Introduction to Machine Learning and Data Mining Lecture-7: Model Evaluation and Project Topics

Prof. Eugene Chang

Today

- Model evaluation
- Project topics
- Some slides are based on materials from Prof. Andrew Rosenberg, CUNY, Prof. Raymond J. Mooney, University of Texas at Austin, and Prof Jiawei Han, University of Illinois at Urbana-Champaign

How do you know that you have a good classifier?

- Is a feature contributing to overall performance?
- Is classifier A better than classifier B?
- Internal Evaluation:
  - Measure the performance of the classifier.
- External Evaluation:
  - Measure the performance on a downstream task

Basic Evaluation

- Training data – used to identify model parameters
- Testing data – used for evaluation
- Optionally: Development / tuning data – used to identify model parameters.

Cross validation

- Identify *n* “folds” of the available data.
- Train on *n-1* folds
- Test on the remaining fold.

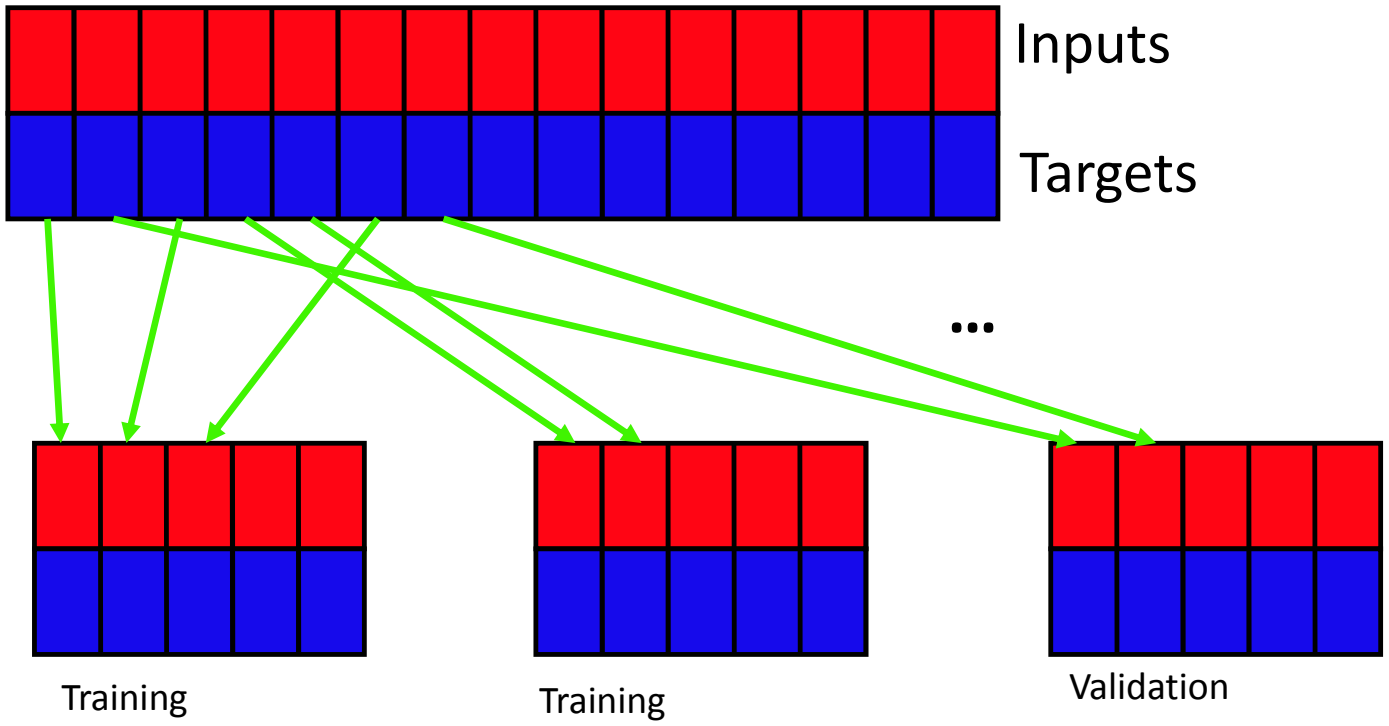
- In the extreme (*n=N*) this is known as “**leave-one-out**” cross validation

- *n*-fold cross validation (xval) gives *n* samples of the performance of the classifier.

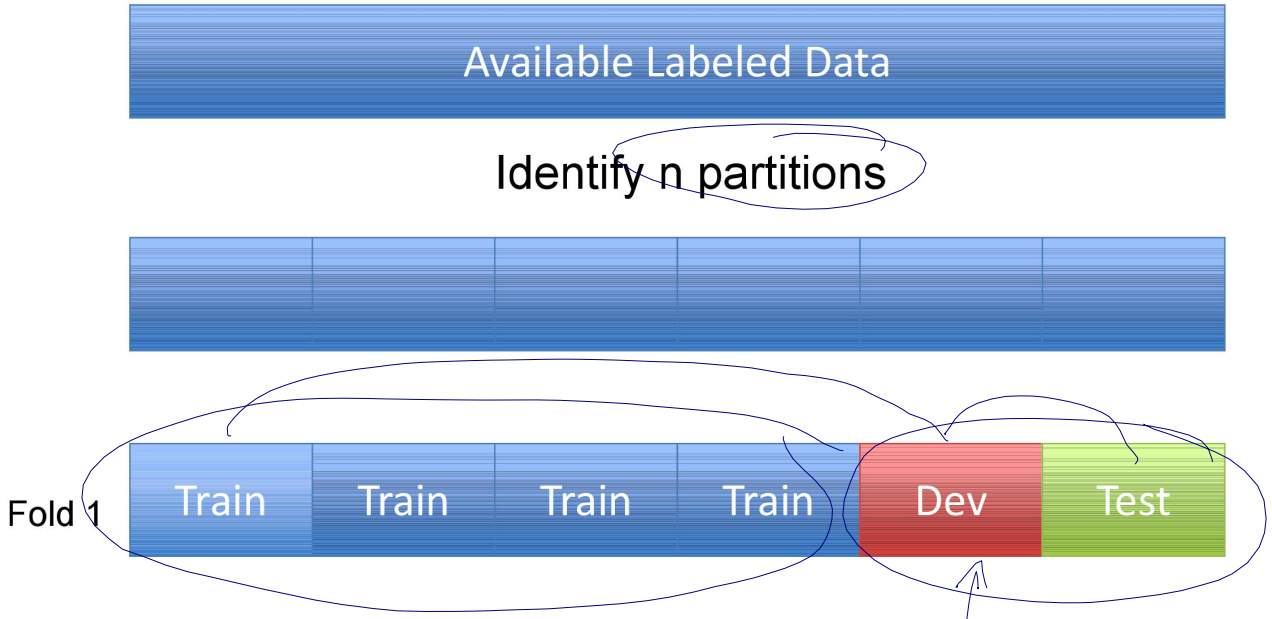
Holdout & Cross-Validation Methods

- **Holdout method**
  - Given data is randomly partitioned into two independent sets
    - Training set (e.g., 2/3) for model construction
    - Test set (e.g., 1/3) for accuracy estimation
  - Random sampling: a variation of holdout
    - Repeat holdout k times, accuracy = avg. of the accuracies obtained
- **Cross-validation** (*k*-fold, where k = 10 is most popular)
  - Randomly partition the data into *k mutually exclusive* subsets, each approximately equal size
  - At *i*-th iteration, use D_i as test set and others as training set
  - Leave-one-out: *k* folds where *k* = # of tuples, for small sized data
  - ***Stratified cross-validation***: folds are stratified so that class dist. in each fold is approx. the same as that in the initial data

Hold Out Cross Validation



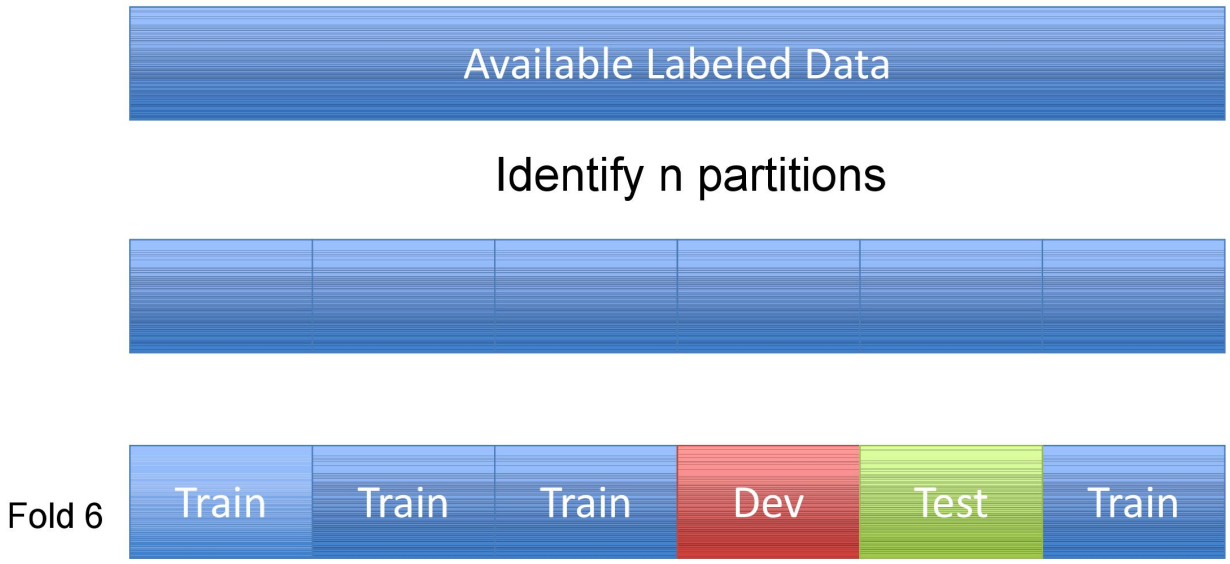
Cross-validation visualized



- loop through this process & choose the combination that has the best score:

Dev consider as Test (pretest) in most case, for tuning purposes

Cross-validation visualized



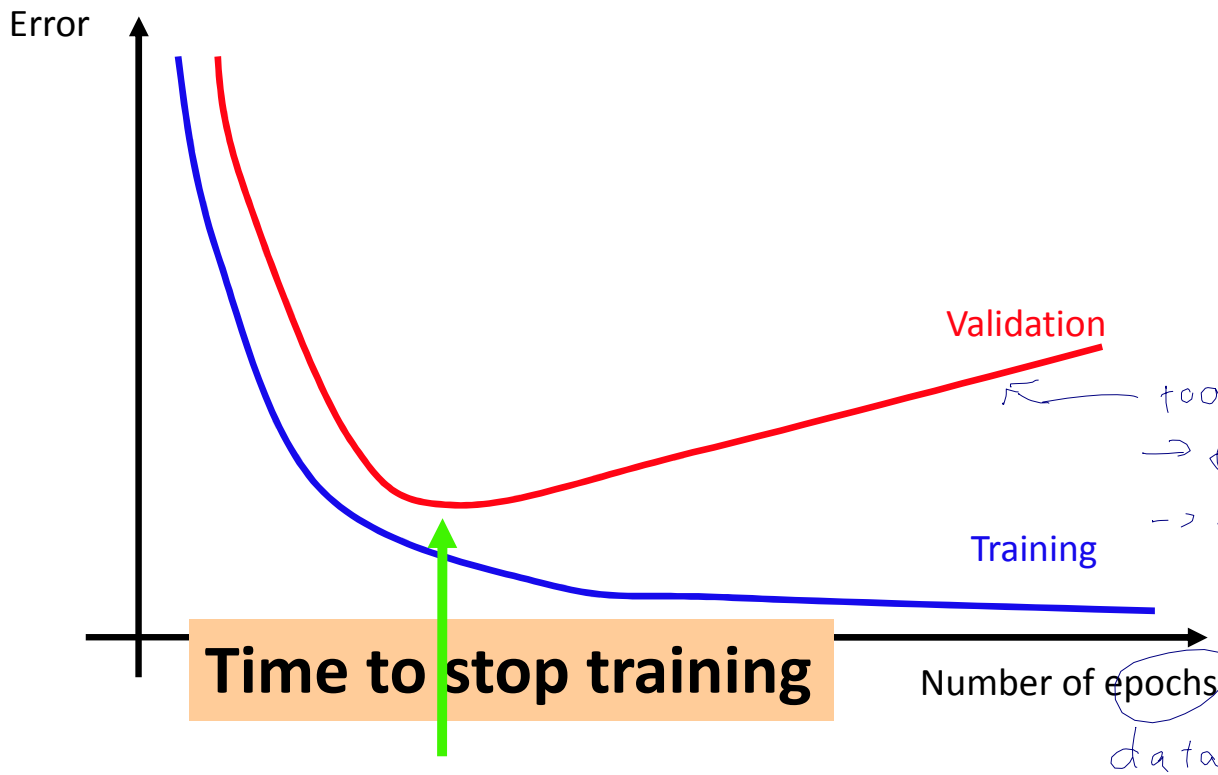
Some criticisms of cross-validation

- While the test data is independently sampled, there is a lot of overlap in training data. (refer to graphic previous slide, due to the training set is bigger.
  - The model performance may be correlated.
  - **Underestimation** of variance.
  - **Overestimation** of significant differences.
- One proposed solution is to repeat **2-fold** cross-validation 5 times rather than **10-fold** cross validation

Early Stopping

- When should we stop training?
  - Could set a minimum training error
    - ✓ Danger of overfitting
  - Could set a number of epochs
    - ✓ Danger of underfitting or overfitting
  - Can use the validation set
    - ✓ Measure the error on the validation set during training

Early Stopping



Types of Errors or Metrics

- False Positives
  - The system predicted **TRUE** but the value was **FALSE**
  - aka “False Alarms” or Type I error
- False Negatives
  - The system predicted **FALSE** but the value was **TRUE**
  - aka “Misses” or Type II error

Simplest Measure: Accuracy

- Easily the most common and intuitive measure of classification performance.

$$Accuracy = \frac{\#correct}{N}$$

both true positive & true negative.

← total no of samples

Problems with accuracy

- Contingency Table

		True Values: Positive	True Values: Negative
Hyp Values	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Problems with Accuracy

- Information Retrieval Example
  - Find the 10 documents related to a query in a set of 100 documents

		True Values	
		Positive	**Negative**
Hyp Values	**Positive**	0	0
	Negative	10	90

$$Accuracy = 90\%$$

the Accuracy measurement. should not be solely used for checking. this is the extreme case

Problems with Accuracy

- Precision: how many hypothesized events were true events
- Recall: how many of the true events were identified
- F-Measure: Harmonic mean of precision and recall

$$P = \frac{TP}{TP + FP}$$

$$R = \frac{TP}{TP + FN}$$

$$F = \frac{2PR}{P + R}$$

		True Values	
		Positive	Negative
Hyp Values	Positive	0	0
	Negative	10	90

From Past: Binary Classification Performance

- Conditional probability
  - P(feature | class-1)
- Sensitivity
  - P(positive | class-1): True Positive (TP) rate or recall rate
  - measures the proportion of positives which are correctly identified
  - False Negative (FN) P(~positive | class-1) = 1-TP
- Specificity
  - P(negative | ~class-1): True negative rate (TN)
  - measures the proportion of negatives which are correctly identified
  - False Positive (FP) P(positive | ~class-1) = 1-TN

F-Measure

- F-measure can be weighted to favor Precision or Recall
  - beta > 1 favors recall
  - beta < 1 favors precision

$$F_\beta = \frac{(1+\beta^2)PR}{(\beta^2 P) + R}$$

Precision, Recall.

		True Values	
		Positive	Negative
Hyp Values	Positive	0	0
	Negative	10	90

Beta = 1

$$P = 0$$

$$R = 0$$

$$F_1 = 0$$

F_1 measure

F-Measure

		True Values	
		Positive	Negative
Hyp Values	Positive	1	0
	Negative	9	90

$$F_\beta = \frac{(1+\beta^2)PR}{(\beta^2 P) + R}$$

$$P = 1$$

$$R = \frac{1}{10}$$

$$F_1 = .18$$

how many cases is classified.

← pretty low

F-Measure

		True Values	
		Positive	Negative
Hyp Values	Positive	10	50
	Negative	0	40

$$F_\beta = \frac{(1+\beta^2)PR}{(\beta^2 P) + R}$$

$$P = \frac{10}{60}$$

$$R = 1$$

$$F_1 = .29$$

F-Measure

		True Values	
		Positive	Negative
Hyp Values	Positive	9	1
	Negative	1	89

$$F_\beta = \frac{(1+\beta^2)PR}{(\beta^2 P) + R}$$

$$P = .9 = \frac{9}{1+9}$$

$$R = .9 = \frac{9}{1+9}$$

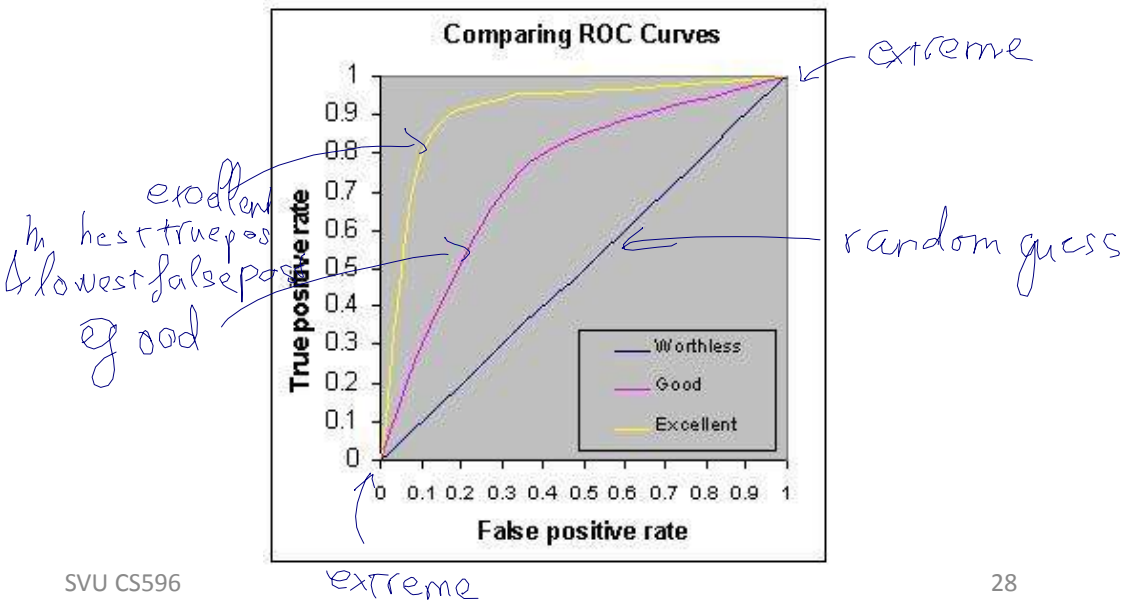
$$F_1 = .9$$

F-Measure

- Accuracy is weighted towards majority class performance.
- F-measure is useful for measuring the performance on minority classes.
- Most popular: F1-measure

ROC Curves

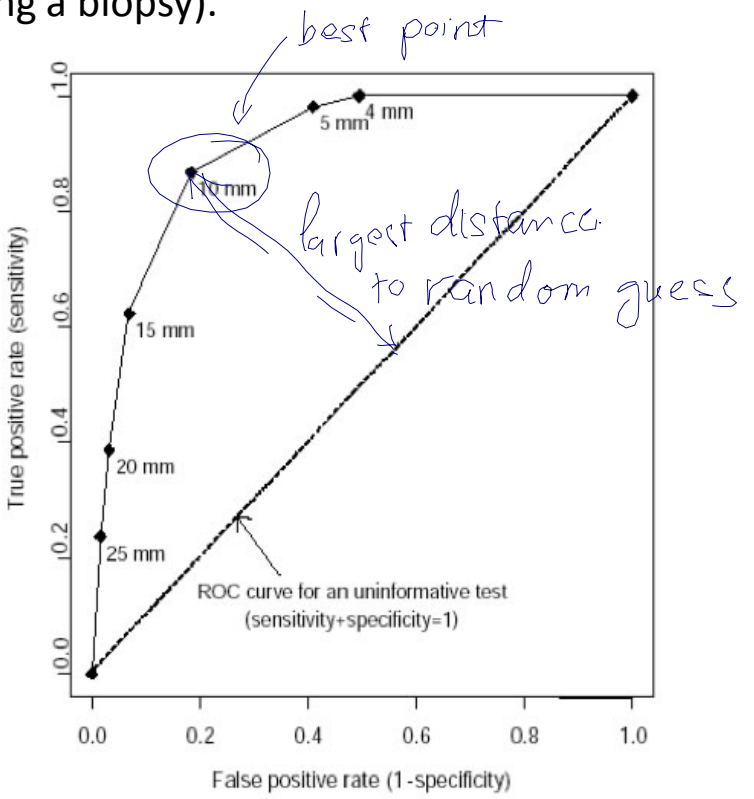
- It is common to plot classifier performance at a variety of settings or thresholds
- Receiver Operating Characteristic (ROC) curves plot true positives against false positives.
- The overall performance is calculated by the Area Under the Curve (AUC)



ROC Curve Example: Endometrial ultrasound

Ultrasound can be used to detect thickening in the lining of the uterus, which may be an early sign of cancer. If abnormal, a biopsy or minor surgical procedure is needed. This is painful and invasive, and has some risk. So our goal is to maximize the number of true positives (correctly diagnosed cancers) with an acceptable number of false positives (false alarms requiring a biopsy).

Cutoff for abnormal wall thickness	Sensitivity (%)	Specificity (%)	1 - Specificity (%)
> 4 mm	99	50	50
> 5 mm	97	61	39
> 10 mm	83	80	20
> 15 mm	60	90	10
> 20 mm	40	95	5
> 25 mm	20	98	2



Classifier Evaluation Metrics: Confusion Matrix

Confusion Matrix: extending true/false to more classes

Actual class\Predicted class	C_1	$\neg C_1$
C_1	**True Positives (TP)**	**False Negatives (FN)**
$\neg C_1$	**False Positives (FP)**	**True Negatives (TN)**

Example of Confusion Matrix:

Actual class\Predicted class	buy_computer = yes	buy_computer = no	Total
buy_computer = yes	**6954**	**46**	7000
buy_computer = no	**412**	**2588**	3000
Total	7366	2634	10000

- Given *m* classes, an entry, $\boldsymbol{CM_{i,j}}$ in a **confusion matrix** indicates # of tuples in class *i* that were labeled by the classifier as class *j*
- May have extra rows/columns to provide totals

Classifier Evaluation Metrics: Accuracy, Error Rate, Sensitivity and Specificity

A\P	C	¬C	
C	**TP**	**FN**	**P**
¬C	**FP**	**TN**	**N**
	P'	**N'**	**All**

- **Classifier Accuracy,** or recognition rate: percentage of test set tuples that are correctly classified

  Accuracy = (TP + TN)/All

- **Error rate:** *1 – accuracy*, or

  Error rate = (FP + FN)/All

- **Class Imbalance Problem**:
  - One class may be *rare*, e.g. fraud, or HIV-positive
  - Significant *majority of the negative class* and minority of the positive class
  - **Sensitivity**: True Positive recognition rate
    - **Sensitivity = TP/P**
  - **Specificity**: True Negative recognition rate
    - **Specificity = TN/N**

Summary: Precision and Recall, and F-measures

- **Precision**: exactness – what % of tuples that the classifier labeled as positive are actually positive

$$precision = \frac{TP}{TP + FP}$$

- **Recall:** completeness – what % of positive tuples did the classifier label as positive?
- Perfect score is 1.0
- Inverse relationship between precision & recall

$$recall = \frac{TP}{TP + FN}$$

- ***F* measure (*F_1* or *F*-score)**: harmonic mean of precision and recall,

$$F = \frac{2 \times precision \times recall}{precision + recall}$$

- ***F_β*:** weighted measure of precision and recall
  - assigns ß times as much weight to recall as to precision

$$F_\beta = \frac{(1 + \beta^2) \times precision \times recall}{\beta^2 \times precision + recall}$$

Classifier Evaluation Metrics: Example

Actual class/Predicted class	buy_computer = yes	buy_computer = no	Total
buy_computer = yes	**6954**	**46**	7000
buy_computer = no	**412**	**2588**	3000
Total	7366	2634	10000

- *Accuracy = (***6954+2588)** / 10000 = 95.42%
- *Precision* = **6954**/7366 = 94.4%
- *Recall* = *Sensitivity* = **6954**/7000 = 99.34%
- Specificity = **2588** /3000 = 86.27%
- F1 = P*R / 2(P+R) = 48.4%

Python Examples

- plot_classifier_comparison.py
  - Compare many classifier from sklearn
- validation.py
  - Illustrate basic sklearn functions
- gp_diabetes_dataset.py, plot_cv_diabetes.py

Extra Bits: Dendogram

- Dendogram: a graphic plot to visualize hierarchical sequence of clustering assignments
- Tree with the following properties
  - Each node represent a grouping
  - Root node is the whole dataset
  - Each leaf node (at bottom) is a singleton (data point)
  - Each internal node has two links connecting to the child nodes
  - Choice of links are determined by the dissimilarity measure
  - If we put the leaf modes at level zero, then each internal node is drawn at the height proportional to the dissimilarity

Dendogram Example → HW3

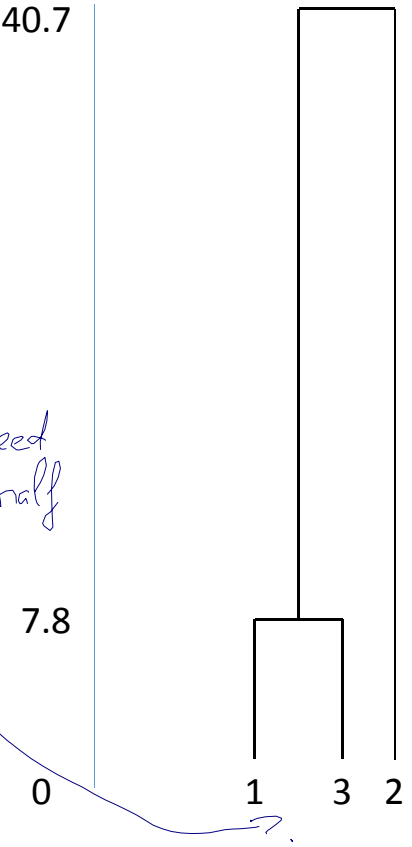
ID	1	2	3
Height	66	73	72
Weight	170	210	165

$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$

distant matrix is needed for dendogram

dissimilarity	1	2	3
1	0	40.7	7.8
2		0	45
3			0

only need upper half



Project Topics

Project Requirements

- Deadlines
  - Sing-up: 07/17
    - Form a team: preferably 1-2, max 3
    - Select a topic
  - Proposal and a short presentation: 07/24
  - Completion and final presentation: 08/14
- Deliverables
  - Code
    - Please use Github (www.github.com) as repository
      - Sign up a free personal accounts
  - Documents
    - Description of problem, solution and any assumption
    - Also put them on Github
  - Demo or presentation in the class (5-10 min)

how to do what to do

Topic-1: Personal Photo Album Mining

- Given the personal photo album folder, classify all the faces in the pictures
- Use the face detection function from OpenCV to locate faces first. Sample code: https://realpython.com/blog/python/face-recognition-with-python/
  - Alternatives: https://github.com/bytefish/facerec and https://www.youtube.com/watch?v=fRiCOxJtsQQ
- Apply face recognition functions from sk-learn sample code: http://scikit-learn.org/stable/auto_examples/applications/face_recognition.html
- Build a simple database (or spreadsheet file) to store the classification results

Topic-2: Segmentation of Blond Hairs

- Given a photo of the upper body of a person
- Use the face detection function from OpenCV to locate the face first. Sample code: https://realpython.com/blog/python/face-recognition-with-python/
- Extend from the face region to detect possible hair region.
- Analyze the properties of the hair region and be able to detect it as a whole.
- Focus on detecting blond hair, as the color is very close to skin colors.

Topic-3: Segmentation of Dark Hairs in Dark Background

- Given a photo of the upper body of a person
- Use the face detection function from OpenCV to locate the face first. Sample code: https://realpython.com/blog/python/face-recognition-with-python/
- Extend from the face region to detect possible hair region.
- Analyze the properties of the hair region and be able to detect it as a whole.
- Focus on detecting dark hair with dark background, as the colors are easily mixed up.

Topic-4: Detection of Hand in Images

- Given a photo of the hand of a person against simple background
- Use the color functions from OpenCV to locate the hand. Sample code: http://creat-tabu.blogspot.com/2013/08/opencv-python-hand-gesture-recognition.html
- Analyze the properties of the hand region and be able to detect it as a whole.

hand shape can change

Topic-5: Training Intelligent Camera for Cat (or dog or rabbit) Detection

- Based on the tutorial found here: https://www.youtube.com/watch?v=DER1Zmx8wY0 and http://nummist.com/opencv/Howse_ISMAR_20140909.pdf
- Use OpenCV and Python to train a camera to detect cat faces in images

Topic-6: Opinion Mining popular.

- Given a product or store name, retrieve all relevant customer reviews from a e-commerce or review site (yelp, cnet, amazon, newegg, macys, for example) ← API from the website.
- Apply machine learning functions to summarize reviews
- Output the results in a simple database (or spreadsheet)
- Samples:
  - https://github.com/Fossj117/opinion-mining
  - http://fjavieralba.com/basic-sentiment-analysis-with-python.html
  - https://github.com/kjahan/opinion-mining
  - http://neuro.imm.dtu.dk/wiki/Sentiment_analysis

Topic-7: Social Media Mining

- Given your social media account, retrieve all the available information from your friends using the social media web service APIs
- Apply machine learning functions to cluster or classify your friends according to the features your defined
- Output the results in a simple database (or spreadsheet)
- Sample codes
  - http://marcobonzanini.com/2015/03/02/mining-twitter-data-with-python-part-1/
  - http://curiositybits.com/python-for-mining-the-social-web/python-tutorial-mining-facebook-fan-page-getting-posts-and-comments/
  - https://github.com/ptwobrussell/Mining-the-Social-Web

Topic-8: Improvements or Addons for SK-learn

- Any pre-processing, post-processing, or visualization tools to work with sk-learn functions
- Spectral Clustering on Color Images
  - http://stackoverflow.com/questions/29630656/pythonscikit-adapting-spectral-clustering-example-code-given-to-use-arbitr
  - http://lagis-vi.univ-lille1.fr/~lm/classpec/reunion_28_02_08/ictta08_pah_lm_color_segmentation.pdf
  - http://www.ijcaonline.org/volume3/number9/pxc3871082.pdf

Topic-9: E-mail Data Mining

- Given your email account (remote or local), use system APIs to retrieve data files with metadata for each email
- Apply machine learning text clustering and classification functions to analyze the contents
- Output the results in a simple database (or spreadsheet)
- Samples:
  - http://www.magiksys.net/pyzmail/
  - http://blog.magiksys.net/parsing-email-using-python-header
  - https://indico.io/blog/email-sentiment/

Other Topics

- Any active project in https://www.kaggle.com/competitions
- Your own ideas
  - Please submit a rough proposal of your ideas to me by e-mail before next week and be prepared to discuss with me in next class
  - I will provide my feedback after mid-term exam