

14. Least squares classification

Outline

Classification

Least squares classification

Multi-class classifiers

Classification

- ▶ data fitting with outcome that takes on (non-numerical) values like
 - TRUE OR FALSE
 - SPAM OR NOT SPAM
 - DOG, HORSE, OR MOUSE
- ▶ outcome values are called *labels* or *categories*
- ▶ data fitting is called *classification*
- ▶ we start with case when there are two possible outcomes
- ▶ called *Boolean* or *2-way* classification
- ▶ we encode outcomes as +1 (TRUE) and −1 (FALSE)
- ▶ classifier has form $\hat{y} = \hat{f}(x), f : \mathbf{R}^n \rightarrow \{-1, +1\}$

Applications

- ▶ email spam detection
 - x contains features of an email message (word counts, ...)
- ▶ financial transaction fraud detection
 - x contains features of proposed transaction, initiator
- ▶ document classification (say, politics or not)
 - x is word count histogram of document
- ▶ disease detection
 - x contains patient features, results of medical tests
- ▶ digital communications receiver
 - y is transmitted bit; x contain n measurements of received signal

Prediction errors

- ▶ data point (x, y) , predicted outcome $\hat{y} = \hat{f}(x)$
- ▶ only four possibilities:
 - *True positive.* $y = +1$ and $\hat{y} = +1$.
 - *True negative.* $y = -1$ and $\hat{y} = -1$.(in these two cases, the prediction is *correct*)
 - *False positive.* $y = -1$ and $\hat{y} = +1$.
 - *False negative.* $y = +1$ and $\hat{y} = -1$.(in these two cases, the prediction is *wrong*)
- ▶ the errors have many other names, like Type I and Type II

Confusion matrix

- ▶ given data set $x^{(1)}, \dots, x^{(N)}$, $y^{(1)}, \dots, y^{(N)}$ and classifier \hat{f}
- ▶ count each of the four outcomes

	$\hat{y} = +1$	$\hat{y} = -1$	Total
$y = +1$	N_{tp}	N_{fn}	N_{p}
$y = -1$	N_{fp}	N_{tn}	N_{n}
All	$N_{\text{tp}} + N_{\text{fp}}$	$N_{\text{fn}} + N_{\text{tn}}$	N

- ▶ off-diagonal terms are prediction errors
- ▶ many error rates and accuracy measures are used
 - *error rate* is $(N_{\text{fp}} + N_{\text{fn}})/N$
 - *true positive (or recall) rate* is $N_{\text{tp}}/N_{\text{p}}$
 - *false positive rate (or false alarm rate)* is $N_{\text{fp}}/N_{\text{n}}$
- ▶ a proposed classifier is judged by its error rate(s) on a test set

Example

- ▶ spam filter performance on a test set (say)

	$\hat{y} = +1$ (SPAM)	$\hat{y} = -1$ (not SPAM)	Total
$y = +1$ (SPAM)	95	32	127
$y = -1$ (not SPAM)	19	1120	1139
All	114	1152	1266

- ▶ error rate is $(19 + 32)/1266 = 4.03\%$
- ▶ false positive rate is $19/1139 = 1.67\%$

Outline

Classification

Least squares classification

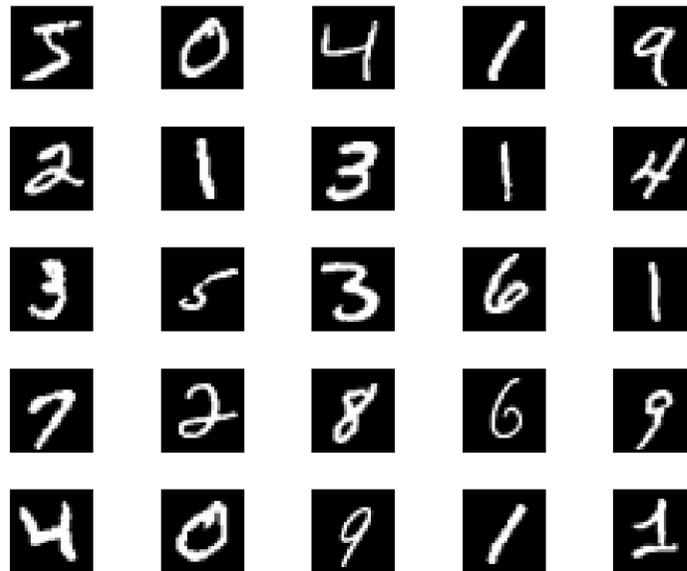
Multi-class classifiers

Least squares classification

- ▶ fit model \tilde{f} to encoded $(\pm 1) y^{(i)}$ values *using standard least squares data fitting*
- ▶ $\tilde{f}(x)$ should be near $+1$ when $y = +1$, and near -1 when $y = -1$
- ▶ $\tilde{f}(x)$ is a *number*
- ▶ use model $\hat{f}(x) = \mathbf{sign}(\tilde{f}(x))$
- ▶ (size of $\tilde{f}(x)$ is related to the ‘confidence’ in the prediction)

Handwritten digits example

- ▶ MNIST data set of 70000 28×28 images of digits 0, ..., 9



- ▶ divided into training set (60000) and test set (10000)
- ▶ x is 494-vector, constant 1 plus the 493 pixel values with nonzero values in at least 600 training examples
- ▶ $y = +1$ if digit is 0; -1 otherwise

Least squares classifier results

- ▶ training set results (error rate 1.6%)

	$\hat{y} = +1$	$\hat{y} = -1$	Total
$y = +1$	5158	765	5923
$y = -1$	167	53910	54077
All	5325	54675	60000

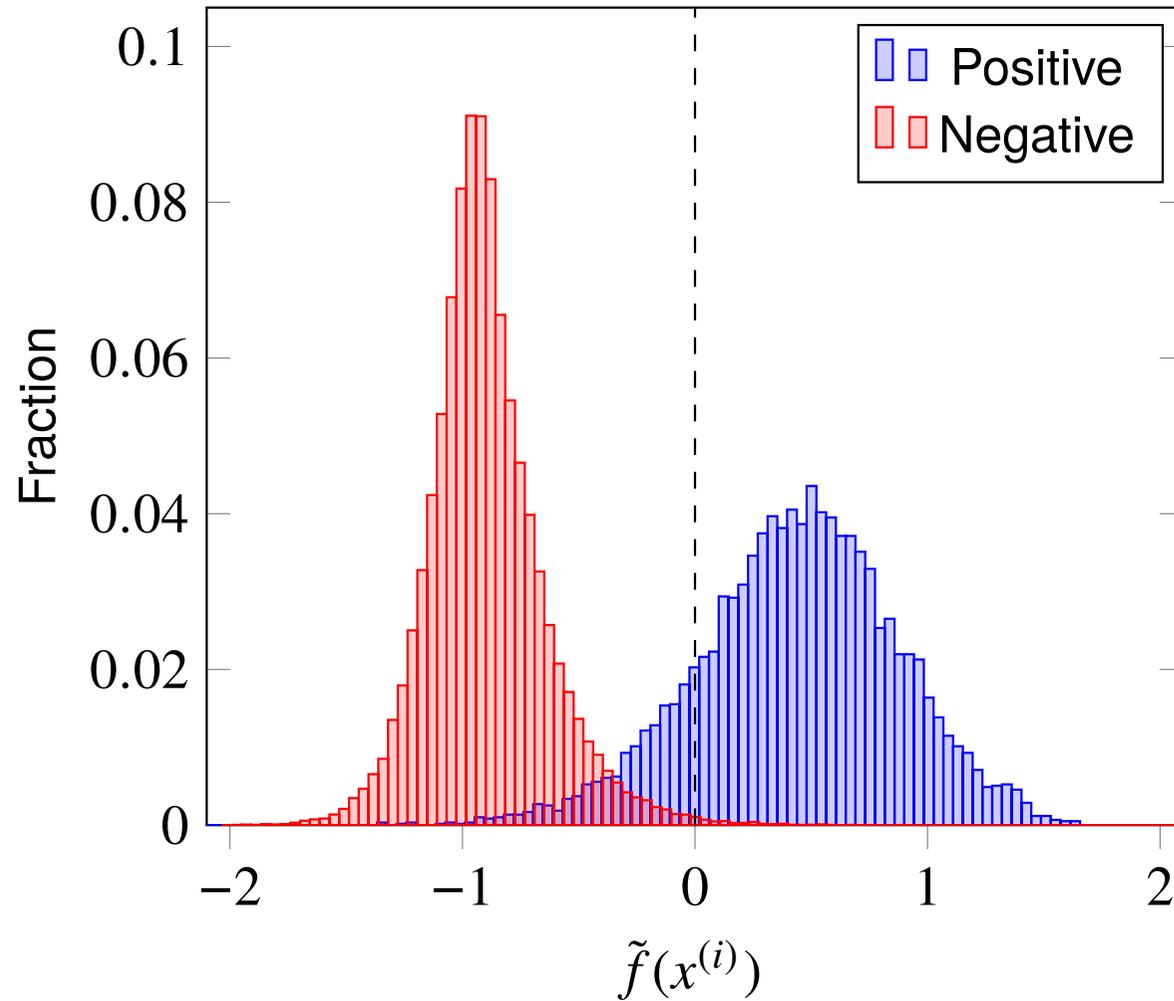
- ▶ test set results (error rate 1.6%)

	$\hat{y} = +1$	$\hat{y} = -1$	Total
$y = +1$	864	116	980
$y = -1$	42	8978	9020
All	906	9094	10000

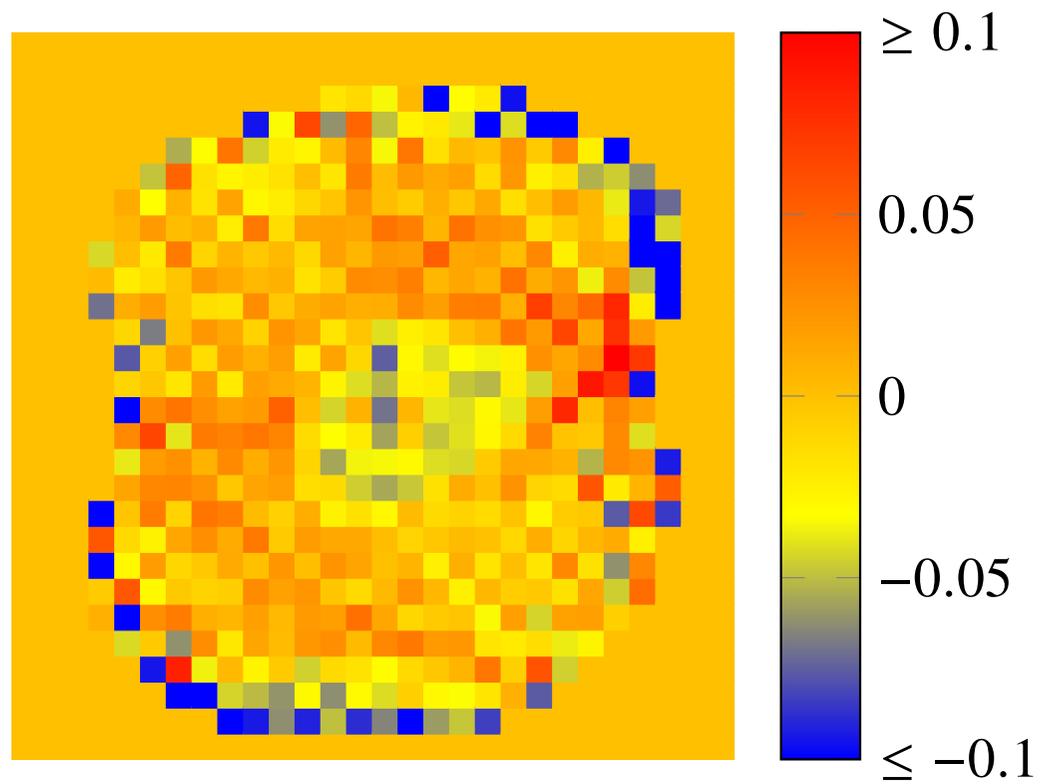
- ▶ we can likely achieve 1.6% error rate on unseen images

Distribution of least squares fit

distribution of values of $\tilde{f}(x^{(i)})$ over training set



Coefficients in least squares classifier



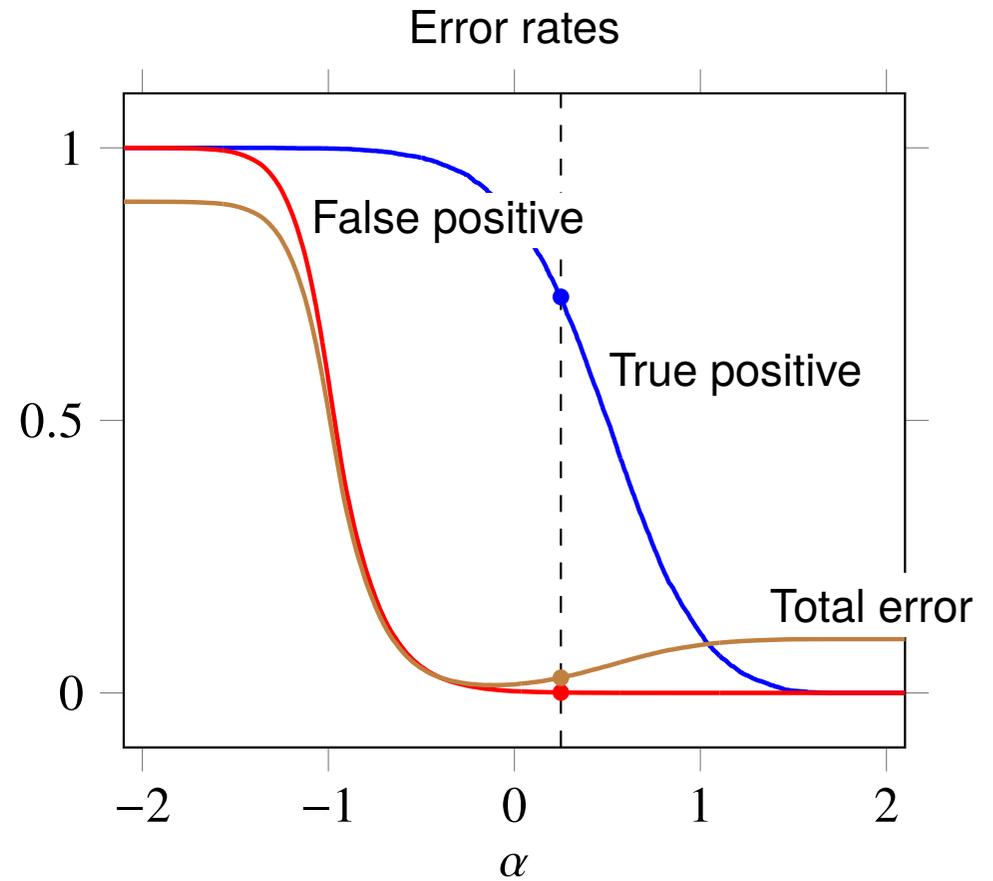
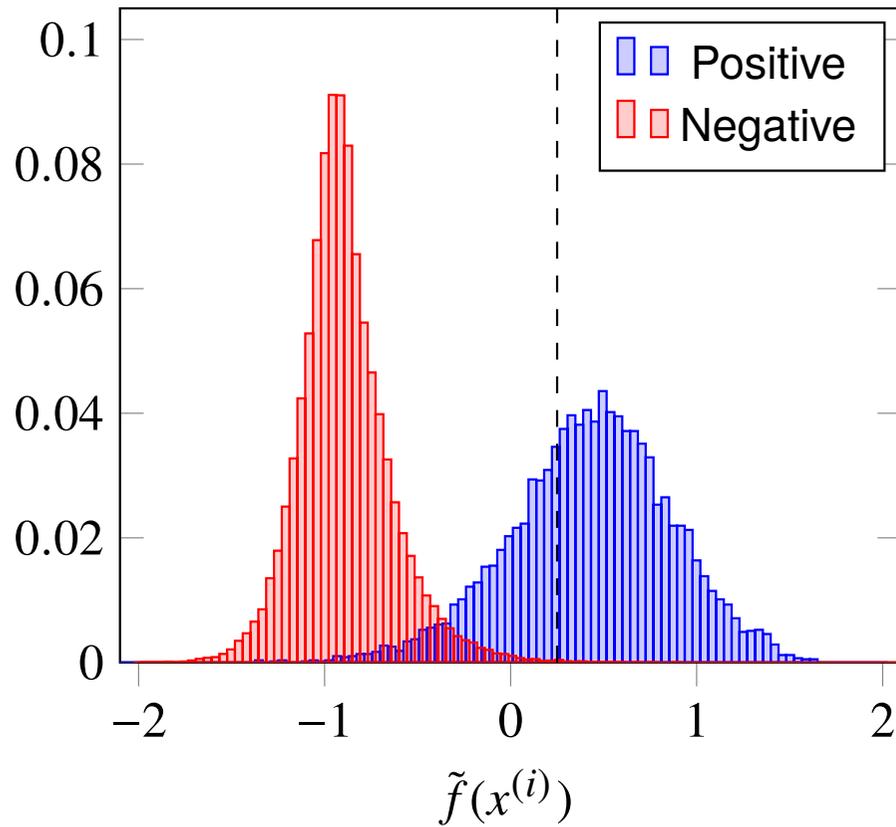
Skewed decision threshold

- ▶ use predictor $\hat{f}(x) = \mathbf{sign}(\tilde{f}(x) - \alpha)$, *i.e.*,

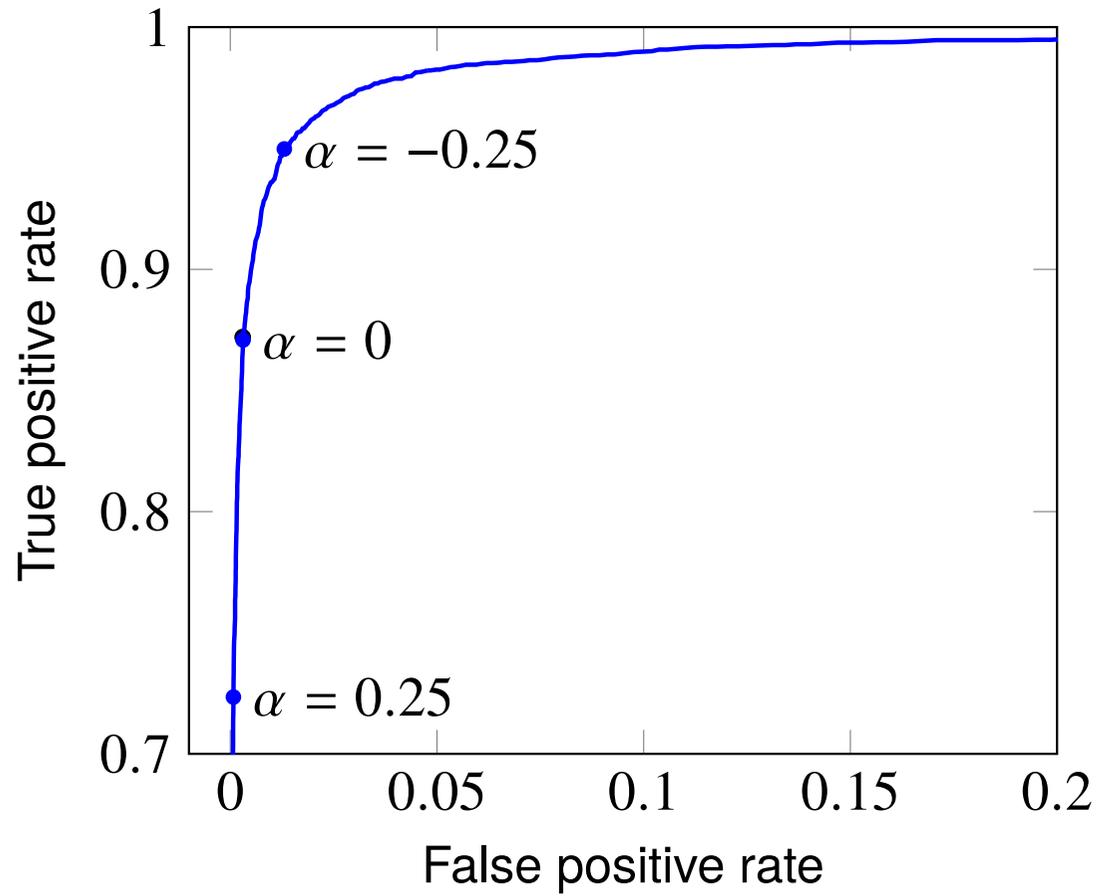
$$\hat{f}(x) = \begin{cases} +1 & \tilde{f}(x) \geq \alpha \\ -1 & \tilde{f}(x) < \alpha \end{cases}$$

- ▶ α is the *decision threshold*
- ▶ for positive α , false positive rate is lower but so is true positive rate
- ▶ for negative α , false positive rate is higher but so is true positive rate
- ▶ trade off curve of true positive versus false positive rates is called *receiver operating characteristic* (ROC)

Example



ROC curve



Outline

Classification

Least squares classification

Multi-class classifiers

Multi-class classifiers

- ▶ we have $K > 2$ possible labels, with label set $\{1, \dots, K\}$
- ▶ predictor is $\hat{f} : \mathbf{R}^n \rightarrow \{1, \dots, K\}$
- ▶ for given predictor and data set, confusion matrix is $K \times K$
- ▶ some off-diagonal entries may be much worse than others

Examples

- ▶ handwritten digit classification
 - guess the digit written, from the pixel values
- ▶ marketing demographic classification
 - guess the demographic group, from purchase history
- ▶ disease diagnosis
 - guess diagnosis from among a set of candidates, from test results, patient features
- ▶ translation word choice
 - choose how to translate a word into several choices, given context features
- ▶ document topic prediction
 - guess topic from word count histogram

Least squares multi-class classifier

- ▶ create a least squares classifier for each label versus the others
- ▶ take as classifier

$$\hat{f}(x) = \operatorname{argmax}_{\ell \in \{1, \dots, K\}} \tilde{f}_\ell(x)$$

(i.e., choose ℓ with largest value of $\tilde{f}_\ell(x)$)

- ▶ for example, with

$$\tilde{f}_1(x) = -0.7, \quad \tilde{f}_2(x) = +0.2, \quad \tilde{f}_3(x) = +0.8$$

we choose $\hat{f}(x) = 3$

Handwritten digit classification

confusion matrix, test set

Digit	Prediction										Total
	0	1	2	3	4	5	6	7	8	9	
0	944	0	1	2	2	8	13	2	7	1	980
1	0	1107	2	2	3	1	5	1	14	0	1135
2	18	54	815	26	16	0	38	22	39	4	1032
3	4	18	22	884	5	16	10	22	20	9	1010
4	0	22	6	0	883	3	9	1	12	46	982
5	24	19	3	74	24	656	24	13	38	17	892
6	17	9	10	0	22	17	876	0	7	0	958
7	5	43	14	6	25	1	1	883	1	49	1028
8	14	48	11	31	26	40	17	13	756	18	974
9	16	10	3	17	80	0	1	75	4	803	1009
All	1042	1330	887	1042	1086	742	994	1032	898	947	10000

error rate is around 14% (same as for training set)

Adding new features

- ▶ let's add 5000 random features (!), $\max\{(Rx)_j, 0\}$
 - R is 5000×494 matrix with entries ± 1 , chosen randomly
- ▶ now use least squares classification with 5494 feature vector
- ▶ results: training set error 1.5%, test set error 2.6%
- ▶ can do better with a little more thought in generating new features
- ▶ indeed, even better than humans can do (!!)

Results with new features

confusion matrix, test set

Digit	Prediction										Total
	0	1	2	3	4	5	6	7	8	9	
0	972	0	0	2	0	1	1	1	3	0	980
1	0	1126	3	1	1	0	3	0	1	0	1135
2	6	0	998	3	2	0	4	7	11	1	1032
3	0	0	3	977	0	13	0	5	8	4	1010
4	2	1	3	0	953	0	6	3	1	13	982
5	2	0	1	5	0	875	5	0	3	1	892
6	8	3	0	0	4	6	933	0	4	0	958
7	0	8	12	0	2	0	1	992	3	10	1028
8	3	1	3	6	4	3	2	2	946	4	974
9	4	3	1	12	11	7	1	3	3	964	1009
All	997	1142	1024	1006	977	905	956	1013	983	997	10000