Projects
Machine Learning Course
Fall 2018

EPFL
School of Computer and Communication Sciences
Martin Jaggi & Rüdiger Urbanke
mlo.epfl.ch/page-157255-en-html/
epfmlcourse@gmail.com

Mock Midterm Exam - Nov 19, 2018

1 Subgradient Descent

Derive the (sub)gradient descent update rule for a one-parameter linear model using the Mean Absolute Error,

$$\mathcal{L}_{\mathsf{MAE}}(\mathbf{X}, \mathbf{y}, w) = \frac{1}{N}\sum_{n=1}^{N} |wx_n - y_n|.$$

Hint: The function $f(x) = |ax|$ is a composition of two simpler function. Use the chain rule!

Solution: Our cost function is $\mathcal{L}(\mathbf{X}, \mathbf{y}, w) = \frac{1}{N}\sum \mathcal{L}_n(x_n, y_n, w)$ where $\mathcal{L}_n(x_n, y_n, w) = |wx_n - y|$. We have that

$$\frac{\partial \mathcal{L}(\mathbf{X}, \mathbf{y}, w)}{\partial w} = \frac{1}{N}\sum_{n=1}^{N} \frac{\partial \mathcal{L}_n(x_n, y_n, w)}{\partial w}.$$

Let a, e be two functions such that $a(x) = |x|$ and $e(x, y, w) = wx - y$. We can rewrite $\mathcal{L}_n$ as $a \circ e$. Let us find the derivative of $\mathcal{L}_n$ using the chain rule.

$$\begin{aligned}\frac{\partial \mathcal{L}_n(x_n, y_n, w)}{\partial w} &= \frac{\partial a(e(x_n, y_n, w))}{\partial w} \\ &= \frac{\partial a(e(x_n, y_n, w))}{\partial e}\frac{\partial e(x_n, y_n, w)}{\partial w}\end{aligned}$$

We have that $\frac{\partial e}{\partial w}(x_n, y_n, w) = x_n$, but $|x|$ is not differentiable at $x = 0$. We will have to use subgradients.

Remember that a vector $\mathbf{g}$ is a subgradient of the function f in $\mathbf{x}$ if $f(\mathbf{y}) \geq f(\mathbf{x}) + \mathbf{g}^\top(\mathbf{y} - \mathbf{x})$, for all $\mathbf{y}$. Note that in our one dimensional case, for a subgradient of $|x|$ in $x = 0$, we need to find g such that

$$|y| \geq gy, \ \forall y.$$

Any $g : |g| \leq 1$ will do, but since we want the error to go to 0 and stay here, we will use $g = 0$. We therefore have that

$$\frac{\partial a(x)}{\partial x} = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases} \text{ and } \frac{\partial e(x, y, w)}{\partial w} = x$$

Which gives us the following expression for the gradient

$$\frac{\partial \mathcal{L}(x, y, w)}{\partial w} = \frac{1}{N}\sum_{n=1}^{N} \begin{cases} x_n & \text{if } wx_n - y_n > 0 \\ 0 & \text{if } wx_n - y_n = 0 \\ -x_n & \text{if } wx_n - y_n < 0 \end{cases}$$

Therefore, one step of gradient descent with step size γ is given by $w^{(i+1)} = w^{(i)} - \frac{\gamma}{N}\begin{cases} x & \text{if } wx - y > 0 \\ 0 & \text{if } wx - y = 0 \\ -x & \text{if } wx - y < 0 \end{cases}$

2 Multiple-Output Regression

Let $S = \{(\mathbf{y}_n, \mathbf{x}_n)\}_{n=1}^N$ be our training set for a regression problem with $\mathbf{x}_n \in \mathbb{R}^D$ as usual. But now $\mathbf{y}_n \in \mathbb{R}^K$, i.e., we have K outputs for each input. We want to fit a linear model for each of the K outputs, i.e., we now have K regressors $f_k(\cdot)$ of the form

$$f_k(\mathbf{x}) = \mathbf{x}^\top \mathbf{w}_k,$$

where each $\mathbf{w}_k^\top = (w_{k1}, \cdots, w_{kD})$ is the weight vector corresponding to the k-th regressor. Let $\mathbf{W}$ be the $D \times K$ matrix whose columns are the vectors $\mathbf{w}_k$.

Our goal is to minimize the following cost function $\mathcal{L}$:

$$\mathcal{L}(\mathbf{W}) = \sum_{k=1}^K \sum_{n=1}^N \frac{1}{2\sigma_k^2}(y_{nk} - \mathbf{x}_n^\top \mathbf{w}_k)^2 + \frac{1}{2}\sum_{k=1}^K \|\mathbf{w}_k\|_2^2,$$

where the σ_k are known real-valued scalars. Let $\boldsymbol{\sigma} = (\sigma_1, \cdots, \sigma_K)$.

For the solution, let $\mathbf{X}$ be the $N \times D$ matrix whose rows are the feature vectors $\mathbf{x}_n$.

1. Write down the normal equations for $\mathbf{W}^\star$, the minimizer of the cost function. I.e., what is the first-order condition that $\mathbf{W}^\star$ has to fulfill in order to minimize $\mathcal{L}(\mathbf{W})$.

   Solution: Note that the cost function $\mathcal{L}(\mathbf{W})$ is the sum of K cost functions, $\mathcal{L}(\mathbf{w}_k)$, each of which only depends on its own parameter $\mathbf{w}_k$. So if we compute the gradient with respect to $\mathbf{w}_k$ then this only involves the term $\mathcal{L}(\mathbf{w}_k)$ and we get

   $$\frac{1}{\sigma_k^2}\mathbf{X}^\top(\mathbf{X}\mathbf{w}_k - \mathbf{y}_k) + \mathbf{w}_k = 0.$$

   This is essentially the expression we had for ridge regression.

2. Is the minimum $\mathbf{W}^\star$ unique? Assuming it is, write down an expression for this unique solution.

   Solution: We show that the problem is strictly convex and therefore has a unique minimizer. This follows from the fact that the first double sum term (the cost function) is convex in $\mathbf{W}$, and that the second term, the squared norm regularizer, is strictly convex in $\mathbf{W}$. Therefore the sum of both is strictly convex. We have the solution

   $$\mathbf{w}_k^\star = \left(\frac{1}{\sigma_k^2}\mathbf{X}^\top\mathbf{X} + \mathbf{I}_D\right)^{-1} \frac{1}{\sigma_k^2}\mathbf{X}^\top \mathbf{y}_k.$$

3. Write down a probabilistic model, so that the MAP solution for this model coincides with minimizing the above cost function. Note that this will involve specifying the the likelihoods as well as a suitable prior (which will give you the regression term).

   Solution: You are asked to derive a probabilistic model under which is the maximum a posteriori estimate. However, since "Posterior probability ∝ Likelihood × Prior probability", the question asks specifically for the prior and the likelihood only. Knowing that the maximization over a Gaussian is equivalent to minimizing the mean square error, one can check that $\mathbf{w}^\star_{\text{MAP}} = \arg\max_{\mathbf{w}} p(\mathbf{y} \mid \mathbf{X}, \mathbf{w})p(\mathbf{w})$ is equivalent to the above cost minimization $\mathbf{w}^\star_{normal} = \arg\min_{\mathbf{w}} \mathcal{L}(\mathbf{w})$ if:

   $$p(\mathbf{y} \mid \mathbf{X}, \mathbf{w}) = \prod_{n=1}^N \prod_{k=1}^K \mathcal{N}(y_{nk} \mid \mathbf{w}_k^\top \mathbf{x}_n, \sigma_k^2)$$

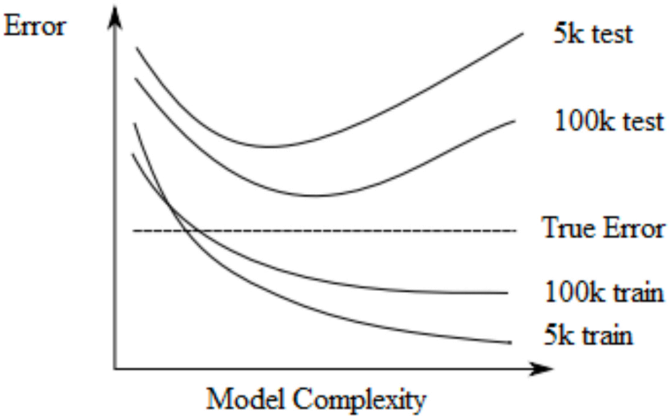
   and

   $$p(\mathbf{w}) = \prod_{k=1}^K \mathcal{N}(\mathbf{w}_k \mid \mathbf{0}, \mathbf{I}_D)$$

3 Bias Variance Trade-off (Due to Alex Smola)

Assume that you have two data sets that contain iid samples from the same distribution, call them S_1 and S_2. S_1 contains 5000 samples, whereas S_2 contains 100000 samples. You randomly split each of the data sets into a training and a testing set, where eighty percent of the data is assigned to the training set. You then train and test on a family of increasing complexity.

In the figure below draw four curves, two that show the *training error* as a function of the model complexity (for S_1 and S_2) and two that show the *testing error* as a function of the model complexity (for S_1 and S_2). Label each of the 4 curves clearly. The constant curve labeled "true error" corresponds to the error due to the inherent noise in the samples and is drawn as a reference curve. *Solution:*



Solution:

- The training error decreases when increasing the model complexity, while the test error decreases first but then increases due to overfitting.
- Given the same model complexity, the model has larger training samples is less likely to overfit than the one with less samples. So the two curves representing 100k samples are nearer the dash line the other two curves.

4 Exponential Families

Consider the Poisson distribution with parameter λ. It has a probability mass function given by $p(i) = \frac{\lambda^i e^{-\lambda}}{i!}$, $i = 0, 1, \cdots$.

(i) Write $p(i)$ in the form of an exponential distribution $p(i) = h(i)e^{\eta\phi(i)-A(\eta)}$. Explicitly specify h, η, ϕ, and $A(\eta)$.

(ii) Compute $\frac{dA(\eta)}{d\eta}$ and $\frac{d^2A(\eta)}{d\eta^2}$? Is this the result you expected?

Solution:

1. $$p(i) = \frac{\lambda^i e^{-\lambda}}{i!} = \frac{1}{i!}e^{\ln(\lambda)i-\lambda} = h(i)e^{\eta\phi(i)-A(\eta)},$$

   with $h(i) = 1/i!$, $\phi(i) = i$, $\eta = \ln(\lambda)$, and $A(\eta) = \lambda = e^\eta$.

2. $\frac{dA(\eta)}{d\eta} = \frac{d^2A(\eta)}{d\eta^2} = A(\eta) = \lambda$.

5 Multiple Choice Questions and Simple Problems

Mark the correct **answer(s)**. More than one answer can be correct!

- In regression, "complex" models tend to

  1. overfit
  2. have large bias
  3. have large variance

  Solution: 1 and 3 are correct

- In regression, "simple" models tend to

  1. overfit
  2. have large bias
  3. have large variance

  Solution: 2 is correct

- We are given a data set $S = \{(\mathbf{x}_n, y_n)\}$ for a binary classification task where $\mathbf{x}_n$ in $\mathbb{R}^D$. We want to use a *nearest-neighbor* classifier. In which of the following situations do we have a reasonable chance of success with this approach? [Ignore the issue of complexity.]

  1. n is fixed, $D \to \infty$
  2. $n = D^2$, $D \to \infty$
  3. $n \to \infty$, $D \ll \ln(n)$
  4. $n \to \infty$, D is fixed

  Solution: If the number of data points is exponential in the dimension then we have a chance that a nearest neighbor classifier works. Therefore when $n \to \infty$ and D is either fixed or very small compared to $\ln(n)$ then we have a chance. Hence, 3 and 4 are correct.

- We add a regularization term because

  1. this sometimes renders the minimization problem of the cost function into a strictly convex/concave problem
  2. this tends to avoid overfitting
  3. this converts a regression problem into a classification problem

  Solution: 1 and 2 are correct

- The k-nearest neighbor classifier

  1. typically works the better the larger the dimension of the feature space
  2. can classify up to k classes
  3. typically works the worse the larger the dimension of the feature space
  4. can only be applied if the data can be linearly separated
  5. has a misclassification rate of at most two times the one of the Bayes classifier if we have lots of data
  6. has a misclassification rate that is two times better than the one of the Bayes classifier

  Solution: 3 and 5 are correct

- A real-valued scalar Gaussian distribution

  1. is a member of the exponential family with one scalar parameter
  2. is a member of the exponential family with two scalar parameters
  3. is not a member of the exponential family

  Solution: 2 is correct

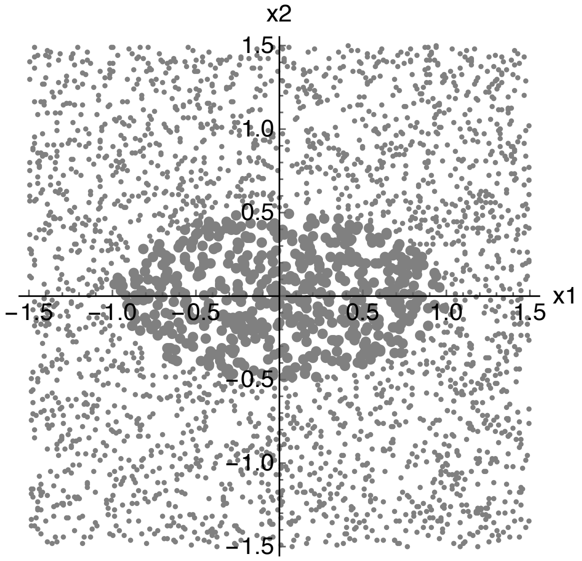


Figure 1: Some 2D data for classification. The two classes are indicated by different point sizes.

- Which of the following statements is correct, where we assume that all the stated minima and maxima are in fact taken on in the domain of relevance.

  1. $\max\{0, x\} = \max_{\alpha \in [0,1]} \alpha x$
  2. $\min\{0, x\} = \min_{\alpha \in [0,1]} \alpha x$
  3. Let $g(x) := \min_y f(x, y)$. Then $g(x) \leq f(x, y)$
  4. $\max_x g(x) \leq \max_x f(x, y)$
  5. $\max_x \min_y f(x, y) \leq \min_y \max_x f(x, y)$

  Solution: all are correct

- Which of the following statements are correct?

  1. The training error is typically smaller than the test error.
  2. The SVM (support vector machine) formulation we discussed can be optimized using SGD.
  3. One iteration of SGD for ridge regression costs roughly $\Theta(ND)$, where N is the number of samples and D is the dimension.
  4. Logistic regression as formulated in class can be optimized using SGD.

  Solution: 1, 2, and 4 are correct

- You have given the 2D data shown in Figure 1. You are allowed to add one component to your data (in addition to a constant component) and then must use a linear classifier. What component should you pick?

  1. $x_1 + x_2$
  2. $1/|x_1 + x_2|$
  3. $x_1 + 4x_2$
  4. $4x_1 + x_2$
  5. $4x_1^2 + x_2^2$
  6. $x_1^2 x_2^2$
  7. $x_1^2 + 4x_2^2$

  Solution: The decision region is an ellipsoid with description $x_1^2 + 4x_2^2 = 1$. Hence 7 is correct.

- The following functions are convex:

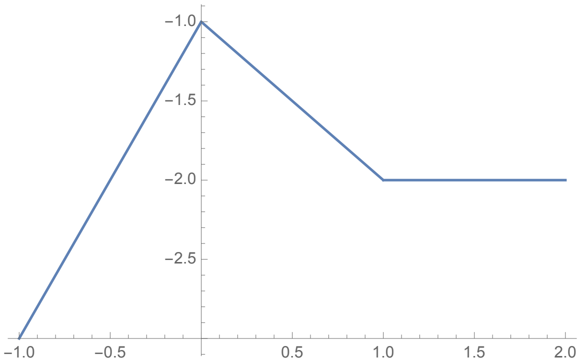


Figure 2: What is the subgradient of this function at $x = 1$?

1. $f(x) := x^2$, $x \in \mathbb{R}$
2. $f(x) := x^3$, $x \in [-1, 1]$
3. $f(x) := -x^3$, $x \in [-1, 0]$
4. $f(x) := e^{-x}$, $x \in \mathbb{R}$
5. $f(x) := e^{-x^2/2}$, $x \in \mathbb{R}$
6. $f(x) := \ln(1/x)$, $x \in (0, \infty)$
7. $f(x) := g(h(x))$, $x \in \mathbb{R}$, where g, h are convex and increasing over $\mathbb{R}$

Solution: 1, 3, 4, 6, and 7 are correct

- Let $f : \mathbb{R}^D \to \mathbb{R}$ be the function $f(\mathbf{w}) := \exp(\mathbf{x}^\top \mathbf{w})$, where $\mathbf{x} \in \mathbb{R}^D$. What is $\nabla_{\mathbf{w}} f$? *Solution:*

$$\nabla_{\mathbf{w}} f(\mathbf{w}) = f(\mathbf{w})\,\mathbf{x} \quad \in \mathbb{R}^D$$

- Which of the following scalars g is a subgradient for the function shown in Figure 2 at the point $x = 1$?

1. $g = -1$
2. $g = -\frac{1}{2}$
3. none exists
4. $g = 0$

Solution: None exists. In order to be a subgradient the linear function defined by the operating point and the slope g should be a global lower bound on the function.