Thursday, May 18, 2017 1:34 PM

types of ML

- supervised - can be formalized as function approximation
  - regression - response variable continuous
  - classification/pattern recodnition - response variable is discrete (classes)
    - binary
    - multiclass
- unsupervised
  - no obvious error metric
  - formalized as density estimation
  - more widely applicable (doesn't require human expert to label the data)
- reinforcement learning - learning how to act or behave when given occasional reward or punishment signals (based on decision theory)

regression vs classification

learn mapping from x -> y

level). When y_i is categorical, the problem is known as **classification** or **pattern recognition**, and when y_i is real-valued, the problem is known as **regression**. Another variant, known as **ordinal regression**, occurs where label space $\mathcal{Y}$ has some natural ordering, such as grades A–F.

unsupervised as density estimation

for each input. Instead, we will formalize our task as one of **density estimation**, that is, we want to build models of the form $p(\mathbf{x}_i|\boldsymbol{\theta})$. There are two differences from the supervised case. First, we have written $p(\mathbf{x}_i|\boldsymbol{\theta})$ instead of $p(y_i|\mathbf{x}_i, \boldsymbol{\theta})$; that is, supervised learning is conditional density estimation, whereas unsupervised learning is unconditional density estimation. Second, $\mathbf{x}_i$ is a vector of features, so we need to create multivariate probability models. By contrast, in supervised learning, y_i is usually just a single variable that we are trying to predict. This means that for most supervised learning problems, we can use univariate probability models (with input-dependent parameters), which significantly simplifies the problem. (We will discuss

- simplest density estimation tool - histogram
- first dimension - inputs, second dimension - group to which it belongs (possible infinity many groups, i.e. intervals in 1D), third - probability that given input belongs to a class
- so it's very similar to multi-label classification
- unsupervised doesn't really make much sense for a single class - what we are looking for is the classes itself (not only correct mapping) (first for their number, and then for their shape)

unsupervised learning similar to human

When we're learning to see, nobody's telling us what the right answers are — we just look. Every so often, your mother says "that's a dog", but that's very little information. You'd be lucky if you got a few bits of information — even one bit per second — that way. The brain's visual system has 10^{14} neural connections. And you only live for 10^9 seconds. So it's no use learning one bit per second. You need more like 10^5 bits per second. And there's only one place you can get that much information: from the input itself. — Geoffrey Hinton, 1996 (quoted in (Gorder 2006)).

classification

We will denote the probability distribution over possible labels, given the input vector $\mathbf{x}$ and training set $\mathcal{D}$ by $p(y|\mathbf{x}, \mathcal{D})$. In general, this represents a vector of length C. (If there are just two

$0|\mathbf{x}, \mathcal{D}) = 1$.) In our notation, we make explicit that the probability is conditional on the test input $\mathbf{x}$, as well as the training set $\mathcal{D}$, by putting these terms on the right hand side of the

conditioning bar |. We are also implicitly conditioning on the form of model that we use to make

$$\hat{y} = \hat{f}(\mathbf{x}) = \underset{c=1}{\overset{C}{\operatorname{argmax}}}\, p(y = c|\mathbf{x}, \mathcal{D}) \tag{1.1}$$

This corresponds to the most probable class label, and is called the **mode** of the distribution $p(y|\mathbf{x}, \mathcal{D})$; it is also known as a **MAP estimate** (MAP stands for **maximum a posteriori**). Using

ignorance of spatial layout

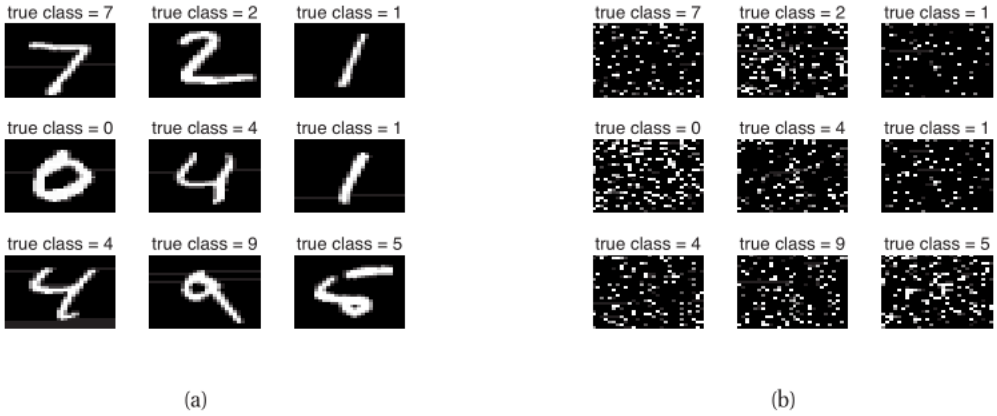


Figure 1.5 (a) First 9 test MNIST gray-scale images. (b) Same as (a), but with the features permuted randomly. Classification performance is identical on both versions of the data (assuming the training data is permuted in an identical way). Figure generated by `shuffledDigitsDemo`.

Many generic classification methods ignore any structure in the input features, such as spatial layout.

This flexibility is both a blessing (since the methods are general purpose) and a curse (since the methods ignore an obviously useful source of information)

interesting applications of regression

- Predict tomorrow's stock market price given current market conditions and other possible side information.
- Predict the age of a viewer watching a given video on YouTube.
- Predict the location in 3d space of a robot arm end effector, given control signals (torques) sent to its various motors.
- Predict the amount of prostate specific antigen (PSA) in the body as a function of a number of different clinical measurements.
- Predict the temperature at any location inside a building using weather data, time, door sensors, etc.

unsupervised learning

Two steps

- determining number of clusters (*model selection*)
- actual clustering

examples

- In e-commerce, it is common to cluster users into groups, based on their purchasing or web-surfing behavior, and then to send customized targeted advertising to each group (see e.g., (Berkhin 2006)).

matrix completion

(a) (b)

Sometimes we have missing data, that is, variables whose values are unknown. For example, we might have conducted a survey, and some people might not have answered certain questions. Or we might have various sensors, some of which fail. The corresponding design matrix will

discovering graph structure - nie rozumiem tego…

Sometimes we measure a set of correlated variables, and we would like to discover which ones are most correlated with which others. This can be represented by a graph G, in which nodes represent variables, and edges represent direct dependence between variables (we will make this precise in Chapter 10, when we discuss graphical models). We can then learn this graph structure from data, i.e., we compute $\hat{G} = \operatorname{argmax} p(G|\mathcal{D})$.

As with unsupervised learning in general, there are two main applications for learning sparse graphs: to discover new knowledge, and to get better joint probability density estimators. We now give somes example of each.

- Much of the motivation for learning sparse graphical models comes from the systems biology community. For example, suppose we measure the phosphorylation status of some proteins in a cell (Sachs et al. 2005). Figure 1.11 gives an example of a graph structure that was learned from this data (using methods discussed in Section 26.7.2). As another example, Smith et al. (2006) showed that one can recover the neural "wiring diagram" of a certain kind of bird from time-series EEG data. The recovered structure closely matched the known functional connectivity of this part of the bird brain.

- In some cases, we are not interested in interpreting the graph structure, we just want to use it to model correlations and to make predictions. One example of this is in financial portfolio management, where accurate models of the covariance between large numbers of different stocks is important. Carvalho and West (2007) show that by learning a sparse graph, and then using this as the basis of a trading strategy, it is possible to outperform (i.e., make more money than) methods that do not exploit sparse graphs. Another example is predicting traffic jams on the freeway. Horvitz et al. (2005) describe a deployed system called JamBayes for predicting traffic flow in the Seattle area; predictions are made using a graphical model whose structure was learned from data.

collaborative filtering

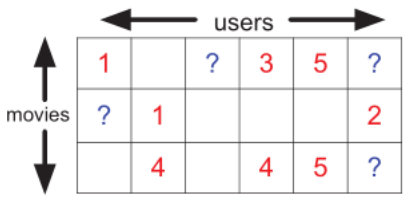


Figure 1.13 Example of movie-rating data. Training data is in red, test data is denoted by ?, empty cells are unknown.

The key idea
is that the prediction is not based on features of the movie or user (although it could be), but merely on a ratings matrix

Note
that most of the entries in X will be missing or unknown, since most users will not have rated most movies

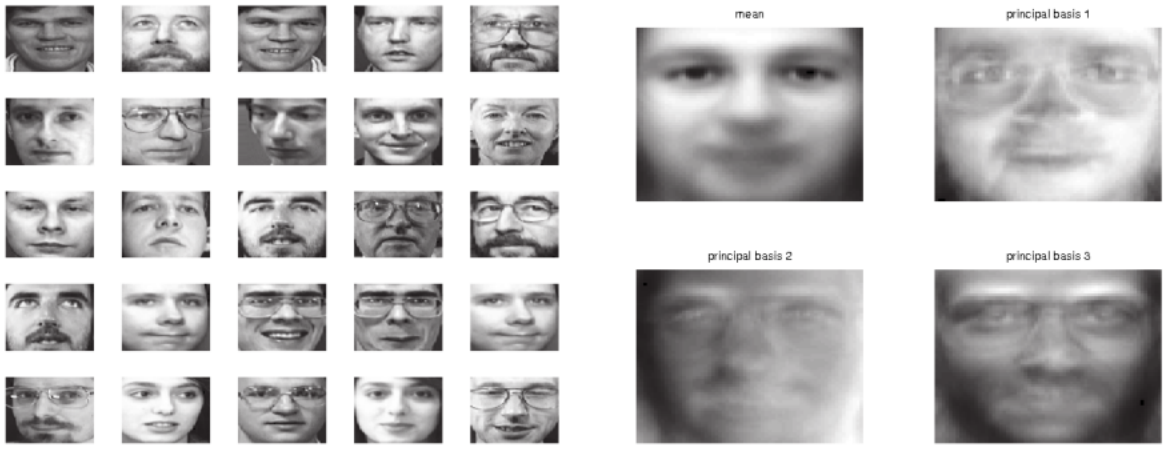
market basket analysis

In commercial data mining, there is much interest in a task called **market basket analysis**. The data consists of a (typically very large but sparse) binary matrix, where each column represents an item or product, and each row represents a transaction. We set $x_{ij} = 1$ if item j was purchased on the i'th transaction. Many items are purchased together (e.g., bread and butter), so there will be correlations amongst the bits. Given a new partially observed bit vector, representing a subset of items that the consumer has bought, the goal is to predict which other bits are likely to turn on, representing other items the consumer might be likely to buy. (Unlike

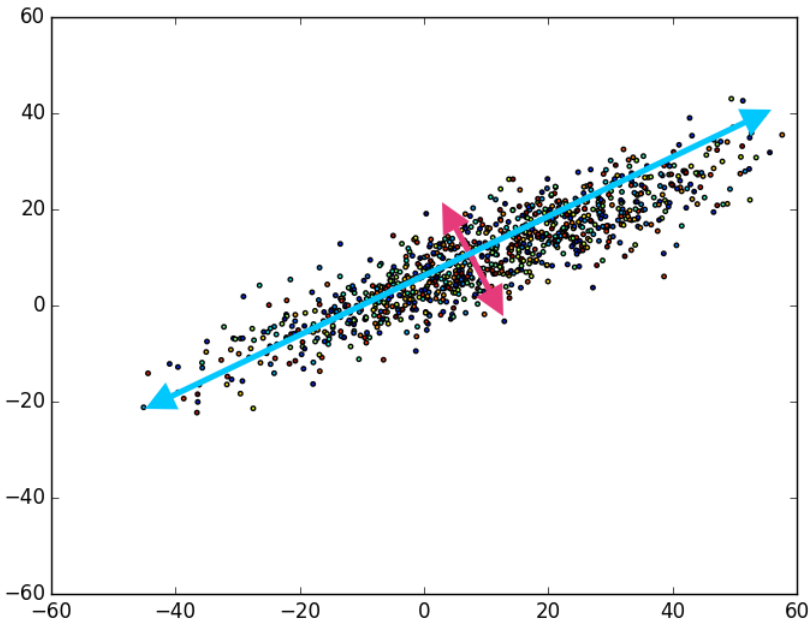
dimensionality reduction (sidenote)

The motivation behind this technique is that although the data may appear high dimensional, there may only be a small number of degrees of variability, corresponding to **latent factors**. For example, when modeling the appearance of face images, there may only be a few underlying latent factors which describe most of the variability, such as lighting, pose, identity, etc, as

useful for feature selection?



How to interpret principal basis?
go back to simple geometrical example

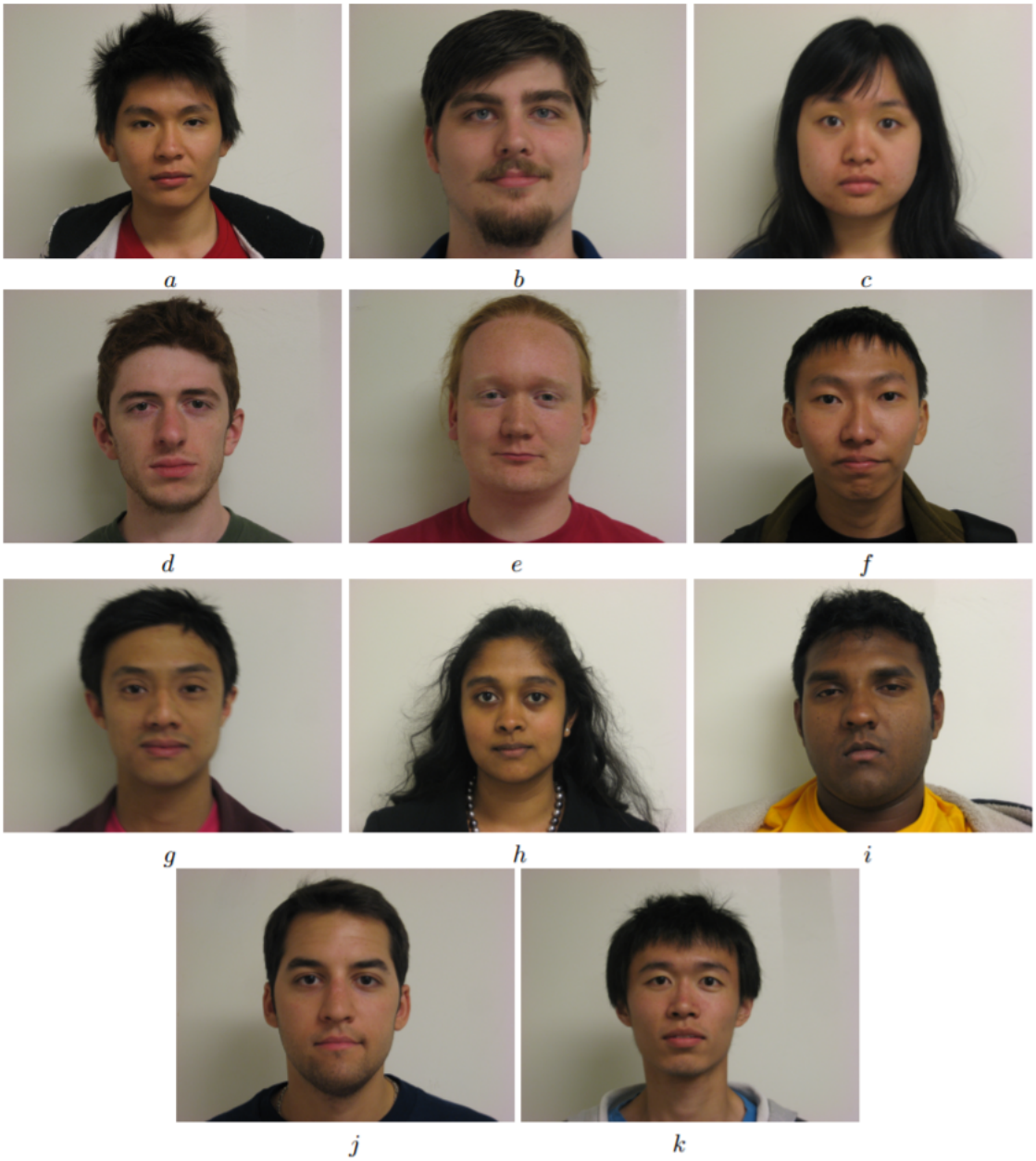


such a vector has *high* (or high *absolute value*) values on axis on which we have most variance

so we need to look at our base vector and see where values are high
in this case we have high values around the mouth, forehead

on the other hand eigenvalues show the importance of base vectors

does mean value influence pca?
normalization importance
we don't want our model to picj up on the differences in the noise, but in the meaningful data



Once each facial image was imported, it was observed that the positioning of each face was not consistent between images. For example, face b takes up a noticeable greater amount of the frame of the picture. In contrast, the relative face sizes of faces a, f, k are smaller. Consequently, the relative position of facial features within the picture frame were very inconsistent from facial image to facial image. This was most likely

http://digitalrepository.trincoll.edu/cgi/viewcontent.cgi?article=1221&context=theses