

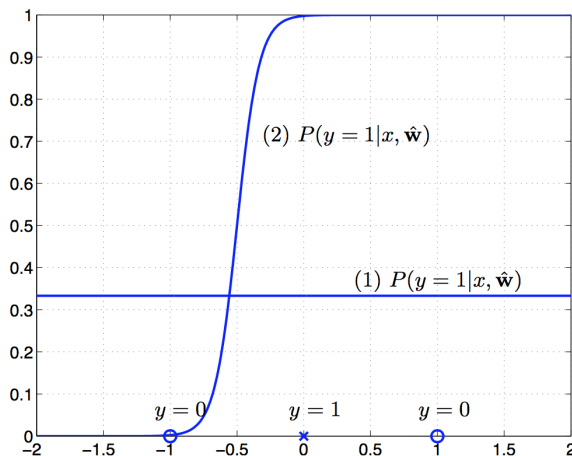
6.867: Exercises (Week 4)

Sept 29, 2016

Contents

1	Logistic regression: basic intuition with one-dimensional data	2
2	Logistic Regression: maximum likelihood	3
3	More on Logistic regression	3
4	Logistic regression vs LDA/QDA	6
5	Softmax	7
6	SVMs: basic intuition with one-dimensional data	8
7	SVMs: Where's the hyperplane?	9
8	SVMs: primal scream	9
9	SVMs: modifying basic formulation	9
10	SVM-like training methods	12
11	Review: Regression and model selection	14

1 Logistic regression: basic intuition with one-dimensional data



Consider a simple one dimensional logistic regression model

$$P(y = 1 \mid x, w) = \sigma(w_0 + w_1 x)$$

where $\sigma(z) = (1 + \exp(-z))^{-1}$ is the logistic function.

The figure above shows two possible conditional distributions $P(y = 1 \mid x, w)$, viewed as a function of x , that we can get by changing the parameters w .

Assume we have a data set $\mathcal{D} = \{(-1, 0), (0, 1), (1, 0)\}$.

1. Please indicate the number of classification errors for each conditional given \mathcal{D} .

Conditional (1) makes () classification errors

Conditional (2) makes () classification errors

2. Which of these two hypotheses assigns a higher likelihood to the data?
3. If your loss function for predictions was

$$L(g, a) = \begin{cases} 0 & \text{if } g = a \\ 1 & \text{if } g = 1 \text{ and } a = 0 \\ 10 & \text{if } g = 0 \text{ and } a = 1 \end{cases}$$

- What output would you predict for $x = -1$ when conditional (1) is the result of your learning?
- What output would you predict for $x = -1$ when conditional (2) is the result of your learning?

2 Logistic Regression: maximum likelihood

Bishop 4.14

Show that for a linearly separable data set, the maximum likelihood solution for the logistic regression model is obtained by finding a vector w whose decision boundary $w^T \phi(x) = 0$ separates the classes and then taking the magnitude of w to infinity.

3 More on Logistic regression

3.1 Regular guys

We are interested in regularizing the terms separately in logistic regression.

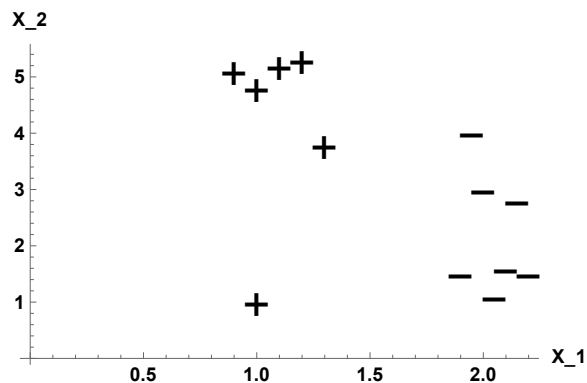
(a) Consider the data in the figure below where we fit the model

$$P(y = 1 \mid x, w) = \text{Sigmoid}(w_0 + w_1 x_1 + w_2 x_2)$$

Suppose we fit the model by maximum likelihood, that is, we minimize

$$J(w) = -\log \Pr(D_{\text{train}}; w)$$

Sketch a possible decision boundary corresponding to w^* .



(b) Is your decision boundary unique?

(c) How many classification errors does it make on the training set?

(d) Now suppose we regularize only the w_0 parameter; that is, we minimize

$$J(w) = -\log \Pr(D_{\text{train}}; w) + \lambda w_0^2$$

with λ approaching ∞ . Sketch a possible decision boundary corresponding to w^* .

(e) How many classification errors does it make on the training set?

(f) Now suppose we regularize only the w_1 parameter; that is, we minimize

$$J(w) = -\log \Pr(D_{\text{train}}; w) + \lambda w_1^2$$

with λ approaching ∞ . Sketch a possible decision boundary corresponding to w^* .

(g) How many classification errors does it make on the training set?

(h) Now suppose we regularize only the w_2 parameter; that is, we minimize

$$J(w) = -\log \Pr(D_{\text{train}}; w) + \lambda w_2^2$$

with λ approaching ∞ .

(i) How many classification errors does it make on the training set?

3.2 Advertising!

We consider here a logistic regression model for classifying potential web ads. The class labels indicate whether the ad will generate a lot of clicks ($y = 1$) or few clicks ($y = -1$). The probabilities over the labels, given the ad x , are assigned according to

$$P(y = 1 | x, \underline{\theta}) = g(\underline{\theta}^T \underline{\phi}(x))$$

where $g(z) = (1 + e^{-z})^{-1}$ is the logistic (sigmoid) function. The feature vectors simply indicate whether a word w appears in the word x :

$$\phi_w(x) = \begin{cases} 1, & \text{if } x \text{ contains word } w \\ 0, & \text{otherwise} \end{cases}$$

There are only two words we are interested in so that $w \in \{\text{weird}, \text{trick}\}$. The ads are first turned into all lowercase letters before evaluating the corresponding feature vectors.

In this problem, we do not add an extra “1” to the feature vectors; they have only 2 components and therefore the weight vectors will only have two components.

We would like to train the logistic regression model based on past ads x_1, \dots, x_n and labels y_1, \dots, y_n (from focus group ratings) by maximizing the regularized log-likelihood of the labels:

$$\sum_{t=1}^n \log P(y_t | x_t, \underline{\theta}) - \frac{\lambda}{2} \|\underline{\theta}\|^2 = \sum_{t=1}^n \log g(\underline{y}_t \underline{\theta}^T \underline{\phi}(x_t)) - \frac{\lambda}{2} \|\underline{\theta}\|^2$$

The problem is a bit hard to solve well, however, since we only had enough money to acquire three labeled ads:

$$\begin{aligned} \underline{\phi}(x_1) &= [1, 1]^T & y_1 &= 1 \\ \underline{\phi}(x_2) &= [1, 0]^T & y_2 &= -1 \\ \underline{\phi}(x_3) &= [0, 0]^T & y_3 &= 1 \end{aligned}$$

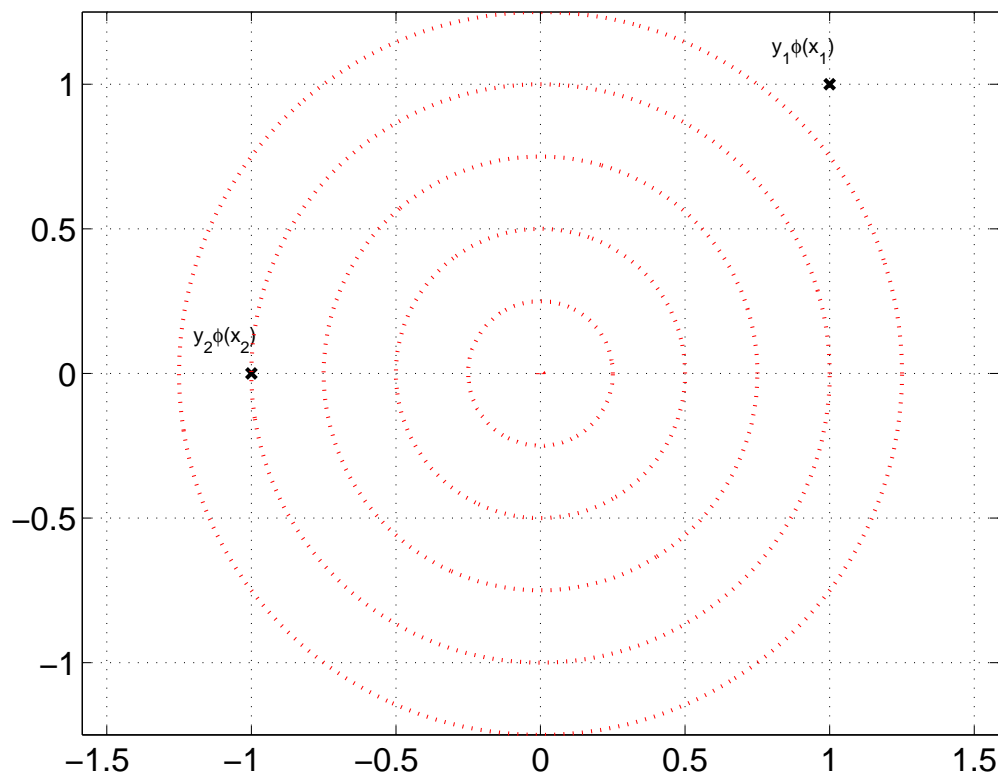
(a) Does it matter how the third ad is labeled?

(b) What would be the value of the resulting training log-likelihood be if we set $\lambda = 0$?

- (c) The logistic regression model associates class probabilities with each point. Does the effect of the regularization penalty on these probabilities depend on the norms $\|\phi(x_t)\|$?
- (d) For large λ (strong regularization), the log-likelihood terms will behave as linear functions of $\underline{\theta}$ (see Figure 4).

$$\log g(y_t \underline{\theta}^T \phi(x_t)) \approx \frac{1}{2} y_t \underline{\theta}^T \phi(x_t)$$

In this regime (large λ), draw how $\hat{\underline{\theta}}$ behaves as a function of λ . In other words, draw $\hat{\underline{\theta}}$ (at any scale) and its direction of change with increasing λ . We will classify correctly only one of the training examples. Why?



Points $y_1\phi(x_1)$ and $y_2\phi(x_2)$ along with the contours of the regularization term $\|\underline{\theta}\|$.

- (e) For this particular case, but for general $\lambda > 0$, will the resulting classification decisions (predicted labels) for new exams depend on the value of λ ? (Assuming the loss is symmetric, that is $y = 1$ if $P(y = 1|x, \underline{\theta}) > 0.5$)

3.3 Being greedy

Here we will look at methods for selecting input features for a logistic regression model

$$P(y = 1 | \mathbf{x}, \mathbf{w}) = g(w_0 + w_1 x_1 + w_2 x_2)$$

The available training examples are very simple, involving only binary valued inputs:

Number of copies	x_1	x_2	y
10	1	1	1
10	0	1	0
10	1	0	0
10	0	0	1

So, for example, there are 10 copies of $\mathbf{x} = [1, 1]^T$ in the training set, all labeled $y = 1$. The correct label is actually a deterministic function of the two features: $y = 1$ if $x_1 = x_2$ and zero otherwise.

We define greedy selection in this context as follows: we start with no features (train only with w_0) and successively try to add new features provided that each addition strictly improves the training log-likelihood. We use no other stopping criterion.

1. Could greedy selection add either x_1 or x_2 in this case? Answer Y or N.
2. What is the classification error of the training examples that we could achieve by including both x_1 and x_2 in the logistic regression model?
3. Suppose we define another possible feature to include, a function of x_1 and x_2 . Which of the following features, if any, would permit us to correctly classify all the training examples when used in combination with x_1 and x_2 in the logistic regression model:
 - $x_1 - x_2$
 - $x_1 x_2$
 - x_2^2
4. Could the greedy selection method choose this feature as the first feature to add when the available features are x_1 , x_2 and your choice of the new feature? Answer Y or N.

4 Logistic regression vs LDA/QDA

(Murphy 4.20)

Suppose we train the following binary classifiers via maximum likelihood.

- (a) GaussI: A generative classifier, where the class conditional densities are Gaussian, with both covariance matrices set to I (identity matrix), i.e., $p(\mathbf{x} | y = c) = \mathcal{N}(\mathbf{x} | \mu_c, I)$. We assume $p(y)$ is uniform.
- (b) GaussX: as for GaussI, but the covariance matrices are unconstrained, i.e., $p(\mathbf{x} | y = c) = \mathcal{N}(\mathbf{x} | \mu_c, \Sigma_c)$.
- (c) LinLog: A logistic regression model with linear features.
- (d) QuadLog: A logistic regression model, using linear and quadratic features (i.e., polynomial basis function expansion of degree 2).

After training we compute the performance of each model M on the training set as follows:

$$L(M) = \frac{1}{n} \sum_{i=1}^n \log p(y^{(i)} | \mathbf{x}^{(i)}, \hat{\theta}, M)$$

(Note that this is the conditional log-likelihood $p(y | \mathbf{x}, \hat{\theta}, M)$ and not joint log-likelihood $p(y, \mathbf{x} | \hat{\theta}, M)$). We now want to compare the performance of each model. We will write $L(M) \leq L(M')$ if model M must have lower or equal log likelihood on the training set than M' , for any training set (in other words, M is worse than M' , at least as far as training set logprob is concerned). For each of the following model pairs, state whether $L(M) \leq L(M')$, $L(M) \geq L(M')$, or whether no such statement can be made (i.e., M might sometimes be better than M' and sometimes worse); also, for each question, briefly (1-2 sentences) explain why.

- (a) GaussI, LinLog
- (b) GaussX, QuadLog
- (c) LinLog, QuadLog
- (d) GaussI, QuadLog
- (e) Now suppose we measure performance in terms of the average misclassification rate on the training set:

$$R(M) = \frac{1}{n} \sum_{i=1}^n I(y^{(i)} \neq \hat{\mathbf{y}}(\mathbf{x}^{(i)}))$$

where $\hat{\mathbf{y}}(\mathbf{x}^{(i)})$ is the predicted y for $\mathbf{x}^{(i)}$. Is it true in general that $L(M) > L(M')$ implies that $R(M) < R(M')$? Explain why or why not.

5 Softmax

Another possible approach to classification is to use a generalized version of the logistic model. Let $\mathbf{x} = [x_1, x_2, \dots, x_d]$ be an input vector, and suppose we would like to classify into k classes; that is, the output y can take a value in $1, \dots, k$. The softmax generalization of the logistic model uses $k(d+1)$ parameters $\theta = (\theta_{ij}), i = 1, \dots, k, j = 0, \dots, d$, which define the following k intermediate values:

$$\begin{aligned} z_1 &= \theta_{10} + \sum_j \theta_{1j} x_j \\ &\dots \\ z_i &= \theta_{i0} + \sum_j \theta_{ij} x_j \\ &\dots \\ z_k &= \theta_{k0} + \sum_j \theta_{kj} x_j \end{aligned}$$

The classification probabilities under the softmax model are:

$$\Pr(y = i \mid x; \theta) = \frac{e^{z_i}}{\sum_{j=1}^k e^{z_j}} .$$

1. Show that when $k = 2$ the softmax model reduces to the logistic model. That is, show how both give rise to the same classification probabilities $\Pr(y \mid x)$. Do this by constructing an explicit transformation between the parameters: for any given set of $2(d+1)$ softmax parameters, show an equivalent set of $(d+1)$ logistic parameters.
2. Which of the decision regions from question 2 can represent decision boundaries for a softmax model?
3. Show that the softmax model, for any k , can always be represented by a Gaussian mixture model. What type of Gaussian mixture models are equivalent to softmax models?
4. A stochastic gradient ascent learning rule for softmax is given by:

$$\theta_{ij} \leftarrow \theta_{ij} + \alpha \sum_t \frac{\partial}{\partial \theta_{ij}} \log \Pr(y^t \mid x^t; \theta) ,$$

where (x^t, y^t) are the training examples. We would like to rewrite this rule as a delta rule. In a delta rule the update is specified as a function of the difference between the target and the prediction. In our case, our target for each example will actually be a vector $y^t = (y_1^t, \dots, y_k^t)$ where $y_i^t = 1$ if $y^t = i$ and 0 otherwise.

Our prediction will be a corresponding vector of probabilities:

$$\hat{y}^t = (\Pr(y = 1 \mid x^t; \theta), \dots, \Pr(y = k \mid x^t; \theta))$$

Calculate the derivative above, and rewrite the update rule as a function of $y - \hat{y}$.

6 SVMs: basic intuition with one-dimensional data

Assume that our training data is four 1-dimensional points, as follows:

index	x	y
1	-2	-1
2	-0.1	-1
3	0.1	1
4	1	1

- (a) Find the values of all the α_i that would be found by the (linear) SVM training algorithm. You should be able to do this without going through the Lagrangian minimization procedure. Think about the conditions for the optimization directly.
- (b) What would the offset be for these values of α_i ?
- (c) What if the value of C were set to 1? What would happen to the values of α_i and the offset? Explain.

7 SVMs: Where's the hyperplane?

(Bishop 7.3) Show that, irrespective of the dimensionality of the data space, a data set consisting of just two data points, one from each class, is sufficient to determine the location of the maximum-margin hyperplane.

8 SVMs: primal scream

The primal *soft-margin* SVM optimization problem given by

$$\begin{aligned} & \text{minimize } \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i \quad \text{subject to} \\ & y^{(i)}(\mathbf{w} \cdot \mathbf{x}^{(i)} + w_0) \geq 1 - \xi_i, \quad i = 1, \dots, n \\ & \xi_i \geq 0, \quad i = 1, \dots, n \end{aligned}$$

involves both \mathbf{w}, w_0 and the slack variables ξ_i . We can rewrite the optimization problem in terms of \mathbf{w}, w_0 alone. This is done by explicitly solving for the optimal values of the slack variables $\xi_i = \xi_i(\mathbf{w}, w_0)$ as functions of \mathbf{w}, w_0 . The values of these slack variables, as functions of \mathbf{w}, w_0 , are “loss-functions” as shown below. What functions $\xi_i(\mathbf{w}, w_0)$ determine the optimal ξ_i ? Are all the margin constraints satisfied with these expressions for the slack variables?

The resulting minimization problem over \mathbf{w}, w_0 can be formally written as

$$\frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i(\mathbf{w}, w_0)$$

where the first (regularization) term biases our solution towards zero in the absence of any data and the remaining terms give rise to the loss functions, one loss function per training point, encouraging correct classification. Do we need any additional constraints? Many learning criteria can be understood and compared in the above regularization + loss form.

9 SVMs: modifying basic formulation

9.1 Lopsided

We are trying to solve a classification problem with support vector machines. In our problem there are only a few positive training examples and we are certain that they are classified correctly. We also have a large number of negative training examples, some of which may be misclassified. We'd like to modify the basic dual form of the SVM optimization problem,

$$(1) \text{maximize } \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y^{(i)} y^{(j)} \mathbf{x}^{(i)\top} \mathbf{x}^{(j)} \quad (2) \text{subject to } \alpha_i \geq 0, \quad \sum_{i=1}^n \alpha_i y^{(i)} = 0$$

to better solve this type of problem. We would like to ensure that we won't misclassify any of the positive examples but could misclassify some of the negative examples. We believe you have to introduce additional parameter(s) (or constants for the purpose of solving the quadratic programming problem) in order to achieve this. In your solution, please use I_+ to index positively labeled examples ($y_i = +1$) and I_- for negative examples ($y_i = -1$). In other words, $i \in I_+$ means that $y_i = +1$, and $|I_+|$ is the number of positive examples.

1. Optimization problem:
 maximize:
 subject to:
 Your solution must be in the dual form. You can refer to (1) and (2) above.
2. Which of the following alternative criteria would work for optimizing your new parameters. We have underlined any differences between the alternatives.
 - (a) We train your SVM algorithm $|I_+|$ times, each time leaving out one of the positive examples, and testing the classifier on the left out example. The parameter(s) are set to minimize the resulting number of misclassified examples.
 - (b) We train your SVM algorithm $|I_-|$ times, each time leaving out one of the negative examples, and testing the classifier on the left out example. The parameter(s) are set to minimize the resulting number of misclassified examples. Briefly explain why this would or would not work:
 - (c) We train your SVM algorithm n times, each time leaving out one of the examples, positive or negative, and testing the classifier on the left out example. The constant is set to minimize the resulting number of misclassified examples.

9.2 Data fusion

You are still trying to build a classifier with data that you gathered on two different days with two different instruments. You trust the labels of the data gathered with instrument 1 twice as much as the labels of the data gathered with instrument 2. You have lots of friends with different opinions about how to handle this.

We will use (x^i, y^i) , $i = 1 \dots n$ to denote data from instrument 1 (more accurate) and (u^i, v^i) to denote data from instrument 2. Slack variables for the instrument 1 data will be ξ and for the instrument 2 data will be ζ . Lagrange multipliers for the instrument 1 data will be α and for the instrument 2 data will be β .

- Pat suggests that you can insert a multiplier of 2 into the slack penalties for the data points gathered with instrument 1, so that the optimization problem is

$$\min_{\theta, \xi, \zeta} \frac{1}{2} \|\theta\|^2 + 2c \sum_{i=1}^n \xi_i + c \sum_{j=1}^m \zeta_j$$

subject to

$$\begin{aligned} y^i(\theta \cdot x^i + \theta_0) &\geq 1 - \xi_i && \text{for all } i \in \{1, \dots, n\} \\ v^j(\theta \cdot u^j + \theta_0) &\geq 1 - \zeta_j && \text{for all } j \in \{1, \dots, m\} \\ \xi_i &\geq 0 && \text{for all } i \in \{1, \dots, n\} \\ \zeta_j &\geq 0 && \text{for all } j \in \{1, \dots, m\} \end{aligned}$$

- Dana suggests that you can insert a multiplier of 2 into the Lagrange multipliers of data points gathered with instrument 1, so that the dual optimization problem is:

$$\max_{\alpha, \beta} \sum_{i=1}^n 2\alpha_i + \sum_{j=1}^m \beta_j - 2 \sum_{i=1}^n \sum_{j=1}^m \alpha_i \alpha_j y^i y^j (x^i \cdot x^j) - 2 \sum_{i=1}^n \sum_{j=1}^m \alpha_i \beta_j y^i v^j (x^i \cdot u^j) - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \beta_i \beta_j v^i v^j (u^i \cdot u^j)$$

subject to

$$\begin{aligned} c &\geq 2\alpha_i \geq 0 && \text{for all } i \in \{1, \dots, n\} \\ c &\geq \beta_j \geq 0 && \text{for all } j \in \{1, \dots, m\} \\ \sum_{i=1}^n 2\alpha_i y^i + \sum_{j=1}^m \beta_j v^j &= 0 \end{aligned}$$

- Robin suggests that you can duplicate the points that you gathered with instrument 1 in the data set, and then proceed as usual.
1. Are these approaches equivalent, in the sense of resulting in the same separator? For each pair, show that they are equivalent or not.

9.3 Objective fusion

We asked a few students to rate their midterm exam according to whether they thought it was difficult ($y = 1$), all right ($y = 2$), or easy ($y = 3$). Each student also provided us with a few pieces of information about themselves such as other courses they had taken, the program they were in, and so on. We could use this additional information to construct a feature vector ϕ_i for each student $i = 1, \dots, n$. On the basis of the rating labels, y_1, \dots, y_n and the feature vectors, ϕ_1, \dots, ϕ_n , we could learn to predict how a particular type of student would react to the exam.

We decided to divide the prediction task into two binary classification tasks

Task 1: whether $y = 1$ (binary label -1) or $y > 1$ (binary label +1)

Task 2: whether $y \leq 2$ (binary label -1) or $y = 3$ (binary label +1)

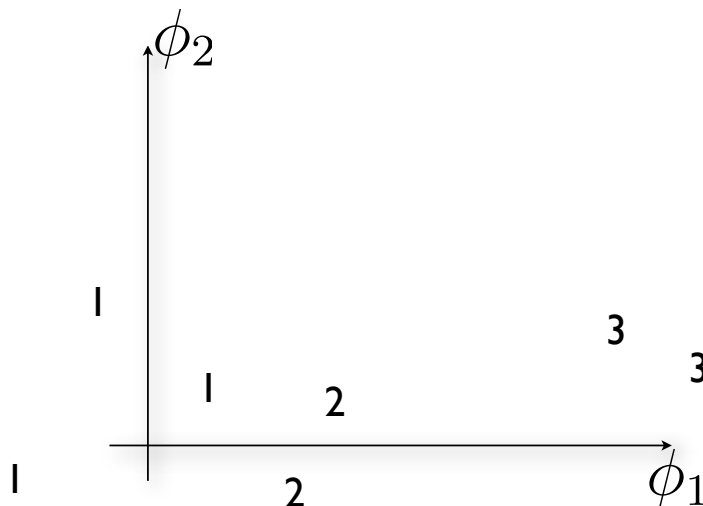
So, we needed two binary classifiers. Since the ratings fall on an ordinal scale, it seemed wise to couple these tasks together. We opted to use common parameters θ for the two tasks but different thresholds b_1 and b_2 for Task 1 and 2, respectively. The corresponding estimation problem is given by

Minimize $\frac{1}{2} \|\underline{\theta}\|^2$ with respect to $\underline{\theta}$, b_1 , and b_2 , subject to

Task 1: $-1(\underline{\theta} \cdot \phi_i - b_1) \geq 1$ if $y_i = 1$, $+1(\underline{\theta} \cdot \phi_i - b_1) \geq 1$ if $y_i > 1$

Task 2: $-1(\underline{\theta} \cdot \phi_i - b_2) \geq 1$ if $y_i \leq 2$, $+1(\underline{\theta} \cdot \phi_i - b_2) \geq 1$ if $y_i = 3$
for all $i = 1, \dots, n$

- Briefly explain why we would like to make sure that $b_1 \leq b_2$?
- If the problem is separable in the sense that the quadratic program has a solution, and all the rating labels occur at least once, are we guaranteed that the solution $\hat{\underline{\theta}}$, \hat{b}_1 , \hat{b}_2 satisfies $\hat{b}_1 \leq \hat{b}_2$?
- Suppose we omit Task 2 constraints altogether and only focus on Task 1 in order to solve for $\hat{\underline{\theta}}$ and \hat{b}_1 . Draw approximately the resulting decision boundary and margin constraints in Figure 1 based on the data in the figure.

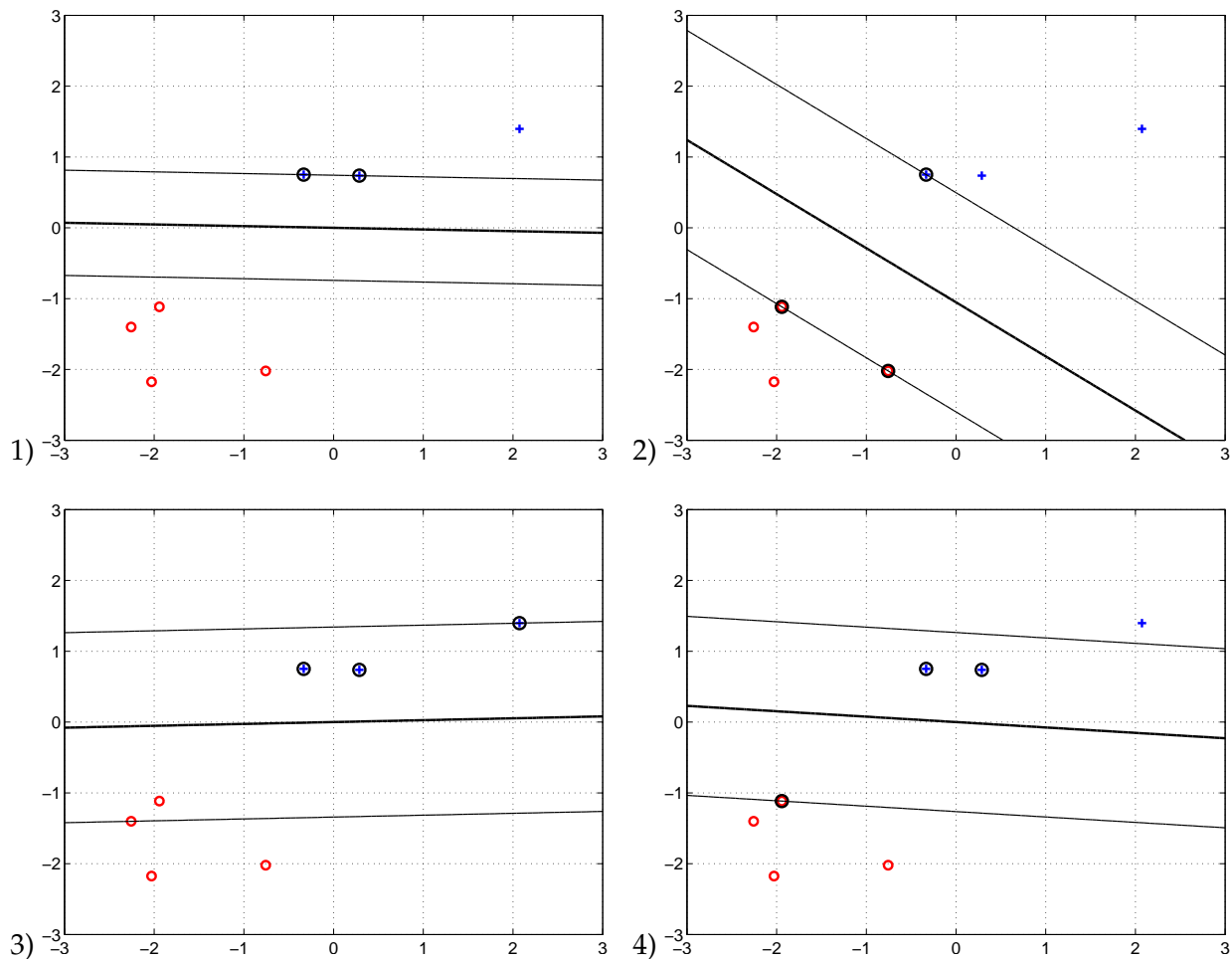


- How could you check if the solution $\hat{\underline{\theta}}$ based on Task 1 constraints alone also works as a solution to the combined task?

10 SVM-like training methods

As you may have suspected, the course staff enjoys writing endless varieties of SVM-like training methods. It is time to sort them out a bit. The figure below shows both decision boundaries and support vectors (circled) from different SVM-like training methods. In all cases, the boundaries correspond to $\hat{\underline{\theta}} \cdot \underline{x} + \hat{\theta}_0 = 0$, where $\hat{\theta}_0 = 0$ unless θ_0 is included in the training method. J_+ and J_- index positive ('x') and negative ('o') training examples, respectively. There are five methods and four figures. For each method, list all figures that show a possible solution for that method. Note that some methods may have more than one possible figure.

Here are plots of $\hat{\theta} \cdot \underline{x} + \hat{\theta}_0 = 0$ for different training methods along with the support vectors. Points labeled +1 are in blue, points labeled -1 are in red. The line $\hat{\theta} \cdot \underline{x} + \hat{\theta}_0 = 0$ is shown in bold; in addition we show the lines $\hat{\theta} \cdot \underline{x} + \hat{\theta}_0 = -1$ and $\hat{\theta} \cdot \underline{x} + \hat{\theta}_0 = 1$ in non-bold. Support vectors have bold circles surrounding them.



(a)

$$\min \frac{1}{2} \|\theta\|^2 + C \sum_{t=1}^n \xi_t \quad \text{s.t. } \xi_t \geq 0, \quad y_t(\theta^T \underline{x}_t + \theta_0) \geq 1 - \xi_t \quad t = 1, \dots, n$$

where $C = \infty$.

(b)

$$\min \frac{1}{2} \|\theta\|^2 + C \sum_{t=1}^n \xi_t \quad \text{s.t. } \xi_t \geq 0, \quad y_t(\theta^T \underline{x}_t) \geq 1 - \xi_t, \quad t = 1, \dots, n$$

where $C = \infty$.

(c)

$$\min \frac{1}{2} \|\theta\|^2 + C \sum_{t=1}^n \xi_t \quad \text{s.t. } \xi_t \geq 0, \quad y_t(\theta^T \underline{x}_t) \geq 1 - \xi_t, \quad t = 1, \dots, n$$

where $C = 1$.

(d)

$$\min \frac{1}{2} \|\underline{\theta}\|^2 + C_+ \sum_{t \in J_+} \xi_t + C_- \sum_{t \in J_-} \xi_t \quad \text{s.t. } \xi_t \geq 0, \quad y_t(\underline{\theta}^T \underline{x}_t) \geq 1 - \xi_t, \quad t = 1, \dots, n$$

where $C_+ = 1$ and $C_- = 0$.

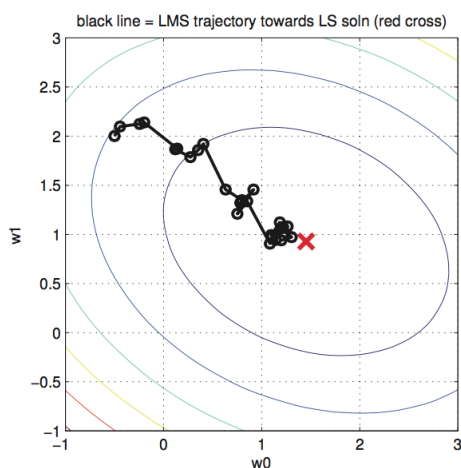
(e)

$$\min \frac{1}{2} \|\underline{\theta}\|^2 + C_+ \sum_{t \in J_+} \xi_t + C_- \sum_{t \in J_-} \xi_t \quad \text{s.t. } \xi_t \geq 0, \quad y_t(\underline{\theta}^T \underline{x}_t) \geq 1 - \xi_t, \quad t = 1, \dots, n$$

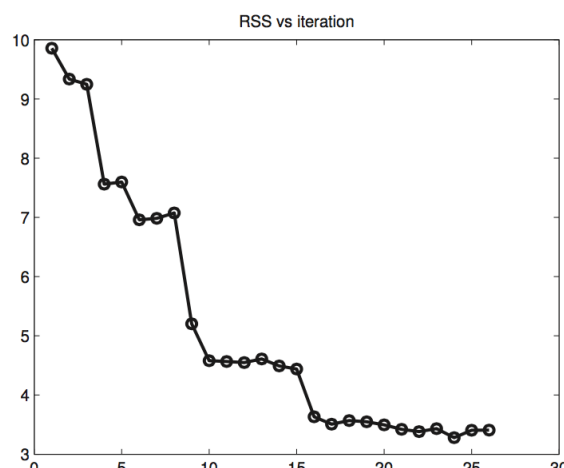
where $C_+ = \infty$ and $C_- = 0$.

11 Review: Regression and model selection

11.1 A bumpy hill



(a)



(b)

This is figure 8.8 from Murphy's textbook. It shows the progress of a gradient descent algorithm, minimizing squared error in a regression problem. Rather than doing the "batch" gradient descent method, where each step goes down the gradient of the squared error, this algorithm randomly selects a training example and takes a step down the gradient of squared error *for that example only*.

Which of the following are true? Why?

- Gradient descent steps always decrease the objective.
- Gradient descent steps with appropriately chosen step size always decrease the objective.
- Stochastic gradient descent steps always decrease the objective.
- Stochastic gradient descent steps with appropriately chosen step size always decrease the objective.

- There is a training method that at each step, randomly choose a element from the training set and take a gradient step based only on the error for that example. This method is an instance of stochastic gradient descent (using empirical error as metric).
- There are circumstances in which stochastic gradient descent is to be preferred to exact gradient descent.
- Training by randomly choosing an element of your training set, randomly perturbing the x values, and taking a gradient step based only on the error for that example is an instance of stochastic gradient descent with the same metric as exact gradient descent on the empirical error.
- Training by randomly choosing an element of your training set, randomly perturbing the x values, and taking a gradient step based only on the error for that example is an instance of stochastic gradient descent with the same metric as exact gradient descent on the ridge regression error.

11.2 Model selection

Dana wants to use ridge regression to fit a model to data, use a model-selection procedure to select an appropriate value of λ , and produce a good estimate of how the resulting model will perform on unseen data. There are three data sets available: D_{train} , D_{validate} , D_{test} , *all drawn from the same distribution*. Define

$$J(w, \lambda, D) = \sum_i (y^{(i)} - w^T x^{(i)})^2 + \lambda \|w\|_2^2$$

Write the following answers in terms of J and the data sets.

- Provide the definition of a function $W^*(\lambda)$ that specifies the optimal value of w for given a value of λ , according to the ridge regression criterion.
- Provide an expression for λ^* , which is the optimal value of λ . You may use the function W^* in your expression.
- Provide an expression for \hat{E} , the estimated error of the final resulting regression hypothesis on unseen data. You may use the function W^* and or the value λ^* in your definition.

11.3 Hip to be square

Consider a regression problem where the two dimensional input points $\mathbf{x} = [x_1, x_2]^T$ are constrained to lie within the unit square: $x_i \in [-1, 1]$, $i = 1, 2$. The training and test input points \mathbf{x} are sampled uniformly at random within the unit square. The target outputs y are governed by the following model

$$y \sim N(x_1^3 x_2^5 - 10x_1 x_2 + 7x_1^2 + 5x_2 - 3, 1)$$

In other words, the outputs are normally distributed with mean given by

$$x_1^3 x_2^5 - 10x_1 x_2 + 7x_1^2 + 5x_2 - 3$$

and variance 1.

We learn to predict y given x using linear regression models with 1st through 10th order polynomial features. The models are nested in the sense that the higher order models will include all the lower order features. The estimation criterion is the mean squared error.

We first train a 1st, 2nd, 8th, and 10th order model using $n = 20$ training points, and then test the predictions on a large number of independently sampled points.

1. Select all the appropriate model(s) for each column. If you think the highest, or lowest, error would be shared among several models, be sure to list all models.

	Lowest training error	Highest training error	Lowest test error (typically)
1st order	()	()	()
2nd order	()	()	()
8th order	()	()	()
10th order	()	()	()

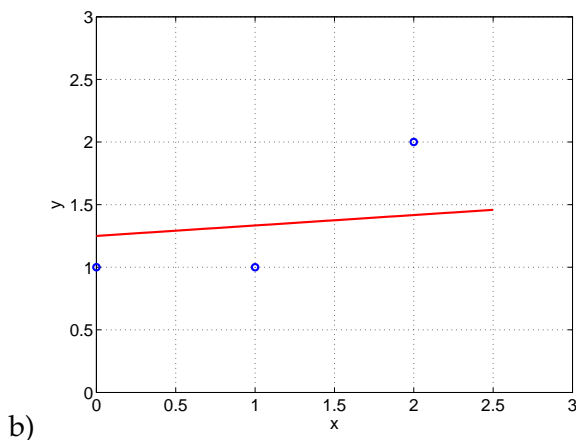
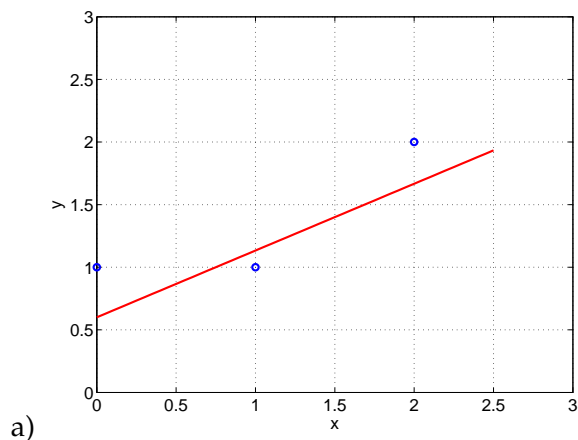
Briefly explain your selection in the last column, i.e., the model you would expect to have the lowest test error:

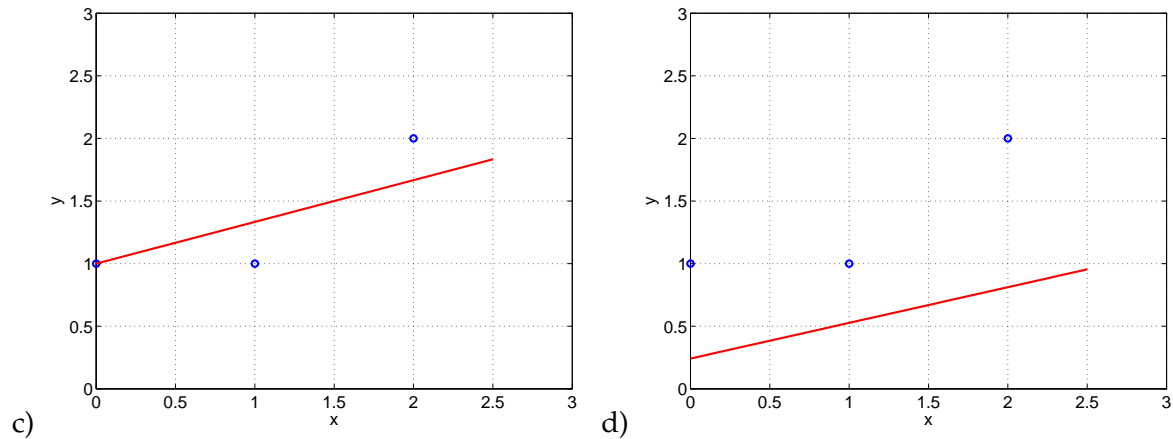
2. We now train the polynomial regression models using $n = 10^6$ (one million) training points. Again select the appropriate model(s) for each column. If you think the highest, or lowest, error would be shared among several models, be sure to list all models.

	Lowest bias	Lowest variance	Lowest test error
1st order	()	()	()
2nd order	()	()	()
8th order	()	()	()
10th order	()	()	()

3. True or False: The bias of a polynomial regression model depends on the number of training points.
4. True or false: The variance of a polynomial regression model depends on the number of training points.

11.4 Regularization in linear regression





The figure below plots linear regression results on the basis of only three data points. We used various types of regularization to obtain the plots (see below) but got confused about which plot corresponds to which regularization method. Please assign each plot to one (and only one) of the following regularization method.

1. $\sum_{t=1}^3 (y_t - \theta x_t - \theta_0)^2 + \lambda \theta^2$ where $\lambda = 1$
2. $\sum_{t=1}^3 (y_t - \theta x_t - \theta_0)^2 + \lambda \theta^2$ where $\lambda = 10$
3. $\sum_{t=1}^3 (y_t - \theta x_t - \theta_0)^2 + \lambda (\theta^2 + \theta_0^2)$ where $\lambda = 1$
4. $\sum_{t=1}^3 (y_t - \theta x_t - \theta_0)^2 + \lambda (\theta^2 + \theta_0^2)$ where $\lambda = 10$