Learn

learning re-visited

…. unsupervised learning – ‘business’ rules

……… features and classes together (recommendations)

……………..learning ‘facts’ from collections of text (web)

……………………what is ‘knowledge’?

learning re-visited: classification

data has (i) *features* $x_1 \ldots x_N = X$

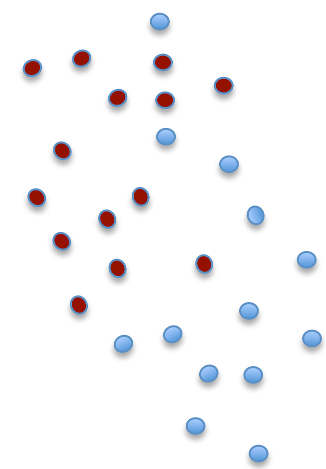
(e.g. query terms, words in a comment)

and (ii) *output variable(s)* Y, e.g. class y, classes $y_1 \ldots y_k$

(e.g. buyer/browser, positive/negative: y=0/1,

in general need not be binary)

classification:



suppose we define a function:

$$f(X) = E[Y|X]$$

i.e., expected value of Y given X

e.g. if Y = y, and y is 0/1; then

$$f(X) = 1{*}P(y=1|X) + 0{*}P(y=0|X) = P(y=1|X)$$

– which we earlier estimated using Naïve Bayes + a training set

examples: old and new

queries

R	F	G	C	Buy?
n	n	y	y	y
y	n	n	y	y
y	y	y	n	n
y	y	y	n	y
y	y	y	n	n
y	y	y	y	n
.....				
......				

(Y, X) = (B, R, F, G, C)
binary variables

comments

Words	Sentiment
like, lot	positive
hate, waste	negative
enjoying, lot	positive
enjoy, lot, [not]	negative
[not], enjoy	negative

(Y, X) = (S, all words)
binary variables

animals

size	head	noise	legs	animal
L	L	roar	4	lion
S	S	meow	4	cat
XL	XL	trumpet	4	elephant
M	M	bark	4	dog
S	S	chirp	2	bird
M	S	bark	4	dog
M	M	speak	2	human
M	S	squeal	2	bird
L	M	roar	4	tiger

(Y, X) = (A, S, H, N, L)
fixed set of multi-valued, categorical variables

transactions:

Items Bought
milk, diapers, cola
diapers, beer
milk, cereal, beer
soup, pasta, sauce
beer, nuts, diapers

(Y, X) = (_ , items)
variable set of multi-valued categorical variables

how do classes emerge? clustering

groups of 'similar' users/user-queries based on terms

groups of similar comments based on words

groups of animal observations having similar features

clustering

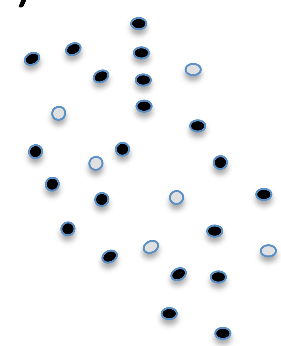
find regions that are *more* populated than *random* data

i.e. regions where $r = \frac{P(X)}{P_0(X)}$ is large (here $P_0(X)$ is uniform)

set y = 1 for all data; then add data *uniformly* with y = 0

then $f(X) = E[y|X] = \frac{r}{1+r}$;

now find regions where this is large



how to cluster? k-means, agglomerative, even LSH ! ....

rule mining: clustering features

like & lot => positive; not & like => negative

searching for flowers => searching for a cheap gift

bird => chirp or squeal; chirp & 2 legs => bird

diapers & milk => beer

statistical *rules*

find regions *more* populated than if x_i's were *independent*

- so this time $P_0(X) = \prod_i P(x_i)$, i.e., assuming feature independence
- again, set y = 1 for all real data
- add y = 0 points, choosing each x_k *uniformly* from the *data* itself
- $f(X) = E[y|X]$ again estimates $\frac{r}{1+r}; r = \frac{P(X)}{P_0(X)}$;
- its extreme regions are those of with *support* and potential rules

association rule mining

infer rule A, B, C => D if

(i) high *support*: *P*(A,B,C,D) > *s*

(ii) high *confidence*: *P*(D|A,B,C) > *c*

(iii) high *interestingness*: $\frac{P(D \mid A,B,C)}{P(D)}$ > *i*

how? key observation:

if A,B has support > *s* then so does A:

- scan all records for support > s values
- scan this subset for all support > s pairs
- … triples, etc. until no sets with support > s
- then check each set for confidence and interestingness

Note:

just counting, so map-reduce is ideal

Items Bought
milk, diapers, cola
diapers, beer
milk, cereal, beer
soup, pasta, sauce
beer, nuts, diapers

size	head	noise	legs	animal
L	L	roar	4	lion
S	S	meow	4	cat
XL	XL	trumpet	4	elephant
M	M	bark	4	dog
S	S	chirp	2	bird
M	S	bark	4	dog
M	M	speak	2	human
M	S	squeal	2	bird
L	M	roar	4	tiger

problems with association rules

characterization of classes

- small classes get left out

➢ use decision-trees instead of association rules

based on mutual information - costly

learning rules from data

- high support means negative rules are lost:

e.g. milk and *not* diapers => *not* beer

➢ use 'interesting subgroup discovery' instead

"Beyond market baskets: generalizing association rules to correlations"
ACM SIGMOD 1997
Sergey Brin, Rajeev Motwani, and Craig Silverstein

unified framework and big data

we defined $f(X) = E[Y|X]$ for appropriate data sets

- y_i=0/1 for classification; problem A: becomes estimating f
- added *random* data for clustering
- added *independent* data for rule mining
- problem B: becomes finding regions where f is large

now suppose we have ‘really big’ data (long, not wide)

- i.e., lots and lots of examples, but limited number of features
- problem A reduces to *querying* the data
- problem B reduces to finding high *support* regions
- just counting … map-reduce (or Dremel) work by brute force

… [*wide* data is still a problem though]

dealing with the long-tail

no particular book-set has high support; in fact *s* ≈ 0!

“customers who bought …”

how are customers compared?

people have *varied* interests

people — books

documents — words

experiences — features

- ‘see animal’ — - legs, noise

classes and features emerge?

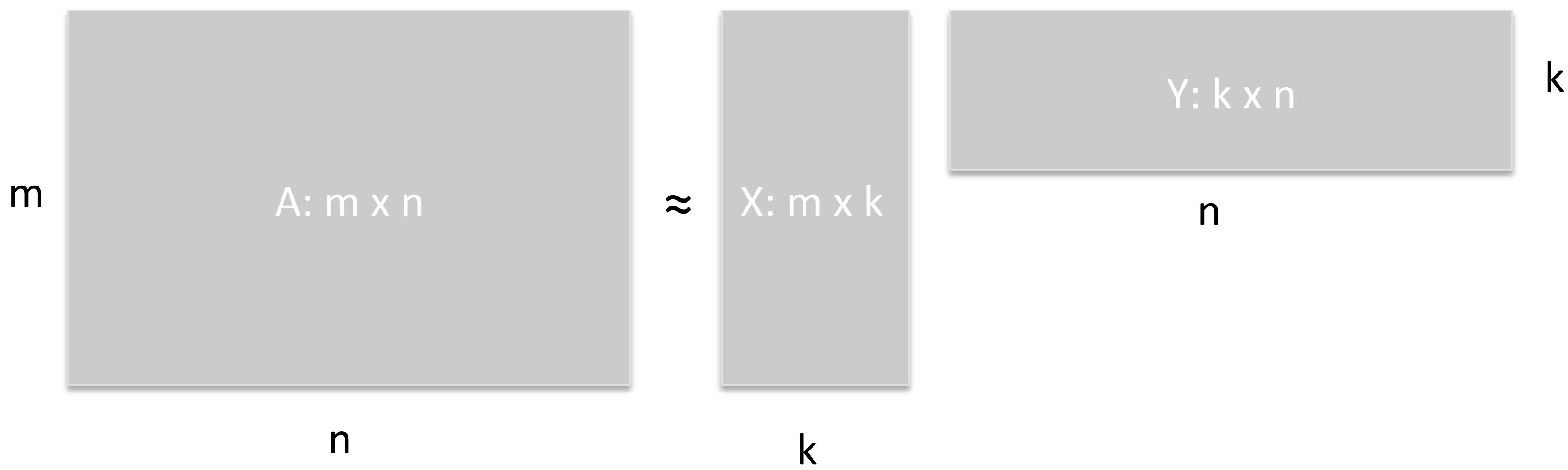
observations — perceptions

collaborative filtering

latent semantic models

“hidden structure”

one approach to latent models: NNMF



matrix A needs to be written as

$$A \approx X Y$$

since X and Y are 'smaller', this is a almost always an approximation
so we minimize $\| A - XY \|_F$

(here $_F$ means sum of squares)

subject to all entries being *non-negative* – hence NNMF

other methods – LDA (latent dirichlet allocation), SVD, etc.

back to our hidden agenda

classes can be learned from experience

features can be learned from experience

e.g. genres, i.e., classes as well as roles, i.e., features

merely from "experiences"

what is the minimum capability needed?

1. lowest level of perception: pixels, frequencies
2. <u>subitizing</u>

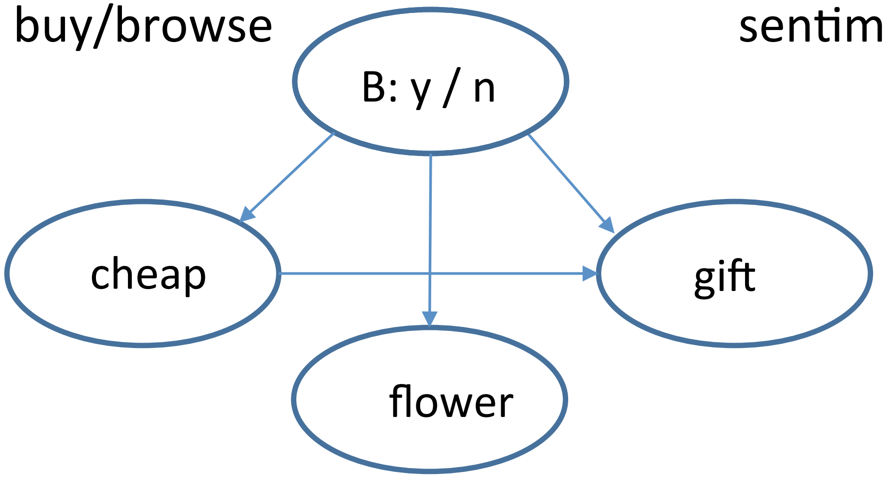
i.e., counting or distinguising between one and two things

being able to break up temporal experience into *episodes*

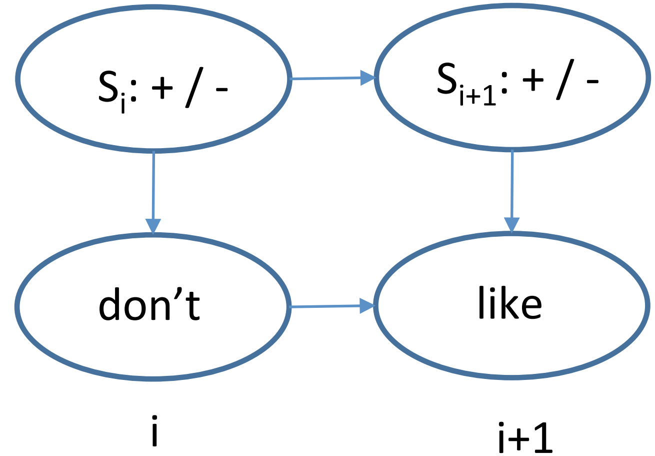
theoretically, this works; in practice …. lots of research …

beyond independent features

buy/browse



sentiment



if 'cheap' and 'gift' are *not* independent, $P(G|C,B) \neq P(G|B)$

(or use $P(C|G,B)$, depending on the order in which we *expand* $P(G,C,B)$)

"I don't like the course" and "I like the course; don't complain!"

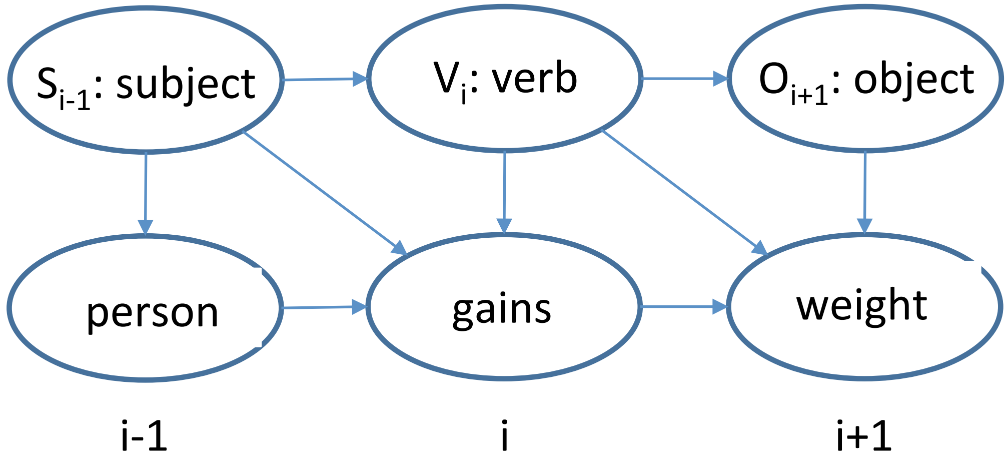
first, we might include "don't" in our list of features (also "not" …)

still – might not be able to disambiguate: need *positional order*

$P(x_{i+1}|x_i, S)$ for each position *i*: hidden markov model (HMM)

we may also need to accomodate 'holes', e.g. $P(x_{i+k}|x_i, S)$

learning 'facts' from text



suppose we want to learn *facts* of the form <subject, verb, object> from text

single class variable is not enough; (i.e. we have many y_j in data [Y,X])

further, positional order is important, so we can use a (different) HMM ..

e.g. we need to know $P(x_i | x_{i-1}, S_{i-1}, V_i)$

whether 'kills' following 'antibiotics' is a verb will depend on whether 'bacteria' is a subject

more apparent for the case <person, gains, weight>, since 'gains' can be a verb or a noun

problem reduces to estimating *all* the a-posterior probabilities $P(S_{i-1}, V_i, O_{i+1})$ for every i , and also allowing 'holes' (i.e., $P(S_{i-k}, V_i, O_{i+p})$) and find the *best* facts from a collection of text? .... many solutions; apart from HMMs - CRFs

after finding all facts from lots of text, we cull using support, confidence, etc.

open information extraction

Cyc (older, semi-automated): 2 billion facts

Yago – largest to date: 6 billion facts, linked i.e., a graph

e.g. <Albert Einstein, wasBornIn, Ulm>

Watson – uses facts culled from the web internally

REVERB – recent, lightweight: 15 million S,V,O triples

e.g. <potatoes, are also rich in, vitamin C>

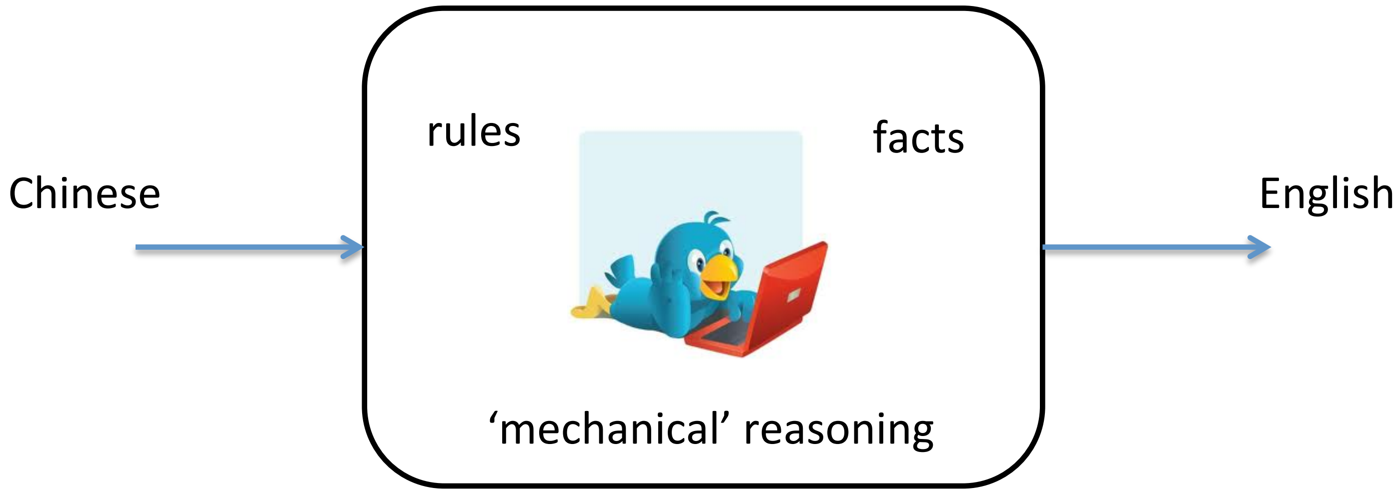
1. part-of-speech tagging using NLP classifiers (trained on labeled corpora)
2. focus on verb-phrases; identify nearby noun-phrases
3. prefer proper nouns, especially if they occur often in other facts
4. extract more than one fact if possible:

“Mozart was born in Salzburg, but moved to Vienna in 1781” yields

<Mozart, moved to, Vienna>, in addition to <Mozart, was born in, Salzburg>

to what extent have we ‘learned’?

Searle’s Chinese room:



does the translator ‘know’ Chinese?

much of machine translation uses similar techniques, as well as HMMs, CRFs, etc. to parse and translate

recap and preview

learning, or 'extracting':

classes from data – unsupervised (clustering)

rules from data - unsupervised (rule mining)

big data – counting works (unified *f*(X) formulation)

classes & features from data – unsupervised (latent models)

next week

facts from text collections – *supervised* (Bayesian n/w, HMM)

what use are these rules and facts?

reasoning using rules and facts to 'connect the dots'

logical, as well as probabilistic, i.e., reasoning under uncertainty

semantic web