ECE-175A

Elements of Machine Intelligence - I

Nuno Vasconcelos
(Ken Kreutz-Delgado)

ECE Department, UCSD
The course

- The course will cover basic, but important, aspects of machine learning and pattern recognition.

- We will cover a lot of ground, at the end of the quarter you’ll know how to implement a lot of things that may seem very complicated today.

- Homework: 20%.

- Exams: 1 mid-term, date TBA- 35%
  1 final – 45% (covers everything)
Homework policy

Homework is individual

Homework sets

- include pen+paper and computer problems
- pen+paper are not due, for practice only
- computer problems are due
- issue and due dates specified on course web site
- it is your responsibility to keep track of the dates

Homework is due on the specified due-date. No exceptions, unless there is an extension to the entire class.

The final course grade will be curve-based.

- do not get discouraged if a homework problem is hard, just put some extra effort into solving it. Try your best.
Academic integrity

» is taken very seriously at UCSD and in this course
» all work done for grade must be individual
» not allowed to
  • talk to friends or classmates about graded problems
  • use any ECE175A material not explicitly handed out by us
  • this includes consulting any websites other than the class web site, piazza, or canvas

» be aware that
  • we plant files on websites and can figure out violations
  • we refer violations to UCSD Academic Integrity Office
  • penalty ranges from F in the class to expulsion from UCSD
  • despite these warnings there are always 3-4 cases a year
Homework policies

- homework is **individual**
- two types of problems
  - ungraded
    - you will be given the solutions
    - OK to discuss with anyone
  - graded
    - think of these as part of an exam
    - don’t ask questions you would not ask on exam room
    - don’t ask “does my result look OK?,” or “did anyone get something like this?”
- homework is **due on the dates specified on the website.**
Resources

Course web page:
http://www.svcl.ucsd.edu/~courses/ece175/
- all materials, except homework and exam solutions will be available there.

Discussion forum:
- We will be using piazza. You will get email.

Course Instructor:
- Nuno Vasconcelos, nuno@ece.ucsd.edu, EBU 1-5602
- office hours: TBA
Texts

► Required:
  • “Introduction to Machine Learning”
  • Ethem Alpaydin, MIT Press, 2004

► Various other good, but optional, texts:
  • “Pattern Classification”, Duda, Hart, Stork, Wiley, 2001
  • “Elements of Statistical Learning”, Hastie, Tibshirani, Fredman, 2001

► Prerequisites you must know well:
  • “Linear Algebra”, Gilbert Strang, 1988
Why Machine Learning?

Good question! After all, many systems & processes in the world that are well-modeled by deterministic equations

- E.g. \( f = ma \); \( V = IR \), Maxwell’s equations, and other physical laws.
- There are acceptable levels of “noise”, “error”, and other “variability”.
- In such domains, we don’t need statistical learning.

However, learning is necessary when there is a need for predictions about, or classification of, poorly known and/or random vector data \( Y \), that

- represents important events, situations, or objects in the world;
- which may (or may not) depend on other factors (variables) \( X \);
- is impossible or too difficult to derive an exact, deterministic model for;
Examples and Perspectives

- The “Data-Mining” viewpoint:
  - huge amounts of data that **does not follow deterministic rules**
  - E.g. given an history of thousands of customer records and some questions that I can ask you, how do I predict that you will pay on time?
  - Impossible to derive a theory for this, must be learned

While many associate learning with data-mining, it is by no means the only important application or viewpoint.

- The Signal Processing viewpoint:
  - Signals combine in ways that depend on **“hidden structure”** (e.g. speech waveforms depend on language, grammar, etc.)
  - Signals are usually subject to significant amounts of **“noise”** (which sometimes means “things we do not know how to model”)

Examples - Continued

Signal Processing viewpoint (Cont’d)

• E.g. the Cocktail Party Problem:
  • Although there are all these people talking loudly at once, you can still understand what your friend is saying.
  • How could you build a chip to separate the speakers? (As well as your ear and brain can do.)
  • Model the hidden dependence as
    – a linear combination of independent sources + noise

• Many other similar examples in the areas of wireless, communications, signal restoration, etc.
Examples (cont’d)

- The Perception/AI viewpoint:
  - It is a complex world; one cannot model everything in detail
  - Rely on **probabilistic models** that explicitly account for the variability
  - Use the laws of probability to make inferences. E.g.,
    - $P(\text{burglar} \mid \text{alarm, no earthquake})$ is high
    - $P(\text{burglar} \mid \text{alarm, earthquake})$ is low
  - There is a whole field that studies “perception as Bayesian inference”
  - In a sense, perception really is “confirming what you already know.”
  - priors + observations = robust inference
Examples (cont’d)

The Communications Engineering viewpoint:

• Detection problems:

• You observe Y and know something about the statistics of the channel. What was X?
• This is the canonical detection problem.
• For example, face detection in computer vision: “I see pixel array Y. Is it a face?”
What is Statistical Learning?

- **Goal:** Given a relationship between a feature vector $x$ and a vector $y$, and data samples $(x_i, y_i)$, find an approximating function $f(x) \approx y$

\[
x \xrightarrow{f(\cdot)} \hat{y} = f(x) \approx y
\]

- This is called **training or learning**.

- **Two major types of learning:**
  - **Unsupervised:** only $X$ is known, usually referred to as clustering.
  - **Supervised:** both $X$ and target value $Y$ are known during training, only $X$ is known at test time. Usually referred to as classification or regression.
Supervised Learning

X can be anything, but the type of known data Y dictates the type of supervised learning problem:

- Y in \{0,1\} is referred to as Detection or Binary Classification
- Y in \{0, ..., M-1\} is referred to as (M-ary) Classification
- Y continuous is referred to as Regression

Theories are quite similar, and algorithms similar most of the time.

We will usually emphasize classification, but will talk about regression when particularly insightful.
Example

Classification of Fish:
- Fish roll down a conveyer belt
- Camera takes a picture
- Decide if is this a salmon or a sea-bass?

Q: What is X? E.g. what features do I use to distinguish between the two fish?

This is somewhat of an art-form. Frequently, the best is to ask domain experts.

E.g., expert says use overall length and width of scales.
Classification/Detection

- Two major types of classifiers
- Discriminant: determine the decision boundary in feature space that best separates the classes;
- Generative: fit a probability model to each class and then compare the probabilities to find a decision rule.
- A lot more on the relationship between these two approaches later!
Caution

- How do we know learning has worked?
- We care about generalization, i.e. accuracy outside the training set
- Models that are “too powerful” can lead to over-fitting:
  - E.g. in regression one can always exactly fit n pts with polynomial of order n-1.
  - Is this good? how likely is the error to be small outside the training set?
  - Similar problem for classification
- Fundamental Rule: only hold-out test-set performance results matter!!!
Generalization

- Good generalization requires controlling the trade-off between training and test error:
  - training error large, test error large
  - training error smaller, test error smaller
  - training error smallest, test error largest

- This trade-off is known by many names

- In the generative classification world it is usually due to the bias-variance trade-off of the class models
Generative Model Learning

- Each class is characterized by a probability density function (class conditional density),
- A model is adopted, e.g. a Gaussian.
- Training data is used to estimate the class pdfs.
- Overall, the process is referred to as density estimation
- A nonparametric approach would be to estimate the pdfs using histograms:
Decision rules

- Given class pdfs, Bayesian Decision Theory (BDT) provides us with optimal rules for classification.
- “Optimal” here might mean minimum probability of error, for example.

We will

- Study BDT in detail,
- Establish connections to other decision principles (e.g. linear discriminants)
- Show that Bayesian decisions are usually intuitive

Derive optimal rules for a range of classifiers
Features and dimensionality

- For most of what we have said so far
  - Theory is well understood
  - Algorithms are available
  - Limitations can be characterized
- Usually, good features are an art-form
- We will survey traditional techniques
  - Bayesian Decision Theory (BDT)
  - Linear Discriminant Analysis (LDA)
  - Principal Component Analysis (PCA)
- and (perhaps) some more recent methods
  - Independent Components Analysis (ICA)
  - Support Vectors Machines (SVM)
Discriminant Learning

Instead of learning models (pdf’s) and deriving a decision boundary from the model, learn the boundary directly.

There are many such methods. The simplest case is the so-called *separating* hyperplane classifier:

- Simply find the hyperplane that best separates the classes, assuming linear separability:
Support Vector Machines

How do we do this efficiently in high-dimensional feature spaces?

A popular approach is based on the use of support vectors.

- One transforms the data into linearly separable features using kernel functions.
- The best performance is obtained by maximizing the margin.
- This is the distance between hyperplane and closest point on each side.
Support Vector Machine (SVM)

- For separable classes, the training error can be made zero by classifying each point correctly.
- This can be implemented by solving the optimization problem

$$\mathbf{w}^* = \text{arg max}_w \text{ margin } (w)$$

s.t. $x_l$ is correctly classified $\forall l$

- This is an optimization problem with many constraints, not trivial but solvable.
- The resulting classifier is the “support-vector machine”.
- The points on the margin are the “support vectors”.
- What if a hyperplane is not good enough?
Kernels and Linear Separability

- The trick is to map the feature space to a higher dimensional feature space:
  - Non-linear boundary in original space
  - Becomes hyperplane in transformed space
- This can be done efficiently by the introduction of a kernel function
- Classification problem is mapped into a reproducing kernel Hilbert space
- Kernels are at the core of the success of SVM classification
- Most classical linear techniques (e.g. PCA, LDA, ICA, etc.) can be kernel-ized with significant improvement
Unsupervised learning

- So far, we have talked about supervised learning:
  - We know the class of each point

- In many problems this is not the case (e.g. image segmentation)
Unsupervised learning

In these problems we are given X, but not Y

The standard algorithms for this are iterative:

• Start from best guess
• Given Y-estimates fit class models
• Given class models re-estimate Y-estimates

This procedure usually converges to an optimal solution, although not necessarily the global optimum

Performance worse than that of supervised classifier, but this is the best we can do.
Reasons to take the course

► To learn about Classification and Statistical Learning
  • tremendous amount of theory
  • but things invariably go wrong
  • too little data, noise, too many dimensions, training sets that do not reflect all possible variability, etc.

► To learn that good learning solutions require:
  • knowledge of the domain (e.g. “these are the features to use”)
  • knowledge of the available techniques, their limitations, etc.
  • In the absence of either of these, you will fail!

► To learn skills that are *highly valued* in the marketplace!
Any questions?