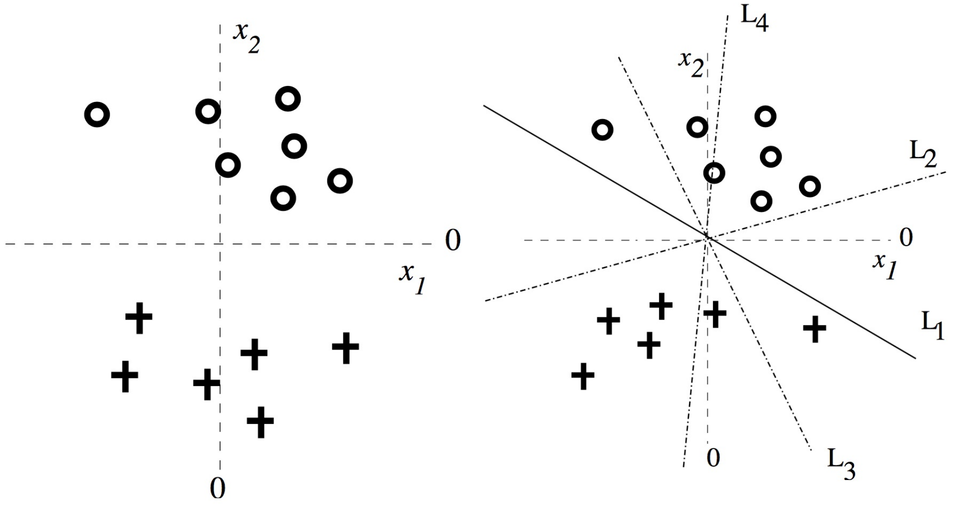
1. Assume the binary classification task depicted in Figure 3, which we attempt to solve with the simple linear logistic regression model

$$\widehat{\Pr}(Y = 1|X = x) = \hat{p}(x) = g(\beta_1 x_1 + \beta_2 x_2) = \frac{1}{1 + \exp(\beta_1 x_1 + \beta_2 x_2)}$$

for simplicity we do not use the parameter β_0. The training data is linearly separable, and the line L_1 is the result of logistic regression, with zero training error..



Assume that we would like to find the classifier by *maximizing* the following regularized objective function, in which *only* β_2 is regularized.

$$\prod_{i=1}^{n} [p(x_i)]^{y_i} [1 - p(x_i)]^{1-y_i} - \lambda \beta_2^2 = L(\beta_1, \beta_2) - \lambda \beta_2^2$$

(a) Assume that λ is large. Which of the four lines $\mathrm{L}_2, \mathrm{L}_3$ or L_4 determine whether it can result from regularizing β_2. Explain very briefly your reasons.

(b) If we change the form of regularization to one-norm (absolute value) and also regularize β_2 we get the following penalized log-likelihood

$$\prod_{i=1}^{n} [p(x_i)]^{y_i} [1 - p(x_i)]^{1-y_i} - \lambda(|\beta_1| + |\beta_2|) = L(\beta_1, \beta_2) - \lambda(|\beta_1| + |\beta_2|)$$

As we increase the regularization parameter λ which of the following scenarios is expected to be observed? Explain why.

i. First β_1 will become 0, then β_2.
ii. β_1 and β_2 will become zero simultaneously.
iii. First β_2 will become 0, then β_1.
iv. None of the weights will become exactly zero, only smaller as λ increases

Solution:[1]

(a) When we regularize β_2, the resulting boundary can rely less on the value of x_2 and therefore becomes more vertical.

- L2 here seems to be more horizontal than the unregularized solution so it cannot come as a result of penalizing β_2.
- When β_2 is small relative to β_1^2 (as evidenced by high slope), and even though it would assign a rather low log-probability to the observed labels, it could be forced by a large regularization parameter λ. So L3 can arise as a result.
- For very large λ, we obtain a separator that is very close to vertical (with negative slope) and in the limit, entirely vertical (line $x_1 = 0$ or the x_2 axis). L_4 here is reflected across the x_2 axis and has a positive slope, and therefore represents a poorer solution than its counterpart on the other side. For moderate regularization we have to get the best solution that we can construct while keeping β_2 small. L_4 is not the best and thus cannot come as a result of regularizing β_2.

(b) The data can be classified with zero training error and therefore also with high log-probability by looking at the value of x_2 alone, i.e. making $\beta_1 = 0$. Initially we might prefer to have a non-zero value for β_1 but it will go to zero rather quickly as we increase regularization. Note that we pay a regularization penalty for a non-zero value of β_1 and if it does not help classification why should the penalty be paid? The $\mathscr{L}_1$ regularization ensures that β_1 will indeed go to exactly zero. As λ increases further, even β_2 will eventually become zero. We pay higher and higher cost for setting β_2 to a non-zero value. Eventually this cost overwhelms the gain from the log-probability of labels that we can achieve with a non-zero β_2. Note that when $\beta_1 = \beta_2 = 0$, the log-probability of labels is a finite value $n \log(0.5)$.

[1] Important Note: Posting the course material to online forums or sharing it with other students is strictly prohibited. Instances will be reported to USC officials as academic dishonesty for disciplinary action.

2. A statistician is working on the amount of funding that companies obtain on a crwod-sourcing website and has developed the following model. She used 26 companies to obtain the model

$$\hat{y} = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + b_4x_4 + b_5x_5$$
$$\hat{y} = 964.8 + 700.2x_1 + 317.5x_2 - 200.2x_3 + 15.3x_4 + 17.1x_5$$

The standard errors are:

$$s_{b_1} = 12.$$
$$s_{b_2} = 22.5$$
$$s_{b_3} = 101.8$$
$$s_{b_4} = 45.3$$
$$s_{b_5} = 2.3$$

- $\hat{y}$: the amount of funding obtained by a company in 1000 dollars
- x_1: the average annual salary of the founders
- x_2: the number of employees the startup hired
- x_3: a dummy vaiable that is 1 when the company's field is information technology and 0 otherwise
- x_4: the age of the company
- x_5 is a dummy variable taking value 1 if the founders had previous failures and 0 otherwise

(a) Interpret the estimated coefficients $b_0 = 964.8$ and $b_3 = -200.2$ (10 pts)

(b) Test, at the 2% level, the null hypothesis that the true coefficient on the dummy variable x_5 is 0 against the alternative that it is not 0. (10 pts)

(c) Find and interpret a 99.8% confidence interval for the parameter β_4. (10 pts)

(d) If for the model, SSR=18147.5 (Regression Sum of Squares) and SSE = 17136.5 (Residual Sum of Squares), test the hypothesis that all the coefficients of the model are 0 (test overal significance of the model) using $\alpha = 5\%$. (10 pts).

Solutions:

(a) b_0 is the amount of the dependent variable that was not explaned by the independent variables. b_3 means that if the field of the company is information technology, on average, the funding tat it wil receive frm the website will decrease by 200.2 units (1000 dollars)

(b) $\mathbf{H_0} : \beta_5 = 0,\ \mathbf{H_1} : \beta_5 \neq 0,\ t_{b_5} = \frac{b_5 - 0}{s_{b_5}} = \frac{17.1}{2.3} = 7.44$

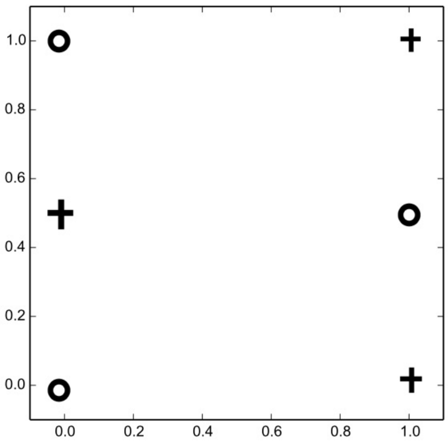
The rejection region is $t > t_{n-K-1,\alpha/2} = t_{26-5-1,.01}$ or $t < -t_{n-K-1,\alpha/2} = -t_{26-5-1,.01}$.

But from the table, $t_{26-5-1,.01} = 2.528$, therefore, we reject the null hypothsi that $\beta_5 = 0$.

(c) $t_{n-K-1,\alpha/2} = t_{26-5-1,0.001} =$, so the Confidence interval for β_4 is: $[b_4 - t_{n-K-1,\alpha/2} s_{b_4}, b_4 + t_{n-K-1,\alpha/2} s_{b_1}] = [15.3-(3.552)(45.3), 15.3+(3.552)(45.3)] = [-145.61, 176.21]$.

(d) $F = \frac{SSR/K}{SSE/(n-K-1)} = \frac{18147.5/5}{17136.5/(26-5-1)} = 4.2$. The rejection region is $F > F_{K,n-K-1,\alpha} = F_{5,20,0.05} = 2.7109$, which means that we reject the null hypothesis that all coefficients are 0.

3. For the two dimensional training data shown below, determine whether or not each of the classification methods below, when trained appropriately, will have zero errors on the training set. In each case, briefly justify your answer. Moreover, provide a reasonable confusion matrix for each case.



   (a) Logistic Regression

   (b) SVM with Linear Kernel

   (c) SVM with RBF Kernel

   (d) Decision Tree

   (e) 3-Nearest-Neighbor Classifier (with Euclidean Distance).

Solution:

(a) Logistic Regression and Linear SVM: linear decision boundaries,hence no.

(b) SVM with RBF kernel: yes.

(c) 3-NN: the 3 nearest neighbors of any point in our training set are 1 of the same class and 2 of the opposite class, hence 3-NN will be systematically wrong.

(d) DT: yes, one can partition the space with lines orthogonal to the axes so that every sample ends up in a different region.

For methods with zero training error, the Confusion Matrix would be:

	Class o	Class +
Class o	3	0
Class +	0	3

For KNN:

	Class o	Class +
Class o	0	3
Class +	3	0

For SVM, LR:

	Class o	Class +
Class o	2	1
Class +	1	2

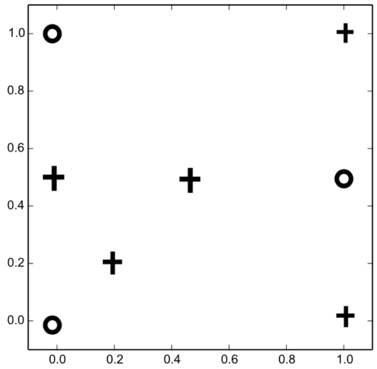
4. For a numeric input, instead of a binary split in a decision tree, one can use a ternary split with two thresholds and three branches as $X_j < s_1, s_1 \leq X_j < s_2, X_j \geq s_2$.

   (a) Propose a modification of the tree learning method to adjust the two thresholds, s_1 and s_2.

   (b) What are the advantages and the disadvantages of such a node over a binary node?

   (c) How would you choose between a binary and a trenary decision tree for a given data set?

   (d) Perform two iterations of the trenary tree algoritm on the tree shown in below and draw the corresponding tree



Solution:

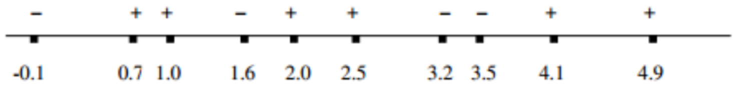
(a) For the numeric attributes, instead of one split threshold, we need to try all possible pairs of split thresholds and choose the best. When there are two splits, there are three children, and in calculating the entropy/ Gini index after the splits, we need to sum up over the three sets corresponding to the instances taking the three branches.

(b) The computational complexity of finding the best pair is higher and each node stores two thresholds instead of one and has three branches instead of two. The advantage is that one ternary node splits an input into three, whereas this requires two successive binary nodes.

(c) Which one is better depends on the data at hand; if we have a ground truth that requires bounded intervals (e.g., rectangles), a ternary node may be advantageous. One has to choose between binary and trenary nodes using *Cross Validation*.

(d) Left to the students.

5. Consider the following dataset with one real-valued input and one binary output (+ or -). The following questions assume that we are using k- nearest-neighbor learning with Euclidean distance to predict Y for an input X. What is the leave-one-out cross-validation error of 1-NN and 3-NN on this dataset, and decide which k is better.



Solution: For each X_i, consider the majority vote of 1 nearest neighbors.

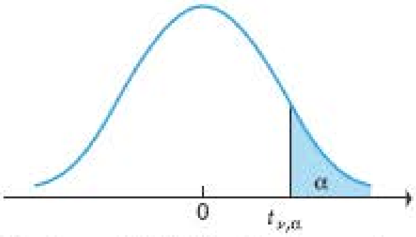
$E_{LOOCV} = 0.4$

For each X_i, consider the majority vote of 3 nearest neighbors.

$E_{LOOCV} = 0.8$

The 1-NN method is better

Upper Critical Values of Student's t Distribution with ν Degrees of Freedom



For selected probabilities, α, the table shows the values $t_{\nu,\alpha}$ such that $P(t_\nu > t_{\nu,\alpha}) = \alpha$, where t_ν is a Student's t random variable with ν degress of freedom. For example, the probability is .10 that a Student's t random variable with 10 degrees of freedom exceeds 1.372.

	Probability of Exceeding the Critical Value					
ν	0.10	0.05	0.025	0.01	0.005	0.001
1	3.078	6.314	12.706	31.821	63.657	318.313
2	1.886	2.920	4.303	6.965	9.925	22.327
3	1.638	2.353	3.182	4.541	5.841	10.215
4	1.533	2.132	2.776	3.747	4.604	7.173
5	1.476	2.015	2.571	3.365	4.032	5.893
6	1.440	1.943	2.447	3.143	3.707	5.208
7	1.415	1.895	2.365	2.998	3.499	4.782
8	1.397	1.860	2.306	2.896	3.355	4.499
9	1.383	1.833	2.262	2.821	3.250	4.296
10	1.372	1.812	2.228	2.764	3.169	4.143
11	1.363	1.796	2.201	2.718	3.106	4.024
12	1.356	1.782	2.179	2.681	3.055	3.929
13	1.350	1.771	2.160	2.650	3.012	3.852
14	1.345	1.761	2.145	2.624	2.977	3.787
15	1.341	1.753	2.131	2.602	2.947	3.733
16	1.337	1.746	2.120	2.583	2.921	3.686
17	1.333	1.740	2.110	2.567	2.898	3.646
18	1.330	1.734	2.101	2.552	2.878	3.610
19	1.328	1.729	2.093	2.539	2.861	3.579
20	1.325	1.725	2.086	2.528	2.845	3.552
21	1.323	1.721	2.080	2.518	2.831	3.527
22	1.321	1.717	2.074	2.508	2.819	3.505
23	1.319	1.714	2.069	2.500	2.807	3.485
24	1.318	1.711	2.064	2.492	2.797	3.467
25	1.316	1.708	2.060	2.485	2.787	3.450
26	1.315	1.706	2.056	2.479	2.779	3.435
27	1.314	1.703	2.052	2.473	2.771	3.421
28	1.313	1.701	2.048	2.467	2.763	3.408
29	1.311	1.699	2.045	2.462	2.756	3.396
30	1.310	1.697	2.042	2.457	2.750	3.385
40	1.303	1.684	2.021	2.423	2.704	3.307
60	1.296	1.671	2.000	2.390	2.660	3.232
100	1.290	1.660	1.984	2.364	2.626	3.174
∞	1.282	1.645	1.960	2.326	2.576	3.090
ν	0.10	0.05	0.025	0.01	0.005	0.001

F - Distribution ($\alpha = 0.05$ in the Right Tail)

Denominator Degrees of Freedom

df_2 \ df_1	Numerator Degrees of Freedom								
	1	2	3	4	5	6	7	8	9
1	161.45	199.50	215.71	224.58	230.16	233.99	236.77	238.88	240.54
2	18.513	19.000	19.164	19.247	19.296	19.330	19.353	19.371	19.385
3	10.128	9.5521	9.2766	9.1172	9.0135	8.9406	8.8867	8.8452	8.8123
4	7.7086	9.9443	6.5914	6.3882	6.2561	6.1631	6.0942	6.0410	6.9988
5	6.6079	5.7861	5.4095	5.1922	5.0503	4.9503	4.8759	4.8183	4.7725
6	5.9874	5.1433	4.7571	4.5337	4.3874	4.2839	4.2067	4.1468	4.0990
7	5.5914	4.7374	4.3468	4.1203	3.9715	3.8660	3.7870	3.7257	3.6767
8	5.3177	4.4590	4.0662	3.8379	3.6875	3.5806	3.5005	3.4381	3.3881
9	5.1174	4.2565	3.8625	3.6331	3.4817	3.3738	3.2927	3.2296	3.1789
10	4.9646	4.1028	3.7083	3.4780	3.3258	3.2172	3.1355	3.0717	3.0204
11	4.8443	3.9823	3.5874	3.3567	3.2039	3.0946	3.0123	2.9480	2.8962
12	4.7472	3.8853	3.4903	3.2592	3.1059	2.9961	2.9134	2.8486	2.7964
13	4.6672	3.8056	3.4105	3.1791	3.0254	2.9153	2.8321	2.7669	2.7144
14	4.6001	3.7389	3.3439	3.1122	2.9582	2.8477	2.7642	2.6987	2.6458
15	4.5431	3.6823	3.2874	3.0556	2.9013	2.7905	2.7066	2.6408	2.5876
16	4.4940	3.6337	3.2389	3.0069	2.8524	2.7413	2.6572	2.5911	2.5377
17	4.4513	3.5915	3.1968	2.9647	2.8100	2.6987	2.6143	2.5480	2.4943
18	4.4139	3.5546	3.1599	2.9277	2.7729	2.6613	2.5767	2.5102	2.4563
19	4.3807	3.5219	3.1274	2.8951	2.7401	2.6283	2.5435	2.4768	2.4227
20	4.3512	3.4928	3.0984	2.8661	2.7109	2.5990	2.5140	2.4471	2.3928
21	4.3248	3.4668	3.0725	2.8401	2.6848	2.5727	2.4876	2.4205	2.3660
22	4.3009	3.4434	3.0491	2.8167	2.6613	2.5491	2.4638	2.3965	2.3419
23	4.2793	3.4221	3.0280	2.7955	2.6400	2.5277	2.4422	2.3748	2.3201
24	4.2597	3.4028	3.0088	2.7763	2.6207	2.5082	2.4226	2.3551	2.3002
25	4.2417	3.3852	2.9912	2.7587	2.6030	2.4904	2.4047	2.3371	2.2821
26	4.2252	3.3690	2.9752	2.7426	2.5868	2.4741	2.3883	2.3205	2.2655
27	4.2100	3.3541	2.9604	2.7278	2.5719	2.4591	2.3732	2.3053	2.2501
28	4.1960	3.3404	2.9467	2.7141	2.5581	2.4453	2.3593	2.2913	2.2360
29	4.1830	3.3277	2.9340	2.7014	2.5454	2.4324	2.3463	2.2783	2.2229
30	4.1709	3.3158	2.9223	2.6896	2.5336	2.4205	2.3343	2.2662	2.2107
40	4.0847	3.2317	2.8387	2.6060	2.4495	2.3359	2.2490	2.1802	2.1240
60	4.0012	3.1504	2.7581	2.5252	2.3683	2.2541	2.1665	2.0970	2.0401
120	3.9201	3.0718	2.6802	2.4472	2.2899	2.1750	2.0868	2.0164	1.9588
∞	3.8415	2.9957	2.6049	2.3719	2.2141	2.0986	2.0096	1.9384	1.8799