COMP 790: Machine Learning in Computational Biology

Meet:	Tuesdays and Thursdays 12:30-1:45, FB 007
Instructor:	Vladimir Jojic (vjojic@cs.unc.edu)
Prerequisites:	Probability/Statistics and Linear Algebra
	Some programming (Matlab/R/Python)
Useful:	Generalized Linear Models, Optimization, LaTeX
Web:	`http://www.cs.unc.edu/~vjojic/comp790`

Topics covered

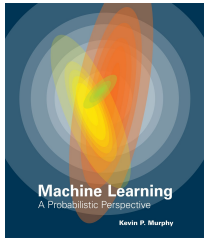
- Linear models in classification and regression
- Subspace models Factor Analysis, Principal Component Analysis, Independent Component Analysis
- Graphical models, Inference, Learning, EM algorithm
- HMM, tree structured models
- Approximate inference techniques: Variational methods and Markov Chain Monte Carlo
- Structure learning in Gaussian models
- Max margin methods

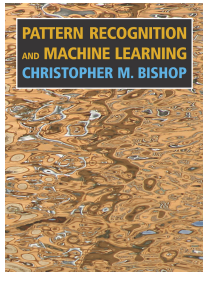
Optional subjects, depending on progress we make

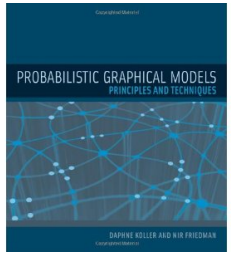
- Bayesian non parametrics (Dirichlet processes, Gaussian processes etc.)
- Random projections and compressed sensing

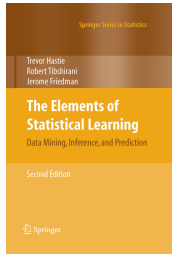
Books

- "Machine Learning: A Probabilistic Perspective," Kevin P. Murphy
- "Pattern Recognition and Machine Learning," Chris M. Bishop
- "Probabilistic Graphical Models," Daphne Koller and Nir Friedman
- "The Elements of Statistical Learning: Data Mining, Inference, and Prediction," T. Hastie, R. Tibshirani, J. Friedman
- "Information Theory, Inference, and Learning Algorithms," David MacKay

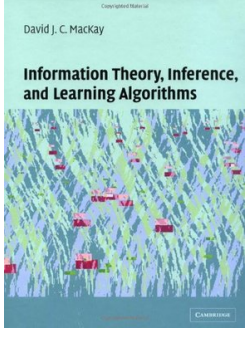








downloadable



downloadable

Credit and Grades

This course is worth 3 credits. Make sure you registered appropriately.

Grading:

- 40% 4 Homework assignments (10% per HW)
- 60% 1 project due on the last day of exams

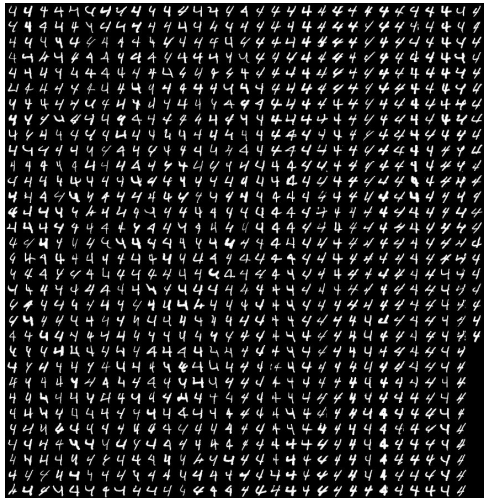
Framing machine learning

Machine Learning is a field that covers a broad set of techniques aimed at "learning" how to accomplish a *task*.

Chief prerequisite for a machine learning application is having *data* illustrating that *task*.

Some examples of tasks and data

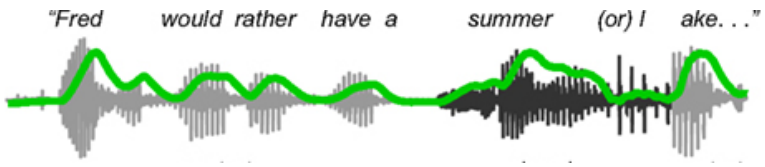
Most frequently referenced and studied task is handwritten digit recognition



Training data: images of handwritten digit 4

Task: given an image of a digit say whether it is 4 or not

Some examples of tasks and data



Training data: segmented speech wave forms and corresponding word

Task: Recognize spoken words from recorded sound

Some examples of tasks and data

Training data: Images of objects

Task: Recognize those objects in an image

Some examples of tasks and data

More complex examples:

1. Order search results or products based on your inferred preferences (google and amazon)
2. Medical diagnosis, imaging analysis, assisted surgery
3. Locating and tracking objects in video
4. Financial market prediction and analysis
5. Automating biological research
6. Predicting voting outcomes

It is essential that you have access to examples of successful execution of these tasks. Plenty of them!

Project: The problem you want to solve

What problem do you want to solve?
Is it an important problem and why?
What do you need to solve this problem?
Has anyone else tried solving this problem?
What is your secret sauce?
How will you know if you did solve the problem?

In next couple of weeks write down several paragraphs answering these (in scientific paper prose.)

For inspiration read: You and Your Research by Richard Hamming

Projects: Writeup

8 pages 11pt document that covers

- The problem and its relevance
- Related and relevant work survey
- The data used in analysis
- The model
- Inference/Learning/Optimization technique
- Evaluation on synthetic and real data
- A discussion of results and method

Quiz: Which of these are you familar with?

Raise your hand as I read off the concepts

- Eigenvectors, eigenvalues, singular value decomposition
- Least squares/linear regression
- p-values, permutation tests
- Multivariate gaussian distribution
- Covariance matrix
- Newton method for optimization
- Categorical, multinomial, Dirichlet distribution

Machine Learning as a different way of developing programs

How do we develop software?

- Specification: function f should return a real value that is three times the input.
- Implementation:

```
function y = f(x)
    y = 3.0*x;
end
```

- Tests: `f(0.0) == 0.0`, `f(NaN) == NaN`, `f(1.0) == 3.0`, `abs(f(1.000000000001) - 3.000000000003)<1e-12` ...

Machine Learning as a different way of developing programs

Suppose that instead of a specification you get a list of input/output pairs – the Data

```
Input (x):  0.2760    0.6797    0.6551    0.1626    ...
Output (y): 0.6147    1.2381    2.3510    0.6959    ...
```

Generate a program that reproduces behavior captured by these pairs.

Machine Learning as a different way of developing programs

First approach: a lookup table.

```
function y = f(x)
    for i=1:length(inputExamples)
        if abs(x-inputExamples(i))<1e-12
            y = outputExamples(i);
            return
        end
    end
    y = NaN;
end
```

What is bad about this solution?
What is good about this solution?

Machine Learning approach

1. Assume a template for the function

   ```
   function y = f(x,beta)
       y = beta*x;
   end
   ```

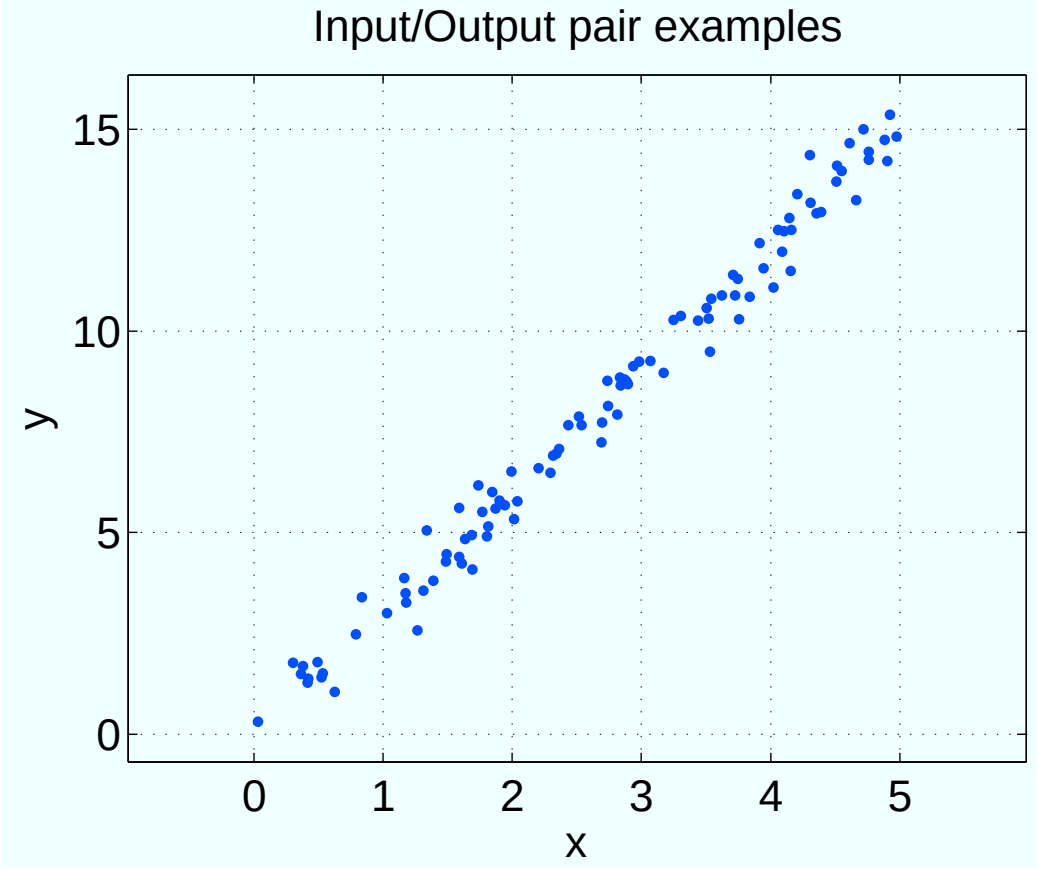
   we call `beta` a **parameter**.

2. Specify a cost $C(\texttt{beta}, \mathrm{Data})$ that tells you how well `f(x,beta)` **fits** the Data.

$$C(\texttt{beta}, \mathrm{Data}) = \sum_{(x,y)\in \mathrm{Data}} (y - \texttt{f}(x, \texttt{beta}))^2$$

3. Find `beta` for which cost $C(\texttt{beta}, \mathrm{Data})$ is the smallest. This is called **learning** or **training**.

Machine Learning as a different way of developing programs



Demo

Machine Learning concepts overview

Machine learning splits into **supervised** and **unsupervised** learning.

In **supervised** learning the data is captured as a list of training pairs (x_i, y_i) and our learning procedure needs to produce a mapping from x_i (features or predictors) into y_i (target or label).

Ex.

$x =$

history of cancer	smoker	gender
Yes	No	Male
No	No	Female
Yes	No	Female
Yes	Yes	Male
No	Yes	Male
...	...	...

$y =$

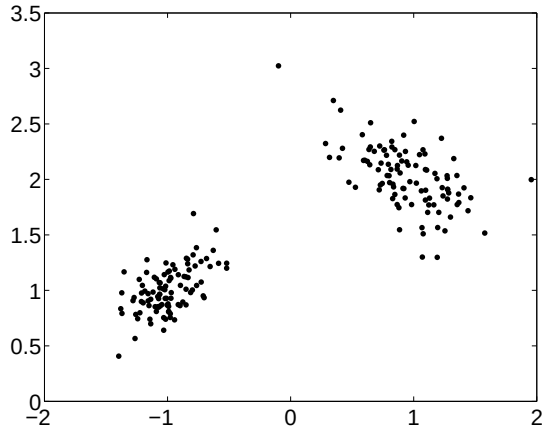
throat cancer
No
No
No
Yes
Yes
...

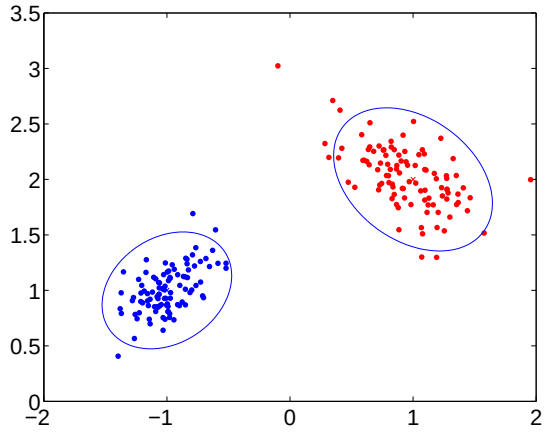
Machine Learning concepts overview

In **unsupervised** learning the data is captured purely in terms of x. The distinction between features and targets melts away.

Instead we learn a joint model of all entries in x

$$x = \begin{bmatrix} 1.0437 & 2.0115 \\ 1.4860 & 1.8677 \\ -1.2375 & 0.8035 \\ -0.5928 & 1.1722 \\ -1.1316 & 1.1769 \\ \dots & \dots \end{bmatrix}$$



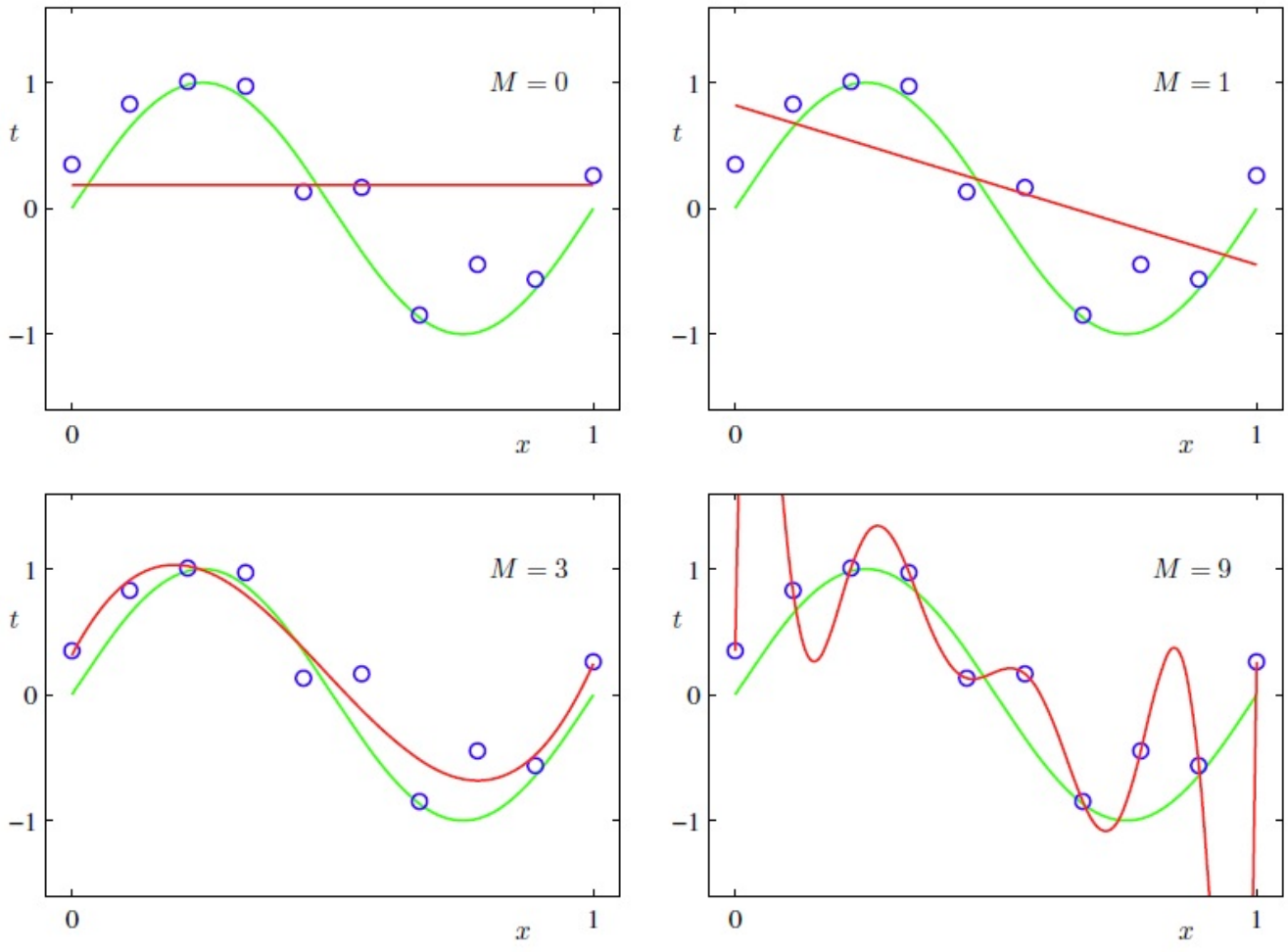


In a joint model any part of the data is a target that can be predicted.

Machine Learning concepts overview - Overfitting

The data are (x, y) coordinates of the blue circles.

We fit polynomials of varying degree
$\sum_{d=0}^{M} a_d x^d = a_0 + a_1 x + a_2 x^2 \ldots$ to the data.



Machine Learning concepts overview

How well does your method perform on new data, data you have not seen during learning?

If you get a new data instance, for example

$$(\text{No history of cancer, Smoker, Male})$$

can you assess how good is your throat cancer predictor?

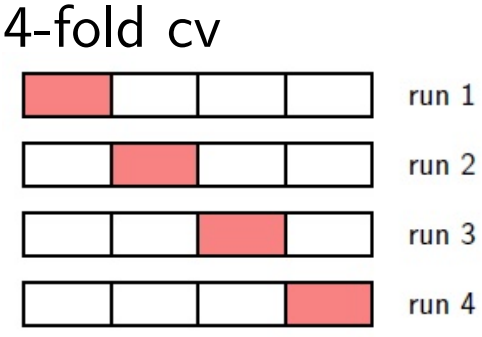
1. Divide data into **training set** and **testing set**.
2. Use **training set** as input to your learning procedure to produce a predictor
3. Use **testing set** to evaluate performance of the resulting predictor.

Most common mistake is to let information about test bleed into train data.

Machine Learning concepts overview

What if data set is too small to split it into test and train?

k-fold Cross-validation: split data into k subsets; then in turn treat each subset as held-out and train on the rest.



We covered

- COMP 790-124
  `http://www.cs.unc.edu/~vjojic/comp790`
- Machine Learning approach to developing algorithms
- Supervised and unsupervised learning
- Overfitting
- Cross-validation