Learning from Data
Lecture 6: Linear Regression
Semi-Supervised Learning

Malvina Nissim
m.nissim@rug.nl
room 1311.421

19 December 2016

Topics

1. Linear Regression
2. Semi-supervised learning
   - Bootstrapping
   - Co-training
   - Active learning
   - Distant Supervision
3. Feature analysis
4. Shared Task

Classification vs Regression

create models of prediction from gathered data

- classification
  the dependent variables are categorical
  - input x: feature vector
  - output: **discrete class label**

- regression
  the dependent variables are numerical
  - input x: feature vector
  - output y: **continuous value**

Example: Predicting housing prices in the Netherlands

Data

training data

size m^2	price
57	150,000
90	210,000
30	90,000

Data

training data

size m^2	price
57	150,000
90	210,000
30	90,000

x (input) y (output)

Data

training data

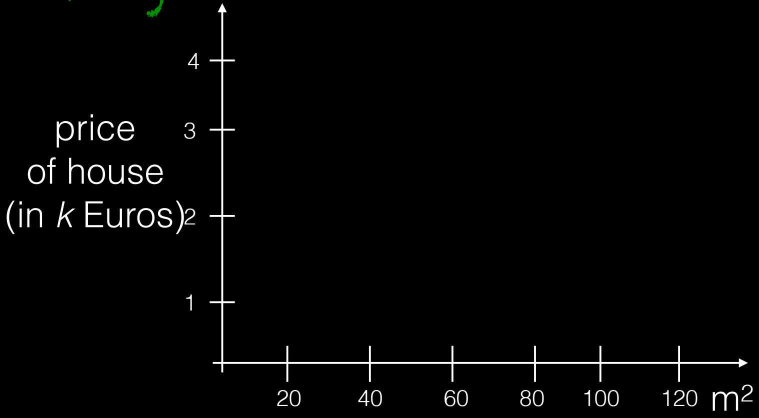
size m^2	price
57	150,000
90	210,000
30	90,000

x (input) y (output)

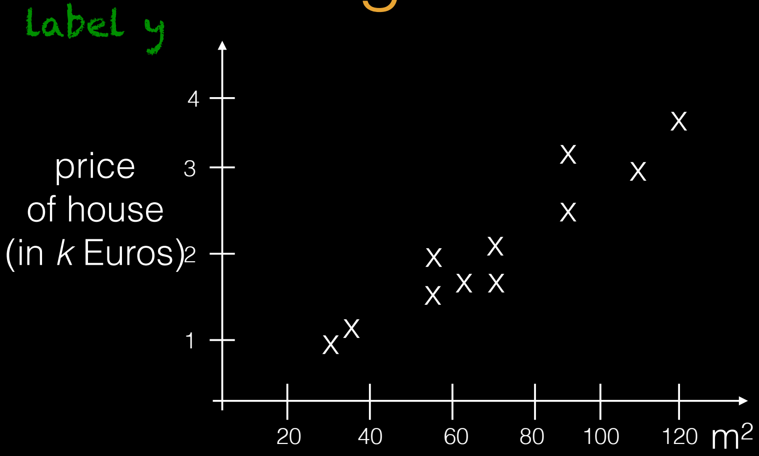
$$\langle x, y \rangle$$

Regression

label y

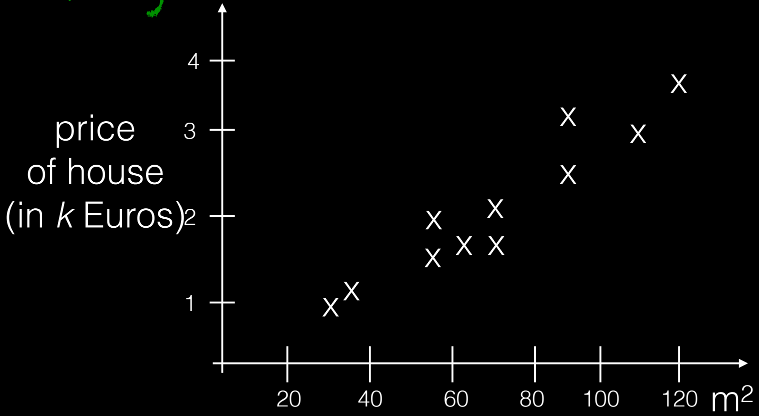


Regression



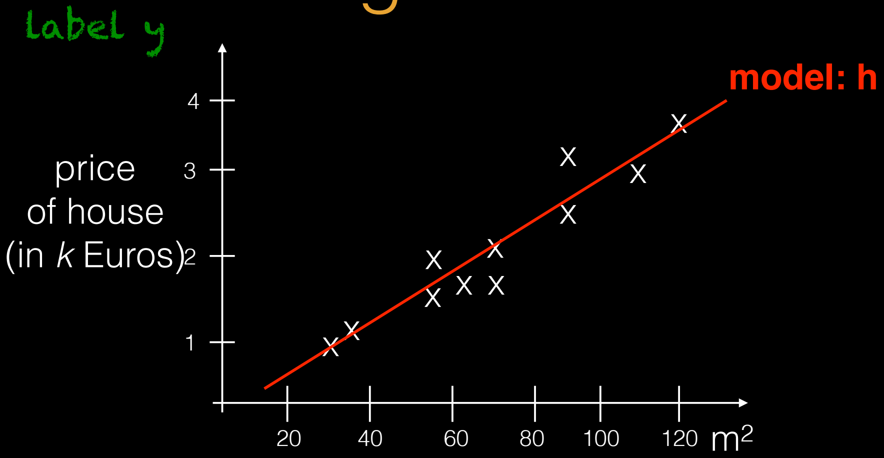
Regression

label y



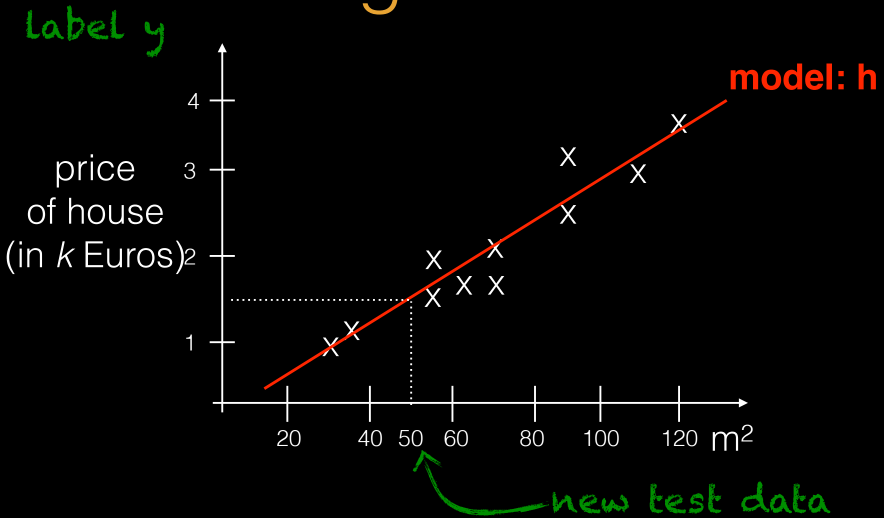
regression: predict numeric (continuous) output

Regression



regression: predict numeric (continuous) output

Regression



regression: predict numeric (continuous) output

Generalisation

experience
(training data)

size m^2	price
57	150,000
90	210,000
30	90,000

Generalisation

experience
(training data)

size m^2	price
57	150,000
90	210,000
30	90,000

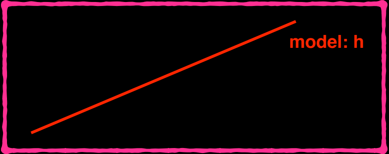
abstract
representation

Generalisation

experience
(training data)

size m^2	price
57	150,000
90	210,000
30	90,000

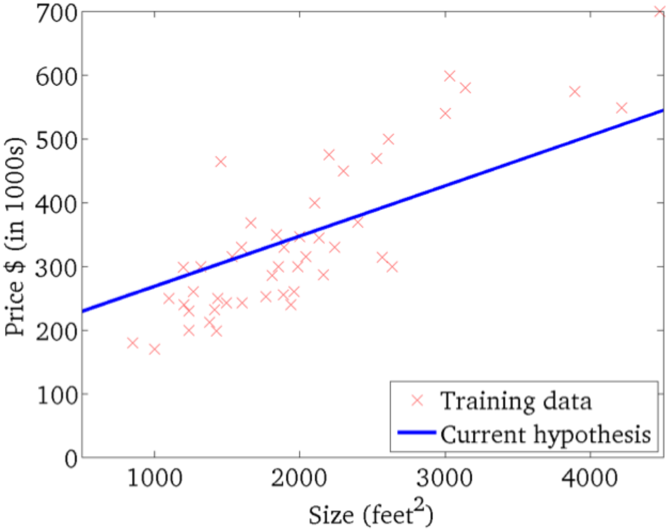
abstract
representation



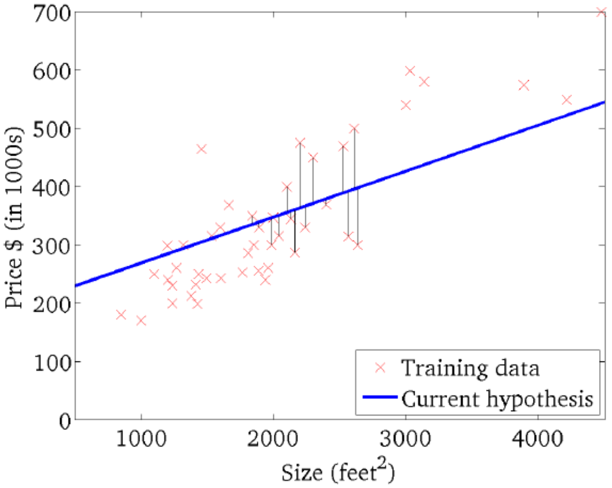
Modelling in LR

- Fitting a model to training data that generalises well to unseen data
- Hypothesis/model/function
- The model is a function that knows how to map x to y

One-feature example



Error as vertical lines



Mean Squared Error (MSE)

How close is a regression line to a set of points?

General idea:

- take the distances from the actual points to the regression line (distance = error)
- square them (necessary to remove any negative signs; also gives more weight to larger differences)
- take average

Mean Squared Error (MSE)

Steps to calculate the MSE from a set of X and Y values:

- find the regression line
- insert your X values into the linear regression equation to find the new Y values (Y)
- subtract the new Y value from the original to get the error
- square the errors
- add up the errors
- find the mean

interpretation: the smaller the MSE, the closer to the best fit

Regression in `scikit`

- training (fitting)
- testing
- evaluating

```
>>> from sklearn import linear_model
>>> from sklearn.metrics import mean_squared_error

>>> lr = linear_model.LinearRegression()
>>> lr.fit(Xtrain, Ytrain)

>>> Yguess = lr.predict(Xtest)
>>> mean_squared_error(Ytest, Yguess)

# worked out:
>>> np.mean((Yguess - Ytest) ** 2))
```

Hypothesis

- Parameters (weights) represented by Theta, Θ
- With one feature only:

$$h_\Theta(x) = \Theta_0 + \Theta_1 x_1$$

- Multiple features:

$$h_\Theta(x) = \Theta_0 + \Theta_1 x_1 + \Theta_2 x_2 + ... + \Theta_n x_n$$

for convenience, $x_0 = 1$

$$h_\Theta(x) = \Theta_0 x_0 + \Theta_1 x_1 + \Theta_2 x_2 + ... + \Theta_n x_n$$

example interpretation:
price $= \Theta_0 + \Theta_1$Size $+ \Theta_2$Age $+ \Theta_3$#Floors...

Hypothesis

- Parameters (weights) represented by Theta, Θ
- With one feature only:

$$h_\Theta(x) = \Theta_0 + \Theta_1 x_1$$

- Multiple features:

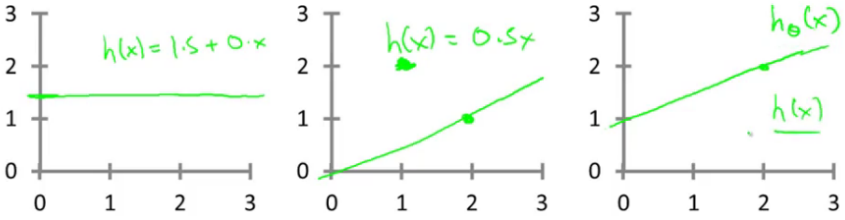
$$h_\Theta(x) = \Theta_0 + \Theta_1 x_1 + \Theta_2 x_2 + ... + \Theta_n x_n$$

for convenience, $x_0 = 1$

$$h_\Theta(x) = \Theta_0 x_0 + \Theta_1 x_1 + \Theta_2 x_2 + ... + \Theta_n x_n$$

example interpretation:
price $= \Theta_0 + \Theta_1$Size $+ \Theta_2$Age $+ \Theta_3$#Floors...

$$h_\theta(x) = \theta_0 + \theta_1 x$$



$\rightarrow \theta_0 = 1.5$	$\rightarrow \theta_0 = 0$	$\rightarrow \theta_0 = 1$
$\rightarrow \theta_1 = 0$	$\rightarrow \theta_1 = 0.5$	$\rightarrow \theta_1 = 0.5$

https://www.coursera.org/learn/machine-learning/lecture/rkTp3/cost-function

Cost function J

How to fit the best possible model to our training data?

- find Θs that minimise the cost
- and because cost is squared error, then
- minimising squared difference between predicted output and true output $(h_{\Theta}(x) - y)^2$

semi-supervised learning

What if there is no y?

x $\longrightarrow$?

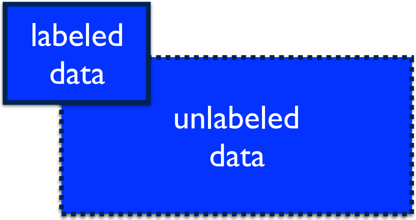
What if there is no y?

$\mathbf{x} \longrightarrow \mathbf{?}$

unlabeled
data

What if there is no y?

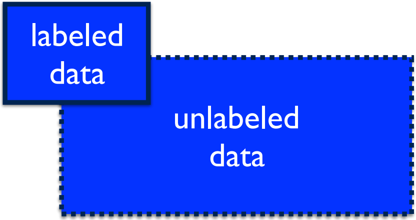
$$\mathbf{x} \longrightarrow \mathbf{?}$$



What if there is no y?

$$\mathbf{x} \longrightarrow \mathbf{?}$$

unsupervised
learning



What if there is no y?

$$\mathbf{x} \longrightarrow \mathbf{?}$$

unsupervised learning

unlabeled data

semi-supervised learning

labeled data

unlabeled data

What is SSL? More formally

- Learning from both labeled and unlabeled data:
  - l labeled instances $\{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$ and
  - u unlabeled instances $\{\mathbf{x}_j\}_{j=l+1}^{l+u}$, usually $u >> l$
- **Goal**: better classifier than from labeled data alone

Anti-SSL arguments

- "We'll find the time and money to annotate more labeled data"
- Hmm, but:
  - Annotating PT WSJ took a decade!
  - What about building a NER for, say, Irish? Who is going to annotate it for me?

Semi-supervised learning

use a little amount of labelled data + a large amount of unlabelled data

Semi-supervised learning

use a little amount of labelled data $+$ a large amount of unlabelled data

it's a matter of getting help

- the classifier gets help from itself $\rightarrow$ bootstrapping
- the classifier gets help from another classifier $\rightarrow$ co-training
- the classifier gets help from a human $\rightarrow$ active learning

bootstrapping

Bootstrapping

aka **self-training**: the classifier uses its own predictions to teach itself

Procedure (only one classifier is required, with no split of features):

Bootstrapping

aka **self-training**: the classifier uses its own predictions to teach itself

Procedure (only one classifier is required, with no split of features):

1. start with a set of labeled data, and build a classifier, which is then applied on the set of unlabeled data.
2. only those instances with a labeling confidence exceeding a certain threshold are added to the labeled set.
3. the classifier is then retrained on the new set of labeled examples, and the process continues for several iterations.

Bootstrapping

aka **self-training**: the classifier uses its own predictions to teach itself

Procedure (only one classifier is required, with no split of features):

1. start with a set of labeled data, and build a classifier, which is then applied on the set of unlabeled data.
2. only those instances with a labeling confidence exceeding a certain threshold are added to the labeled set.
3. the classifier is then retrained on the new set of labeled examples, and the process continues for several iterations.

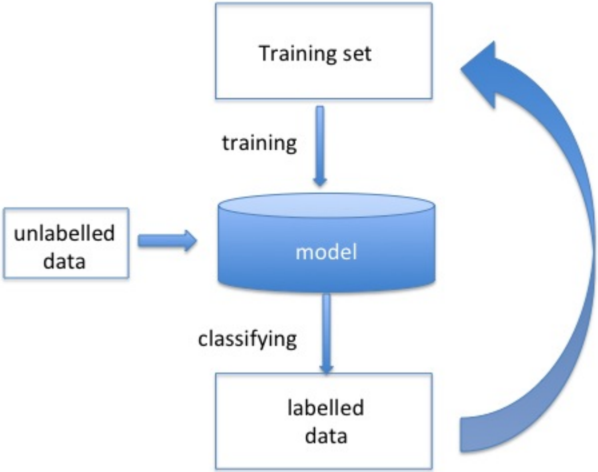
Bootstrapping

aka **self-training**: the classifier uses its own predictions to teach itself

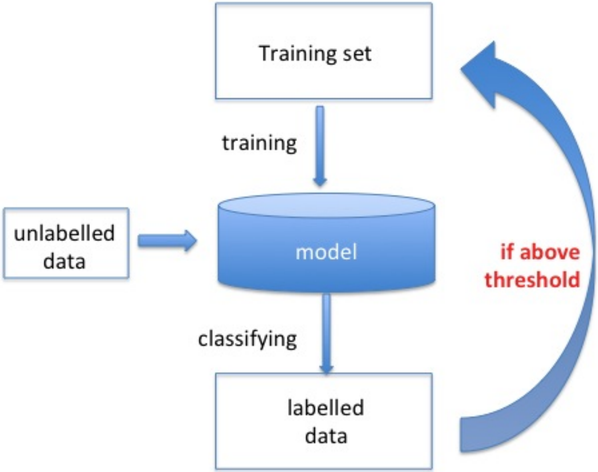
Procedure (only one classifier is required, with no split of features):

1. start with a set of labeled data, and build a classifier, which is then applied on the set of unlabeled data.
2. only those instances with a labeling confidence exceeding a certain threshold are added to the labeled set.
3. the classifier is then retrained on the new set of labeled examples, and the process continues for several iterations.

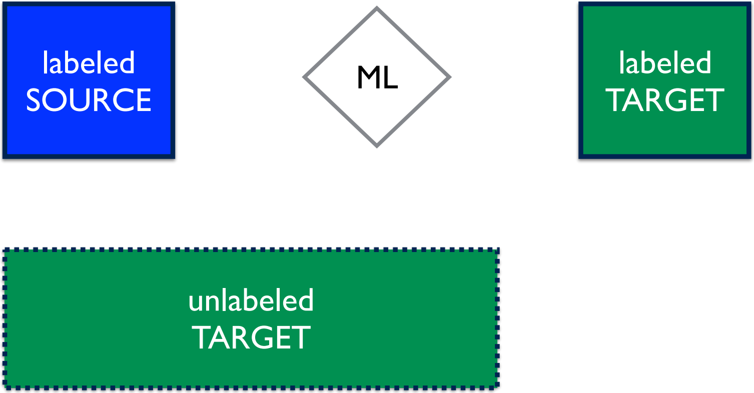
Bootstrapping



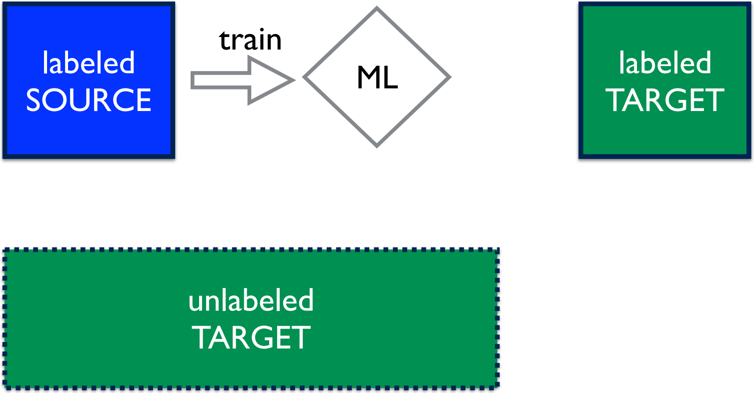
Bootstrapping



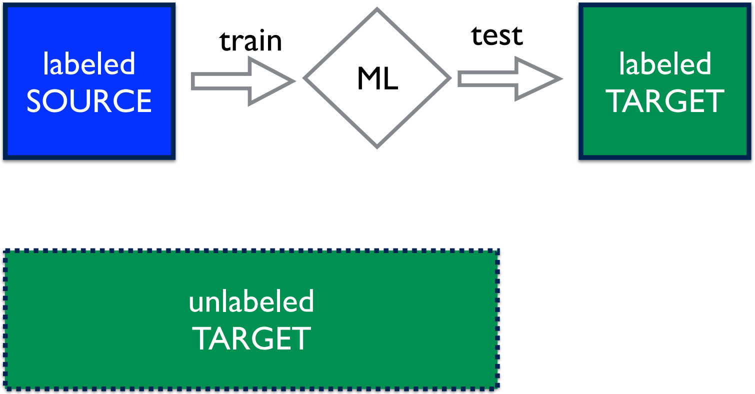
Self-training



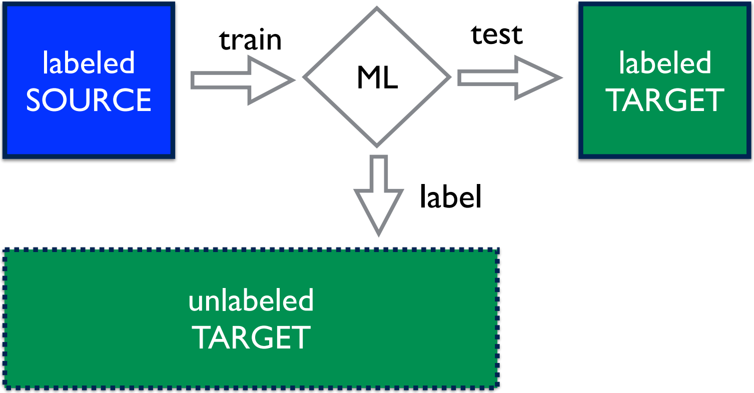
Self-training



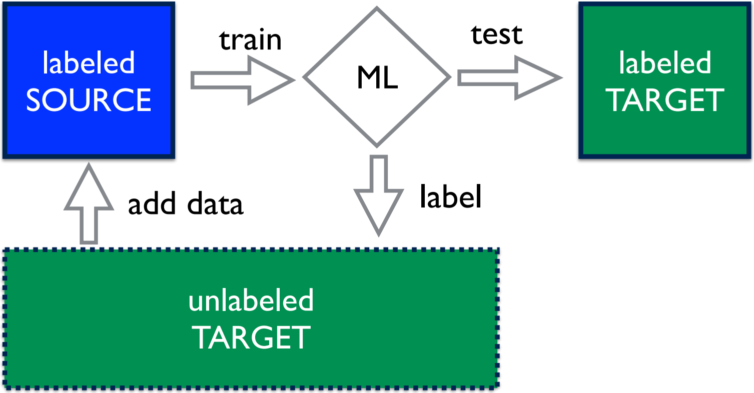
Self-training



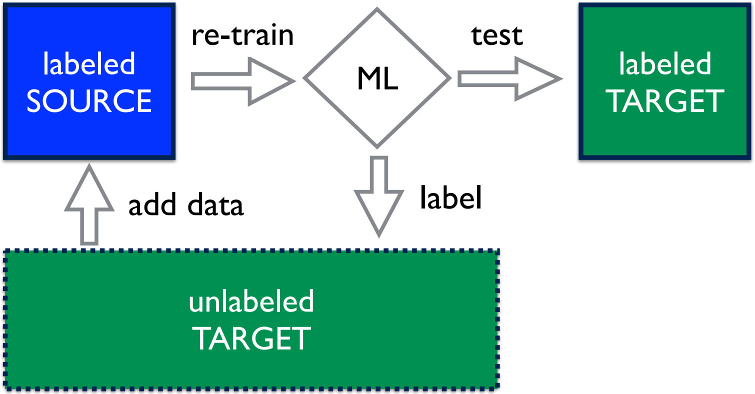
Self-training



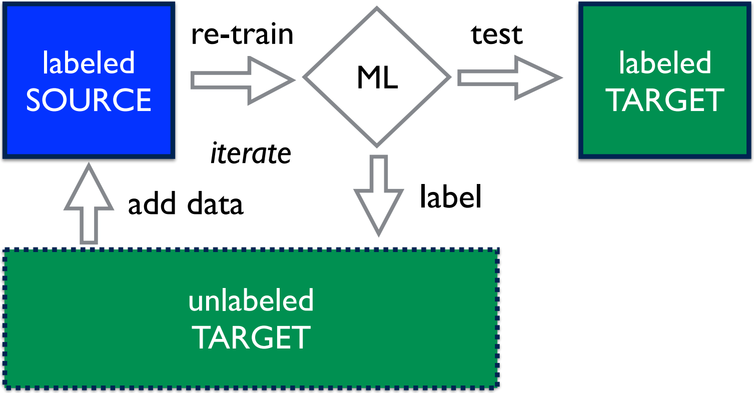
Self-training



Self-training



Self-training



Bootstrapping or Self-training

parameters:

- iterations: number of iterations
- pool size: number of examples selected from the unlabeled set U for annotation at each iteration.
- growth size: number of most confidently labeled examples that are added at each iteration to the set of labeled data L.

Bootstrapping or Self-training

- useful when very little but very good data is available and it's too costly to annotate more
- if the data is very skewed it can be problematic to safely assign the low frequency class(es)
- the choice of initial seed examples is crucial

→ this is different from "bootstrap" as used in statistics!

co-training

Co-training

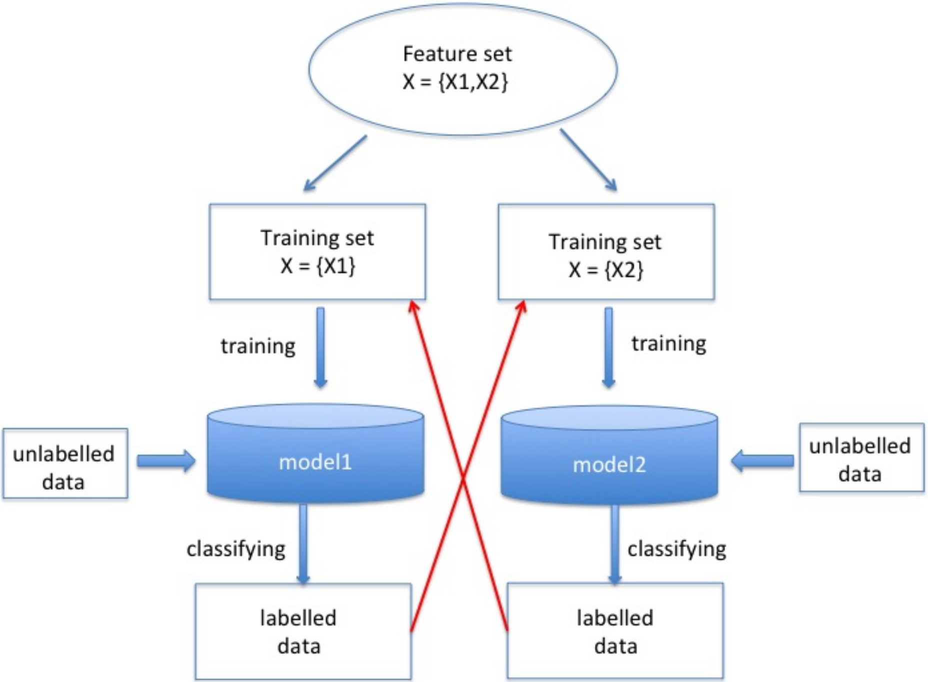
- two sufficient and independent sets of features: an instance X is $X = (X_1, X_2)$

  $P(X_1|X_2, Y) = P(X_1|Y)$

  $P(X_2|X_1, Y) = P(X_2|Y)$

- independent sets are different *views*
- independent views can be informative
- exploit two views of the same phenomenon to acquire more labelled data for training

(ref: Blum & Mitchell 1998)


Feature set
X = {X1,X2}
Training set
X = {X1}
Training set
X = {X2}
training
training
unlabelled
data
model1
model2
unlabelled
data
classifying
classifying
labelled
data
labelled
data

active learning

Active learning

excellent slides by Piyush Rai

distant supervision

Distant Supervision

- so far: little amount of gold data, large amounts of unlabelled data

Distant Supervision

- so far: little amount of gold data, large amounts of unlabelled data
- what if we have ZERO labelled data to start with?

Distant Supervision

- so far: little amount of gold data, large amounts of unlabelled data
- what if we have ZERO labelled data to start with?
- we can *approximate* labelled data (and get large amounts)

Distant Supervision

- so far: little amount of gold data, large amounts of unlabelled data
- what if we have ZERO labelled data to start with?
- we can *approximate* labelled data (and get large amounts)

Distant Supervision

- so far: little amount of gold data, large amounts of unlabelled data
- what if we have ZERO labelled data to start with?
- we can *approximate* labelled data (and get large amounts)

use *reasonably safe proxies* to obtain training labels

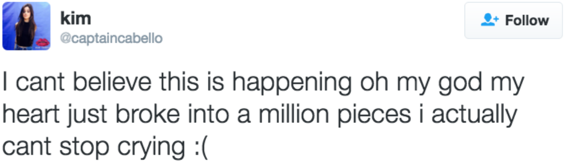
Distant Supervision: examples

Distant Supervision: examples


Weston Dennis
@G2Westballz
Follow

I'll be back even stronger. See me at g4. GG's to hbox and every one else I played. Fun tourney great crowd. I missed this :)

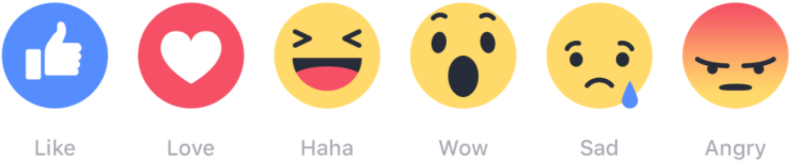
⇒ positive


kim
@captaincabello
Follow

I cant believe this is happening oh my god my heart just broke into a million pieces i actually cant stop crying :(

⇒ negative

Distant Supervision: examples



how to understand the contribution of features

Choosing features

pos tagging identity of the word being processed, identity of the words immediately to the left and right, part-of-speech tag of the word to the left, function/content word

sentiment analysis positive/negative trait in a lexicon, id of the speaker, discourse relations (contrast, concession, ...)

authorship verification character n-grams, pos n-grams, sentence length, punctuation, ...

named entity recognition ...

Feature analysis

- leave one out (feature ablation): remove one single feature at the time and re-train and re-test the classifier to compare results with and without that feature.

  $\rightarrow$ The most useful features are those that cause the biggest drop in performance

- single feature classifier: train and test the classifier with just one feature at the time.

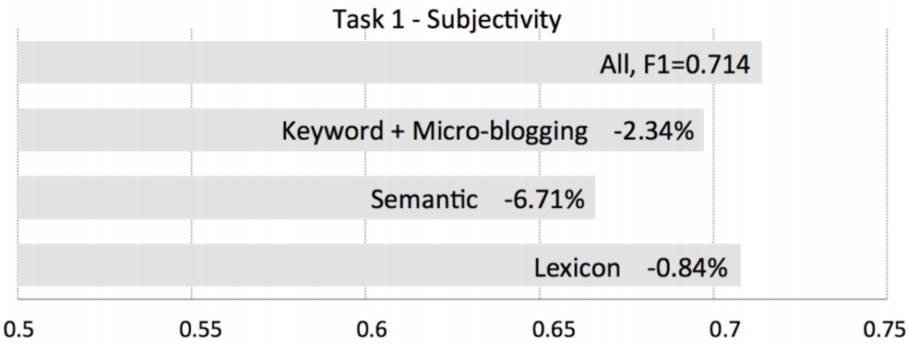
  $\rightarrow$ The most useful features are those that yield the highest performance

Leave one out

- also known as *ablation*
- it helps assessing the contribution of one feature
- often used with *groups* of features

Leave one out

- also known as *ablation*
- it helps assessing the contribution of one feature
- often used with *groups* of features



Single feature classifier

- it helps assessing the contribution of one feature
- often used with *groups* of features

shared task

Shared Task

also known as *challenge*

- evaluating systems on the same data for a proper comparison
- assessing (and sharing) state of the art systems and methods
- converging efforts on common interests
- creating data

Shared Task

Procedure:

- research groups or programme committees propose a series of tasks
- the shared task's organisers make a sample available
- teams register
- teams receive training data
- teams develop their systems
- teams receive test data
- after about one week teams return their outputs/systems to the organisers
- organisers evaluate systems according to predefined metrics
- teams and organisers write reports
- workshop(s)

Shared Task

also known as *challenge*

- evaluating systems on the same data for a proper comparison
- assessing (and sharing) state of the art systems and methods
- converging efforts on common interests
- creating data

Semeval 2016
Semeval 2017