
- Best means giving lowest variance of estimates as compared to other linear unbiased estimators
- Assumptions
 - Linearity: Linear relation among predictors and y
 - No multicollinearity of predictors
 - Homoscedasticity: Residuals follow normal distr. and exhibit constant variance
 - No autocorrelation: No correlation between i^{th} and j^{th} residual terms
 - Exogeneity of independent variables: Covariance between X's and residual term is 0
 - Expected value of residual vector is 0

Total Deviation

$$\sum (y - \bar{y})^2 = \sum (\hat{y} - \bar{y})^2 + \sum (y - \hat{y})^2$$

$$TSS = ESS + RSS$$

$$\text{Total deviation} = \text{explained deviation} + \text{unexplained deviation}$$

Selected Statistics

- Adjusted R^2
 - Represents proportion of variability of y explained by the X's; R^2 is adjusted so that models with different number of variables can be compared
 - $$\bar{R}^2 = 1 - \frac{RSS/(n-p-1)}{TSS/(n-1)}$$
- F-test
 - Significant F indicates linear relationship between y and at least one of the X's
 - $H_0: \beta_1 = \beta_2 \dots \beta_p = 0$

- t-test of each partial regression coefficient
→ Significant t indicates that variable in question influences response variable while controlling for other explanatory variables

Considering Nominal Predictor Variables

→ Code (eg Gender) binary

Then in the regression equation $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2$ when

$x_{1,1} = 1$ the value of Y indicates what is obtained for female gender;

When $x_{1,1} = 0$ the value of Y indicates what is obtained for males.

Regression Model Building

- Setting: Possibly large set of predictor variables (including interactions)
- Goal: Fit parsimonious model that explains variation in Y with small set of predictors
- Automated procedures
 - Backward elimination (top down)
 - Forward selection (bottom up)
 - Best subset (among all exponentially many)
 - Stepwise regression (combines forward/backward)

Backward Elimination

- Select significance level to stay in model
- Fit full model with all possible predictors
- Consider predictor with lowest t-statistic (highest p-value)
 - If $p > \text{sign. level}$, remove predictor and fit model without
 - If $p \geq \text{sign. level}$, stop and keep current model
- Continue until all predictors have p-values below sign. level

Model Specification

- Specification: Process of developing a regression model
→ Consists selecting appropriate functional form for the model and choosing which variables to include
- Non-linear models are challenging
 - Quadratic models: $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2^2 + \varepsilon$ use $z_2 = x_2^2$
 - Models with interaction terms $y = \beta_0 + \beta_1 x_1 x_2$ use $z_1 = x_1 x_2$
 - Exponential terms $y = \alpha x^\beta \varepsilon$ can be transformed using the logarithm to
 $\ln(y) = \ln(\alpha) + \beta \ln(x) + \ln(\varepsilon)$

3. MULTI-LINEAR REGRESSION

→ Gauss-Markov Assumptions

3.1 Linearity

When it doesn't hold

- Transform either X or Y or both variables
- Polynomial regression
- Non-linear regression, eg with constant c
- Piecewise linear regression

Outliers

- Unusually small or large
- Reasons: Error/ point doesn't belong to sample/ valid

3.2 No multicollinearity of predictors

Multicollinearity

- Rank of data matrix X is p, number of columns
- $p < n$, the number of observations
- No exact linear relation among independent variable: $\text{rank}(X)=p$
- Basic check: Calculate correlation coefficient for each pair of predictor variables
 - Large correlation \rightarrow problem
 - Large: Greater than correlation between predictors and response
 - Possible that pairwise correlations small, and linear dependence exists among three/more variables \rightarrow Use variance inflation factor

Variance Inflation Factors

- $\text{VIF} = \frac{1}{1 - R_k^2}$ where R_k^2 is value when predictor in question (k) is set as dependent variable
- Eg if $\text{VIF}=10 \rightarrow R_k^2 = 90\%$ meaning 90% of the variance in predictor in question can be explained by other independent variables
- Rule of thumb: Remove variables with $\text{VIF} > 10$

Collinearity

- If one/more variables have big VIF's, regression is called collinear
- Caused by one/more variables being almost linear combinations of others
- Sometimes indicated by high correlations between IVs
- Results in imprecise estimation of coefficients
- Standard errors are high, so t-statistics small, variables non-significant

Non-Significance

- If variable has non-significant t-value, then
 - Variable not related to response \rightarrow Small t-value, small VIF, small correlation with response
 - Variable related to response, but not required in regression bc strongly related to third variable (don't need both) \rightarrow Small t-value, big VIF, big correlation with response
- Remedy
 - Drop one/more variable from model
 - Breaks linear relationship between the variables
 - Leads to problem of subset selection

3.3 Homoscedasticity

- Heteroscedasticity: No constant variance
 - \rightarrow Check with Breusch-Pagan test or White test
- Homoscedasticity: Constant variance (spread of the data points doesn't change much)

3.4 No autocorrelation

Autocorrelation

- Assumption: No autocorrelation between ith and jth residual terms
- No pattern should be observed if errors are independent
- Can be detected by graphing the residuals against time, or Durbin-Watson statistic

- DW to test for first order autocorrelation

$$DW = \frac{\sum_{i=2}^n (e_i - e_{i-1})^2}{\sum_{i=1}^n e_i^2}$$

→ [0,4], for no serial correlation value [1.5-2.5] is expected

Modeling Seasonality

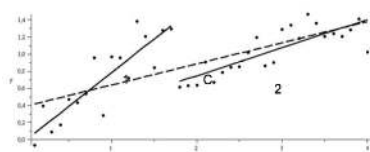
- Uses regression to estimate both trend and additive seasonal indexes
 - Create dummy variables which indicate season
 - Regress on time and seasonal variables
 - Use multiple regression model to forecast
- For any season (eg season 1), create column with 1 for time periods which are season 1, and zero for others → season – 1 dummy variables required
- Model (for quarterly data)

$$y_t = \beta_0 + \beta_1 t + \beta_2 Q_1 + \beta_3 Q_2 + \beta_4 Q_3$$

→ Additive model, allows to test for seasonality

Testing for Structural Breaks in Time Series

- Tests to compare regression models eg encompassing tests or Chow test
- Null Hypotheses Chow test: Coefficients of both models 1 and 2 are the same than those of combined model C
 - n: Number of observations in group, p: Number of parameters
 - Test follows F-distribution with p parameters and n1+n2-2p degrees of freed.



$$\frac{RSS_C - (RSS_1 + RSS_2))/p}{(RSS_1 + RSS_2)/(n_1 + n_2 - 2p)}$$

3.5 Exogeneity of independent variables

Omitted Variables

- Example: Acceptance rate of men and women in college: Men 55%, women 44% accepted
- Now broken down by type of school: Computer science (M 15%, W 20%), School of Management (M 75%, W 80%)
- Exactly opposite phenomenon → No discrimination
- Explanations
 - Women rather applied to schools with lower acceptance rates
 - Example for Simpson's paradox: When the omitted (confounding) variable (here type of school) is ignored the data seem to suggest discrimination

Panel Data vs. Cross-Section Data

- CSD: Data observing many subjects at same point of time or without regard differences in time → Might be omitted variables describing characteristic of individuals
- PD (longitudinal data set): Repeated observations on same units
 - Overcome problem of omitted variables bias caused by unobserved heterogeneity
 - Balanced panel: Every unit is surveyed in every time period
 - Unbalanced: Some individuals haven't been recorded in some period

Omitted Variable Bias in Panel Data

- Endogeneity given when IV is correlated with error term and covariance != 0
 - GM-Assumptions state that error term is uncorrelated with the regressors
 - Reason for endo: Relevant variables omitted from model

- Eg: Enthusiasm/ Willingness to take risks of an individual in panel describe unobserved hetero
- Various techniques to address endogeneity panel data

Treatment of Individual Effects

→ Two options

- Fixed effects: Assume λ_i are constants (there is endo)
→ Individual-specific effects correlated to other covariates
- Random effects: Assume λ_i are drawn independently from some probability distribution
→ Individual-specific effects uncorrelated to other covariates
→ Hausman test helps to decide whether on one or the other

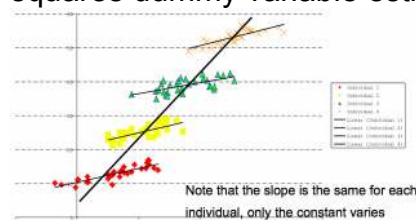
Fixed Effect Model

Treat λ_i as constant for each individual

$$y_{it} = (\beta_0 + \lambda_i) + \beta_1 x_{1it} + \beta_2 x_{2it} + \dots + \beta_p x_{pit} + \varepsilon_{it}$$

λ_i is part of constant, but varies by individual i

→ Various estimators for fixed effect models: First differences, within, between, least squares dummy variable estimator



First-Differences Estimator

Eliminating unobserved heterogeneity by taking first differences

$$y_{it} - y_{it-1} = \beta_0 + \lambda_i + \beta_1 x_{1it} + \beta_2 x_{2it} + \dots + \beta_p x_{pit} + \varepsilon_{it} \\ - \beta_0 - \lambda_i - \beta_1 x_{1it-1} - \beta_2 x_{2it-1} - \dots - \beta_p x_{pit-1} - \varepsilon_{it-1}$$

Lag one period and subtract

Constant and individual effects eliminated

Transformed equation

$$\Delta y_{it} = \beta_1 \Delta x_{1it} + \beta_2 \Delta x_{2it} + \dots + \beta_p \Delta x_{pit} + \Delta \varepsilon_{it}$$

Alternative Estimators

- Within estimator (for more than 2 periods)
→ Take deviations from individual means and apply least squares
 $y_{it} - \bar{y}_i = \beta_1 (x_{1it} - \bar{x}_{1i}) + \dots + \beta_p (x_{pit} - \bar{x}_{pi}) + (\varepsilon_{it} - \bar{\varepsilon}_i)$
→ Relies on variations within individuals rather than between
- Between estimator
→ Uses only info on individual means
 $\bar{y}_i = \beta_1 \bar{x}_{1i} + \dots + \beta_p \bar{x}_{pi} + \bar{\varepsilon}_i$
- Least squares dummy variable estimator
→ Uses dummy variable for each individual

Random Effects Model

- Fixed effect assumption: Individual specific effect is correlated with IVs

- Random effects assumption: Individual specific effects are uncorrelated with IVs

Original equation

$$y_{it} = \beta_0 + \beta_1 x_{1it} + \beta_2 x_{2it} + \dots + \beta_p x_{pit} + \varepsilon_{it}$$

$$y_{it} = \beta_0 + \beta_1 x_{1it} + \beta_2 x_{2it} + \dots + \beta_p x_{pit} + \lambda_i + u_{it}$$

λ_i is part of error term in random effects models

Error Components Model

$$y_{it} = \beta_0 + \beta_1 x_{1it} + \beta_2 x_{2it} + \dots + \beta_k x_{kit} + \varepsilon_{it}$$

Explanatory variables

$$\varepsilon_{it} = \lambda_i + u_{it}$$

Normally distributed error
 $u_{it} \sim N(0, \sigma_u^2)$

Constant across individuals
 $E(u_{it}) = E(\lambda_i) = 0;$
 $E(x_{pit} \lambda_i) = 0$ for all p, t, i

Composite error term

Random effects models also require special estimators.

3.6 Expected value of residual vector is 0

$$E(\varepsilon) = E \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

4. LOGISTIC AND POISSON REGRESSION

4.1 Logistic Regressions

- Logit models extend linear models to treat dichotomous and categorical target variables → Discrete choice modeling
- Application (Eg): Customers choose to fly or not:
Choose to fly iff $U_{fly} \geq 0$
 $-U_{fly} = \beta_0 + \beta_1 Cost + \beta_2 Time + \beta_3 Income + \varepsilon$

Linear Probability Model

- In the OLS regression:
 $Y_i = \beta_0 + \beta_1 X_{1i} + \varepsilon$; where $Y = \{0, 1\}$
- Predicted probabilities of linear model can be greater than 1 or less than 0
- ε not normally distributed because Y takes only two values
- Error terms are heteroscedastic

Linear Regression Model

- Logit model solves problems of linear model

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

$p(X)$ is the probability that the event Y occurs given X , $\Pr[Y = 1|X]$

$\frac{p(X)}{1-p(X)}$ describes the "odds"

– The 20% probability of winning describes odds of .20/.80 = .25

– A 50% probability of winning leads to odds of 1

$\ln \left(\frac{p(x)}{1-p(x)} \right)$ is the *log odds*, or "logit"

– $p = 0.50$, then logit = 0

– $p = 0.70$, then logit = 0.84

– $p = 0.30$, then logit = -0.84

Logistic Function

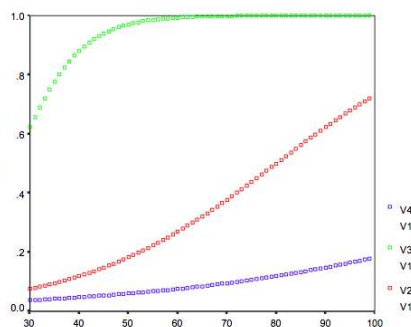
- The logistic function $\Pr[Y|X]$ constraints the estimated probabilities to lie between 0 and 1 ($0 \leq \Pr[Y|X] \leq 1$)

$$\Pr[Y|X] = p(X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

- $\Pr[Y|X]$ is the estimated probability that the i^{th} case is in a category and $\beta_0 + \beta_1 X$ is the regular linear regression equation
- This means that the probability of a success ($Y = 1$) given the predictor variable (X) is a non-linear function, specifically a logistic function
 - if you let $\beta_0 + \beta_1 X = 0$, then $p(X) = .50$
 - as $\beta_0 + \beta_1 X$ gets really big, $p(X)$ approaches 1
 - as $\beta_0 + \beta_1 X$ gets really small, $p(X)$ approaches 0

- The values in the regression equation β_1 and β_0 have slightly different meanings
 - β_0 : The regression constant (moves curve left and right)
 - β_1 : The regression slope (steepness of curve)
 - $\frac{\beta_0}{\beta_1}$: Threshold, where probability of success = 0.5
- Fixed regression constant, different slopes

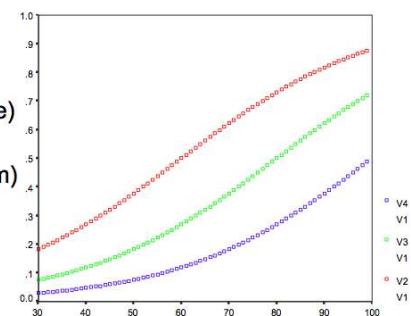
- v3: $\beta_0 = -4.00$
 $\beta_1 = 0.15$ (top)
- v2: $\beta_0 = -4.00$
 $\beta_1 = 0.05$ (middle)
- v4: $\beta_0 = -4.00$
 $\beta_1 = 0.025$ (bottom)



$p(X)$	$\frac{p(X)}{1-p(X)}$
0,1	0,1111
0,2	0,2500
0,3	0,4286
0,4	0,6667
0,5	1,0000
0,6	1,5000
0,7	2,3333
0,8	4,0000
0,9	9,0000
1	INF

- Constant slopes with different regression constants

- v2: $\beta_0 = -3.00$
 $\beta_1 = 0.05$ (top)
- v3: $\beta_0 = -4.00$
 $\beta_1 = 0.05$ (middle)
- v4: $\beta_0 = -5.00$
 $\beta_1 = 0.05$ (bottom)



$p(X)$	$\frac{p(X)}{1-p(X)}$	Logit
0	0	$-\infty$
0,1	0,11	-2,20
0,2	0,25	-1,39
0,3	0,43	-0,85
0,4	0,67	-0,41
0,5	1,00	0,00
0,6	1,50	0,41
0,7	2,33	0,85
0,8	4,00	1,39
0,9	9,00	2,20
1	∞	∞

Odds and Logit

- Logistic regression equation can be written in terms of an odds of success

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

- Odds: $[0, \infty]$
- If left side is
 - less than 1: Less than 0.5 probability
 - greater than 1: Greater than 0.5 probability

The Logit

- Writing equation in terms of logits (log-odds)

$$\ln\left(\frac{\Pr[Y = 1|X]}{1 - \Pr[Y = 1|X]}\right) = \ln\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X$$

- Probability: $[0, 1]$
- Log-odds are linear function of the predictors
- Logit now $[-\infty, +\infty]$ (as dependent variable of linear regression)
- Regression coefficients go back to old interpretation (kind of)
- Amount the logit (log-odds) changes, with a one unit change in X

Estimating the Coefficients of a Logistic Regression

- MLE is statistical method for estimating the coefficients of a model
- Likelihood function (L) measures probability of observing the particular set of dependent variable values that occur in the sample
- MLE involves finding coefficients that make the log of the likelihood function ($LL < 0$) as large as possible

Likelihood Function for Logit Model

For the logit model we specify

$$\Pr(Y_i = 1) = F(\beta_0 + \beta_1 X_{1i}) = \frac{e^{(\beta_0 + \beta_1 X_{1i})}}{1 + e^{(\beta_0 + \beta_1 X_{1i})}} = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_{1i})}}$$

- $\Pr(Y_i = 1) \rightarrow 0$ as $\beta_0 + \beta_1 X_{1i} \rightarrow -\infty$
- $\Pr(Y_i = 1) \rightarrow 1$ as $\beta_0 + \beta_1 X_{1i} \rightarrow \infty$
- Probabilities from the logit model will be between 0 and 1

$$L = \prod_{i=1}^n \left(\frac{1}{1 + e^{-(\beta_0 + \beta_1 X_{1i})}} \right)^{Y_i} \left(1 - \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_{1i})}} \right)^{1-Y_i} \rightarrow \max$$

$$L = p_1^{Y_1} (1 - p_1)^{1-Y_1} p_2^{Y_2} (1 - p_2)^{1-Y_2} \dots p_n^{Y_n} (1 - p_n)^{1-Y_n}$$

← sequence of Bernoulli trials

$$= \prod_{i=1}^n p_i^{Y_i} (1 - p_i)^{1-Y_i}$$

and

$$\ln(L) = \sum_{i=1}^n Y_i \ln p_i + (1 - Y_i) \ln(1 - p_i)$$

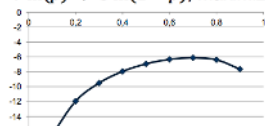
$$\text{if } p_i = F(\beta_0 + \beta_1 X_{1i}) = \frac{e^{\beta_0 + \beta_1 X_{1i}}}{1 + e^{\beta_0 + \beta_1 X_{1i}}} \quad \leftarrow \text{i.e., the cumulative distribution function } F() \text{ might be this logistic function}$$

$$LL = \ln(L) = \sum_{i=1}^n \{Y_i \ln F(\beta_0 + \beta_1 X_{1i}) + (1 - Y_i) \ln(1 - F(\beta_0 + \beta_1 X_{1i}))\}$$

The log-likelihood functions are now differentiated wrt. the coefficients and the partial derivatives are set to zero and solved for β_1 (using numerical methods such as Newton-Raphson as there is no closed-form solution.)

Example of MLE

- Suppose 10 individuals make travel choices between auto (A) and public transit (T).
 - All travelers are assumed to possess identical attributes (unrealistic), and so the probabilities are not functions of β_i but simply a function of p , the probability p of choosing auto.
- $$-L = p^x (1 - p)^{n-x} = p^7 (1 - p)^3$$
- $$-LL = \ln(L) = 7 \ln(p) + 3 \ln(1 - p), \text{ maximized at } 0.7$$



Goodness of Fit

- Saturated model
→ Assumes each data point has its own parameters (have n parameters to estimate) → Likelihood of the model = 1
- Null model
→ Assumes one parameter (intercept) for all of data points

- Fitted model
→ Assumes you can explain data points with p parameters and intercept term (p+1 parameters)
- Null deviance: $-2\ln(L(\text{null}))$
→ How much is explained by model with only intercept
- Residual deviance: $-2\ln(L(\text{fitted}))$
→ Small values mean fitted model explains data well
- Likelihood ratio test

$$D = -2\ln\left(\frac{L(\text{null})}{L(\text{fitted})}\right) = -2(LL(\text{null}) - LL(\text{fitted}))$$
 - Logarithm of this likelihood ratio (ratio of fitted model to saturated model) will produce negative value → Need for negative sign
 - D follows χ^2 distribution (the smaller, the better)
 - Non-significant χ^2 values indicate that significant amount of variance is unexplained
 - Test can also be used to assess individual predictors
- Wald test
 - Analogous to t-test for linear regression
 - Used to test statistical significance of each coefficient in the model hypothesis that $\beta_i = 0$

McFadden R^2 as example of Pseudo R^2

$$R^2_{\text{McFadden}} = 1 - \frac{LL(\text{fitted})}{LL(\text{null})}$$

- If full model does much better than just a constant, in discrete-choice model value is close to 1
- If full model doesn't explain much, value will be close to 0
- Typically values are lower than those of R^2 in linear regression and need to be interpreted with care → >0.2 acceptable, >0.4 ok

Calculating Error Rate from Logistic Regression

- If estimated $p(x) \geq 0.5$: Event is expected to occur (otherwise: Not occur)
- Assigning these probabilities 0s and 1s and comparing these to actual 0s and 1s, the % correct Yes, % correct No and overall % correct scores are calculated

Interpreting Coefficients of Logistic Regression

- If $\beta_1 < 0$ then an increase in $X_1 \Rightarrow (0 < e^{\beta_1} < 1)$
– then odds go down
- If $\beta_1 > 0$ then an increase in $X_1 \Rightarrow (e^{\beta_1} > 1)$
– then odds go up
- Always check for significance of the coefficients

Example Results: Campaign Response and Age

$$-\ln(p(x)/(1-p(x))) = \beta_0 + \beta_1 \text{ Age}$$

Variable	Estimated Coefficient	Standard Error
Age	0.135	0.036
Constant	-6.54	1.73

→ How to interpret value 0.135?

$$\ln(\text{odds response person \#2}) = \beta_0 + \beta_1(X_1 + 1) = \beta_0 + \beta_1(X_1) + \beta_1$$

$$\ln(\text{odds response person \#1}) = \beta_0 + \beta_1(X_1)$$

→ The difference is β_1 (which describes estimator here)

$$\text{So, } \beta_1 = \ln(\text{odds "response" person \#2})$$

$$- \ln(\text{odds "response" person \#1})$$

- "Reversing" a property of logs:

$$\ln\left(\frac{x}{y}\right) = \ln(x) - \ln(y)$$

$$\beta_1 = \ln\left(\frac{\text{odds_of_response_person\#2}}{\text{odds_of_response_person\#1}}\right)$$

- $\beta_1 = \ln(\text{odds ratio for person \#2 compared to person \# 1})$
 $= \ln(\text{odds ratio comparing two age groups who differ by one year in age})$
- So, $\beta_1 = \ln(\text{odds ratio})$, then we can get the estimated odds ratio, OR, by e^{β_1}
- So, in our example, $OR = e^{0.135} = 1.14$
- Example:
 - If we were to compare 2 people (or two groups of people) 60 years old and 59 years old respectively, the odds ratio for response of the 60-year-old to the 59-year-old is 1.14
- In fact, if we compared any two people (groups) who differed by year of age, older to younger, the odds ratio would be 1.14 . . .
 - 27 to 26 year olds
 - 54 to 53 year olds . . . etc . .

Multicollinearity and Irrelevant Variables

- Presence of MC won't lead to biased coefficients, but effect on standard errors
 - If variable which you think should be statistically significant is not, consult the correlation coefficients or VIF
 - If two variables are correlated at rate greater than 0.6 then try dropping least theoretically important of the two
- Inclusion of irrelevant variables can result in poor model fit → Consult your Wald statistics and remove irrelevant variables

Multiple Logistic Regression

- More than one independent variables

$$\ln\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n$$
- Interpretation of β_1 : Increase in log-odds for one unit increase in x_i with all other $x_j, j \neq i$ constant

$$\Pr[Y = 1|X] = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n)}}$$

Multinomial Logit Models

- Dependent variable (Y) is discrete variable that represents a choice/ category from a set of mutually exclusive choices/ categories (Eg brand selection, transportation mode selection)
- Model: Choice between $J > 2$ categories; Dependent variable $y = 1, 2, \dots, J$
- If characteristics that vary over alternatives (eg prices) the multinomial logit is called conditional logit
- Ordered logit models have ordinal dependent variables

Generalized Linear Models (GLM)

- Logit model is example
- GLMs are general class of linear models made of: Random, Systematic and Link Function
 - Random: Identifies dependent variable (Y) and its probability distribution
 - Systematic: Identifies set of explanatory variables (X_1, \dots, X_k)
 - Link Function: Identifies function of the mean that is linear function of explanatory variables

$$g(\mu) = \beta_0 + \beta_1 X_1 + \dots + \beta_k X_k$$

- Common Link Functions

Identity link (form used in *normal* regression models):

$$g(\mu) = \mu$$

Log link (used when μ cannot be negative as when data are *Poisson* counts):

$$g(\mu) = \log(\mu)$$

Logit link (used when μ is bounded between 0 and 1 as when data are binary):

$$g(\mu) = \log\left(\frac{\mu}{1-\mu}\right)$$

Count Variables as Dependent Variables

→ Many dependent variables are counts: Non-negative integers

- # Crimes person has committed
- # Children living in household,...
- Count variables can be modeled with OLS regression but
 - Linear models can yield negative predicted values
 - Count variables often highly skewed (eg # crimes: most people 0/ very low, few people very high) → Extreme skew violates normality assumption

4.2 Poisson Regressions

Count Models

- Two most common count models
 - Poisson regression model (log-linear model)
 - Negative binominal regression model
- Both assume observed count is distributed according to Poisson distribution
 - μ = expected count (and variance)
 - y = observed count

$$\Pr[y|\mu] = \frac{e^{-\mu} \mu^y}{y!}$$

Poisson Regression for Count Data

- Model log of μ as function of X
- Log form avoids negative values

$$\ln(\mu) = \sum_{j=1}^K \beta_j X_{ji}$$

- Can be written as

$$\mu = e^{\sum_{j=1}^K \beta_j X_{ji}}$$

- Distribution: Poisson (Restriction: $E(Y) = \text{Var}(Y)$)
- Link function

$$g(\mu) = \ln(\mu) = \beta_0 + \beta_1 X_1 + \dots + \beta_k X_k$$

$$\Rightarrow \mu(X_1, \dots, X_k) = e^{\beta_0 + \beta_1 X_1 + \dots + \beta_k X_k}$$

Interpreting Coefficients

- y is typically conceptualized as a rate (pos. coefficients → higher rate, negative → lower rate)
- Poisson models non-linear (like logit)
- Model has log form (like logit); Exponentiation aids interpretation
 - Exponentiated coefficients are multiplicative
 - Analogous to odds ratios (but called incidence rate ratios)

Example: Purchasing Decision

$$\ln(\mu) = -0.282 + (0.388)(X_{\text{Student}})$$

- No student

- $\ln(\mu) = -0.282 + (0.388)(0) = -0.282$

- $\mu = e^{-0.282} = 0.754$

- $\mu = 0.75$ tickets bought

- Student

- $\ln(\mu) = -0.282 + (0.388)(1) = 0.106$

- $\mu = e^{0.106} = 1.112$

- $\mu = 1.11$ tickets bought

Interpreting Coefficients

- Exponentiated coefficients: Indicate effect of unit change of X on rate
 - $e^{\beta} = 2.0$: Rate doubles for each unit change in X
 - $e^{\beta} = 0.5$: Rate drops by half for each...
- Recall: Exponentiated coeffs are multiplicative → If $e^{\beta} = 5.0$, a 2-point change in X isn't 10, it is $5 \times 5 = 25$
- Can be converted to % change → Formula: $(e^{\beta} - 1) \times 100\%$

Poisson Model Assumptions

- $E(Y) = \text{Var}(Y) = \mu \rightarrow$ Often not met in real data (Variance greater than $\mu \rightarrow$ Overdispersion)
- Consequence of overdispersion
 - Standard errors underestimated
 - Potential for overconfidence in results; Rejecting Null Hypotheses when shouldn't
- Negative binomial regression as alternative to Poisson regression

Zero-Inflation

- If outcome variable has many zeros it tends to be highly skewed → Negative binomial regressions help
- But if LOTS of zeros, even that not sufficient (Eg # violent crimes committed by a person a year)
- Logic of zero-inflated models: Assume 2 types of groups in sample
 - A: Always 0
 - $\sim A$: Non-zero chance of positive count variable (prob. is variable, but $\neq 0$)
 - Use logit to model group membership
 - Use Poisson or NB regression to model counts for those in $\sim A$
 - Compute probabilities based on those results

5. NAIVE BAYES AND BAYESIAN NETWORKS

Formal Definition of Classification

- Classification problem: Define a mapping where each item/tuple is assigned to one class → Logistic regression can be used for it
- Prediction: Similar, but usually implies mapping to numeric values instead of a class

Algorithms for Classification

- (Logistic) Regression
- Rudimentary Rules (eg 1R)
- Statistical Modeling (eg Naïve Bayes)
- Decision Trees: Divide and Conquer,...

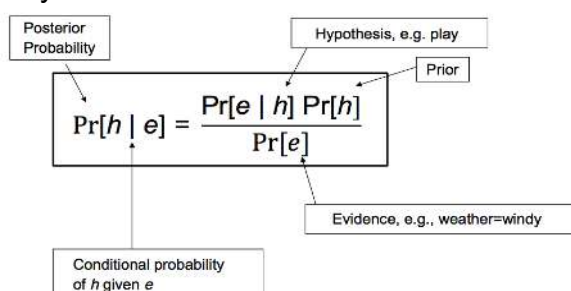
1-Rule (1R)

- Generate one level decision tree
- One attribute – easy to explain
- Rules testing single attribute → Classify according to frequency in training data → Evaluate error rate for each attribute → Choose best attribute

Naïve Bayes Classifier

- Allows attributes to contribute equally (not only single attribute)
- Assumptions
 - All attributes equally important
 - All attributes independent (knowledge about value of attribute doesn't tell us anything about value of another attribute)

Bayes Theorem



- $\Pr[e]$: Prior/ unconditional probability that proposition e is true (Eg: $\Pr[e]=0.1$ means there is 10% chance that given customer is a high credit risk)
- Missing values
 - Training: Instance not included in frequency count for attribute value-class combination
 - Classification: Attribute will be omitted from calculation
- Dealing with numeric attributes
 - Assumption: Attributes have normal probability distribution
 - Probability density function

The sample mean μ :

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i$$

The standard deviation σ :

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \mu)^2}$$

The density function $f(x)$:

$$f(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- Numeric data: Unknown distribution
 - Consider random variable X whose distribution $f(X)$ is unknown, but sample with non-uniform distribution $\{x_1, x_2, \dots, x_n\}$
 - Want to derive function $f(x)$ such that
 - (1) $f(x)$ is a probability density function, i.e.

$$\int f(x) dx = 1$$

- (2) $f(x)$ is a smooth approximation of the data points in X
- (3) $f(x)$ can be used to estimate values x^* which are not in $\{x_1, x_2, \dots, x_n\}$

- Rosenblatt-Parzen Kernel-Density-Estimator

$$f(x) = \frac{1}{n} \sum_{i=1}^n K(x - x_i, h)$$

Where

$$K(t, h) = \frac{1}{\sqrt{2\pi}h} e^{-\frac{1}{2}(\frac{t}{h})^2}$$

- Discussion
 - Works well even if independence assumption is clearly violated
 - Adding too many redundant attributes → Problems
 - Time complexity

Bayesian (Belief) Networks: Multiple Variables with Dependency

- Describe conditional independence among subsets of attributes: Combining prior knowledge about dependencies among variables with observed training data
- Graphical representation: Directed acyclic graph (DAG) → One node for each attribute
 - Overall probability distribution factorized into component distributions
 - Nodes hold component distributions (conditional distributions)
- Probability laws

Conditional independence

$$-\Pr[h|e_1, e_2] = \Pr[h|e_2]$$

Chain rule

$$-\Pr[e_1, e_2, \dots, e_n] = \prod_{i=1, \dots, n} \Pr[e_i | e_{i-1}, \dots, e_1]$$

$$-\Pr[A, B, C, D, E] = \Pr[A] \Pr[B|A] \Pr[C|A, B] \Pr[D|A, B, C] \Pr[E|A, B, C, D]$$

- The joint distribution is independent of the ordering

BN assumption

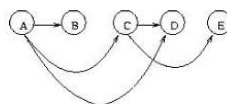
$$-\Pr[e_1, e_2, \dots, e_n] =$$

$$\prod_{i=1, \dots, n} \Pr[e_i | e_{i-1}, \dots, e_1] = \prod_{i=1, \dots, n} \Pr[e_i | \text{Parents}(e_i)]$$

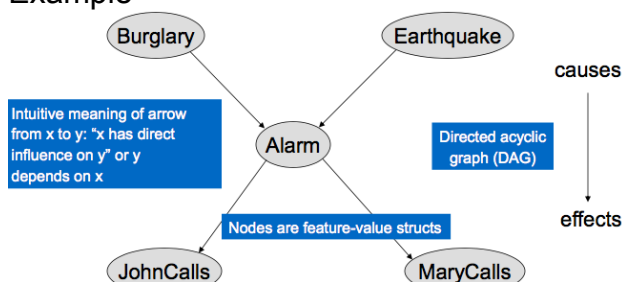
$$-\Pr[A, B, C, D, E] =$$

$$\Pr[A] \Pr[B|A] \Pr[C|A, B] \Pr[D|A, B, C] \Pr[E|A, B, C, D]$$

$$= \Pr[A] \Pr[B|A] \Pr[C|A] \Pr[D|A, C] \Pr[E|C]$$

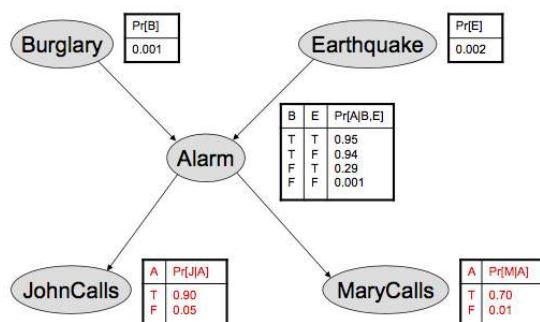


- Example



→ Alarm system against burglary, but can set off by earthquake
 → Mary and John call you if alarm (but not guaranteed) and sometimes just call to chat

- Probability tables



- Inference
 - May want to infer probability of an event, given observations about subset of other variables
 - Need to consider evidence and topology of the network
- Inference rules (example)
 - If alarm isn't observed, B and M calls are dependent: My knowing that B has taken place increases my belief on M; My knowing that M called increases my belief von B
 - If alarm is observe, B and M are conditionally independent: If I already know that alarm went off: My further knowing that burglary has taken place wouldn't increase my belief on Marys call; My further knowing that Mary called wouldn't increase my belief von burglary
- Learning Bayes nets
 - Method for evaluating goodness for given network
 - Measures maximize the joint probability of training data given the network
 - Summarize Log-Likelihood of training data based on network
 - Method for searching through space of possible networks: Amounts to searching through sets of edged because node are fixed
- Discussion
 - Handle dependencies among attributes
 - Complex (Network structure given or not)
 - Subject of much current research

6. DECISION TREE CLASSIFIERS

6.1 Choosing a splitting attribute in decision trees

Regression Trees

- Leaf nodes are numbers
- High accuracy
- Large and possibly awkward

Model Trees

- Leaf nodes are linear models

Decision Trees

- Internal node: Test on an attribute
- Branch: Outcome of the test
- Leaf node: Class label/ class label distribution
- At each node: One attribute chosen to split training examples into classes as much as possible
- New case: Classified by following a matching path to leaf node

Building Decision Trees

- Top-down
 - At start, all training examples at the root
 - Partition examples recursively by choosing one attribute each time
- Bottom-up
 - Remove subtrees or branches, in bottom-up manner, to improve estimated accuracy on new cases (if overfitting)

Choosing the Splitting Attribute

- Goodness function used to evaluate available attributes at each node
- Typical goodness functions
 - Information gain
 - Information gain ratio
 - Gini index

Criterion for Attribute Selection

- Best attribute: One which results in smallest tree (Purest node!)
- Information gain increases with average purity of subsets
- Choose attribute that results in greatest information gain

Computing Information

- Info is measured in bits
- Entropy: Gives info required to predict an event (eg play yes/no) in bits
- Formula for computing information entropy

$$\text{entropy}(p_1, p_2, \dots, p_n) = -p_1 \log_2 p_1 - p_2 \log_2 p_2 \dots - p_n \log_2 p_n$$

Computing information Gain

- Info gain = (Info before split) – (info after split)

Wish List for a Purity Measure

- Properties required from purity measure
 - If node pure \rightarrow Measure = 0
 - If impurity maximal (all classes equally likely) \rightarrow Measure = Max (1 for Boolean values)
 - Multistage property: $\text{info}[2,3,4] = \text{info}[2,7] + 7/9 \text{info}[3,4]$
- Entropy is function that satisfies these properties

$$\text{entropy}(S) = \sum_{i=1}^n -p_i \log_2 p_i$$

(scales from 0 to $\max \log_2 n$)

Annotations:

- Number of classes $\rightarrow n$
- Training data (instances) $\rightarrow S$
- Probability of S being classified to i $\rightarrow p_i$

- Entropy describes randomness of a system ($=0 \rightarrow$ perfectly ordered system)

Expected Information Gain

$$\text{gain}(S, a) = \text{entropy}(S) - \sum_{v \in \text{Values}(a)} \frac{|S_v|}{|S|} \text{entropy}(S_v)$$

$S_v = \{s \in S : a(s) = v\}$

All possible values for attribute a

$\text{gain}(S, a)$ is the information gained adding a sub-tree
(Reduction in number of bits needed to classify an instance)

Highly-Branching Attributes

- Problem: Attributes with large number of values (extreme case: ID code)
- Info gain biased towards choosing attributes with large number of values (more likely to be pure if large number of values)
- May result in overfitting

- Modification on info gain reduces this bias (takes number and size of branches into account)
→ Takes intrinsic info of a split into account (how much info do we need to tell which branch an instance belongs to)
- Important of attribute decreases as intrinsic info grows

Gain Ratio: Problem

→ May overcompensate

- Max choose attribute just because its intrinsic info is very low
- Standard fix
 - First, only attributes with greater than average info gain
 - Then, compare gain ratio

Gini Index for 2 Attribute Values

- Eg 2 classes (Pos and Neg), dataset S with p pos elements and n neg elements → Frequency:

$$P = p / (p + n)$$

$$N = n / (p + n)$$

$$Gini(S) = 1 - P^2 - N^2 \quad \in [0,0.5]$$

If dataset S is split into S_1, S_2 then

$$Gini_{split}(S_1, S_2) = (p_1 + n_1)/(p + n) \cdot Gini(S_1) + (p_2 + n_2)/(p + n) \cdot Gini(S_2)$$

Example

$$Gini(p) = 1 - \sum_j p_j^2$$

Numbers of Cases		Proportion of Cases				Gini Index
A	B	A	B			
		p_A	p_B	p_A^2	p_B^2	$1 - p_A^2 - p_B^2$
400	400	0.5	0.5	0.25	0.25	0.5

Select the split that decreases the Gini Index most. This is done over all possible places for a split and all possible variables to split.

Number of Cases		Proportion of Cases				Gini Index	Info required
A	B	A	B				
		p_A	p_B	p_A^2	p_B^2	$1 - p_A^2 - p_B^2$	
300	100	0.75	0.25	0.5625	0.0625	0.375	0.1875
100	300	0.25	0.75	0.0625	0.5625	0.375	0.1875
						Total	0.375
200	400	0.33	0.67	0.1111	0.4444	0.4444	0.3333
200	0	1	0	1	0	0	0
						Total	0.3333

0.5*Gini(i)

0.75*Gini(i)

Complexity of Basix Algorithm (DT)

- m attributes, n instances
- Depth of tree $O(\log n)$
- $O(n \log n)$ work for single attribute over entire tree
- Total cost: $O(mn \log n)$

Scalability of DT Algorithms

- Need to design for large amount of data
- Large number of attributes leads to larger tree and takes long time

6.2 Relational rules

Instances of the Shapes Problem

Width	Height	Sides	Class
2	4	4	Standing
3	6	4	Standing
4	3	4	Lying
7	8	3	Standing
7	6	3	Lying
2	9	4	Standing
9	1	4	Lying
10	2	3	Lying

If width ≥ 3.5 and height < 7.0 then lying
 If height ≥ 3.5 then standing

Relational Rules

If width $>$ height then lying
 If height $>$ width then standing

- Rules comparing attributes to constants: Propositional rules
- Relational rules: More expressive
 - Define relations
 - Most DM techniques don't consider relational rules
- Can introduce additional attributes, describing if width $>$ height \rightarrow Allows using conventional propositional learners

Propositional Logic

- Decision trees can represent any function in propositional logic
 - A, B, C: Propositional variables
 - And, or, not, \Rightarrow (implies), \Leftrightarrow (equivalent): Connectives
- Proposition is statement that is either true or false
- DT example for propositional learner

6.3 Numeric attributes

C4.5 An Industrial-Strength Algorithm

- Useful algorithm must
 - Permit numeric attributes
 - Allow missing values
 - Be robust in presence of noise
- Basic algorithm needs to be extended to fulfill these requirements

Numeric Attributes

- Unlike nominal attributes, every attribute has many possible split points (Standard method: Binary split, eg temp < 45)
- Solution
 - Evaluate info gain for every possible split point of attribute
 - Choose best split point
 - Info gain for best split is highest info gain for attribute
- Numerical attributes can be used several times in DT, nominal only once

Binary Splits on Numeric Attributes

- Splitting (multi-way) on nominal attribute exhausts all info in that attribute \rightarrow Nominal attribute is tested once on any path in tree
- \Leftrightarrow Binary splits for numeric attributes: May be tested several times \rightarrow Disadvantage: Tree hard to read
- Remedy
 - Pre-discretize numeric attributes or
 - Allow for multi-way splits instead of binary ones using the info gain criterion

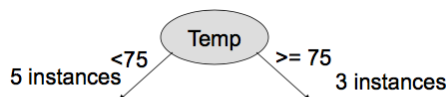
6.4 Missing values

Handling Missing Values

- Ignore instances with missing values
- Ignore attributes with missing values
- Treat missing values as another nominal value
- Estimate missing values

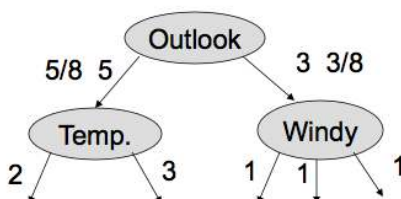
Handling Missing Values / Classification

- Follow the leader
→ Instance with missing value for testes attribute (temp) is sent down the branch with most instances



Instance included on the left branch

- Partition the instance
 - Branches show # instances
 - Send down parts in instance (eg 3/8 on windy and 5/8 on sunny) proportional to # training instances
 - Resulting leaf nodes get weighted in result



Overfitting

- 2 sources of abnormalities: Noise (randomness) and outliers (measurement errors)
- Chasing every abnormality causes overfitting
 - DT gets too large and complex
 - Good accuracy in training set, poor one on test set
 - Doesn't generalize to new data any more
- Solution: Prune the tree

7. DATA PREPARATION

7.1 Pruning

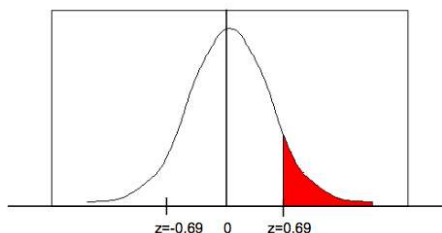
- Prepruning: Decide a priori when to stop creating subtrees
- Postpruning: Simplify existing decision tree
→ Subtree replacement: Replace subtree with single leaf node

When to Prune?

- Replace node if its error rate is less than combined rates of its children
- Error on the training data is NOT useful estimator
- Use hold-out set for pruning (reduced-error pruning) → Limits data you can use for training
- C4.5 method
 - Derive CI from training data
 - Use heuristic limit for error rate, derived from CI for pruning
 - Shaky statistical assumptions (bc based on training data), but works well

Confidence Limits

- Confidence limits c for standard normal distribution with 0 mean and variance 1



$c = \Pr[X \geq z]$	z
0.1%	3.09
0.5%	2.58
1%	2.33
5%	1.65
10%	1.28
20%	0.84
25%	0.68
40%	0.25

- 25% probability of X being > 0.68
 $\Pr[-0.68 \leq X \leq 0.68]$
- To use this f must be reduced to have 0 mean and unit variance
- Transforming f

Standardized value for observed error rate f :

$$\frac{\bar{X} - \mu}{\sigma / \sqrt{n}} = \frac{f - p}{\sqrt{p(1-p)/n}}$$

(Standardization: subtract mean and divide by the standard deviation)

Binomial conf. interval: $\Pr\left[\frac{f - p}{\sqrt{p(1-p)/n}} > z\right] = c$

Solving for p provides limits for the confidence factor c :

$$p = \left(f + \frac{z^2}{2n} \pm z * \sqrt{\frac{f - \frac{f^2}{n} + \frac{z^2}{4n^2}}{n}} \right) / \left(1 + \frac{z^2}{n} \right)$$

You prune the tree stronger

- If c goes down $\Rightarrow z$ goes up and also p goes up
- If n goes down $\Rightarrow p$ goes up
- with p as an estimator for the error rate

C4.5 Method

- Error estimate e for a node (:= upper bound for CI)

$$e = p = \left(f + \frac{z^2}{2n} + z * \sqrt{\frac{f - \frac{f^2}{n} + \frac{z^2}{4n^2}}{n}} \right) / \left(1 + \frac{z^2}{n} \right)$$

- If confidence limit $c = 25\%$ then $z = 0.69$ (from Normal distribution)
- f is the error on the training data
- n is the number of instances covered by the node

- Even with positive infogain, e might increase as well
- Error estimate for subtree is weighted sum of error estimates for all its leaves

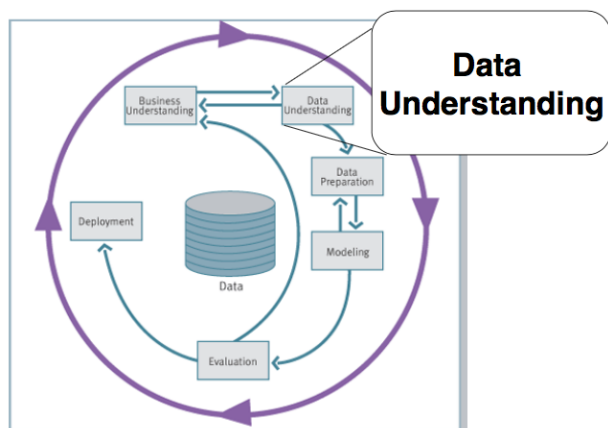
From Trees to Rules

- One rule for each leaf
- C4.5 rules: Greedily prune conditions from each rule if this reduces its estimated error
- Look at each class in turn and consider rules for that class
- Find good subset
- Remove rules (greedily) if this decreases error on training data

C4.5 and C4.5 rules: Summary

- C4.5 decision tree algorithm has 2 important parameters
 - Confidence value (default 25%): Lower values incur having pruning
 - Minimum number of instances in 2 most popular branches (default 2)
- Classification rules
 - Slow for large and noisy datasets
 - Commercial version C5.0 rules uses different pruning technique (Faster, more accurate)

7.2 Knowledge Discovery Process



Data Understanding

- # of instances (records)/ attributes/ targets
- Visualization
- Data summaries (Attribute means/ variation/ relationships)

Data Preparation

- Estimated to take 70-80% of the time and effort
- Cleaning: Missing values
- Conversion: Ordered to Numeric
 - Ordered attributes (eg grade) can be converted to numbers preserving natural order (eg A \rightarrow 4.0) \rightarrow Important to allow meaningful comparisons (eg grade > 3.5)
- Conversion: Nominal, Few Values
 - Multi-valued, unordered attributes with small # of values (eg color = red)
 - For each value v create binary flag variable C_v which is 1 if color = v, 0 otherwise

ID	Color	...
371	red	
433	yellow	

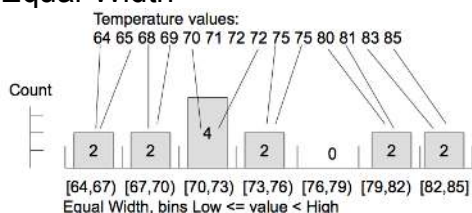
→

ID	C_red	C_orange	C_yellow	...
371	1	0	0	
433	0	0	1	

- Nominal, many values Ignore ID-like fields whose values are unique for each record

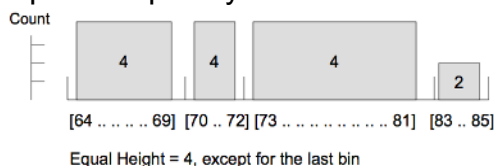
Data Cleaning: Discretization

- Reduces number of values for continuous attribute
- Why?
 - Some methods can only use nominal data
 - Helpful if data needs to be sorted frequently (eg when constructing DT)
 - Some methods that handle numerical attributes assume normal distribution which isn't always appropriate
- Useful for generating summary of the data
- Equal-Width



\rightarrow May produce clumping (if many observations in one/two bins and zero in the other ones) \rightarrow Lose information

- Equal-Frequency

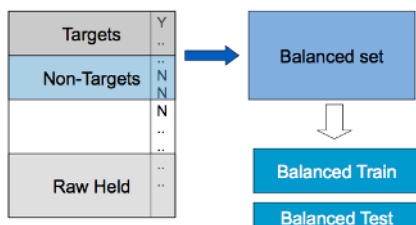


- Class Dependent

- Eg based on info gain of the class variable (see C4.5)
- | | | | | | | | | | | | | | |
|-----|----|-----|-----|-----|----|----|-----|-----|-----|----|-----|-----|----|
| 64 | 65 | 68 | 69 | 70 | 71 | 72 | 72 | 75 | 75 | 80 | 81 | 83 | 85 |
| Yes | No | Yes | Yes | Yes | No | No | Yes | Yes | Yes | No | Yes | Yes | No |
- Treating numerical attributes as nominal discards the potentially valuable ordering info
 - Alternative: Transform the k nominal values to k-1 binary attributes
 - The (i-1)th binary attribute indicates whether discretized attribute is < i

Unbalanced Target Distribution

- Sometimes, classes have very unequal frequency
- Similar situation with multiple classes
- Majority class classifier can be 97% correct, but useless
- Building balanced train sets



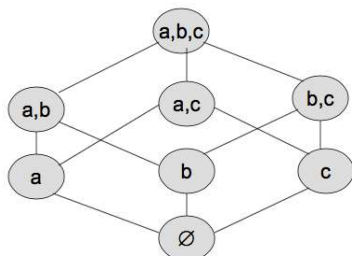
Attribute Selection

- If too many attributes: Select subset that is most relevant
- Remove redundant and/or irrelevant attributes
- Reasons
 - Simpler model (More transparent, easier to interpret)
 - Fast model induction
 - Accuracy?
 - Adding random attribute to a DT learner can decrease accuracy by 5-10%
 - Instance-based methods are particularly weak in presence of irrelevant attributes (compared with only few instances)

Attribute Selection Heuristics

- Stepwise forward selection
 - Start with empty attribute set
 - Add best of attributes (eg using entropy); Add best remaining attributes
 - Repeat. Take top k (certain threshold value)
- Stepwise backward selection
 - Start entire attribute set
 - Remove worst of attributes
 - Repeat until k are left
- Using entropy for attribute selection
 - Calculate info gain of each attribute
 - Select k attributes with highest info gain
- Experiences
 - Lead to local, not necessarily global optima, but perform well
 - BS performed better than FS

- FS leads to smaller attributes sets and easier models
- Actually it is search problem: Select subset of attribute giving most accurate model



Basic Approaches to Attribute Selection

- Remove attributes with no/ little variability
- Remove false predictors (leakers)

Causal Inference

- $Y \sim$ outcome (DV)
- $T \sim$ treatment indicator
- $X \sim$ Covariate (pretreatment)
→ What would have happened to those who received treatment, if they haven't received treatment (or vice versa)?
- Example: The true advertising lift

Individual treatment effect:

$$Y_{1i} - Y_{0i}$$

Average treatment effect:

$$E(Y_{1i} - Y_{0i})$$

Subgroup treatment effect:

$$E(Y_{1i} - Y_{0i} | X)$$

- 1) A customer has a high probability of buying a car (based on his attributes)
- 2) I show a customer a display ad on a web site (treatment)
- 3) The customer buys the car (outcome)
- 4) Actually 10% of those, who have seen the ad buy a car as compared to 1% in the group who has not seen a display ad.

Q) Is it because I showed the display ad?

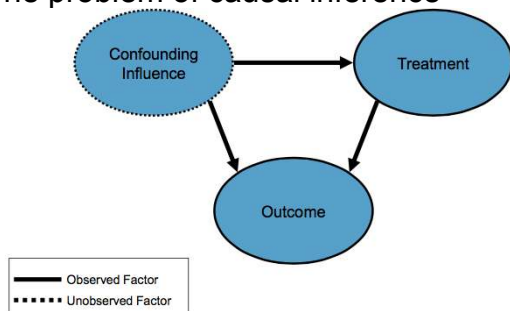
A) We'll never know – it is *counterfactual* – for the individual

This is a problem of causal inference

- Y_{1i} : Outcome of individual i given being treated
- Y_{0i} : Outcome of i given being control
- $\Delta_i = Y_{1i} - Y_{0i}$: Treatment effect on i

Sub.	X	Y_1	Y_0	Δ
A	40	15	10	5
B	30	13	8	5
C	30	13	8	5
D	20	9	4	5

- The problem of causal inference



Sampling of Data

- Selection bias: Results from method of collecting samples
- Randomized controlled trials (RCTs): Experiments, where each subject is randomly assigned to treated group or control group
 → Min selection bias and different comparison groups allow to determine any effects of treatment when compared with control group
- Observation studies: Inferences from sample to population where IV isn't under control of the researcher
 - Cross-sectional study: Data collection from population or representative subset, at one specific point in time
 - Cohort study/ Panel study: Form of longitudinal study (group patients is monitored over time)
 - Longitudinal study: Correlational study that involves repeated observations of same variables over long periods of time

Methods for Observational Studies

- Propensity Score Matching (PSM): Compares outcomes of similar units where only difference is treatment (discards rest)

First stage: regress treatment on observables

$$T_{it} = X_{it}\beta + \varepsilon_{it}$$

Second stage: form individual probabilities of treatment and save observations where there is overlap

$$\Pr(T_{it} = 1) = \Phi(-X_{it}\hat{\beta}) = \hat{p}_{it}$$

Third stage: compare outcomes of treated observations to similar non-treated observations. Less weight is given, the less the similarity.

Matching algorithm (e.g., nearest neighbour) iteratively finds the pair of subjects with the shortest distance. The goal is to balance the pretreatment covariates distribution.

- Fixed Effects: Eliminates alternative explanations that are fixed across units
- Regression
- Discontinuity: Subsample for which assignment to treatment is random
- Instrumental variables
- Control Function Approach

8. EVALUATION OF CLASSIFIERS AND LEARNING THEORY

8.1 Bias-Variance Tradeoff

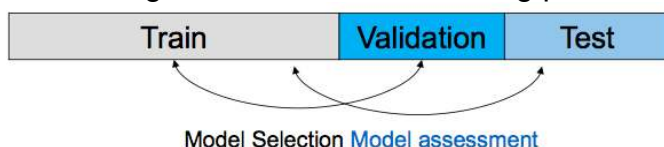
Supervised Learning

- Inferring a function from labeled training data

- Training: Given training set of labeled examples, estimate prediction function f by minimizing prediction error on training set
- Testing: Apply f to never before seen test example x and output predicted value

Model Selection and Model Assessment

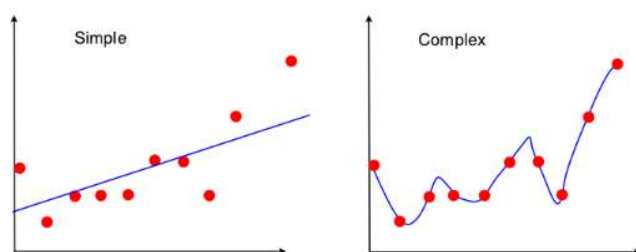
- MS: Estimating performances of different models to choose best one (min test error)
- MA: Having chosen model, estimating prediction error on new data



Holdout Procedure

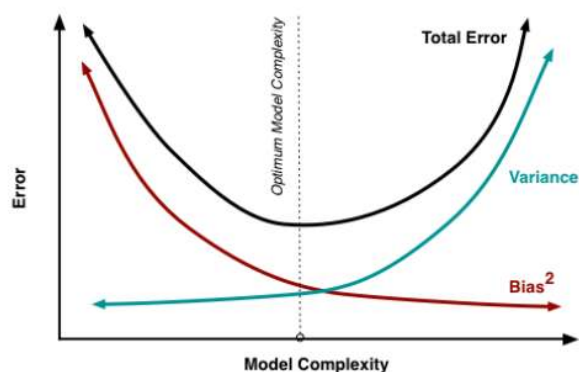
- Holdout procedure (validation set approach)
 - Reserve some data for testing ($\sim 1/3$), use remaining data for training
 - Use test data set to estimate error rate (select a model)
- Stratified holdout
 - Guarantee that classes are ca. proportionally represented in test and training data set
- Repeated holdout (in addition)
 - Randomly select holdout set several times and average error rate estimates

Bias-Variance Tradeoff



Models with too few parameters are inaccurate because of a large bias (not enough flexibility).

Models with too many parameters are inaccurate because of a large variance (too much sensitivity to the sample).



Generalization Errors

→ Components

- Bias: How much differs average model over all training sets from true model (might be due to inaccurate assumptions/ simplifications made by model)
- Variance: How much differ models estimated from different training set from each other
- Underfitting: Model too simple to represent all relevant characteristics
→ High bias, low variance, high training error, high test error
- Overfitting: Model too complex and fits irrelevant characteristics
→ Low bias, high variance, low training error, high test error

Which Model Should be Selected?

- Bias-variance tradeoff provides conceptual framework for determining good model (but not directly useful)
- Aim: Model that minimizes criterion: f (fitting error from given data) + g (model complexity) with f & g increasing functions
- All methods based on tradeoff between fitting error (high variance) and model complexity (low bias)

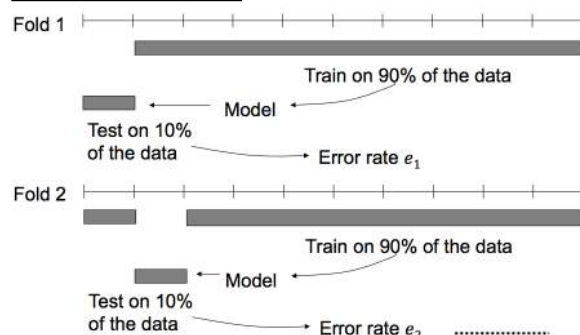
Model Assessment and Model Selection

Popular methods:

- Akaike Information Criterion ($AIC = sk - 2\ln(L)$)
- Minimum description length
- Resampling techniques to estimate error rate (Cross validation; Bootstrap)
- Cross validation popular: Predicts performance of a model on validation set using computation

8.2 Resampling

Cross Validation



k-fold Cross-Validation

- Fixed number of k partitions of the data (fold)
- Each partition used for testing and remaining instances for training → Each instance used for testing once
- May use stratification
- Standard practice: Stratified ten-fold cross-validation
- Error rate estimated by taking average of error rates
- Select model that performs best over all test subsets

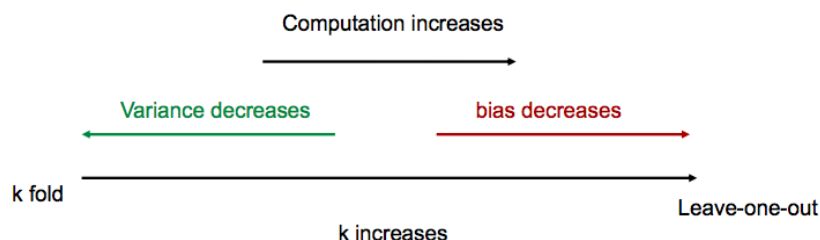
$$\hat{e} = \frac{1}{k} \sum_{i=1}^k e_i$$

Leave-One-Out Holdout

- n-Fold Cross-Validation
 - n instances in data set

- Use all but one instance for training
- Each iteration evaluated by predicting omitted instance
- (Dis-)Advantages
 - Maximum use of data for training
 - Deterministic (no random sampling of test sets)
 - High computational cost
 - Non-stratified sample

How many Folds?



Bootstrap

- Number observations $1, 2, \dots, n$
- Draw random sample of size n with replacement
- Calculate statistic (eg error rate)
- Repeat steps 1-3 many times (eg 500)
- Calculate variance of your statistic directly from sample of 500 statistics
- Can also calculate CI directly from sample of 500 statistics
- Example: Re-sample 500 samples of $n=50$ with replacement, run logistic regression and examine the distribution of error rates (or other metrics)

Comparing Error Rates

- Suppose 2 algorithms
 - Obtain 2 different models
 - Estimate error rates
 - Compare estimates
- $\hat{e}^{(1)} < \hat{e}^{(2)}$?
- Select better one
- Problem: Significant differences in error rates?
- Estimated error rate is just estimate (random)
- Students t-test shows if means different
- Construct t-test statistic: Need variance and point estimates

$$t = \frac{\bar{d}}{s_d / \sqrt{k}}$$

Observed standard deviation of diff. in error rate

Average of differences of error rates

H_0 : Difference = 0

- If 2 algorithms are compared on same test set (test sets not independent), then binomial test preferred

Measuring Errors

		Predicted class	
		Yes	No
Actual class	Yes	True positive	False negative (Type I error)
	No	False positive (Type II error)	True negative

Error rate = # of errors / # of instances = $(FN+FP) / N$

Recall = # of found positives / # of positives

= $TP / (TP+FN)$ = sensitivity = hit rate

Precision = # of found positives / # of found

= $TP / (TP+FP)$

Specificity = $TN / (TN+FP)$

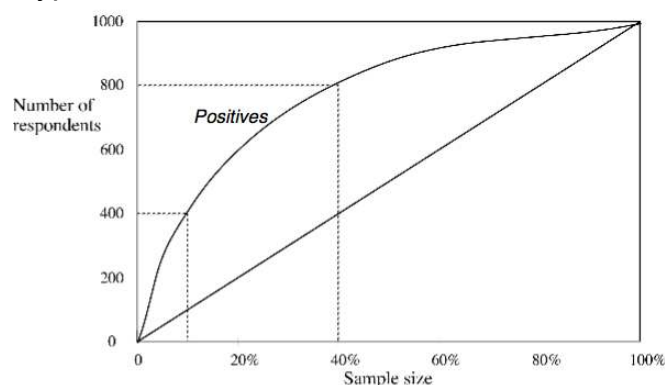
False alarm rate = $FP / (FP+TN) = 1 - \text{Specificity}$

8.3 Gain and ROC Curves

Direct Marketing

- Find most likely prospects to contact
- Not everybody needs to be contacted
- Number of targets usually much smaller than number of prospects
- Typical applications: Retailers, catalogues, customer acquisitions,...
- Accuracy on entire dataset isn't right measure
- Approach: Develop target model; Score all prospects and rank them by decreasing score; Select top q% of prospects for action

Hypothetical Gain Curve



Generating Gain Curve

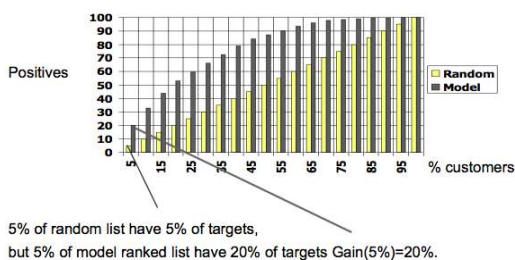
→ Visualize results of different cutoffs

- Instances sorted according to predicted probability

Rank	Predicted probability	Actual class
1	0.95	Yes
2	0.93	Yes
3	0.93	No
4	0.88	Yes
...

3 hits in top 5% of the list
If there are 15 targets overall, then top 5 has 3/15=20% of targets

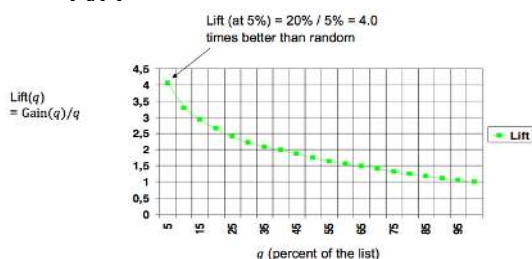
- x-axis: Sample size (# of instances); y-axis: # of positives
- Random List vs. Model-Ranked list



Lift Curve

→ Displays factor between classifier and random value for every part of gain curve

- x-axis: # of instances (or percentage of the data set); y-axis: Factor: gain at X/X

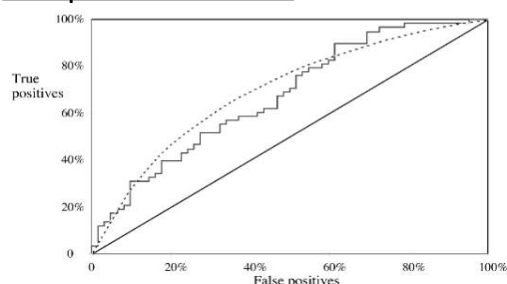


ROC Curves

→ Displays ratio of false positive rate and true positive rate

- Differences to gain chart
 - y axis: Percentage of true positives in sample (rather than absolute number):
TP rate = tp = 100 * TP/(TP + FN)
 - x axis: Percentage of false positives (rather than sample size):
FP rate = fp = 100 * FP/(FP + TN)
- ROC curves similar to gain curves
 - ROC stands for receiver operating characteristic
 - Used in signal detection to show tradeoff between hit rate and false alarm rate over noisy channel
 - Go through all sizes of a sample and plot tp vs. fp

Sample ROC Curve



Jagged curve - one set of test data

Smooth curve - use cross-validation and average

→ Every point that is beginning of a step can be cutoff value

ROC Curves for Two Classifiers

- For small, focused sample, use method A (eg only interest in 40% of tp)
- For larger one, use method B

8.5 Kolmogorov Complexity and MDL

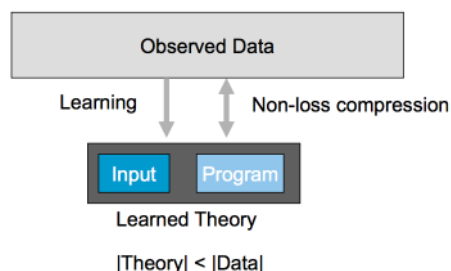
Traditional Model Selection Criteria in Science

- Model selection criteria attempt to find good compromise between model complexity and its prediction accuracy on the training data
- Occam's Razor: Best theory is smallest one that describes all facts

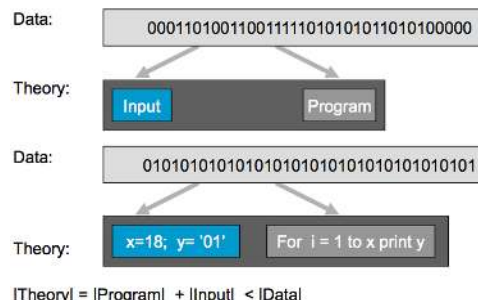
MDL Principle

- Model selection criterion (Minimum description length)
- Description length (DL) defined as:
space required to describe a theory + the one to describe theory's mistake
- Our case: Theory is classifier and mistakes are errors on training data
- Aim: Classifier with minimal DL

Inductive Learning Systems



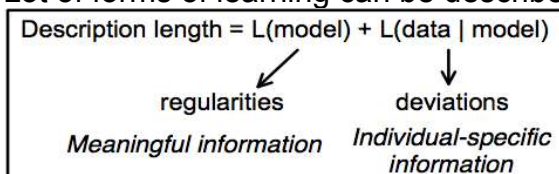
Example:



$$|Theory| = |Program| + |Input| < |Data|$$

Randomness vs. Regularity

- Random string: Incompressible → Maximal information
- Regular: Allows compression → If training set is representative then regularities in training set will be representative for regularities in unseen test set
- Lot of forms of learning can be described as data compression:



Kolmogorov Complexity

- KC (K) of binary object is length of shortest program that generates this object in universal Turing machine
 - Random strings aren't compressible
 - Message with low K complexity is compressible
- K as complexity measure is incomputable
 - In practical applications: Needs to be approximated

9. ENSEMBLE METHODS AND CLUSTERING

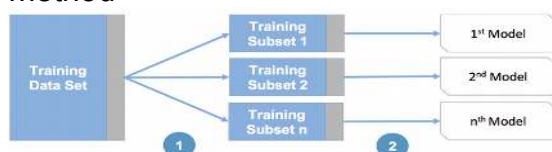
9.1 Ensemble Methods

- Methods used for classification
- Use multiple models to obtain better predictive performance → Combine multiple weak learners to produce strong learner
- Recall Bias-Variance tradeoff: Ensemble of models with low bias and high variance might have reduced overall variance (Generalization Error = Bias + Variance is lower)

Bagging (Bootstrap Aggregation)

- Overcome problems with particular sample
- Sample with replacement from training data set
- Apply learning algorithm (eg C4.5) to each set
- Vote on prediction (classification/ numeric)
 - All trees applied to test data set
 - Select class with highest number of votes/ average result (if numeric)
- Reduces variance by aggregation

- Useful for unstable algorithms (DT, rule learners)
 - Small changes in input data can change attributes in DT
 - Less useful if learning algorithms produce stable results (eg linear regression)
- How it works
 - Training models
 - Sample randomly n training subsets of same size from initial training set
 - Train one model for each training subset using same machine learning method



→ Classifying instances

- Each model is giving vote for a classification → n independent predictions
- If one class receives majority: Taken as correct one



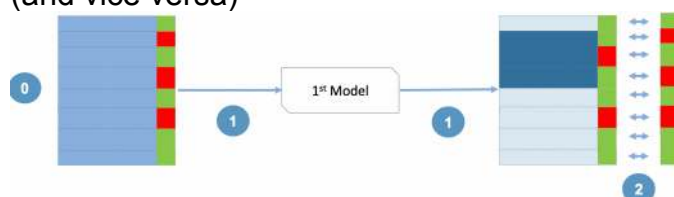
Random Forests

- Constructing multitude of DTs at training time and outputting class that is mode of the classes (classification) or mean prediction (regression) of individual trees
- Correct overfitting
- Classifier generation
 - Select ntree (# of trees to grow) and mtry (# of variables)
 - For $i = 1$ to ntree:
 - Draw bootstrap sample from data, call those not in this sample “out-of-bag” data
 - Grow random tree (at each node best split is chosen among mtry random selected variables)
 - Store resulting DT
- Classification
 - For each of the ntree DT: Predict class of instance
 - Return class predicted most often

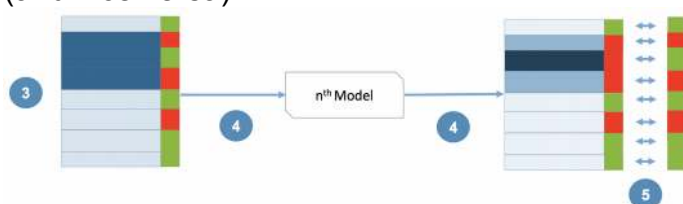
Boosting

- Start by applying some method (eg C4.5) to learning data
- Compute predicted classifications, apply weights to instances in learning sample
- Apply classifier again to weighted data and continue with next iteration (until error-rate = 0)
- Model generation (AdaBoost.M1)
 - Assign equal weights to each training instance
 - Apply learning algorithm to weighted dataset and store model
 - Compute error e and store error
 - If $e = 0$ or $e > 0.5$ terminate
 - For every instance: If classified correctly weight = weight * $e/(1-e)$
 - Weight for misclassified instances remains the same
 - Normalize weight of all instances
- Classification

- Assign zero weight to each class
- For ever model generated: Add $-\log(e/(1-e))$ to weight of class predicted by model \rightarrow Classifiers with low error rate get higher weight
- Return class with highest overall sum
- Generates sequence of classifiers where each of them is expert in classifying observations that weren't well classified by those preceding it
- Predictions of different classifiers can be combined
- How it works
 - \rightarrow Training models - 1st round



- \rightarrow Training models - nth round
- Precondition: All training data instances have different weights
- nth model is trained on training data set focusing on high weights
- Prediction is evaluated: Weights of correctly classified instances are reduced (and vice versa)



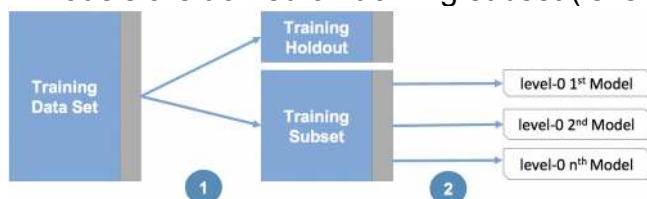
- \rightarrow Classifying instances
- Each model assigned weight according to its error rate during training
- Each model gives vote for classifying test data set \rightarrow n independent predictions
- Final decision based on weighted majority vote



Stacking

- Predictions from different classifiers used as input to meta-learner, which combines them to create final best predicted classification
- Meta learning
 - Holdout part of training set
 - Use remaining data for training level-0 methods (eg DT, naïve Bayes)
 - Use holdout data to train level-1 learning (Meta learner)
 - Retrain level-0 algorithms with all the data
 - Level-1 learning: Use very simple algorithm (eg linear model)
- How it works
 - \rightarrow Training models – level-0
 - Split training data into training subset and holdout subset

- n models are trained on training subset (level-0 classifiers)



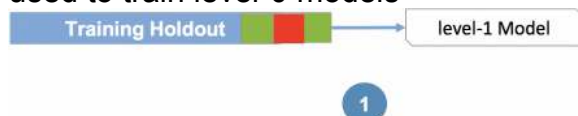
→ Classifying instances – level-0

- Level-0 models predict training holdout's label
- Training holdout data contains now predictions of level-0 classifiers only



→ Training models – level-1

- Training holdout data now serves as training data for single level-1 model; Usually machine learning method used for level-1 model differs from ones used to train level-0 models



→ Classifying instances – level-1

- Level-1 model used to classify test data



9.2 Clustering

Definition

- Data set with n p-dimensional data items
- Find natural partitioning into number of clusters (k) and noise
- Clusters should maximize intra-cluster similarity and minimize inter-cluster similarity
- Methods
 - For numeric and/ or nominal data
 - Deterministic vs. probabilistic
 - Partitional vs. overlapping
 - Hierarchical vs. flat
- Issues
 - Interpreting and evaluating results
 - Outlier handling
 - # of clusters
 - Scalability of algorithms

Partitional clustering via k-means

- Works with numeric data only
- Pick number (k) of random cluster centers
- Assign every item to nearest cluster center
- Move each cluster center to mean of its assigned items
- Repeat until convergence (change in cluster assignments less than threshold)
- Discussion
 - Result can vary significantly depending on initial choice of seeds (-)

- Can get trapped in local minimum (-)
- Must pick # of clusters before hand (-)
- All items forced into cluster → Too sensitive to outliers (-)
- Simple, understandable (+)
- Items automatically assigned to clusters (+)
- To increase chance of finding global optimum: Restart with different random seeds

Hierarchical clustering via MST

- Bottom up: Start with single-instance cluster
- Top down: Start with one universal clusters
- Both: Produce dendrogram
- MST algorithm
 - Compute minimal spanning tree of graph (Minimum-weight tree in weighted graph which contains all of the graphs vertices with minimal total weight)
 - Connect graph components form clusters

Probabilistic clustering via Expectation Maximization (EM)

- Model each cluster with probability distribution (mixture)
 - Simple case: Single numeric attribute, 2 clusters A & B each represented by a normal distribution
 - Start with initial guesses for the parameters $\mu_A, \mu_B, \sigma_A, \sigma_B, \Pr[A]$
 - Calculate cluster probabilities w_i for each instance (expectation)
 - Re-estimate distribution parameters from probabilities (maximization)
 - Repeat
 - Termination
 - EM algorithm converges to a maximum
 - Continue until overall likelihood growth is negligible → Measure for finding the maximum likelihood
- $$\prod_{i=1}^n (\Pr[A] \Pr[x_i | A] + \Pr[B] \Pr[x_i | B])$$
- Maximum found by EM could be local optimum → Repeat several times with different initial values
- Extending the model
 - Multiple clusters
 - Multiple attributes (multiply probabilities of all attributes)
 - For nominal attributes (can't use normal distribution)

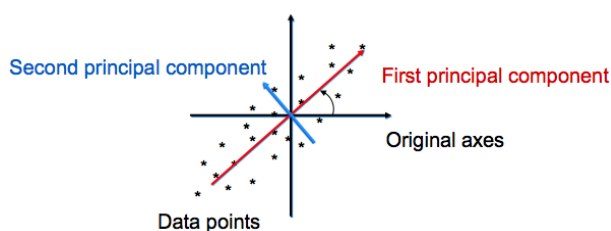
9.3 Classification and Clustering

Classification	Clustering
You have an instances with a known outcome (dependent variable) and would like to know to which class a new instance belongs to.	You have instances where you assume relationships between them and would like to group them accordingly.
Characteristics	Characteristics
<ul style="list-style-type: none"> Supervised learning Target is known Training data 	<ul style="list-style-type: none"> Unsupervised learning Target is unknown No training data
Algorithms/Concepts	Algorithms/Concepts
<ul style="list-style-type: none"> Naïve Bayes Decision Trees Ensemble Methods 	<ul style="list-style-type: none"> K-means Minimal Spanning Tree Expectation Maximization

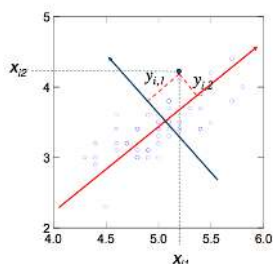
10. DIMENSIONALITY REDUCTION

Principal Component Analysis (PCA)

- Converts set of possibly correlated variables into (smaller) set of values of linearly uncorrelated variables called principle components
- First PC has largest possible variance, each succeeding has next highest variance under the constraint it is orthogonal (uncorrelated) with the preceding
- PCs are Eigenvectors (\rightarrow orthogonal) of the symmetric covariance matrix



- Eigenvalues (describe directions in space): Eigenvalues λ_1 explain proportion of variance explained by PC1
- Scores: Gives coordinates



\rightarrow PCA score for any of the X is its coefficient in each of the Y's

- PCA

From p original variables: x_1, x_2, \dots, x_p :

Produce k (or less) new variables: y_1, y_2, \dots, y_k as linear combinations of the original variables x_i .

$$y_1 = a_{11}x_1 + a_{12}x_2 + \dots + a_{1k}x_k$$

$$y_2 = a_{21}x_1 + a_{22}x_2 + \dots + a_{2k}x_k$$

...

$$y_k = a_{k1}x_1 + a_{k2}x_2 + \dots + a_{kk}x_k$$

y_k 's are the
Principal Components

$\rightarrow y_k$'s have to be uncorrelated (orthogonal) and y_1 has to explain as much as possible of the original variance in data

Eigenvalues and Eigenvectors

Let A be an $p \times p$ matrix with Eigenvalue λ and corresponding Eigenvector v . Thus $Av = \lambda v$. This equation may be written

$$Av - \lambda v = 0$$

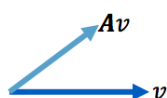
, given

$$(A - \lambda I_p)v = 0$$

- Solving the equation $|A - \lambda I_p| = 0$ for λ leads to all the Eigenvalues of A
- On expanding the determinant $|A - \lambda I_p|$, we get a polynomial in λ
- This polynomial is called the **characteristic polynomial** of A
- The equation $|A - \lambda I_p| = 0$ is called the **characteristic equation** of A

→ Geometric interpretation

Av points in some other direction defined by A



v is an Eigenvector and λ an Eigenvalue if

$$\lambda v = Av$$

PCA – Steps

1. Subtract the mean of each column
2. Calculate covariance matrix which summarizes relationship between variables
 - If non-diagonal elements in matrix are positive, should expect that both the x and the y variables increase together
 - Sum of diagonals is called trace and represents total variance (= mean squared Euclidean distance between object and the centroid in p -dim space)
 - Covariance matrix centers each variable on the mean, but scale matters → Should only be used when variables measured in comparable units and differences in variance are important for interpretation → If variables measured in different units: Use correlation matrix
3. Calculate Eigenvectors and Eigenvalues of the covariance matrix
 - Final data
 $Z = X * \Phi$
 - If you take all Eigenvectors in Φ : Get original data rotated so that Eigenvectors are the axes
4. Reduce dimensionality and form feature vector
 - Rotate such that Eigenvector with highest Eigenvalue is first PC
 - Order Eigenvectors by Eigenvalue (high to low) → Gives you components in order of significance
 - Can ignore components of lesser significance
 - Don't lose much info if Eigenvalues are small

How many PC?

- Take enough Eigenvalues to cover 80-90% of the total variance

Reconstruction of Original Data with one EV

$X \approx \text{PCA Scores} * \text{Eigenvectors} + \text{original mean}$

→ If we reduce dimensionality, then, when reconstructing data, we lose dimensions we chose to discard

Aggregation of Attributes

- Matrix Φ allows aggregating similar attributes
- Each element of Eigenvectors represents contribution of given variable to a component
- Example

	1	2	3	4	5	6	7	8
Volume	0.948	-0.094	-0.129	0.228	0.040	0.036	0.136	0.055
Length	0.906	0.302	-0.064	-0.209	0.128	-0.144	-0.007	-0.050
Width	0.977	-0.128	-0.031	0.032	0.103	-0.017	-0.014	0.129
Depth	0.934	-0.276	-0.061	0.014	0.074	0.129	0.154	-0.038
Speed1	0.552	0.779	-0.196	-0.133	-0.099	0.143	-0.038	0.018
Speed2	-0.520	0.798	-0.157	0.222	0.109	-0.038	0.071	0.004
Radius	0.398	0.311	0.862	0.038	0.008	0.022	-0.002	-0.005

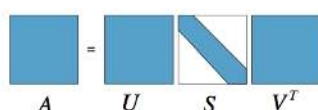
→ First 4 aggregated to latent variable size, next 2 to speed

Assumptions

- Linear relationships among variables
- Euclidian distance among points used, assuming continuous variables (→ discrete: Special techniques necessary)

Singular Value Decomposition (SVD)

For any matrix $A \in \mathbb{R}^{m \times n}$, there exist orthogonal matrices U, V and a diagonal matrix S , such that all the diagonal values σ_i of S are non-negative and

$$A = USV^T$$


SVD and PCA

- Diagonal values of S: Singular values → Sort them by size
- Columns of U: Left singular vectors → Are PCs
- Columns of V: Right singular vectors
- Singular values is square root of the Eigenvalues
- Finding U,S,V is equivalent to finding Eigenvectors

Reminder: Multiple Linear Regression

Equation: $y = X\beta + \varepsilon$

- y : $n \times 1$ vector of observed values
- X : $n \times p$ matrix of independent values
- β : $p \times 1$ vector of regression parameters
- ε : $n \times 1$ vector of residuals

OLS estimator: $\hat{\beta} = (X^T X)^{-1} X^T y$

Multicollinearity in Linear Regression Models

$$\min(\|X\beta - y\|_2^2)$$

MLR solution, requires $n \geq p$
(variable selection) and nearly orthogonal X

If $X^T X$ is not full rank

- No unique solution to normal equations

If the columns of X are highly correlated

- Leads to unstable equation/plane
in the direction with little variability

Solutions to Multicollinearity

- Subset/attribute/feature selection
 - Backward, forward, stepwise selection of features
- Using derived input

- PC regression
- Partial Least Squares
 - Like PC Regression except in how components are computed
 - PC: Weights calculated from covariance matrix of the predictors
 - PLS: Weights reflect covariance structure between predictors and response
 - Like in regression: Goal is to max correlation between response(s) and component scores
- Coefficient shrinkage (smoothing) → Continuous version of subset selection
 - Ridge Regression: Ridge coefficient minimize penalized RSS
 - Lasso: Penalize by absolute value of parameter

Ridge vs. PCA vs. PLS vs. Lasso

- Ridge regression and PCR outperform PLS in prediction
- Lasso outperforms ridge when moderate # of sizeable effects, rather than many small effects; Also produces more interpretable models

11. ASSOCIATION RULES AND RECOMMENDERS

Unsupervised Learning

- Clustering
 - Unsupervised classification (without the class attribute)
 - Want to discover classes
- Association rule discovery
 - Discover correlation among attributes
 - Widely used, later for cross- and up-selling

Association Rule Discovery

- Discover interesting correlations/ relationships in large databases
- Finds rule: If A and B then C and D
- Which attributes included in relation is unknown
- Primary application: Market basket analysis
 - Analyze customer buying habits by finding associations and correlations between different items that customers place in shopping basket
 - Applicable whenever customer purchases multiple things in proximity
- Association rules with minimum support and confidence: Strong rules

Applications to recommender systems

- Collaborative filtering: For given user, find similar users (ratings correlate)
 - Recommend items rated highly by these users (not by our user)
- Important not to trust correlations based on very few co-rated items → Include significance weights based on # of co-rated items
- Problems CF
 - Cold start
 - Sparsity
 - First rater
 - Popularity bias (unique tastes → No recommendation)
- Content-based filtering: Based on info on content of items rather than other users opinions → Uses machine learning algorithm
- Pro: Able to recommend new and unpopular items and unique tastes