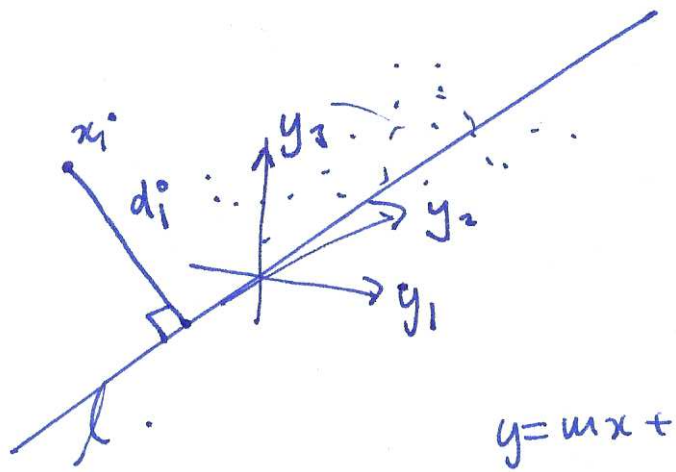
PCA: Principal component analysis

intuition. $X = (x_1, \dots, x_n) \subseteq \mathbb{R}^d$.

$(x_{i1}, x_{i2}, x_{i3}, \dots x_{id})$.



$y = mx + c$

$a_0 + a_1 x + a_2 y = 0$.

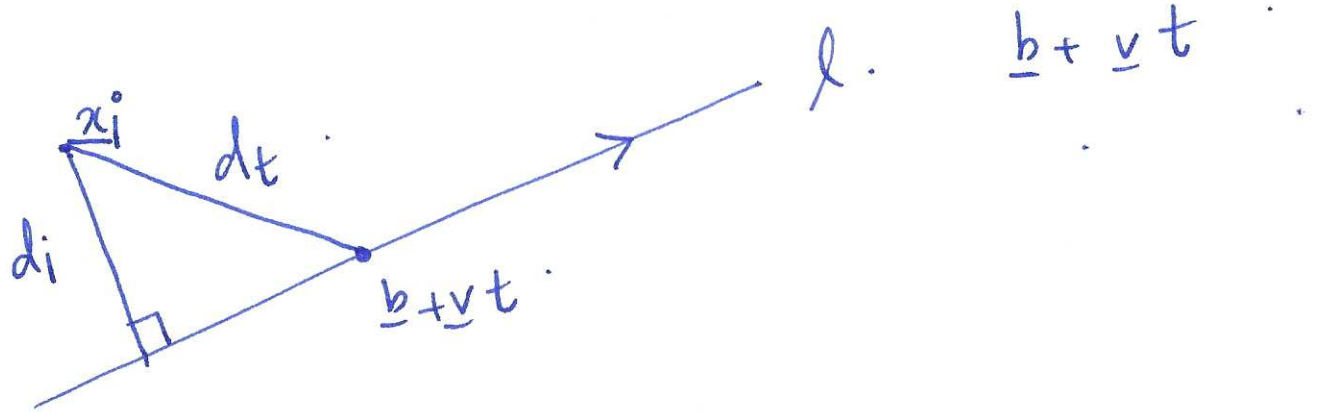
- find the best fit line.

$$a_0 + a_1 y_1 + a_2 y_2 + \dots + a_d y_d = 0.$$

choose a_i to minimize $\frac{1}{n} \sum_{i=1}^{n} d_i^2$

sum of squares error.

can find d_i just from vector algebra:



$$\|d_t\|^2 = \|\underline{b}+\underline{v}t-\underline{x}_i\| = (\underline{b}+\underline{v}t-\underline{x}_i).(\underline{b}+\underline{v}t-\underline{x}_i)$$

choose t to minimise this.

$$\frac{d}{dt}\|d_t\|^2 = 2\underline{v}.(\underline{b}+\underline{v}t-\underline{x}_i) = 0$$

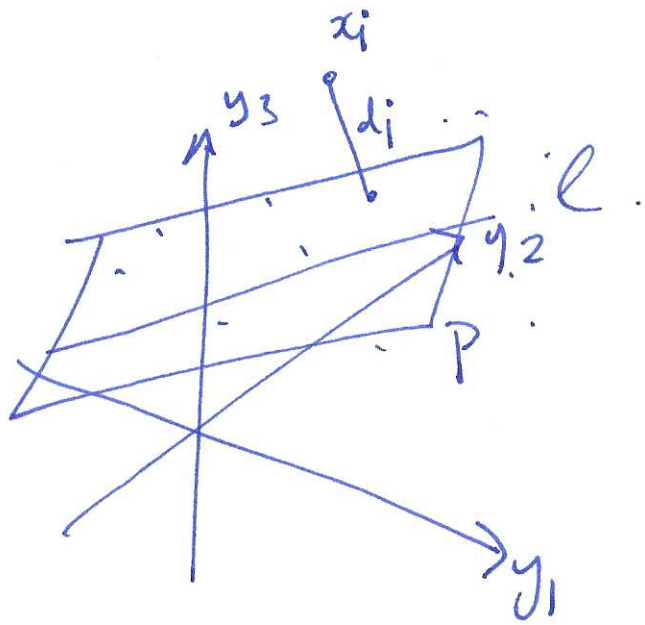
$$\underline{v}.\underline{b}+\underline{v}.\underline{v}t-\underline{v}.\underline{x}_i = 0$$

$$t = \frac{\underline{v}.(\underline{x}_i-\underline{b})}{\|\underline{v}\|^2}$$

can do this for planes etc.

choose P to minimize

$$\frac{1}{n}\sum d_i^2 .$$



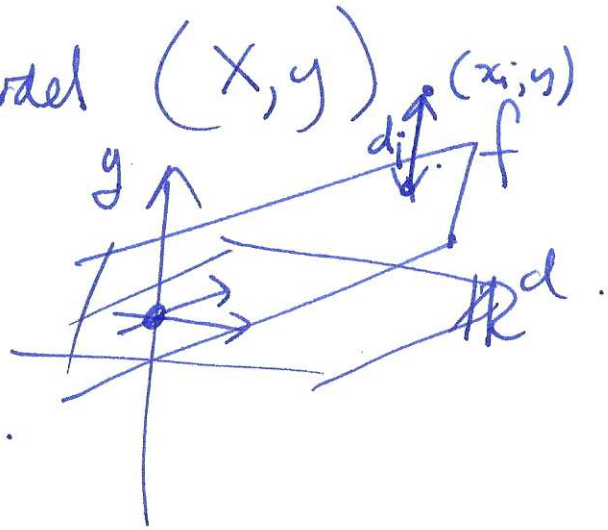
PCA (4 vectors).

— vectors .

— weights.

Lasso . recall linear regression model (X, y)

$X = (x_i) \subseteq \mathbb{R}^d$.



$f(x_i) = a_0 + a_1 x_{i1} + a_2 x_{i2} + \cdots a_d x_{id}$.

choose $a_0, a_1, \ldots, a_d$ to minimise sum of squares error $\frac{1}{n} \sum_{i=1}^{n} d_i$

some a_i's will be small.

Lasso choose $\lambda = 0.01$

minimise $\frac{1}{n} \sum_{i=1}^{n} d_i + \lambda \left| \#\text{ cds} \right|$.

fix $\lambda = 0.01$

$\rightsquigarrow$ get best $\underbrace{a_0, a_1, \ldots a_k}$, rest are zero.

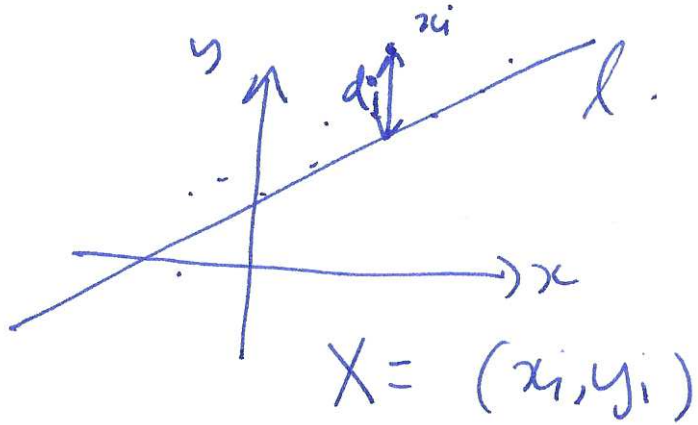
best k-dim linear model.

Lasso ↑ most important columns.

vs

PCA ↑ best fit directions

Bootstrap (type of cross-validation).



recall linear regression.

$y = mx + b$.

- get estimates for m, b.
- get accuracy / standard error. for m, b.

problem: don't get error estimates for any non-parametric model / many parametric models

solution: bootstrap

bootstrap · take lots of random samples from the original data set, compute estimates for these.

$X = (x_1, \dots, x_n)$ ← ~~1500~~ 1000.

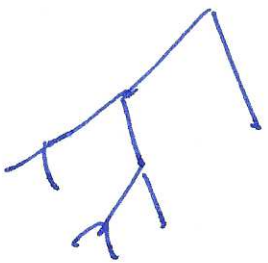
$\cup$

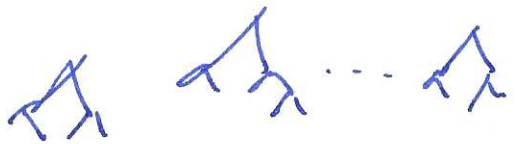
$B_1, B_2, \dots, B_{\text{~~100~~ 1000}}$ ← of size $k \leq n$.

— take variance / standard deviation of these estimate

— in practice this is a good estimate of the variance of the model.

decision trees · bagging / bootstrap aggregation · boostings · random forest





- high variance.
- easy to understand.

bagging

idea: • subsample b times.

$X \supseteq B_1, B_2 \dots, B_b$.

- build a decision tree for each B_i
- take average / majority vote of the b trees.
  (numeric) (categorical)
- tends to be lower variance than a single tree.

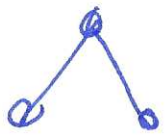
decision trees: boosting data set X. want $f(x_i) \approx y_i$

parameters: b # trees.

$\lambda = 0.01$ (shrinkage parameter).

assume rescaled data.

d # splits. (# leaves $= d+1$) $d=1$.



① start with $f(x) = 0$. and predictions/residuals $r_0 = y$ (X, y).

② for $i = 1$ to b

- fit a tree f^b with d splits to the data. (X, r_{b-1}).
- update. new $f_b := f_{b-1}(x) + \lambda f^b(x)$.
- update the residuals $r_i = r_{i-1} - \lambda f^b(x_i)$.

③ repeat for b times.