Math 251: Statistical & Machine Learning Classification

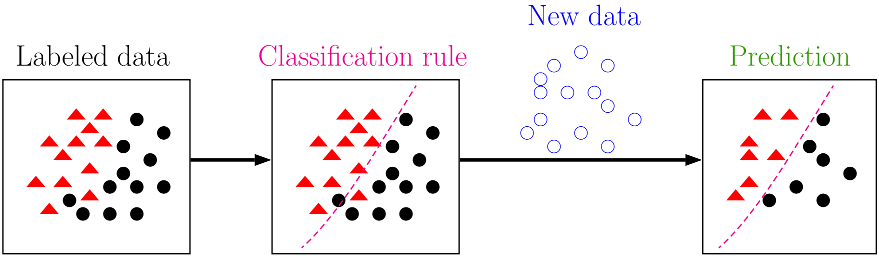
Course introduction and overview (Fall 2018)

Dr. Guangliang Chen

Department of Math & Statistics
San José State University

What is this course about?

An introduction to the machine learning field of classification, the task of assigning labels to new data based on a given set of labeled data.



Terminology:

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

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

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

Major subfields of machine learning:

- Supervised learning (with labeled data)
  - Regression
  - Classification
- Unsupervised learning (no labeled data)
  - Dimensionality reduction
  - Clustering

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

Rational of this course

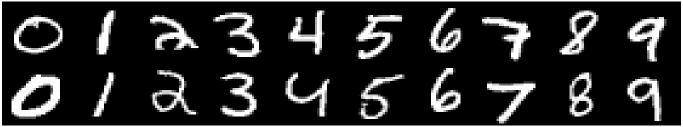
- Learn a subject (**classification**)
- ...through an application (**digits recognition**)
- ...using a benchmark dataset (**MNIST Handwritten Digits**)
- ...via a technical computing language (**MATLAB, R, Python**)
- ...with a data science competition flavor (**Kaggle**).

Goals of this course

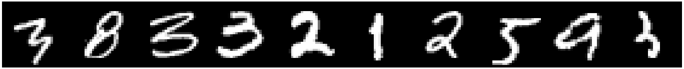
- Introduce the machine learning task of classification with applications
- Present the main ideas and necessary theory of major classification methods in the literature
- Teach how to use specialized software to perform classification tasks while adequately addressing practical challenges (e.g., parameter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

Handwritten digit recognition

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



determine by machine what digits the new images contain:

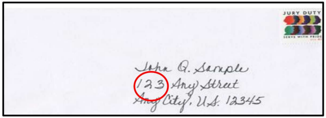


Why digit recognition?

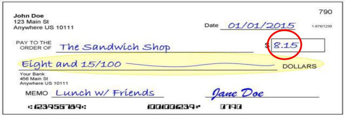
Simple, intuitive to understand, yet practically important

Potential Applications

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


JURY DUTY
SERVE WITH PRIDE

John Q. Sample
123 Any Street
Any City, U.S. 12345



John Doe
123 Main St
Anywhere US 10111

790

Date 01/01/2015 1-678/1239

PAY TO THE ORDER OF The Sandwich Shop $ 8.15

Eight and 15/100 DOLLARS

Your Bank
456 Main St
Anywhere US 10111

MEMO Lunch w/ Friends Jane Doe

⑆123456789⑆ 1001001234⑈ 0790

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 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

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

[5] https://www.kaggle.com/c/digit-recognizer

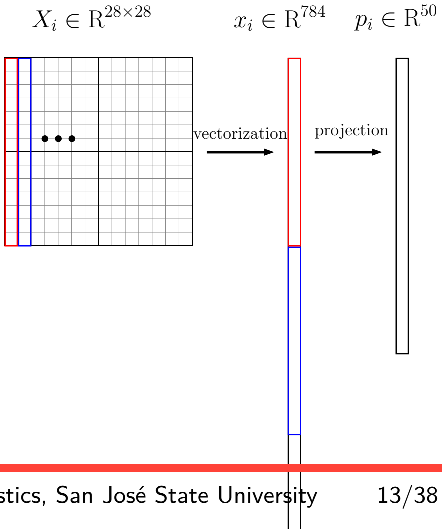
[6] http://yann.lecun.com/exdb/mnist/

[7] http://www.sjsu.edu/faculty/guangliang.chen/Math285S16.html

[8] http://www.sjsu.edu/faculty/guangliang.chen/Math203F15.html

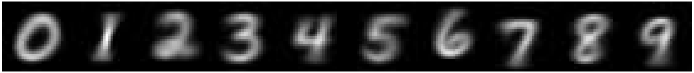
Representation of the digits

- 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.


$X_i \in \mathrm{R}^{28\times 28}$
$x_i \in \mathrm{R}^{784}$
$p_i \in \mathrm{R}^{50}$
vectorization
projection

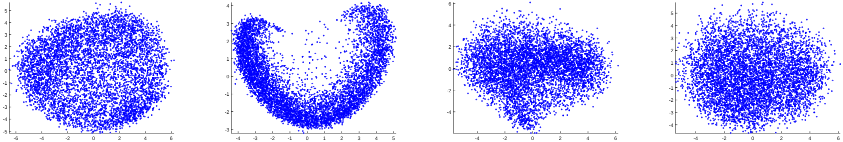
Visualization of the data set

1. The "average" writer



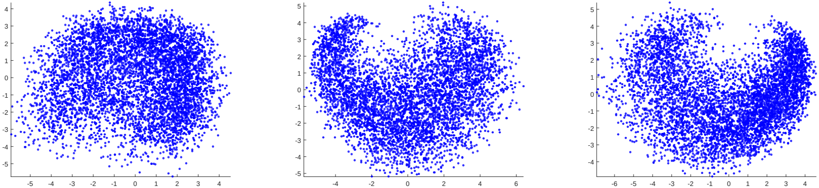
2. The full appearance of each digit class

0 - 3

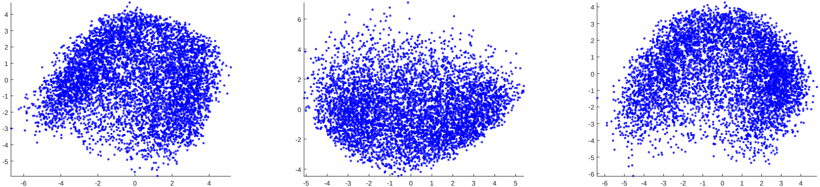


(cont'd on next page)

4-6

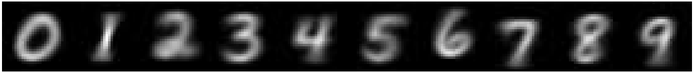

4
3
2
1
0
-1
-2
-3
-4
-5
-5 -4 -3 -2 -1 0 1 2 3 4
5
4
3
2
1
0
-1
-2
-3
-4
-5
-4 -2 0 2 4 6
5
4
3
2
1
0
-1
-2
-3
-4
-6 -5 -4 -3 -2 -1 0 1 2 3 4

7-9


4
3
2
1
0
-1
-2
-3
-4
-5
-6 -4 -2 0 2 4
6
4
2
0
-2
-4
-5 -4 -3 -2 -1 0 1 2 3 4 5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-6 -5 -4 -3 -2 -1 0 1 2 3 4

A very first attempt at classification

Assign labels to test images based on the **closest class centroid**:



We call this classifier the nearest centroid classifier.

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

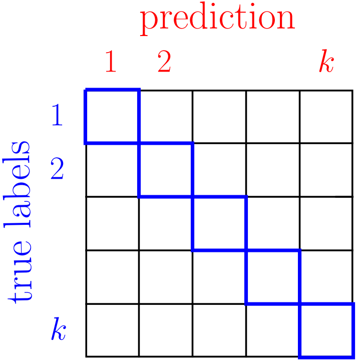
Evaluation criteria

- **Misclassification rate**

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

- **Confusion matrix** ⟶⟶

- **Running time**: CPU time, or wall clock time

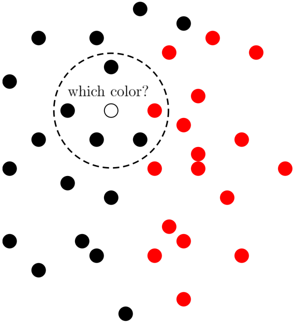


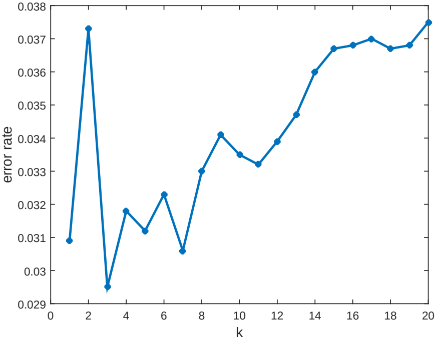
Course introduction and overview

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. $\longleftarrow$ Error rate $= 0.0294$ (when $k = 3$)

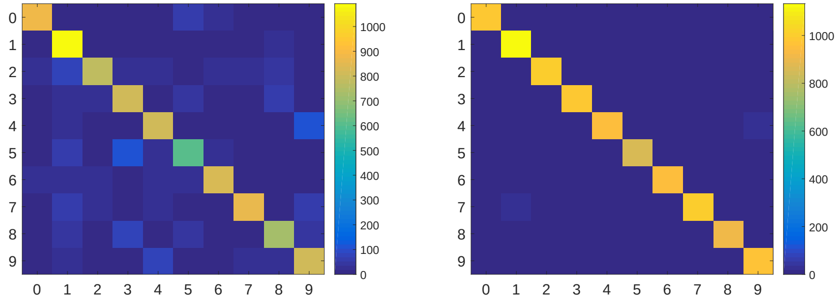




.

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



.

Classifiers covered in this course

- Dimensionality reduction: PCA, Fisher's discriminant analysis, 2DLDA
- Instanced-based classifiers: kNN and variants
- Maximum a posteriori classification: LDA/QDA, Naive Bayes
- Logistic regression
- Support vector machine
- Ensemble methods: trees, bagging, random forest, and boosting
- Neural networks

Additional data to be used

- Small toy data sets created by the instructor
- Data sets from UC Irvine Machine Learning Repository[9] (e.g. *iris*)
- Other digits data[10] (where more data are also available)
  - USPS digits (7,291 + 2,007 = 9298 images of size 16×16)
  - pendigits (7,494 + 3,498 = 10,992 images of size 4×4)

[9] `http://archive.ics.uci.edu/ml/datasets.html`

[10] `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`

Prerequisites of the course

- Math 32 multivariable calculus (method of Lagrange multipliers)
- Math 129A linear algebra (still need to learn more)
- Math 164 mathematical statistics (Bayes' rule and MLE)
- Math 267A or CS 122 or equivalent (programming is crucial)

The programming language

Most classifiers are already coded in any of the following languages (via certain toolbox such as *scikit-learn* in Python):

- **Matlab** (what I am good at)
- **R** (what most of you are familiar with) ⟵ also used by textbook
- **Python** (what we should really use)

You are strongly recommended to form study groups to learn programming from each other.

I may ask some of you to teach programming in class.

Other traits of successful students

- Hard work
- Strong motivation
  - Eager to explore new things
  - Willing to go beyond requirements
- Good communication skills
- Collaborative spirit

Remember that **this is not a traditional course**!

Requirements of this course

- Weekly homework assignments (40%)
- A midterm exam (30%): Monday, Oct. 15
- A final project (30% = 10% oral + 20% report)

Additionally, you are expected to attend all classes and actively participate in classroom activities.

Classroom activities

I plan to embed some classroom activities into the course (e.g., student presentations, or group work). And you will need a laptop to perform them.

So bring your laptop to class every time and if you don't have one, let me know (this classroom has some spare laptops that can be checked out).

Sometimes, you may be chosen to present your homework, or teach something (e.g., a new method, or programming). I will let you know beforehand.

Homework

The homework assignments will typically contain both theory and programming questions.

- You must submit homework on time in order to receive full credit (late homework will receive a 20% penalty for each extra day) .
- You may collaborate on homework but you must write independent codes and solutions.
- You must type your homework and include necessary visuals (e.g., figures, tables).
- Cheating in any form will be reported to the Office of Student Conduct per SJSU policy.

What is allowed

Collaboration is encouraged on homework only for the learning part. This includes (no acknowledgment needed in the following cases):

- Discuss homework questions;
- Come up with a solution together;
- Help each other with certain step or line of code;
- Compare answers/results with each other

You are still required to write independent code and solution (as it is still individual work, not group work).

What is considered cheating

- Copy other people's work
- Use other people's work (such as plots and code) as your own submission (even with acknowledgment)
- Give your work to other people for copying or submission
- One person does all the typing/plotting and shares it with others
- Copy solution or code found online (even with acknowledgment).[11]

[11] However, you can study it and after you fully understand it, rewrite it completely using your own language.

The final project

The course will end with a final project to be selected between each individual student and the instructor (by October 31).

The students will need to give a **10-minute oral presentation** in class to report their findings and meanwhile write **a report of 5+ pages**.

Both the presentation and report will be graded based on correctness, clarity, depth, completeness, and originality.

More details will be given after the midterm.

Textbook

Required Textbooks:

- James, Witten, Hastie and Tibshirani (2015), "An Introduction to Statistical Learning with Applications in R", 6th edition, Springer.[12]
- Michael A. Nielson (2015), "Neural Networks and Deep Learning", Determination Press.[13]

[12] Freely available online at http://www-bcf.usc.edu/~gareth/ISL/

[13] Freely available online at http://neuralnetworksanddeeplearning.com/

Optional Reading:

- Hastie, Tibshirani, and Friedman (2009), "The Elements of Statistical Learning: Data Mining, Inference, and Prediction", 2nd edition, Springer-Verlag.[14]

Meanwhile, lecture slides by the instructor and material from other sources (websites, papers, etc.) will be provided from time to time in class.

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

Some final reminders

This course is

- new (subject to changes as needed)
- challenging (theory and/or programming)[15]
- demanding
- highly rewarding
- very practical and useful

Lastly, this is not an ordinary classroom!

[15] However, the level is still similar to other electives, such as Math 263 and 264.

Assignments

- Install/check software (MATLAB, R, or Python).
- Download all the data sets mentioned in this presentation, and try to understand them as much as you can.
- Explore the Kaggle and old course sites.
- Form study groups based on selected programming language (I will send out another form for you to sign up).
- Do HW0 (due September 5, Wed.)

Waitlist policy

To request an add code, you need to

- **write down your name and sign** on the attendance sheet and
- **complete the background survey**[16] as soon as possible.

I will rely on the survey result to determine your eligibility and give out add codes (when seats become available).

[16] https://goo.gl/forms/6C5WOP7Adq0ks5Zb2

Questions?