San José State University

Math 251: Statistical & Machine Learning Classification

Welcome to the first class!

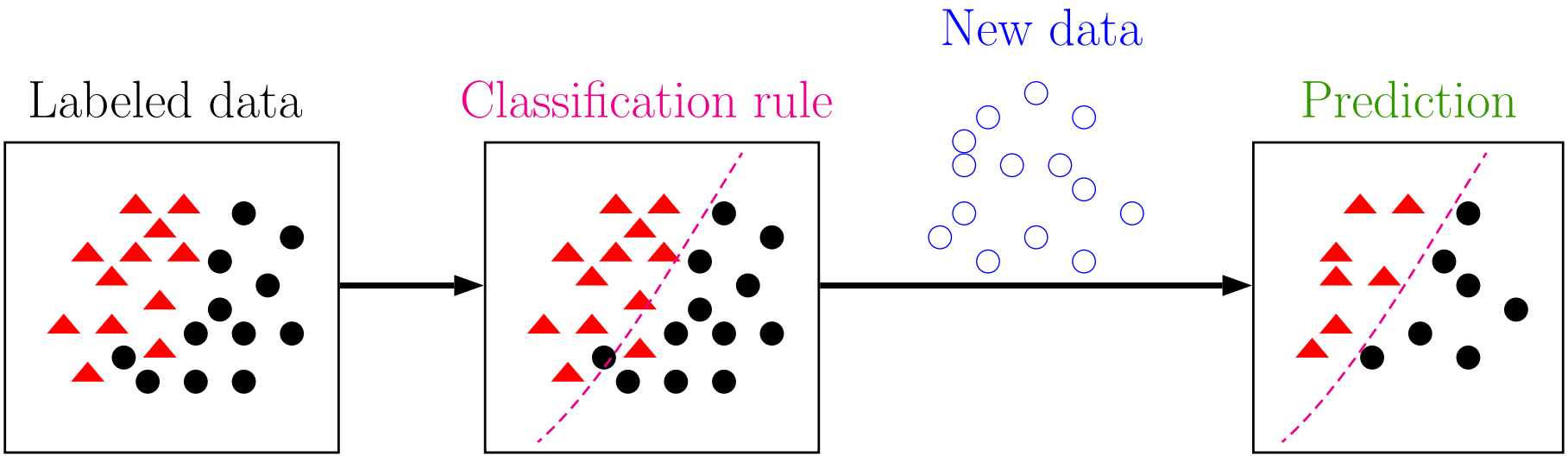
Dr. Guangliang Chen

Agenda

1. Introductions
2. Course overview
3. Syllabus information

What is this course about?

An in-depth survey of the machine learning field of classification, the task of assigning labels to new data based on a given set of examples from the different categories (called labeled data).



In classification settings,

- Labeled data is called **training data**;
- New data is called **test data**;
- A classification algorithm is called a **classifier**.

Classification has numerous applications, e.g., *spam email detection*, *digit recognition*, *face recognition*, and *document classification*.

MANY algorithms have been developed, leading to a VAST literature on classification.

Major branches of machine learning:

- **Supervised learning** (with labeled data)
  - Regression (Math 261A)
  - Classification ⟵ this course
- **Unsupervised learning** (no labeled data)
  - Dimensionality reduction (Math 250)
  - Clustering (Math 252)

History of this course

Fall 2015: **Math 203 CAMCOS** (based on a Kaggle competition: *Digit Recognizer*[1])

Spring 2016: **Math 285 Classification with Handwritten Digits**[2]

Fall 2018: **Math 251 Statistical & Machine Learning Classification**[3]

[1] https://www.kaggle.com/c/digit-recognizer
[2] http://www.sjsu.edu/faculty/guangliang.chen/Math285S16.html
[3] http://www.sjsu.edu/faculty/guangliang.chen/Math251F18.html

Design of this course

To teach a machine learning topic (**classification**)

- ...through an application (**digits recognition**)
- ...using a benchmark dataset (**MNIST Handwritten Digits**)
- ...assisted by a technical computing language (**MATLAB/Python**)
- ...enhanced by a hands-on project.

Goals of this course

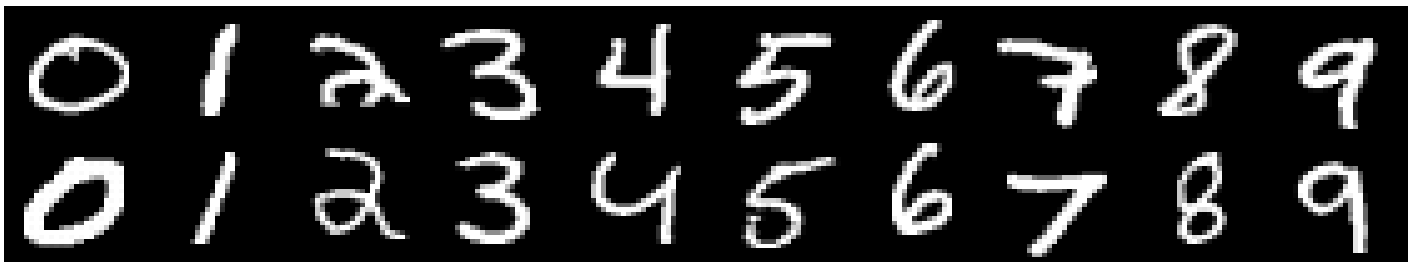
- Introduce the machine learning field of classification with applications
- Present the ideas and mathematics of major classification methods in the literature
- Teach how to use specialized software to perform classification tasks while adequately addressing practical challenges (e.g., parameter tuning, memory and speed)
- Provide students with valuable first-hand experience in handling large, complex data

Prerequisites of the course

- Math 164 Mathematical Statistics (Bayes' decision rule and MLE)
- Math 250 Mathematical Data Visualization (advanced linear algebra, data visualization, dimension reduction, and coding)

Handwritten digit recognition

Problem. Given a set of labeled digits (in image format)



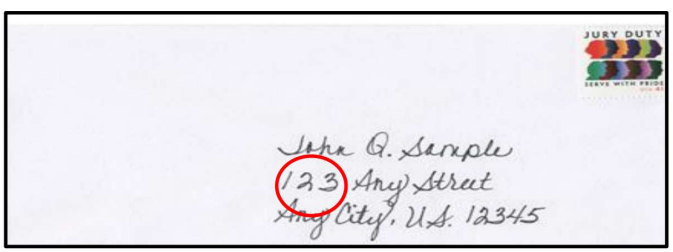
determine by machine what digits the new images contain:

Why digit recognition?

Simple, intuitive to understand, yet practically important

Potential Applications

- **Banking**: Check deposits
- **Surveillance**: license plates
- **Shipping**: Envelopes/Packages



Our main data set: MNIST handwritten digits[4]

It is a benchmark data set for machine learning (due to Yann LeCun), consisting of 70,000 handwriting examples of approximately 250 writers:

- Black/white images of size 28×28
- 60,000 for training and 10,000 for testing

[4] `http://yann.lecun.com/exdb/mnist/`

Why MNIST?

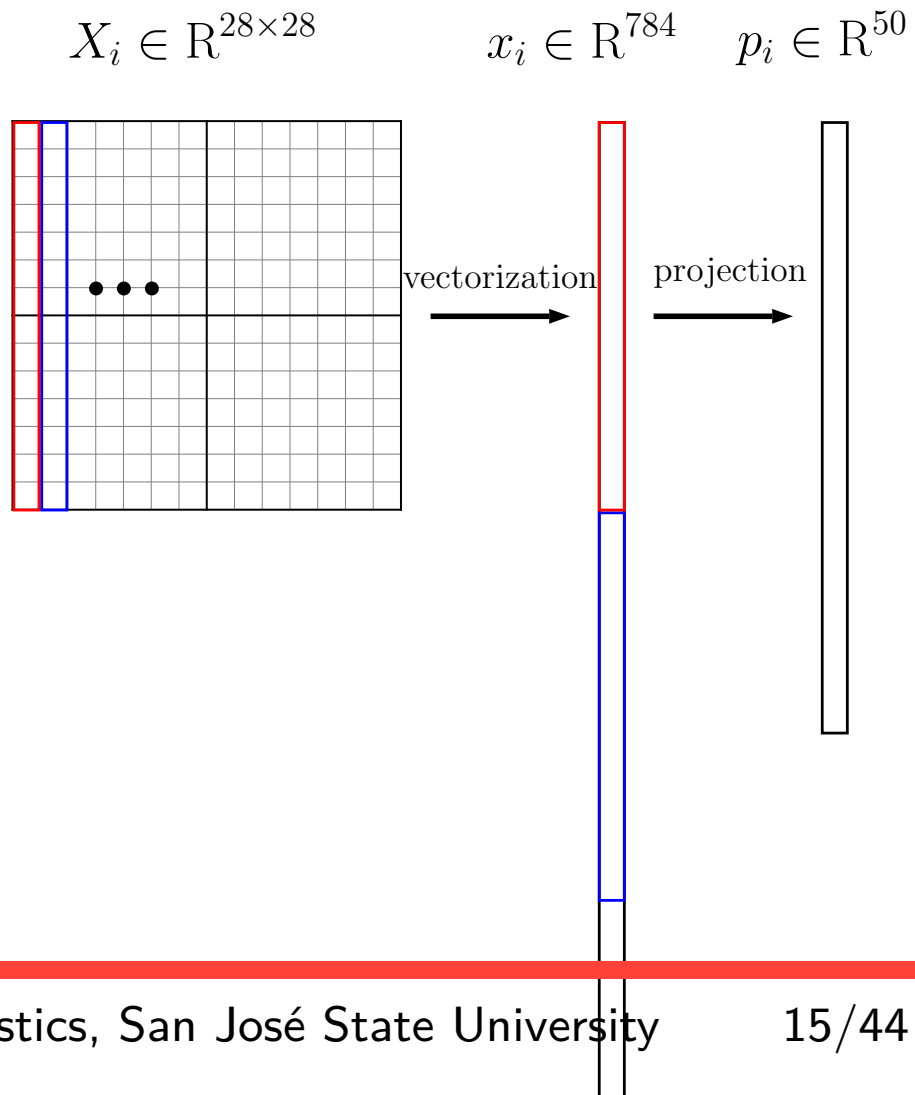
- Well-known
- Simple to understand and easy to use
- But difficult enough for classification
  - Big data (large size and high dimensionality)
  - 10 classes in total $(0, 1, \ldots, 9)$
  - Great variability (due to different ways people write)
  - Nonlinear separation between different classes

- Well studied (thus lots of resources available)
  - The Kaggle competition page[5]
  - Lecun's webpage[6]
  - Math 203 course page from Fall 2015[7]
  - Math 285 course page from Spring 2016[8]

[5] https://www.kaggle.com/c/digit-recognizer
[6] http://yann.lecun.com/exdb/mnist/
[7] http://www.sjsu.edu/faculty/guangliang.chen/Math203F15.html
[8] http://www.sjsu.edu/faculty/guangliang.chen/Math285S16.html

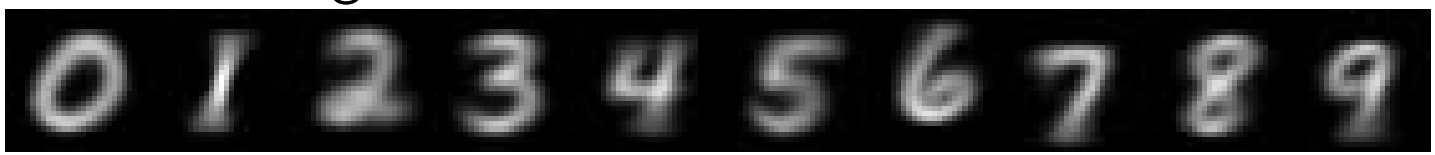
Representation of the digits

- The original format is matrix (of size 28×28);
- Can be converted to vectors (784 dimensional), required by most algorithms.
- Due to high dimensionality, we can further project the data into a low dimensional space.



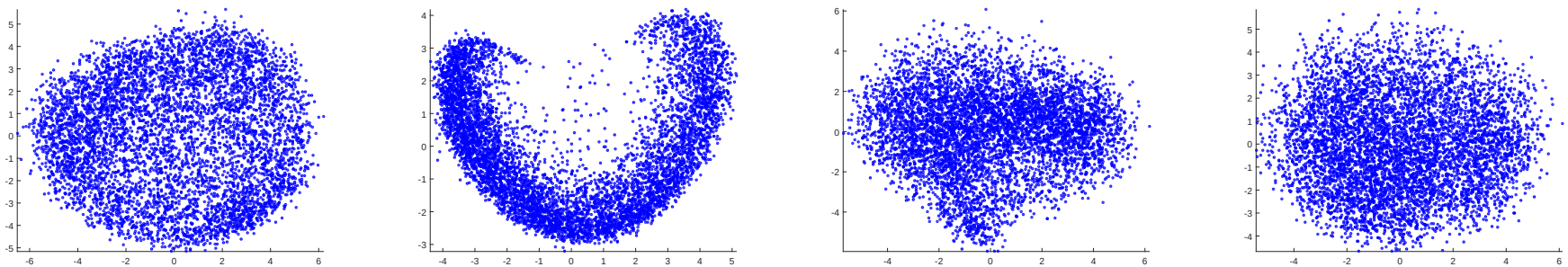
Visualization of the data set

1. The "average" writer

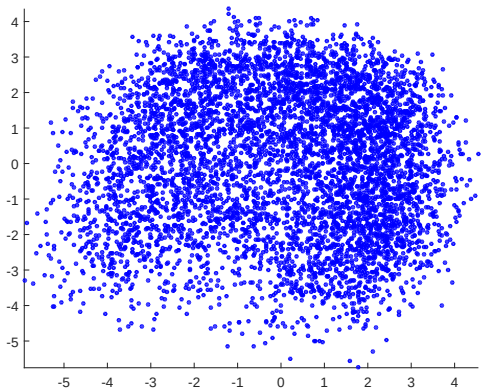
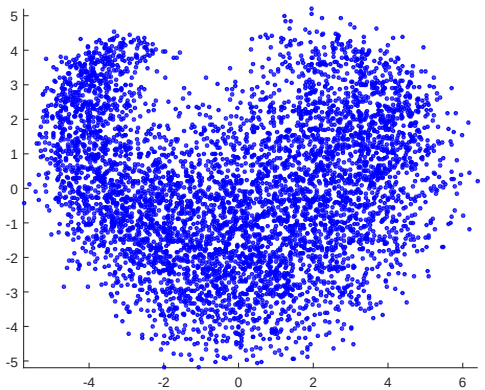
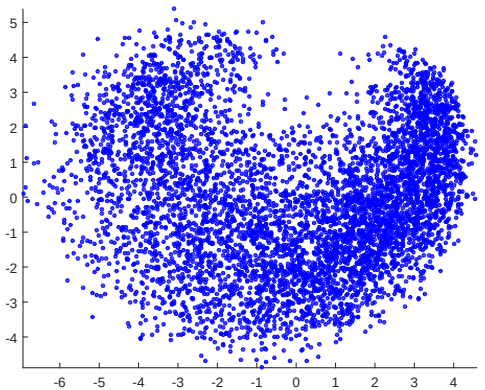


2. Two-dimensional visualization of each digit class

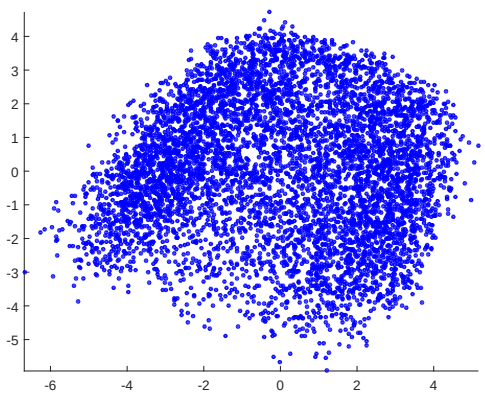
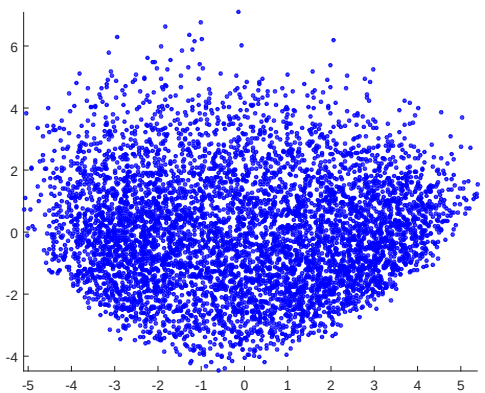
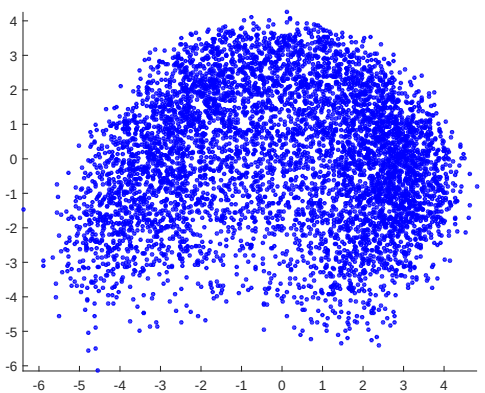
0 - 3



4-6

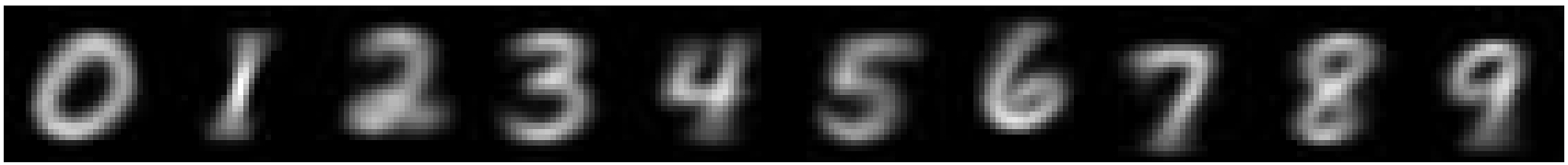




7-9

A very first attempt at classification

Assign labels to test images based on the **closest class centroid** (under some given metric, e.g., Euclidean):



We call this classifier the nearest centroid classifier.

How good is it (under the Euclidean metric): 17.97% error rate (i.e. 1,797 errors out of 10,000)

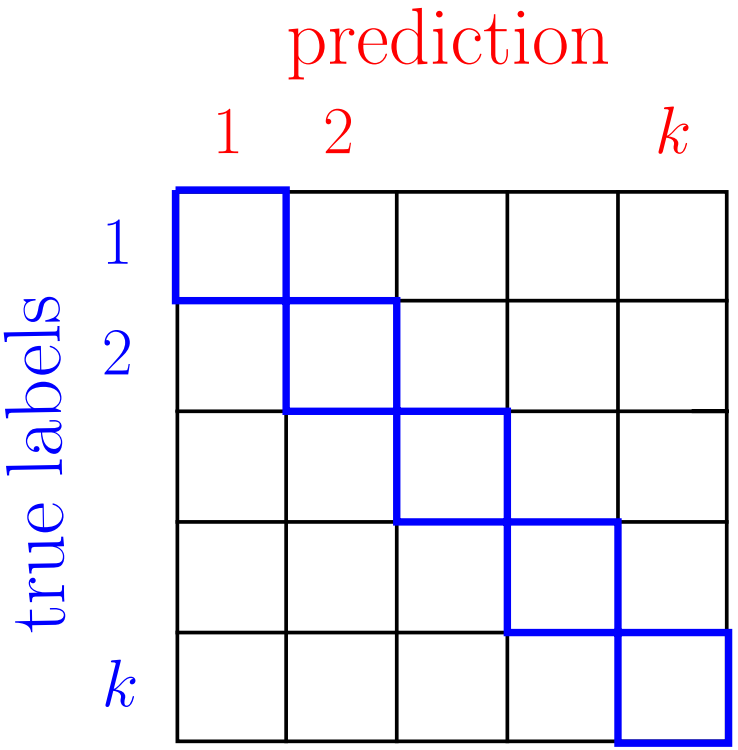
This will be our first baseline.

Evaluation criteria

- **Test error**

$$= \frac{\#\text{misclassified points}}{\#\text{all test points}}$$

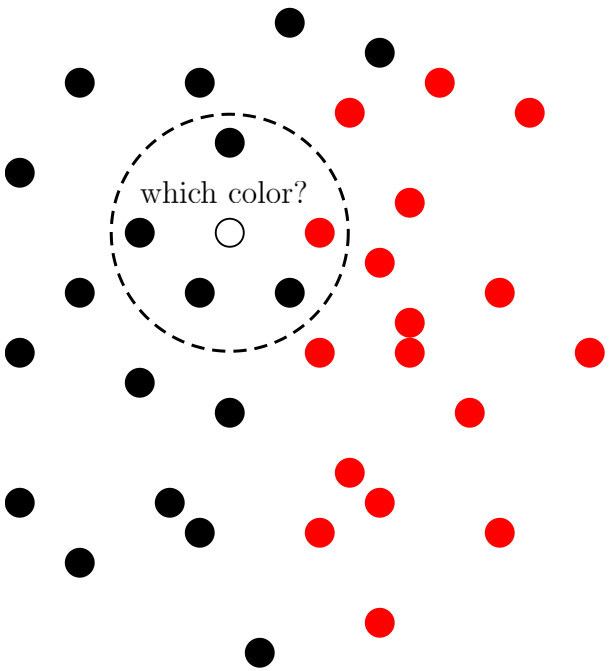
- **Confusion matrix** $\longrightarrow\longrightarrow$
- **Running time**: CPU time, or wall clock time

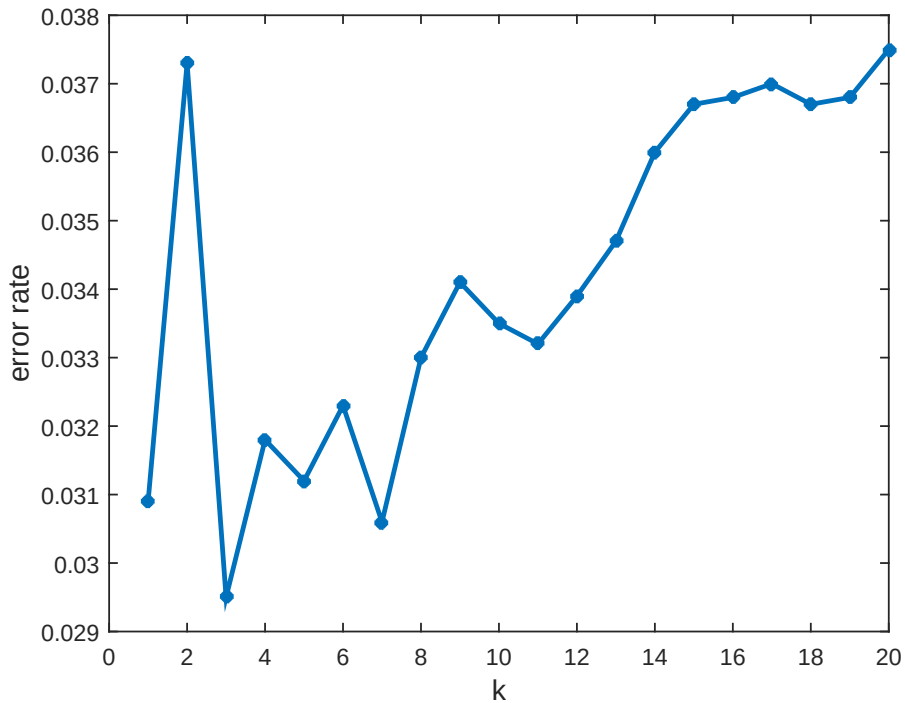


true	**prediction**									
labels	0	1	2	3	4	5	6	7	8	9
0	878	0	7	2	2	58	25	1	7	0
1	0	1092	10	3	0	7	3	0	20	0
2	19	71	781	33	31	3	23	18	50	3
3	4	24	25	814	1	49	8	15	58	12
4	1	22	2	0	811	3	16	1	10	116
5	11	63	2	118	21	612	27	10	13	15
6	18	27	22	0	31	32	827	0	1	0
7	2	59	22	1	20	2	0	856	13	53
8	14	39	11	83	12	36	13	10	718	38
9	15	22	7	10	83	12	1	27	18	814

A second attempt

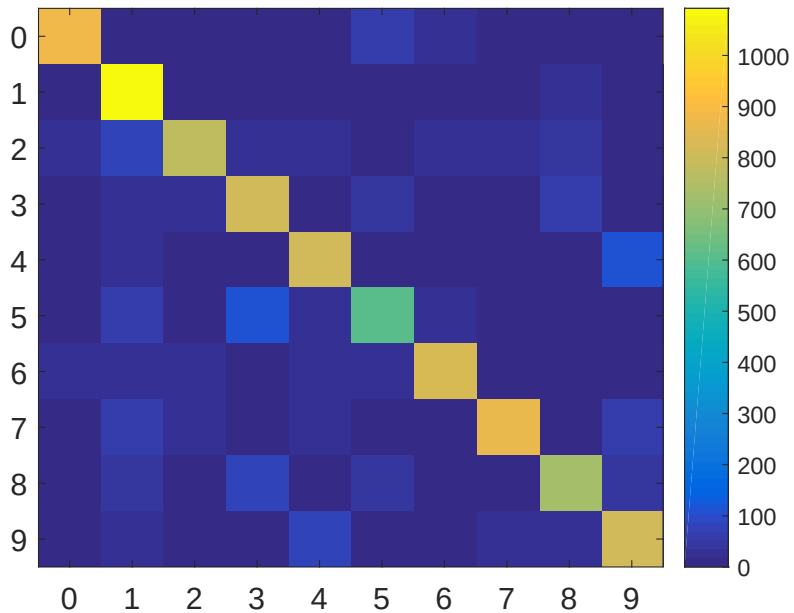
The k nearest neighbors (kNN) classifier assigns labels based on a majority vote around each test point. $\longleftarrow$ Test error $= 2.94\%$ (when $k = 3$)

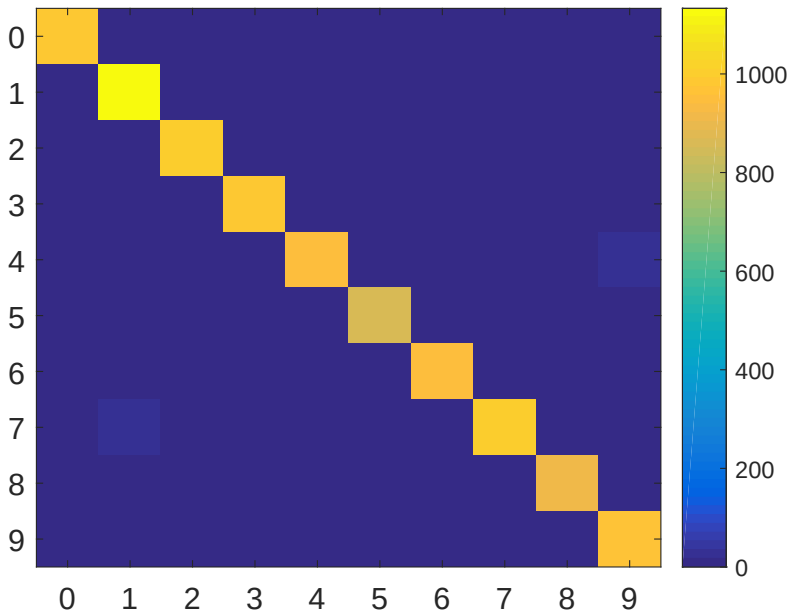




true	**prediction**									
labels	0	1	2	3	4	5	6	7	8	9
0	974	1	1	0	0	1	2	1	0	0
1	0	1133	2	0	0	0	0	0	0	0
2	10	9	993	2	1	0	0	15	2	0
3	0	2	4	974	1	15	1	7	3	3
4	0	6	0	0	951	0	4	2	0	19
5	4	1	0	9	2	863	5	1	3	4
6	4	3	0	0	4	3	944	0	0	0
7	0	21	4	0	1	0	0	992	0	10
8	5	3	4	11	8	15	6	4	914	4
9	3	5	1	6	9	5	1	9	2	968

Confusion matrices displayed as images





More performance metrics

The overall test error is a not a good measure when having imbalanced classes, as the large classes can dominate the small ones (think of a situation where there are two classes with size ratio: 9 to 1).

We will cover the following performance metrics later:

- Precision, Recall or Sensitivity, Specificity
- F1 score
- AUC (area under the ROC curve)

The statistical perspective of classification

Let $\vec{X}, Y$ be two random variables representing the location and label of a data point to be observed. Suppose they have a joint distribution $f_{\vec{X},Y}$, and marginal distributions $f_{\vec{X}}$ (continuous), and f_Y (discrete).

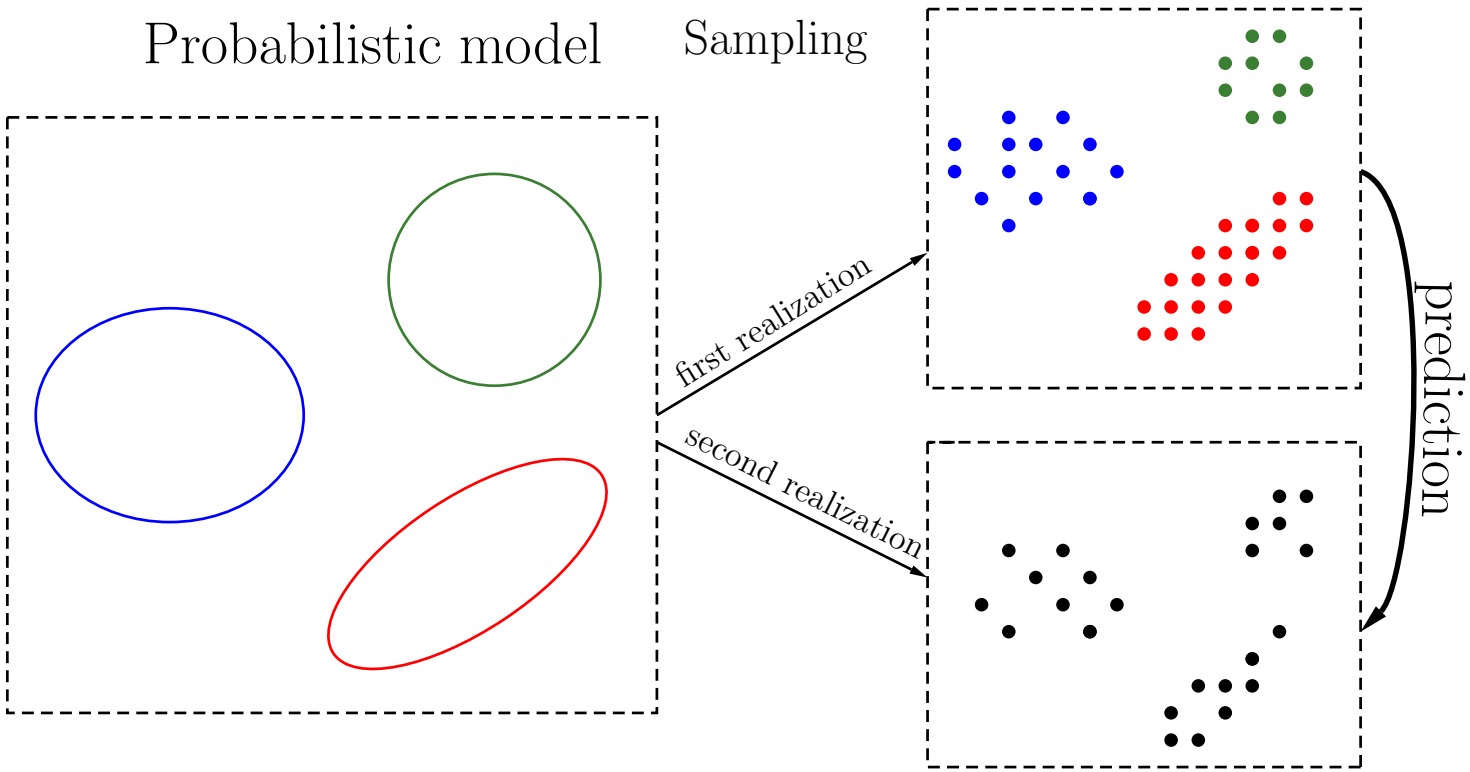
The training data can be modeled by a random sample from the joint distribution :

$$(\vec{X}_1, Y_1), \ldots, (\vec{X}_n, Y_n) \overset{iid}{\sim} f_{\vec{X},Y}$$

The test data is an independent sample $\vec{X}_{n+1}, \ldots, \vec{X}_{n+m}$ from the marginal distribution of $\vec{X}$:

$$\vec{X}_{n+1}, \ldots, \vec{X}_{n+m} \overset{iid}{\sim} f_{\vec{X}}$$

The problem of classification is thus to predict the value of the label Y for each of the test point locations $\vec{X}_{n+j},\ 1 \leq j \leq m$.



Model-based classification using Bayes decision rule

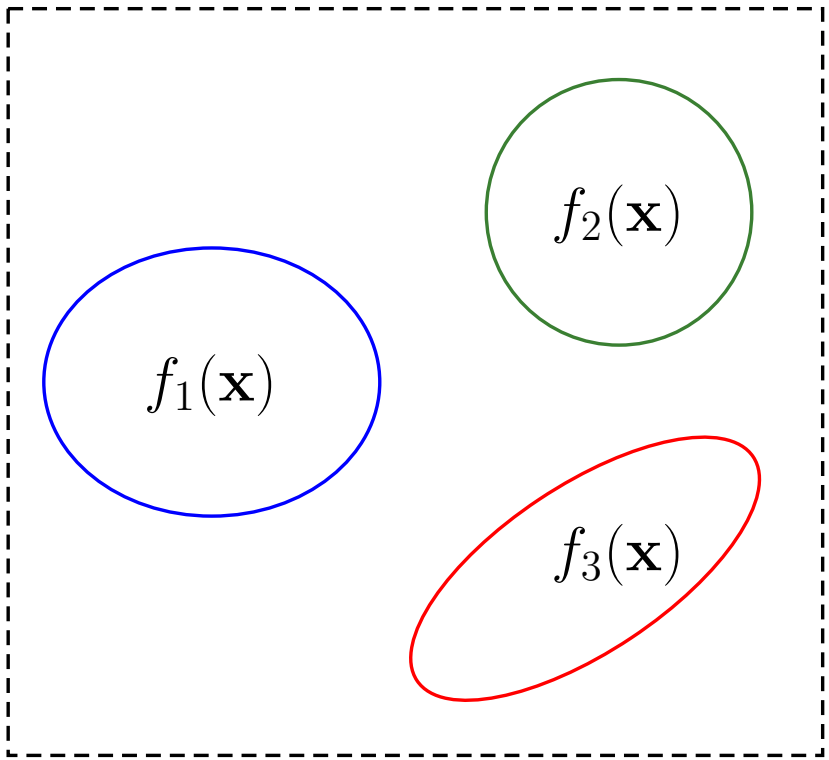
Suppose that for each class j,

$$P(Y = j) = \pi_j$$
$$f(\mathbf{x} \mid Y = j) = f_j(\mathbf{x})$$

both estimated from training data.

Given a new data point $\mathbf{x}$, one can then assign the label based on the posterior probabilities

$$\boxed{\hat{j} = \text{argmax}_j P(Y = j \mid \vec{X} = \mathbf{x})}$$



Classifiers to be covered in this course

- Instanced-based classifiers: kNN and its variants
- Bayes classifiers: LDA/QDA, Naive Bayes
- Logistic regression
- Support vector machine
- Ensemble methods: trees, bagging, random forest, and boosting
- Neural networks

Software requirement

Most classifiers are already implemented in the following languages (via certain toolbox or library):

- **MATLAB**: Statistics and Machine Learning Toolbox
- **Python**: scikit-learn[9]

Proficiency in at least one of the programming languages is required.

You are free to choose either or both to use for your homework and project.

[9] `https://scikit-learn.org/stable/`

Textbook

There is no required textbook, but the following are optional readings:

- James, Witten, Hastie and Tibshirani (2015), "An Introduction to Statistical Learning with Applications in R", 6th edition, Springer.[10]
- Hastie, Tibshirani, and Friedman (2009), "The Elements of Statistical Learning: Data Mining, Inference, and Prediction", 2nd edition, Springer-Verlag.[11]

[10] Freely available online at `https://www.statlearning.com`

[11] Freely available online at `http://statweb.stanford.edu/~tibs/ElemStatLearn/index.html`

- Michael A. Nielson (2015), "Neural Networks and Deep Learning", Determination Press.[12]
- Goodfellow, Bengio, and Courville (2016), "Deep Learning", MIT Press.[13]

Meanwhile, lecture slides by the instructor and material from other sources will be posted on the course webpage.[14]

[12] Freely available online at `http://neuralnetworksanddeeplearning.com/`
[13] Freely available online at `http://www.deeplearningbook.org`.
[14] `https://www.sjsu.edu/faculty/guangliang.chen/Math251.html`.

Requirements of this course

- Homework (30%): Will be assigned regularly
- A midterm exam (35%): Tuesday, October 25, in class.
- A project (35%): You will need to give two presentations to present your problem and results (10%) and additionally complete a report (25%) to describe all the technical details.

You are also expected to attend all classes and actively participate in various course activities.

Homework assignments

... will typically contain both theory and programming questions.

- Please present your work in a clear, logical, organized way.
- For programming questions, you must include your code. Submit both the results (in presentation format) and executable code to Canvas.
- You may collaborate with each other on homework but must write independent codes and solutions (Cheating in any form will be reported to the Office of Student Conduct per SJSU policy).
- You must submit homework on time in order to receive full credit [15]

[15] Late submissions within 24 hours of the due time can still be accepted but will lose 10% of the total grade automatically.

The midterm exam

The midterm exam, to be conducted in this classroom, will cover the conceptual and mathematical aspects of the course.

No make-up exam will be given if you miss the midterm exam unless you have a legitimate excuse such as illness or other personal emergencies and can provide documented evidence.

The course project

The class will be divided into groups of size 2 or 3 to work on the project.

The data set used in your project must be sufficiently large and complex.

Both the group and data set used require advanced approval by the instructor. Preliminary deadline: Thursday, 9/15.

Each group needs to give **two presentations** during the semester:

- Presentation 1 (problem and data set): October 4, Tuesday
- Presentation 2 (final results): December 13, Tuesday, 9:45am–12pm

and additionally write a **technical report** to present all the details.

Grade cutoffs

...will be determined by combining the following **percentages**:

- A+: 97%, A: 93%, A-: 90%
- B+: 86%, B: 80%, B-: 76%
- C+: 73%, C: 68%, C-: 65%
- D: 60%
- F: 59% or less

and **the actual distribution of the class** at the end of the semester.

Academic dishonesty

Students who are suspected of copying homework or cheating during an exam will be referred to the Student Conduct and Ethical Development office and depending on the severity of the conduct, will receive a zero on the assignment or a grade of F in the course.

Learning management system

I will use **Canvas** in various ways:

- Post homework assignments and tests
- Record homework and test scores
- Make announcements (e.g. reminders, clarifications, deadline changes)

Make sure to check your Canvas settings to receive timely notifications. Also, check if your email address in record is still good.

Course webpage

I am maintaining a course webpage[16] for posting the lecture slides and other learning resources.

Please visit the webpage before each class to download the corresponding slides (try refreshing your browser if you don't see them).

[16] `https://www.sjsu.edu/faculty/guangliang.chen/Math251.html`

Piazza

This term we continue using Piazza[17] for online discussions. The system is highly catered to getting you help fast and efficiently from classmates and the instructor.

Rather than emailing questions to me, I encourage you to post your questions on Piazza. If you have any problems or feedback for the developers, email team@piazza.com.

[17] `https://piazza.com/class/l6qzsfu3n7p7gi`

Instructor availability

- **Office hours**: TR 12:20-1:20pm, W 4-5pm (Zoom: 422 306 1605), and by appointment.
- **Piazza**: `https://piazza.com/class/l6qzsfu3n7p7gi`
- **Email**: `guangliang.chen@sjsu.edu`. I check my emails frequently, but you should allow a turnaround time of up to 24 hours (on weekdays) or 48 hours (during weekends).

Special accommodations

If you anticipate needing any special accommodation during the semester (e.g., you have a disability registered with SJSU's Accessible Education Center), please let me know as soon as possible.

Student feedback

I strive to teach in the best ways to facilitate your learning. To achieve this goal, it is very helpful for me to receive timely feedback from you.

You can choose to

- talk to me in person, or
- send me an email, or,
- submit your feedback anonymously through `http://goo.gl/forms/f0wUD5aZSK`.

Questions?