Introduction to Machine Learning in Economics

Zhiyuan Chen

Department of Trade Economics
Remin Business School

December 2, 2020

Two Cultures in using Statistical Modeling

> "There are two cultures in the use of statistical modeling to reach conclusions from data. One assumes that the data are generated by a given stochastic data model. The other uses algorithmic models and treats the data mechanism as unknown.
> .....
> If our goal as a field is to use data to solve problems, then we need to move away from exclusive dependence on data models and adopt a more diverse set of tools." Breiman [2001b], p199.

- Machine learning (ML) methods are beginning to be widely used in empirical works by economists, though in a much slower pace than in statistics
- Economics journals emphasize the use of methods with formal properties which are not delivered by ML methods

An Overview of ML

- The term Machine Learning is used because the computer figures out the model $\hat{f}(x)$ from the data rather than using a pre-specified model $y = \mathbf{x}'\beta + \varepsilon$.
  - The data is not necessarily big, but typically $dim(\mathbf{x})$ is large

$$\text{Machine Learning} \begin{cases} \text{Supervised learning} \begin{cases} \text{Regression: } y \text{ is continuous} \\ \text{Classification: } y \text{ is categorical} \end{cases} \\ \text{Unsupervised learning: Cluster Analysis, } y \text{ unknown} \end{cases}$$

- Economists mostly use supervised learning

Econometrics vs. ML

	Econometrics	ML
Goals	parameter estimates and CI	developing **algorithms** to make predictions and classifications
	e.g.: $\hat{\beta}_{OLS} = argmin_\beta \sum_{i=1}^{N} (Y_i - \mathbf{X}_i'\beta)$	e.g.: $min_\beta \sum_{j=N+1}^{N_m} (Y_j - \hat{Y}_j)^2$
Terminology	sample	*training/test (hold-out)* sample
	regressors, covariates, or predictors	*features*
	estimation	*training*
	prediction	*supervised learning*, *unsupervised learning*, *classification problems*
Methods	estimating parameters in a given model: selecting the "true" model	*cross-validation*: predictive power *and* out-of-sample comparisons

Overfitting

- Overfitting Problem: models "overfit" within sample
  - $\hat{\varepsilon} = \left(\mathbf{y} - \mathbf{x}\hat{\beta}_{OLS}\right) = \left(I - \mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}\right)\varepsilon$: $|\hat{\varepsilon}_i| < |\varepsilon_i|$ on average when overfitting
- Two solutions:
  1. Penalty for overfitting (Regularization): $\bar{R}^2$, AIC, BIC, etc.

$$\text{argmin}_\beta \sum_{i=1}^{N} \left(y_i - X_i'\beta\right)^2 + \lambda \left(\sum_{k=1}^{K} \beta_k^2\right)$$

  2. Cross-validation (CV): choose the optimal λ that yields the best out-of-sample fit using some criterion, usually $K-$fold CV
     1. split data into K(5 or 10) mutually exclusive folds of roughly equal size
     2. for $j = 1, \cdots, K$ fit using all subset excluding j and predict on fold j
     3. Choose λ that minimizes the $\frac{1}{K}\sum_{j=1}^{K} MSE\,(j)$

Supervised Learning

- Linear regression: Lasso, Ridge, and other hybrid methods
- Regression trees and random forests
- Neural networks
- Boosting

Regularized Linear Regression

- A common form of regularization is to add a penalty term that shrinks β_k towards zeros by minimizing

$$argmin_\beta \sum_{i=1}^{N} (y_i - X_i\beta)^2 + \lambda \left(\sum_{k=1}^{K} |\beta_k|^q \right)^{\frac{1}{q}},$$

  where λ is chosen through out-of-sample cross-validation.
  - $q = 1$: LASSO
  - $q = 2$: Ridge regression
  - $q \to 0$: Best subset regression
- Hybrid methods:
  - Elastic nets: combines LASSO and Ridge
  - Relaxed LASSO: combines OLS estimates for parameters selected by LASSO and the LASSO estimates themselves

LASSO vs. Ridge

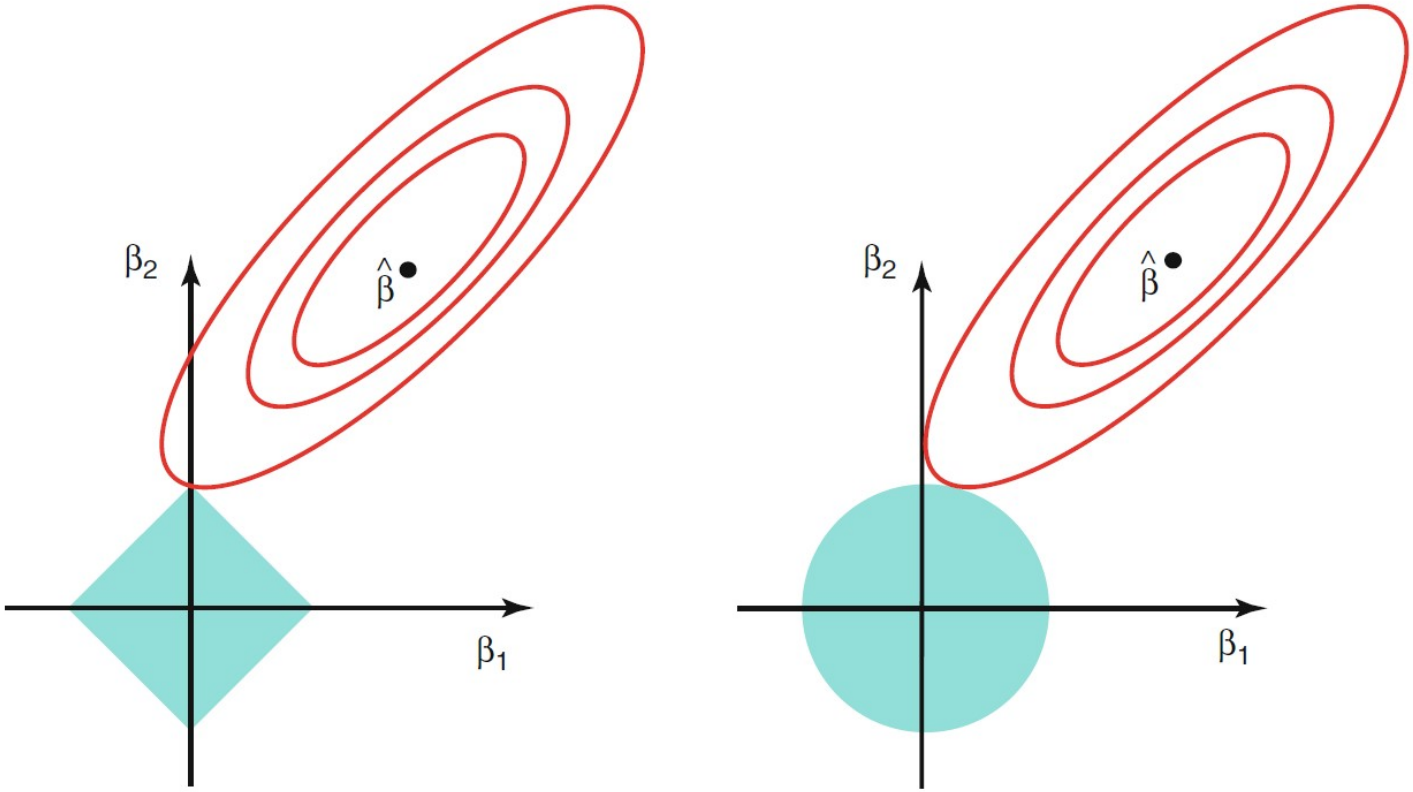


FIGURE 6.7. *Contours of the error and constraint functions for the lasso* (left) *and ridge regression* (right). *The solid blue areas are the constraint regions,* $|\beta_1| + |\beta_2| \leq s$ *and* $\beta_1^2 + \beta_2^2 \leq s$, *while the red ellipses are the contours of the RSS.*

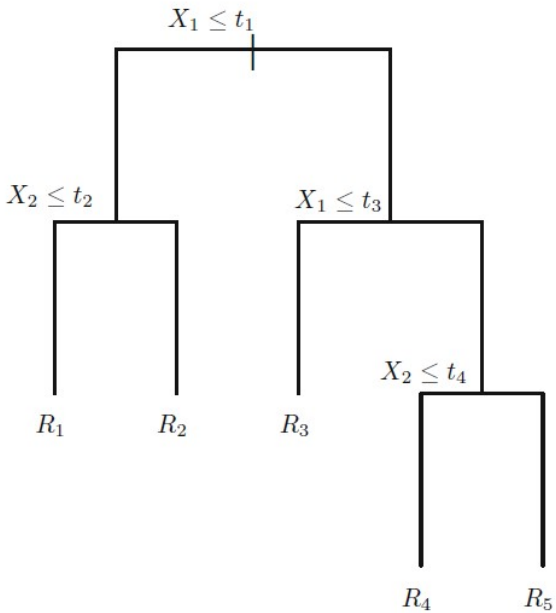
- LASSO is likely to set some coefficients to zero

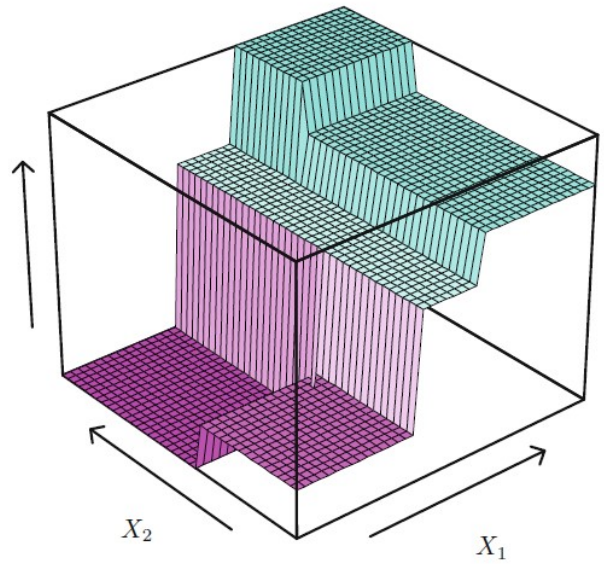
Regression Trees

- Flexibly estimating regression functions when out-of-sample predictive power is important
- Regression trees sequentially split regressors $\mathbf{X}$ into regions that best predict y:
  - Sequentially split $\mathbf{X}$'s into rectangular regions based on a single covariate X_{ik}

1. Before the split, the sum of in-sample squared errors is $Q = \sum_{i=1}^{N} \left(Y_i - \bar{Y}\right)^2$
2. After a split based on X_{ik} and threshold c, $Q(k, c) = \sum_{i:X_{ik} \leq c} \left(Y_i - \bar{Y}^l_{k,c}\right)^2 + \sum_{i:X_{ik} > c} \left(Y_i - \bar{Y}^r_{k,c}\right)^2$
   - ★ X_{ik} and c are chosen to minimize $\{Q(k, c)\}$ in each splitting.
3. Repeat the splitting over subsamples (or leaves) and determine the optimal tree structure
   - ★ Add penalty term λS, where S is the number of leaves, λ is chosen through cross-validation
4. The prediction in each leaf is a sample average

Regression Trees with Two Regressors





Random Forests

- The estimated regression function given by regression trees is discontinuous with substantial jumps
- Random forests induce smoothness by averaging over a large number of trees:
  1. Each tree is based on a bootstrap sample (bagging) or on a subsample
  2. At each stage, the splits are optimized over a random subset of the covariates, changing every split.
- Random forests are related to kernel regression:
  - use of a weighted average of nearby observations:

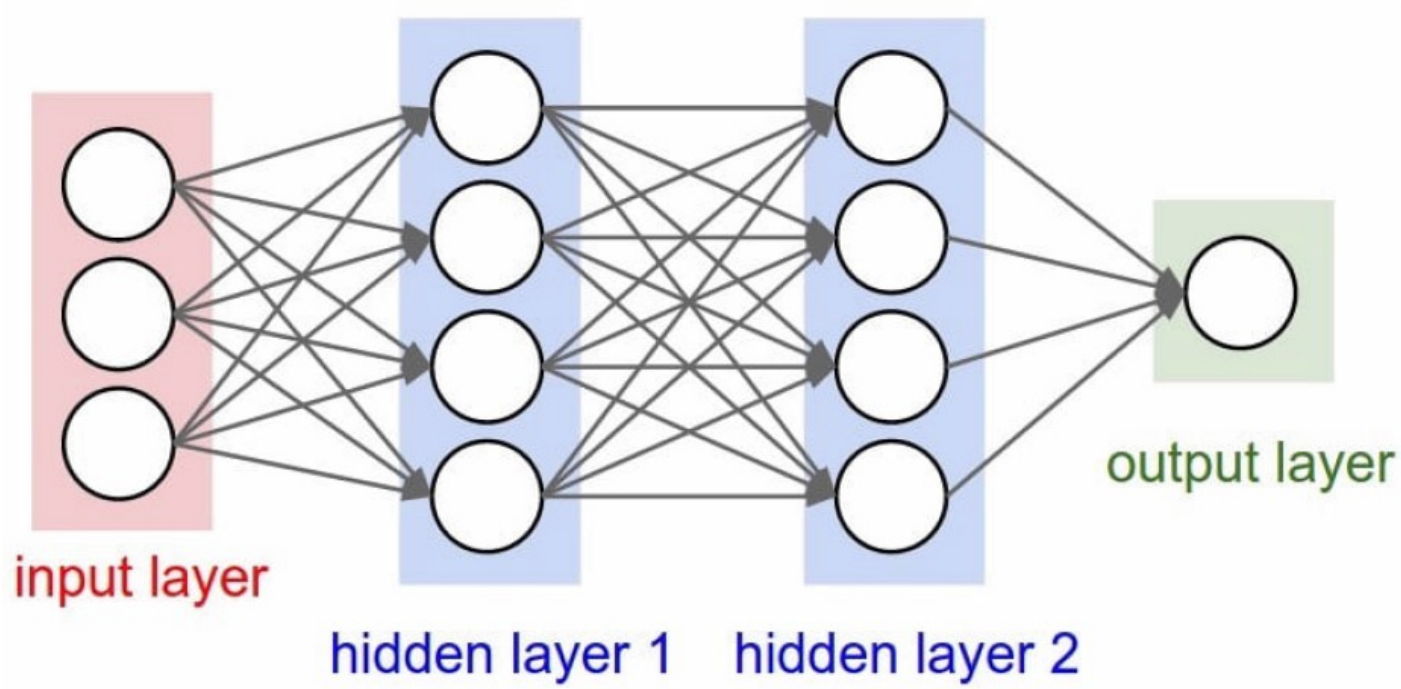
$$\hat{\mu}_{RF}(x) = \sum_{i=1}^{n} \alpha_i(x) y_i, \sum_{i=1}^{n} \alpha_i(x) = 1, \alpha_i(x) \geq 0$$

  - random forests ignore irrelevant covariates more effectively
  - random forests are particularly successful in settings with sparsity (a large number of irrelevant features)

Neural Networks

- A *neural network* involves a series of logit regressions

Figure: Neural nets example



A Neural Network with One Hidden Layer

- Given K features X_{ik}, we model K_1latent/unobserved variables Z_{ik}(hidden nodes):

$$Z_{ik}^{(1)} = \sum_{j=1}^{K} \beta_{kj}^{(1)} X_{ij},\ k = 1, \cdots, K_1$$

- We then modify them using some non-linear transformation:
  - Sigmoid function: $g(z) = 1/(1+\exp(-z))$
  - Rectified linear function: $g(z) = z\mathbf{1}(z > 0)$
- The outcome is modeled as a linear function of the nonlinear transformation:

$$Y_i = \sum_{k=1}^{K_1} \beta_k^{(2)} g\left(Z_{ik}^{(1)}\right) + \varepsilon_i$$

- Parameters are estimated by minimizing the sum of squared residuals, plus a penalty term.
- Back-propagation algorithm is used

Supervised Learning for Classification Problems

- *Digit recognition:* Based on a picture, coded as a set of say 16 or 256 black and white pixels, the challenge is to classify the image as corresponding to one of the ten digits from 0 to 9.
- Algorithms:
  1. Classification Trees and Forests:
  2. Support vector machines and kernels

Classification Trees and Forests

- The main objective is to minimize a impurity function $I(p_1, \cdots, p_M) = -\sum_{m=1}^{M} p_m \ln(p_m)$
- The regularizer is a penalty term on the number of leaves in the tree:

$$L = -\sum_{m=1}^{M} p_m \ln(p_m) + \lambda S$$

- Extension from a single tree to a random forest is similar to the regression case

Support Vector Machines

- N observations on K-dimensional features X_i and a binary label $Y_i \in \{-1, 1\}$
- weights: $\omega \in \mathbb{R}^K$, bias: $b \in \mathbb{R}$
- A hyperplane $\omega' X_i + b = 0$, and $Y_i = 1$ if $\omega' X_i + b = 0 \geq 0$ and $\omega' X_i + b < 0$ if $Y_i = -1$
- We want to choose (ω, b) that maximizes the *margin* (distance to the closest units):

$$\max_{\omega, b} M$$

$$s.t. \sum_{j=1}^{K} \omega_j^2 = 1$$

$$y_i \left(\omega' X_i + b\right) \geq M$$

SVM Graph from ISLR

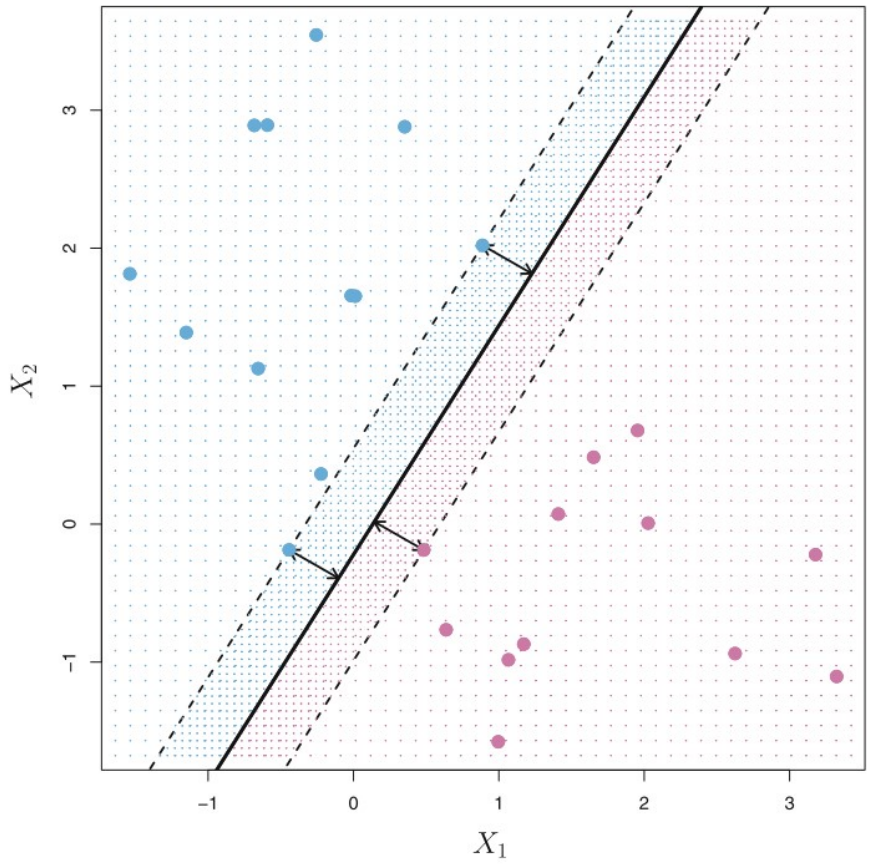


FIGURE 9.3. *There are two classes of observations, shown in blue and in purple. The maximal margin hyperplane is shown as a solid line. The margin is the distance from the solid line to either of the dashed lines. The two blue points and the purple point that lie on the dashed lines are the support vectors, and the distance from those points to the margin is indicated by arrows. The purple and blue grid indicates the decision rule made by a classifier based on this separating hyperplane.*

Some Machine Learning Algorithms

Function class $\mathcal{F}$ *(and its parametrization)*	*Regularizer* $R(f)$
Global/parametric predictors	
Linear $\beta' x$ (and generalizations)	Subset selection $\|\|\beta\|\|_0 = \sum_{j=1}^{k} 1_{\beta_j \neq 0}$
	LASSO $\|\|\beta\|\|_1 = \sum_{j=1}^{k} \|\beta_j\|$
	Ridge $\|\|\beta\|\|_2^2 = \sum_{j=1}^{k} \beta_j^2$
	Elastic net $\alpha\|\|\beta\|\|_1 + (1-\alpha)\|\|\beta\|\|_2^2$
Local/nonparametric predictors	
Decision/regression trees	Depth, number of nodes/leaves, minimal leaf size, information gain at splits
Random forest (linear combination of trees)	Number of trees, number of variables used in each tree, size of bootstrap sample, complexity of trees (see above)
Nearest neighbors	Number of neighbors
Kernel regression	Kernel bandwidth
Mixed predictors	
Deep learning, neural nets, convolutional neural networks	Number of levels, number of neurons per level, connectivity between neurons
Splines	Number of knots, order
Combined predictors	
Bagging: unweighted average of predictors from bootstrap draws	Number of draws, size of bootstrap samples (and individual regularization parameters)
Boosting: linear combination of predictions of residual	Learning rate, number of iterations (and individual regularization parameters)
Ensemble: weighted combination of different predictors	Ensemble weights (and individual regularization parameters)

Unsupervised Learning

- We have information on $\mathbf{X}$ and no outcome y
- We want to partition the sample into a number of subsamples (clusters), or estimating the joint distribution of $\mathbf{X}$:
  - Clustering methods: K-means clustering, hierarchical clustering
  - Generative adverserial networks

Clustering Figure

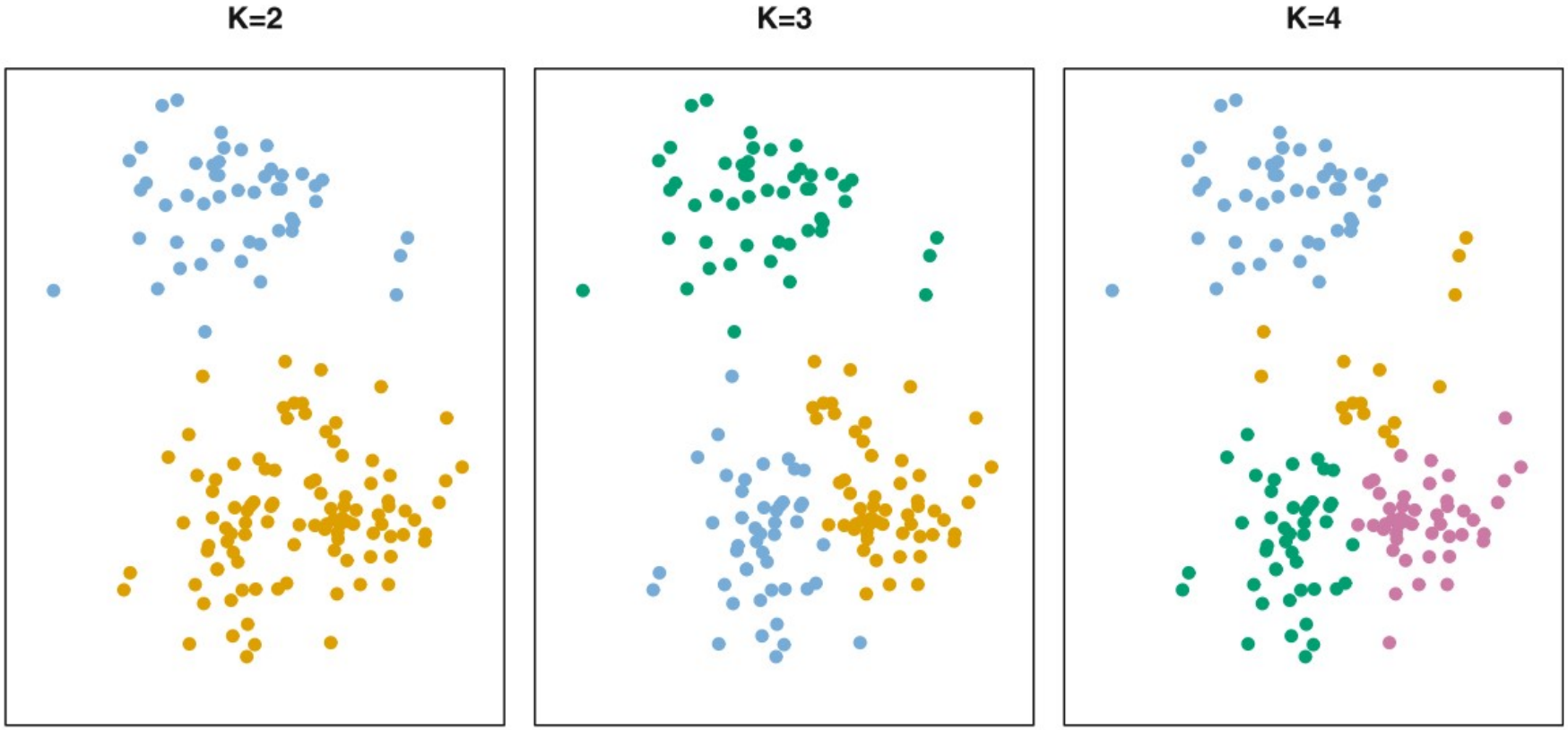


FIGURE 10.5. *A simulated data set with 150 observations in two-dimensional space. Panels show the results of applying K-means clustering with different values of K, the number of clusters. The color of each observation indicates the cluster to which it was assigned using the K-means clustering algorithm. Note that there is no ordering of the clusters, so the cluster coloring is arbitrary. These cluster labels were not used in clustering; instead, they are the outputs of the clustering procedure.*

K-means Clustering

- We wish to partition the feature space into K subspaces or clusters
- We choose centroids $b_1, \cdots, b_K$ and assign units to each cluster based on their proximity to the centroids
- Clustering algorithm:
  1. Start with K centroids, $b_1, \cdots, b_K$, and sufficiently spread out over the space
  2. Assigne each unit to the cluster such that

$$C_i = argmin_{c \in \{1, \cdots, K\}} \|X_i - b_c\|^2$$

  3. Update the centroids as the average of the X_i in each of the clusters:

$$b_c = \sum_{i: C_i = c} X_i / \sum \mathbf{1}(C_i = c)$$

- Choosing K is difficult

Generative Adverserial Networks

- Estimation of a joint distribution using observations on X_i for a random sample of units
- The idea of generative avderserial networks (GAN): find an algorithm to generate data that resemble $X_1, \cdots, X_N$
- Algorithm:
  1. Initialize the discriminator parameter θ_d for $D(\cdot; \theta_d)$ and θ_g for $G(\cdot; \theta_g)$
  2. Sampling m units $\{x^1, \cdots, x^m\}$ from the real data; sampling m noise samples $\{z^1, \cdots, z^m\}$from a chosen distribution D; obtaining generated data $\{\tilde{x}^1, \cdots, \tilde{x}^m\}$ by setting $\tilde{x}^i = G(z^i)$
  3. Update θ_d through maximizing $\tilde{V}(\theta_d) = \frac{1}{m}\sum_{i=1}^m \log\left(D\left(x^i; \theta_d\right)\right) + \frac{1}{m}\sum_{i=1}^m \log\left(1 - D\left(\tilde{x}^i; \theta_d\right)\right)$:

$$\theta_d^{new} = \theta_d + \eta \nabla \tilde{V}(\theta_d)$$

  4. Sampling m noise samples $\{w^1, \cdots, w^m\}$from distribution $D\left(\cdot, \theta_d^{new}\right)$. Update θ_g to maximize $\tilde{W}(\theta_g) = \frac{1}{m}\sum_{i=1}^m \log\left(D\left(G\left(z^i\right)\right)\right)$

$$\theta_g^{new} = \theta_g + \eta \nabla \tilde{W}(\theta_g)$$

Text Analysis[1]

- An explosion of empirical economics research using text as data:
  - use of text from financial news, social media, and company filing to predict stock prices
  - text is used to predict inflation, unemployment, and effects of policy uncertainty
  - text from advertisement and product reviews to study the consumer decision making
  - text from politicians' speeches is used to study the dynamics of political agendas
- Text data is inherently high-dimensional:
  - A w-word long document: if each word is drawn from a vocabulary of p words, the representation of the document has dimension p^w!
  - A sample of thirty-word Twitter mes- sages that use only the one thousand most common words in the English language, for example, has roughly as many dimensions as there are atoms in the universe!

[1] Based on the excellent overview by Gentzkow et al (2019, JEL).

Text analysis in a nutshell

1. Represent raw text $\mathscr{D}$ as a numerical array $\mathbf{X}$;
   - reduce the dimensionality of the data to a manageable level
2. Map $\mathbf{X}$ to predicted values $\hat{\mathbf{Y}}$ to unknown outcomes $\mathbf{Y}$;
   - use high-dimensional statistical methods (e.g., detecting spam emails)
3. Use $\hat{\mathbf{Y}}$ in subsequent descriptive or causal analysis
   - use text to infer causal relationships or structural parameters

From Text to Data

- *Define the document* $\{\mathscr{D}_i\}$ based on the research question:
  - For spam detection, each email is a document
  - For daily stock price prediction, daily news is a document
- *Feature Selection*
  - Strip out elements of the raw text other than words (like punctuation, numbers, HTML tags, and so on)
  - Remove a subset of words that are either too common ("stop words", the, a, and, or,...) or very rare:
    - ★ term frequence-inverse document frequency (tf-idf): $tf_{ij} \times idf_j$

$$tf_{ij} = c_{ij}, \text{ count of occurences of } j \text{ in } \mathscr{D}_i$$

$$idf_j = \log\left(n / \sum_i \mathbf{1}\left(c_{ij} > 0\right)\right)$$

      both very rare words and very common words have low tf-idf.
  - Stemming: replacing words with their root:
    - ★ "economic", "economics", "economically" are all replaced by the stem "economic"

From Text to Data (Continued)

- *n-grams:*
  - A phrase of length n is referred to as an $n-$gram
  - Consider the text of a document $\mathscr{D}_i$ as: *Good night, good night! Parting is such sweet sorrow:*
    - ★ The count of 2-grams (bigrams):
      $c_{ij} = 1$, for $j \in \{night.good, night.part, part.sweet, sweet.sorrow\}$
      $c_{ij} = 2$, for $j \in \{good.night\}$
  - In STATA, the `matchit` command helps

Statistical Methods

- Mapping the document-token matrix $\mathbf{X}$ to predicting $\hat{\mathbf{Y}}$for attributes $\mathbf{Y}$
  - In some cases, the data is partitioned into $\mathbf{X}^{train}$ and $\mathbf{X}^{test}$
  - $\mathbf{X}^{train}$ collects rows for observed $\mathbf{Y}^{train}$, $\mathbf{X}^{test}$ collects rows for unobserved part in $\mathbf{Y}$
- Methods connecting $\mathbf{x}_i$ to $\mathbf{y}_i$:
  1. *Dictionary-based methods*: specify $\hat{\mathbf{y}}_i = f(\mathbf{x}_i)$
  2. *Text regression methods*: $p(\mathbf{y}_i|\mathbf{x}_i)$
  3. *Generative model*: $p(\mathbf{x}_i|\mathbf{y}_i)$: (i) unsupervised; (ii) supervised; (iii) semi-supervised
  4. *Word embeddings*: limited applications in economics, but with great potential

Text Regression

- Mainly applications of standard high-dimensional regression methods to text:
  - Regularized linear models: $\min_{\beta} \left\{ l(\beta) + \lambda \sum_{j=1}^{K} \kappa_j(\beta) \right\}$
  - Dimension reduction: principle components regression (PCR) and partial least squares

$$\min_{\Gamma, B} \left[\left(C - \Gamma B' \right) \left(C - \Gamma B' \right)' \right]$$

$$s.t.\, rank(\Gamma) = rank(B) = K$$

  - Nonlinear text regression: SVM, GLM, Regression trees, Deep learning, Bayesian Regression Methods

Generative Language Models

1. Unsupervised Generative Models
   - No direct observations of the true outcome attributes
   - The counts for document i

$$\mathbf{x}_i \sim MN(\mathbf{q}_i, m_i)$$

     where $\mathbf{q}_i = [q_{i1}, \cdots, q_{ip}]'$ is a document-specific token probability vector, $m_i = \sum_j x_{ij}$ is the document length
     - ★ *Topic model*: $\mathbf{q}_i = \sum_{j=1}^{k} y_{ij}\theta_j$, where $\sum_{l=1}^{p} \theta_{jl} = 1$ for $\theta_j = (\theta_{j1}, \cdots, \theta_{jp})$, and k is the number of topics
2. Supervised Generative Models
   - Attributes are observed in a training set
   - *Naive Bayes classifier*
     - ★ A naive specification of the conditional independent between tokens j: $p(\mathbf{x}_i|\mathbf{y}_i) = \prod_j p_j(x_{ij}|\mathbf{y}_i)$
     - ★ p_j can be estimated as $\hat{p}_j$ using the text database
     - ★ Using the Bayes's rule:

$$p(\mathbf{Y}|\mathbf{x}_i) = \frac{p(\mathbf{x}_i|\mathbf{Y})\pi_y}{\sum_a p(\mathbf{x}_i|a)\pi_a}$$

     where π_y and π_a are prior probabilities.

Word Embeddings

- Representing words as points in a large vector space
  - e.g. six words: {*king*, *queen*, *prince*, *man*, *woman*, *child*}

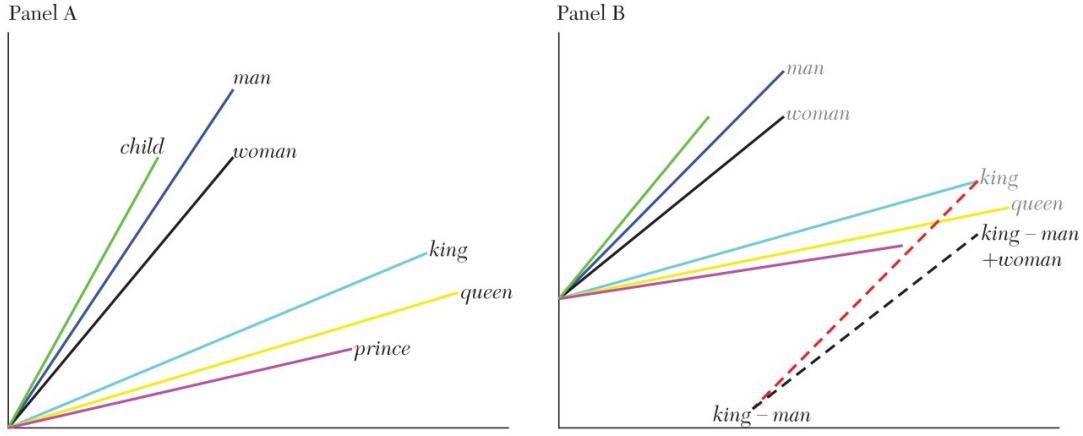


Figure 2. A Graphical Example of Word Embeddings

- Preprocessing the text data to replace word identities with an embedding (location) of each vocabulary in $\mathbb{R}^K$

Dimension	*king*	*queen*	*prince*	*man*	*woman*	*child*
Royalty	0.99	0.99	0.95	0.01	0.02	0.01
Masculinity	0.94	0.06	0.02	0.99	0.02	0.49
Age	0.73	0.81	0.15	0.61	0.68	0.09
...						

Statistical Inference

- For machine learning methods based on a Bayesian modeling approach, posterior distribution is known
- Frequentist view:
  - *non-parametric bootstrap:* not working for non-differentiable loss functions (LASSO)
  - *parametric boostrap:* generates new unrepeated observations for each bootstrap sample using an estimated generative model
  - *subsampling:* data are partitioned into subsamples without replacement; the target parameters are reestimated separately on each subsample
  - *sample splitting:* model selection is performed on one "selection" sample, then the standard inference is performed on the second "estimation" sample conditional upon the selected model

Application I: Text-based market definition (Hoberg and Phillips, 2016)

- The authors develops an time-varying industry classification using text-based analysis of firm product descriptions and show its validity
- They uses 10-K business descriptions to compute pairwise similarity scores for any pair of firms in each year:
  1. Exclude the common words and stop words; focus on nouns and proper nouns
  2. Let P_i be the word vector for firm i, it is normalized as $V_i = \frac{P_i}{\sqrt{P_i \cdot P_i}}$, the pairwise similarity score is

$$\text{Product Cosine Similarty}_{i,j} = (V_i \cdot V_j)$$

  3. Industries are then defined by grouping firms according to their cosine similarities using a clustering algorithm:
     1. Starting with each firm in its own industry
     2. Gradually grouping a firm to the cluster with its nearest neighbor based on similarity score
     3. The algorithm terminates when number of industries reaches a certain value

Application II: Text-based measure of patent quality (Kelly et al., 2019)

- They offer a novel measure of patent quality using patent documents
- They think of two dimensions of patent quality: novel and impactful
  - novel patents: low similarity with the existing stock of patents
  - impactful: high smilarity with subsequent patents
- They show that novelty and similarity scores correlate strongly with the firm's market value

Application III: Economic Policy Uncertainty (Baker et al., 2016

- They come up with a measure of the economic policy uncertainty (EPU) and quantify its economic impacts
- The analyze ten leading newspapers in the US by country-month
- The economic policy uncertainty measure is

$$\frac{\sum_j c_{ij}}{total\ number\ of\ articles_i}$$

  where c_{ij} is a count of the number of articles in newspaper j containing at least one keyword from three categories: (i) economy; (ii) policy; (iii) uncertainty.
- The predicted value $\hat{v}_i$ is a simple average of these scaled counts
- They validate this method using a human audit of 12,000 articles from 1900-2012
- They show that EPU leads to reduced employment, investment, and greater asset price volatility

References

- Athey, S., and G. W. Imbens. (2019): "*Machine Learning Methods That Economists Should Know About*," Annual Review of Economics, 11, 685–725.
- Gentzkow, M., B. Kelly, and M. Taddy. (2019): "*Text as data*," Journal of Economic Literature, 57, 535–74.
- Mullainathan, S., and J. Spiess. (2017): "*Machine learning: An applied econometric approach*," Journal of Economic Perspectives, 31, 87–106.