Introduction to Machine Learning
Final Exam

Aug 9, 2018

Time limit:	120 minutes
Number of pages:	17
Total points:	100

You can use the back of the pages if you run out of space. Collaboration on the exam is strictly forbidden. Please show *all* of your work and always *justify* your answers. The questions are not necessarily ordered by difficulty; if you cannot solve a question, move on to the next one.

Please write your answers with a *pen*.

(1 point) Please fill in your student ID and full name (LASTNAME, FIRSTNAME) in capital letters.

Please leave the table below empty.

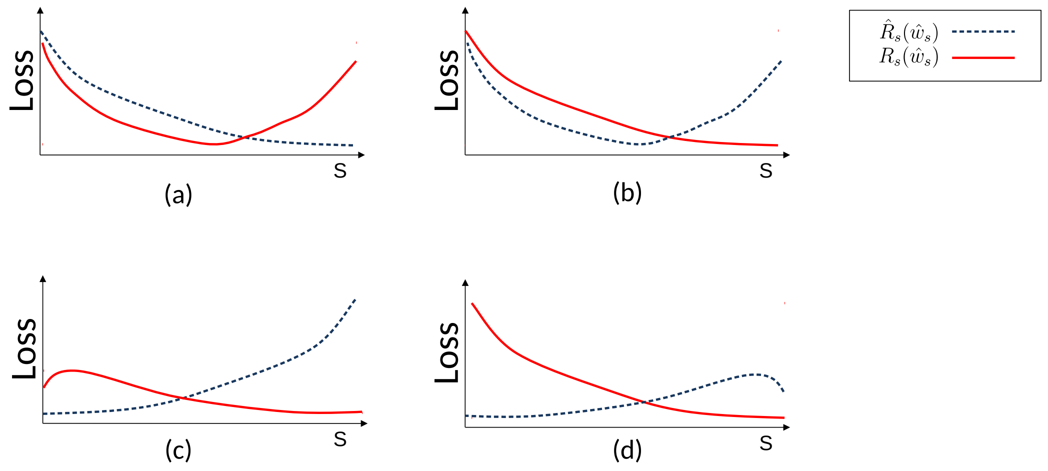
Problem	Maximum points	Obtained
1.	19	
2.	20	
3.	12	
4.	14	
5.	14	
6.	20	
Total	100	

1. Short Questions (19 points)

In each multiple choice question, you get 1 point for a correct answer, 0 points for a blank answer, and -1 point for a wrong answer. There is only **one correct answer for each question**. You cannot get less than 0 points in each part. *You don't get negative points for wrong answers in part ii).*

(3 points) (i) *Short questions on regression.* Suppose you are given a 1-dimensional regression problem on a data set $\mathcal{D} = \{(x_1, y_1), \dots, (x_n, y_n)\}$ with n *i.i.d.* samples from some distribution $P(X, Y)$ on $\mathbb{R} \times \mathbb{R}$. Our goal is to fit a polynomial $p(x; \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \cdots + w_s x^s$ of degree s by minimizing the empirical squared loss $\hat{R}_s(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} (p(x_i; \mathbf{w}) - y_i)^2$. Let $\hat{\mathbf{w}}_s$ be the empirical risk minimizer $\hat{\mathbf{w}}_s = \operatorname{argmin}_{\mathbf{w}} \hat{R}_s(\mathbf{w})$ and denote by $R_s(\mathbf{w})$ the true risk (expected loss).

1. Choose the graph that best describes the qualitative behaviour of the empirical loss $\hat{R}_s(\hat{\mathbf{w}}_s)$ (in blue) and the expected loss $R_s(\hat{\mathbf{w}}_s)$ (in red) as a function of s.



a) ☐ b) ☐ c) ☐ d) ☐

2. To find the empirical risk minimizer $\hat{\mathbf{w}}_s = \operatorname{argmin}_{\mathbf{w}} \hat{R}_s(\mathbf{w})$ one can use gradient descent on $\mathbf{w}$. Which of the following statements is correct?
   ☐ For any s, n, sufficiently small learning rate $\eta > 0$ and any starting point $\mathbf{w}^{(0)}$, gradient descent will always converge (asymptotically) to a minimizer of $\hat{R}_s(\cdot)$.
   ☐ For any s, n, sufficiently small learning rate $\eta > 0$ and any starting point $\mathbf{w}^{(0)}$, gradient descent will always converge (asymptotically) to the same solution.
   ☐ The computational complexity of each step in the gradient descent algorithm is independent of the number of training points n.
   ☐ The loss $R_s(\mathbf{w})$ is non-convex in $\mathbf{w}$, so we need multiple restarts to find a good minimum.

3. In order to use cross-validation on s, we split the data set into a training and a validation set. As we train polynomials of higher degree and begin to overfit, we notice that
   ☐ the training error increases and the validation error stays the same.
   ☐ the training error decreases and the validation error increases.
   ☐ the training error increases and the validation error increases.
   ☐ None of the above.

(6 points) (ii) *Short questions on classification.* Consider the classification problem with two classes which is illustrated by circles and crosses in the plots below. In each of the plots, one of the following classification methods has been used, and the resulting decision boundary is shown:

A) Linear SVM

B) Perceptron

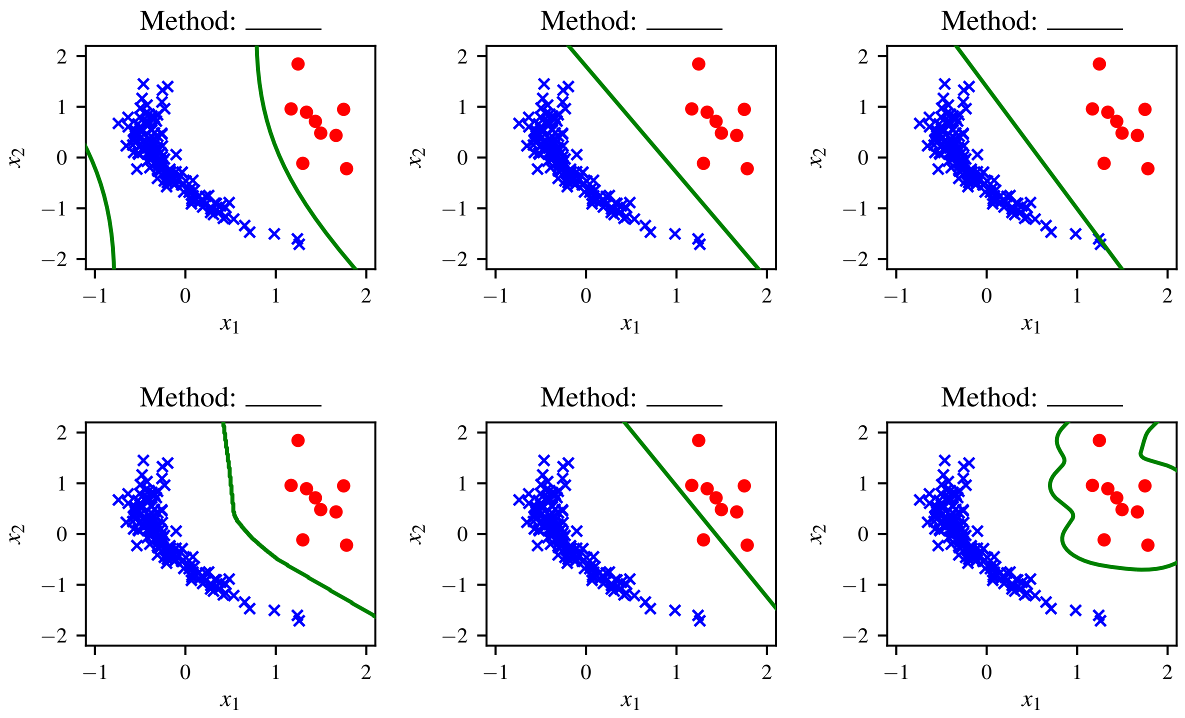
C) Logistic Regression

D) Kernelized SVM (Polynomial kernel of order 2)

E) Kernelized SVM (Gaussian kernel)

F) Neural Network (1 hidden layer with 20 rectified linear units)

Assign each of the methods to exactly one of the following plots (in a one to one correspondence) by annotating the plots with the respective letters.



(4 points) (iii) *Short questions on dimensionality reduction.* Assume we apply PCA with k principal components to a data set $D = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ where $\mathbf{x}_i \in \mathbb{R}^d$.

1. If $k < d$ we can *exactly* reconstruct x_i from k principal components.

   ☐ True ☐ False

2. If $k = d$ we can *exactly* reconstruct x_i from k principal components.

   ☐ True ☐ False

3. If $k > n$ we can *exactly* reconstruct x_i from k principal components.

   ☐ True ☐ False

4. Suppose $\mathbf{X}$ is the $n \times d$ data matrix. Then, the eigenvectors of $\mathbf{X}\mathbf{X}^T$ and $\mathbf{X}^T\mathbf{X}$ are the same.

   ☐ True ☐ False

(2 points) (iv) *Short questions on kernels.*

1. Suppose $k_1 : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and $k_2 : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ are two valid kernels on a data space $\mathcal{X}$. Which of the following is **not** a valid kernel?
   ☐ $k(x, z) := k_1(x, z) + k_2(x, z)$
   ☐ $k(x, z) := ck_1(x, z)$ for any $c \in \mathbb{R}$
   ☐ $k(x, z) := k_1(x, z) + k_2(x, z) + 3 \cdot k_1(x, z) \cdot k_2(x, z)$
   ☐ $k(x, z) := g(x)g(z)$ for any function $g : \mathcal{X} \to \mathbb{R}$

2. Suppose you have trained a kernelized SVM using a Laplace kernel $k(x, y) = \exp\left(-\frac{\|x-y\|}{h}\right)$ with bandwidth h and regularization parameter λ. To reduce overfitting, you should
   ☐ increase λ and reduce h.
   ☐ reduce λ and increase h.
   ☐ reduce λ and reduce h.
   ☐ increase λ and increase h.

(2 points) (v) *Short questions on neural networks.*

1. When training a neural network with *one hidden layer* with stochastic gradient descent,
   ☐ the test error always decreases monotonically.
   ☐ for computing the gradients, a single backwards pass through the network suffices.
   ☐ we do not necessarily find the global optimum, even if we use the square loss.
   ☐ the momentum method incorporates the direction of the second derivative in the update step.

2. Which of the following statements about regularization in neural networks is **false**?
   ☐ Dropout is used for regularization.
   ☐ When using dropout to train a neural network, we need to modify the training phase only. The prediction phase remains the same.
   ☐ Early stopping means monitoring the validation loss and stop training when the loss increases.
   ☐ Adding L_2 regularization on the weights encourages small weights.

(2 points) (vi) *Short questions on generative modeling.*

1. In the Naive Bayes model,
   ☐ the features are conditionally independent given the labels.
   ☐ the labels are conditionally independent given the features.
   ☐ the features are independent.
   ☐ the features follow a Gaussian distribution.

2. You have trained a generative model on binary labels and you want to predict the label for a new data point x. According to your model:

$$p(y = 0) = 0.5, \qquad p(y = 1) = 0.5, \qquad p(x|y = 0) = 0.02, \qquad p(x|y = 1) = 0.03$$

   What is the probability of x belonging to class 0?
   ☐ $p(y = 0|x) = 0.01$
   ☐ $p(y = 0|x) = 0.2$
   ☐ $p(y = 0|x) = 0.4$
   ☐ $p(y = 0|x)$ cannot be calculated without knowing $p(x)$.

2. Huber-Loss Regression (20 points)

You are given a dataset consisting of n labeled training points $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$. A common approach towards linear regression is to find a solution $\mathbf{w} \in \mathbb{R}^d$ which minimizes the *empirical risk* $\hat{R}(\mathbf{w})$,

$$\hat{R}(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} l_\delta(\mathbf{w}^\top \mathbf{x}_i - y_i) \,. \tag{1}$$

Here, l_δ denotes the Huber loss function with parameter $\delta \geq 0$,

$$l_\delta(a) = \begin{cases} \frac{1}{2}a^2 & \text{for } |a| \leq \delta, \\ \delta(|a| - \frac{1}{2}\delta), & \text{otherwise.} \end{cases}$$

(2 points) (i) Qualitatively draw the loss $l_\delta(a)$ for $\delta = 1$. What loss function do you recover for $\delta \to \infty$?

(2 points) (ii) The empirical risk minimizer $\hat{\mathbf{w}}$ is defined as $\hat{\mathbf{w}} = \operatorname{argmin}_{\mathbf{w}} \hat{R}(\mathbf{w})$. Is the solution always unique? Justify your answer.

(3 points) (iii) If the dataset contains outliers, would you rather use the standard squared loss or the Huber loss? Explain your choice. *Hint: Consider the qualitative behaviour of the empirical risk function if one label y_i is perturbed by a large amount.*

(5 points) (iv) In cases where the response y_i cannot be modeled as a linear function of the features $\mathbf{x}_i$, we can apply a kernel method instead. Use the kernel trick to show that the kernelized linear Huber estimator is given by $\hat{\boldsymbol{\alpha}} = \operatorname{argmin}_{\boldsymbol{\alpha} \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^n l_\delta(\boldsymbol{\alpha}^T \mathbf{k}_i - y_i)$ where $\mathbf{k}_i = \big(k(\mathbf{x}_i, \mathbf{x}_1,), \dots, k(\mathbf{x}_i, \mathbf{x}_n)\big)^T$. *Hint: You can assume that the solution to* (1) *can be written in the form* $\hat{\mathbf{w}} = \sum_{i=1}^n \alpha_i x_i$ *for* $\boldsymbol{\alpha} \in \mathbb{R}^n$.

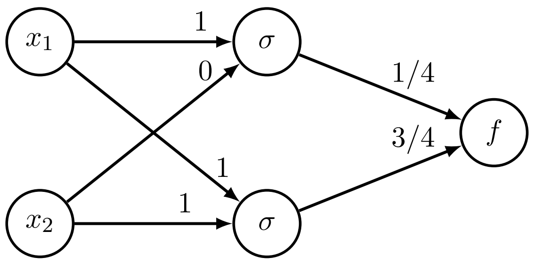
(5 points) (v) One way to find the empirical risk minimizer $\hat{\boldsymbol{\alpha}} = \operatorname{argmin}_{\boldsymbol{\alpha} \in \mathbb{R}^n} \frac{1}{n} \sum l_\delta(\boldsymbol{\alpha}^T \mathbf{k}_i - y_i)$ of the kernelized risk function is to use gradient descent on $\boldsymbol{\alpha}$. Write down the gradient $\nabla_{\boldsymbol{\alpha}} l_\delta(\boldsymbol{\alpha}^T \mathbf{k}_i - y_i)$, and use it to define the update step of gradient descent with a learning rate η.

After your graduation, you are hired by the company *Golden Cookies* as a data scientist to predict the tastiness of new cookie recipes. To this end, you have carefully carried out a study where you asked $n = 200$ volunteers to rate the tastiness of $m = 10$ different cookie recipes. Specifically, you have a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1,\dots,n}$ consisting of 200 data points, where $\mathbf{x}_i$ encodes the ingredients of a recipe and y_i is the corresponding rating of the ith participant of your study. Each participants rated exactly one of the recipes, such that you have 20 answers for each of the 10 recipes tested. Your goal is to predict the *tastiness score* y_{new} for a new recipe $\mathbf{x}_{\text{new}}$.

(3 points) (vi) For this task, a colleague suggests to optimize the parameter δ of the Huber loss using k-fold cross-validation by randomly assigning the data-points to one of $k = 10$ folds. Why is this approach problematic here? Propose a better way to use cross-validation in this setting.

3. Artifical Neural Networks (12 points)

Consider the neural network given in the figure below. The numbers above the lines correspond to the weights of the connections. In the hidden layer, the activation function σ is applied and the network does not have any biases.



(2 points) (i) Read off the initial weights $\mathbf{W}_1$ and $\mathbf{w}_2$ from the network given above, such that the weights correspond to the following matrix notation of the neural network with input $\mathbf{x} = (x_1, x_2)$.

$$f(\mathbf{x}; \mathbf{W}_1, \mathbf{w}_2) = \mathbf{w}_2^\top \sigma(\mathbf{W}_1 \mathbf{x})$$

(2 points) (ii) You are given a data set $\mathcal{D} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ with n data points, where $\mathbf{x}_i \in \mathbb{R}^2$ and $y_i \in \mathbb{R}$ for $i = 1, \dots, n$. Write down the empirical risk $\hat{R}(\mathcal{D}, \mathbf{W}_1, \mathbf{w}_2)$ for training the neural network using the squared loss with added L_2 regularization *only* on the output layer.

(2 points) (iii) For training the neural network, the empirical risk function is often minimized using *stochastic* gradient descent (SGD). What is the difference between SGD and standard gradient descent in terms of computational complexity?

(2 points) (iv) Given $\mathbf{W}_1$ and $\mathbf{w}_2$ as in the diagram on the previous page, find different weights $\tilde{\mathbf{W}}_1$ and $\tilde{\mathbf{w}}_2$, such that the resulting network has the same performance as the original network *(for any activation function σ)*. In other words, find $\tilde{\mathbf{W}}_1$ and $\tilde{\mathbf{w}}_2$ s.t. $f(\mathbf{x}; \mathbf{W}_1, \mathbf{w}_2) = f(\mathbf{x}; \tilde{\mathbf{W}}_1, \tilde{\mathbf{w}}_2)$, but $\mathbf{W}_1 \neq \tilde{\mathbf{W}}_1$ and $\mathbf{w}_2 \neq \tilde{\mathbf{w}}_2$.

(4 points) (v) Show that if σ is the *relu* activation function $\sigma(x) = \max(0, x)$, then the resulting network $f(\mathbf{x}; \mathbf{W}_1, \mathbf{w}_2)$ is a piecewise linear function in $\mathbf{x}$. Specifically, define 4 intervals on the real line where the resulting function is linear, for the weights given in the diagram on the previous page.

4. Evaluation of binary classifiers (14 points)

In collaboration with a team of astrolinguists, you are building a tool to identify whether a message from an alien species is friendly or hostile. Through careful interaction you have collected a set of 300 examples of both friendly (positive) and hostile (negative) messages, and built a classifier. On a test set, you evaluate the quality of the classifications, and record the results in the following contingency table:

	Positive message	Negative message
Classified positive	216	3
Classified negative	54	27

Calculate the following quantities for the classifier *(either as reduced fraction or decimal numbers)*:

(1 point) (i) Accuracy

(1 point) (ii) Sensitivity (also known as recall or true positive rate)

(1 point) (iii) False positive rate

You receive a new transmission from the aliens, and your classifier identifies it as hostile (negative)! The president points at the accuracy of your classifier and says the risk is too high - we must disengage with the aliens.

(1 point) (iv) Given the above contingency table, what is the *base rate*, or prior probability of hostile messages?

(2 points) (v) What is the probability that the new message is hostile?

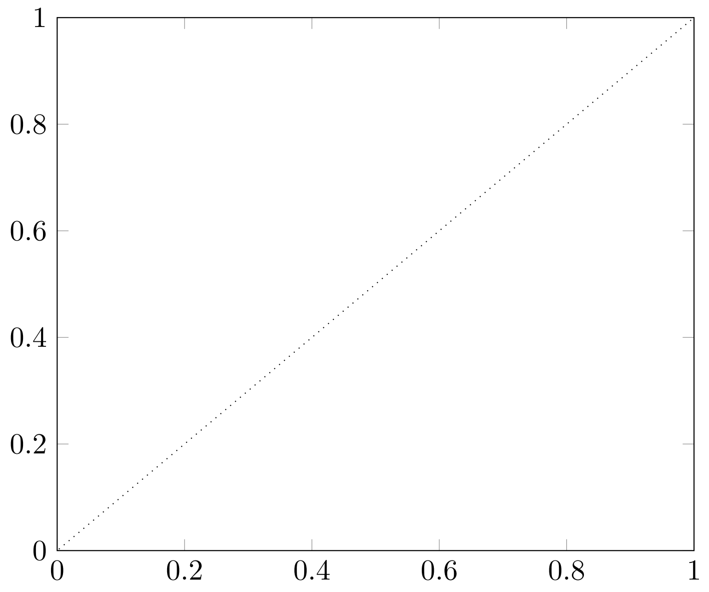
(4 points) (vi) A competing team of reseachers has built a similar message classifier. On the same test set, they report that their accuracy is 0.9 and their recall is 1. They say that their model should be used to screen all future alien communications. Describe a model which would produce this performance on the given dataset. Should such a model be used? Why or why not?

After your system has been deployed, a politician argues that the risk of classifying hostile (negative) messages as friendly (positive) is too high. As a remedy, you propose to use a cost-sensitive classifier instead. Specifically, you use to following modification of the perceptron loss:

$$l(\mathbf{w}; \mathbf{x}, y) = \begin{cases} \max(0, -y\mathbf{w}^T\mathbf{x}) & \text{if } y = 1 \\ c \cdot \max(0, -y\mathbf{w}^T\mathbf{x}) & \text{if } y = -1 \end{cases}$$

where $\mathbf{x}$ and y corresponds to the message and the label, $\mathbf{w}$ represents the classifier's weights, and $c \in [0, \infty)$ trades off the cost of miss-classifying a negative message. You decide to evaluate the effect of the parameter c on the classifier using the receiver operating characteristic (ROC) curve. Assume that the confusion matrix given at the beginning of this task corresponds to $c = 1$.

(4 points) (vii) First, label the axes in the empty plot below; then (qualitatively) draw the resulting ROC curve from varying c. Clearly mark the point $c = 1$ (using the data given by the confusion matrix), and the parts of the curve which correspond to $c < 1$ and $c > 1$.



5. Elastic Net (14 points)

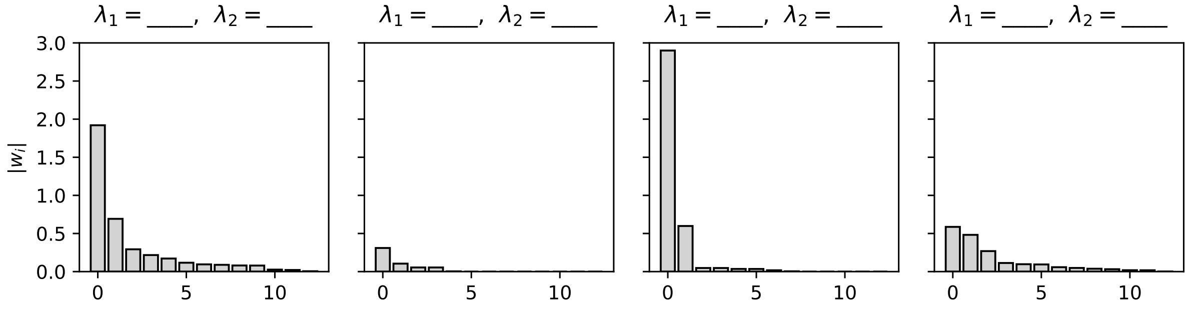
In class you have seen how to use linear regression with regularization. In this task we will use the following regularized risk function, also known as *elastic net*:

$$R(w) = \sum_{i=1}^{n} (\mathbf{w}^T \mathbf{x}_i - y_i)^2 + \lambda_1 \|\mathbf{w}\|_1 + \lambda_2 \|\mathbf{w}\|_2^2$$

The risk function has two regularization parameters $\lambda_1, \lambda_2 \geq 0$. As usual, you are given *i.i.d.* data $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$ for $i = 1, \ldots, n$.

(2 points) (i) For what values of λ_1 and λ_2 do you recover Ridge regression and Lasso regression respectively?

(3 points) (ii) The plots below show the learned weights $w_1, \ldots, w_d$ sorted by their absolute value. The plots were created using the following values for (λ_1, λ_2): $(1, 0), (0, 1), (10, 10), (0, 10)$. Clearly indicate what plot belongs to which model.



(2 points) (iii) Define the *maximum a posteriori* (MAP) estimate for $\mathbf{w}$ in terms of the likelihood $P(y_i|\mathbf{x}_i, \mathbf{w}, \sigma^2)$, a given prior $P(\mathbf{w})$ and the data $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$. Treat σ as fixed parameter.

Assuming that the errors are normally distributed, the likelihood is given by

$$P(y_i|\mathbf{x}_i, \mathbf{w}, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y_i - \mathbf{w}^T\mathbf{x}_i)^2}{2\sigma^2}\right) .$$

As a prior you are given

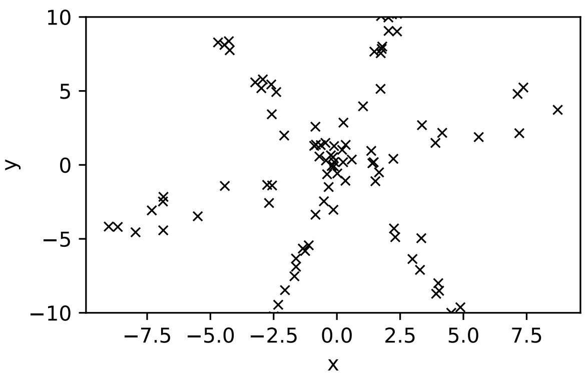
$$P(\mathbf{w}) = \frac{1}{Z} \exp(-\frac{1}{\beta_1}\|\mathbf{w}\|_1 - \frac{1}{2\beta_2^2}\|\mathbf{w}\|_2^2) ,$$

where Z is a normalizing constant.

(7 points) (iv) Show that the MAP estimate of $\mathbf{w}$ using the prior given above, corresponds to the empirical risk minimizer of the *elastic net*, assuming the σ is a known constant. Express the constants β_1 and β_2 in terms of λ_1 and λ_2.

6. Mixtures of Lines (20 points)

For this task you are given a data set $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$ for $i = 1, \ldots, n$. After inspection, you observe that the data points accumulate on m different lines, given by equations of the form $\mathbf{a}_j^T \mathbf{x}_i = y_i$ for $\mathbf{a}_j \in \mathbb{R}^d$ and $j = 1, \ldots, m$. An illustration of such a dataset is given in the plot below for $d = 1$ and $m = 3$.



Our goal is to fit a mixture model with m components using the EM procedure. Using a Gaussian likelihood for each component, the probability of observing label y given $\mathbf{x}$ is given by

$$P(y|\mathbf{x}, \pi_1, \ldots, \pi_m, \mathbf{a}_1, \ldots, \mathbf{a}_m) = \sum_{j=1}^{m} \pi_j \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\mathbf{a}_j^T \mathbf{x} - y)^2}{2\sigma^2}\right) .$$

The mixture weights $\pi_{1:m} = (\pi_1, \ldots, \pi_m)$ are a convex combination, i.e. $\pi_j \geq 0$ and $\sum_{j=1}^{m} \pi_j = 1$. For simplicity, assume that the scale parameter $\sigma > 0$ is known beforehand and *fixed*.

The EM algorithm iteratively updates estimates $\pi_j^{(t)}$ and $\mathbf{a}_j^{(t)}$ for each component $j = 1, \ldots, m$ in rounds $t = 1, 2, 3, \ldots$, starting with initial values $\pi_j^{(0)}$ and $\mathbf{a}_j^{(0)}$. We introduce latent variables $Z_i \in \{1, 2, \ldots, m\}$ denoting the component that $(\mathbf{x}_i, y_i)$ belongs to.

(3 points) (i) In the E-step of Soft EM, we estimate responsibilities of the different components for the data points. We define them to be $\gamma_j^{(t)}(\mathbf{x}_i, y_i) = P(Z_i = j|\mathbf{x}_i, y_i, \pi_{1:m}^{(t-1)}, \mathbf{a}_{1:m}^{(t-1)})$. Write down the explicit update for $\gamma_j^{(t)}(\mathbf{x}_i, y_i)$ given estimated mixture weights $\pi_{1:m}^{(t-1)}$ and parameters $\mathbf{a}_{1:m}^{(t-1)}$.

(1 point) (ii) Write down the class predictions $z_i^{(t)}$ for $(\mathbf{x}_i, z_i)$ in the E-step of Hard EM in terms of $\gamma_j^{(t)}(\mathbf{x}_i, y_i)$.

(2 points) (iii) Assume that we observe the *true* labels $z_1, \dots, z_l$ for the first few data points $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_l, y_l)$ for $l < n$. How can we modify the E-step of Soft EM to incorporate the additional information?

(2 points) (iv) Write down the optimization objective for the M-step in Soft EM for the model parameters $\pi_j^{(t)}$ and $\mathbf{a}_j^{(t)}$ in terms of the responsibilities $\gamma_j^{(t)}(\mathbf{x}_i, y_i)$.

(12 points) (v) Derive the explicit update rules for $\pi_j^{(t)}$ and $\mathbf{a}_j^{(t)}$ used in the M-step of Soft EM.

Hint: You can assume that the matrix $\sum_{i=1}^{n} \gamma_j^{(t)}(\mathbf{x}_i, y_i)\mathbf{x}_i\mathbf{x}_i^T$ *is invertible for all* $j = 1, \ldots, m$.