Exercises
Introduction to Machine Learning
FS 2020

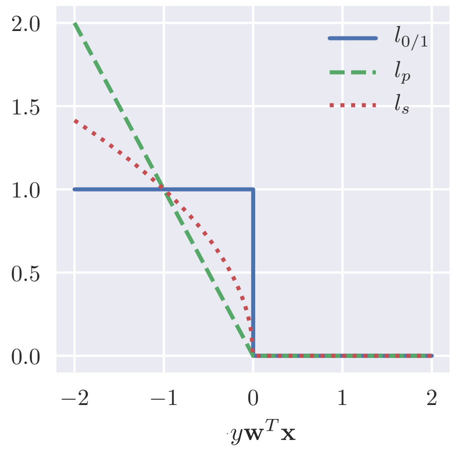
Series 2, Mar 27th, 2020 (Kernel)

Institute for Machine Learning
Dept. of Computer Science, ETH Zürich
Prof. Dr. Andreas Krause
Web: `https://las.inf.ethz.ch/teaching/introml-s20`
For questions, please refer to Piazza.

Problem 1 (SVM):

This exercise is based on an exercise designed by Stephanie Hyland. In its original formulation, the perceptron aims to minimise a $0/1$-loss function (shown below, solid). Because this objective is neither convex nor differentiable, a surrogate loss function is optimised (typically, $l_p(\mathbf{w}; \mathbf{x}, y) = \max(0, -y\mathbf{w}^T\mathbf{x})$, dashed). In this exercise, we consider a different surrogate loss function l_s, which approximates the $0/1$-loss function more closely.

$$l_s(\mathbf{w}; \mathbf{x}, y) = \begin{cases} 0, & \text{for } \operatorname{sign}(\mathbf{w}^T\mathbf{x}) = y \\ \sqrt{-y\mathbf{w}^T\mathbf{x}}, & \text{for } \operatorname{sign}(\mathbf{w}^T\mathbf{x}) \neq y \end{cases}$$



1. Mark the following statements as True or False. Try to justify the answer for yourself.
   (a) l_p is convex.
   (b) l_p is differentiable.
   (c) l_s is convex.
   (d) l_s is differentiable.

Solution:
Only (a) is True.

(a) l_p, known as *the hinge loss*, is convex because it is the maximum of two linear functions, and:
   i. Any linear function is convex.
   ii. The maximum of two convex functions is convex.

(b) Let's differentiate with respect to $y\mathbf{w^T x}$. If $sign(\mathbf{w^T x}) = y, l_p^{'}(\mathbf{w};\mathbf{x},y) = 0$. If $sign(\mathbf{w^T x}) \neq y$, $l_p^{'}(\mathbf{w};\mathbf{x},y) = -1$.
To check differentiability, we need to check the limit at point 0. $\lim_{x\to 0_-} l_p^{'} = -1$. l_p is not differentiable at $y\mathbf{w}^T\mathbf{x} = 0$, since the left and right derivatives are not equal.

(c) l_s is not convex.
To check whether ls is convex, we can look at $f(x) = \sqrt{x}$.
A way to show that $f(x) = \sqrt{(}x)$ is not convex is to show that $-f(x)$ is convex.

$$\sqrt{tx_1 + (1-t)x_2} > t\sqrt{x_1} + (1-t)\sqrt{x_2}$$

$$tx_1 + (1-t)x_2 > t^2x_1 + (1-t)^2x_2 + t1(1-t)\sqrt{x_1x_2}$$

$$x_1 + x_2 > 2\sqrt{x_1x_2}$$

$$(\sqrt{x_1} - \sqrt{x_2})^2 > 0$$

Hence, $f(x) = \sqrt{x}$ is concave and so is l_s.

(d) Let's differentiate with respect to $y\mathbf{w^T x}$. If $sign(\mathbf{w^T x}) = y, l_s^{'}(\mathbf{w};\mathbf{x},y) = 0$. If $sign(\mathbf{w^T x}) \neq y$, $l_s^{'}(\mathbf{w};\mathbf{x},y) = \frac{1}{2}(-y\mathbf{w}^T\mathbf{x})^{\frac{1}{2}}(-1) = \frac{1}{2\sqrt{-yy\mathbf{w}^T\mathbf{x}}}$.
To check differentiability, we need to check the limit at point 0. Let $z = y\mathbf{w}^T\mathbf{x}$. Then, $\lim_{x\to 0_-} -\frac{1}{\sqrt{z}} = -\infty$. Hence, l_s is not differentiable at $y\mathbf{w}^T\mathbf{x} = 0$.

2. Derive $\nabla l_s(w, x, y)$.

(a) $\begin{cases} 0, & \text{if } y = sign(w^Tx) \\ -\frac{yx}{2\sqrt{-y\mathbf{w}^T\mathbf{x}}}, & \text{if } y \neq sign(w^Tx) \end{cases}$

(b) $\begin{cases} 0, & \text{if } y = sign(w^Tx) \\ -\frac{yx}{2\sqrt{y\mathbf{w}^T\mathbf{x}}}, & \text{if } y \neq sign(w^Tx) \end{cases}$

(c) $\begin{cases} 0, & \text{if } y = sign(w^Tx) \\ \frac{yx}{2\sqrt{-y\mathbf{w}^T\mathbf{x}}}, & \text{if } y \neq sign(w^Tx) \end{cases}$

(d) $\begin{cases} 0, & \text{if } y = sign(w^Tx) \\ \frac{yx}{2\sqrt{y\mathbf{w}^T\mathbf{x}}}, & \text{if } y \neq sign(w^Tx) \end{cases}$

Solution:
The correct answer is (a).
Although l_s not differentiable at $y\mathbf{w}^T\mathbf{x} = 0$, the subgradient exists and hence (stochastic) gradient descent converges. To derive the subgradient let's rewrite the function l_s as $l_s(\mathbf{w};\mathbf{x},y) = max(0, \sqrt{-y\mathbf{w}^T\mathbf{x}})$. Now let $f(z) = max(0, -\sqrt{yz}) and g(\mathbf{w}) = \mathbf{w}^T\mathbf{x}$. We use the chain rule

$$\frac{\partial}{\partial w_i} f(g(\mathbf{w})) = \frac{\partial f}{\partial z}\frac{\partial g}{\partial w_i}$$

. We get

$$\frac{\partial f}{\partial z} = \begin{cases} 0, & \text{for sign}(z) = y \\ -\frac{y}{2\sqrt{-yz}}, & \text{for sign}(z) \neq y \end{cases}$$

and $\frac{\partial g}{\partial w_i} = x_i$. Hence,

$$\frac{\partial f(g(\mathbf{w}))}{\partial w_i} = \begin{cases} 0, & \text{for sign}(z) = y \\ -\frac{y\mathbf{x}}{2\sqrt{-y\mathbf{w}^T\mathbf{x}}}, & \text{for sign}(z) \neq y \end{cases}$$

3. The exercise suggests to train an SVM, where we penalise the margin violation given by $(1 - y\mathbf{w}^T\mathbf{x})_+ = max(1 - y\mathbf{w}^T\mathbf{x}, 0)$, not linearly but with the square root instead. Correspondingly, our modified SVM seeks to optimise the following objective

$$L(\mathbf{w}) = \frac{1}{n}\Sigma_{i=1}^{n}\sqrt{(1 - y\mathbf{w}^T\mathbf{x})_+} + \lambda\|w\|^2$$

.

Pick the correct update step for stochastic gradient descent.

(a) Pick $i_t \sim Unif(1, 2, ...n)$.
If $y_{it}\mathbf{w_t^T}\mathbf{x_{it}} < 1$
$w_{t+1} = w_t(1 - \eta_t 2\lambda) + \eta_t \frac{y_i\mathbf{x_i}}{2\sqrt{(1 - y_i\mathbf{w}^T\mathbf{x_i})}}$
Else
$w_{t+1} = w_t(1 - \eta_t 2\lambda)$

(b) Pick $i_t \sim Unif(1, 2, ...n)$.
If $y_{it}\mathbf{w_t^T}\mathbf{x_{it}} < 1$
$w_{t+1} = w_t(1 - \eta_t 2\lambda)$
Else
$w_{t+1} = w_t(1 - \eta_t 2\lambda) + \eta_t \frac{y_i\mathbf{x_i}}{2\sqrt{(1 - y_i\mathbf{w}^T\mathbf{x_i})}}$

(c) Pick $i_t \sim Unif(1, 2, ...n)$.
If $y_{it}\mathbf{w_t^T}\mathbf{x_{it}} < 1$
$w_{t+1} = w_t(1 + \eta_t 2\lambda) + \eta_t \frac{y_i\mathbf{x_i}}{2\sqrt{(1 - y_i\mathbf{w}^T\mathbf{x_i})}}$
Else
$w_{t+1} = w_t(1 + \eta_t 2\lambda)$

(d) Pick $i_t \sim Unif(1, 2, ...n)$.
If $y_{it}\mathbf{w_t^T}\mathbf{x_{it}} < 1$
$w_{t+1} = w_t(1 + \eta_t 2\lambda)$
Else
$w_{t+1} = w_t(1 + \eta_t 2\lambda) + \eta_t \frac{y_i\mathbf{x_i}}{2\sqrt{(1 - y_i\mathbf{w}^T\mathbf{x_i})}}$

Solution:
The correct answer is (a).
For $y_{it}\mathbf{w_t^T}\mathbf{x_{it}} < 1$,

$$\nabla_w L = -\frac{y_i\mathbf{x_i}}{2\sqrt{(1 - y_i\mathbf{w}^T\mathbf{x_i})}} + 2\lambda\mathbf{w_t}$$

Else,

$$\nabla_w L = 2\lambda\mathbf{w_t}$$

Why may this modification not be a good idea? You can see that the weight update due to margin violations getsrescaled as a result of the modification by the factor $\frac{1}{2\sqrt{1-y_i\mathbf{w}^T\mathbf{x_i}}}$. This factor is small when the margin violation is large and large when the margin violation is small, which may make training this modified SVM troublesome.

Problem 2 (Kernels):

Use the basic rules for kernel decomposition discussed in class or otherwise and assuming that $k(x, y)$ is a valid kernel, letting $f : \mathbb{R} \to \mathbb{R}$ in a) and b), $g : \mathcal{X} \to \mathbb{R}_+$ for d), $f : \mathcal{X} \to \mathbb{R}$ for e) and f), and $\phi : \mathcal{X} \to \mathcal{X}'$.

4. Mark the following statements as True or False. Try to justify your answers to yourself.

   (a) $k_a(x, y) = f(k(x, y))$ is a valid kernel, if f is a polynomial with non-negative coefficients.

   (b) $k_b(x, y) = f(k(x, y))$ is a valid kernel, if f is any polynomial.

   (c) $k_c(x, y) = \exp(k(x, y))$ is a valid kernel.

   (d) $k_d(x, y) = g(x)k(x, y)g(y)$ is a valid kernel.

   (e) $k_e(x, y) = f(x)k(x, y)f(y)$ is a valid kernel.

   (f) $k_f(x, y) = k(\phi(x), \phi(y))$ is a valid kernel.

   Solution:
   (a), (c), (d), (e) and (f) are True.

   (a) Since each polynomial term is a product of kernels with non-negative coefficients, the proof follows from the rules of addition and multiplication yielding valid kernels.

   (b) Product of kernels with *negative* coefficients is not necessarily a valid kernel.

   (c) We can use the Taylor expansion around 0:

   $$\begin{aligned} exp(k(x, y)) &= exp(0) + exp(0)k(x, y) + \frac{exp(0)}{2!}(k(x, y))^2 + ... \\ &= 1 + k(x, y) + \frac{1}{2}(k(x, y))^2 + \frac{1}{6}(k(x, y))^3 ... \end{aligned}$$

   (d) and (e) Since k(x, y) is a valid kernel, we can define a feature map $\phi(.)$, such that $k(x, y) = \langle\phi(x), \phi(y)\rangle$.
   Now,

   $$k_e(x, y) = f(x)k(x, y)f(y) = f(y)f(x)\langle\phi(x), \phi(y)\rangle = f(y)\langle f(x)\phi(x), \phi(y)\rangle = \langle f(x)\phi(x), f(y)\phi(y)\rangle$$

   Hence, with the new feature map $\phi_e(.) = f(.)\phi(.)$, $k_e(x, y)$ is a valid kernel (symmetry and positive definiteness properties don't change). This is a solution for (e). (d) follows from this, as it a specific case of the same.

   (f) We know that $k(x, y)$ is a valid kernel and hence, on any set of vectors (also transformed ones) it yields a valid kernel.

5. For $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$, and $K(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T\mathbf{x}' + 1)^2$, identify possible feature maps $\phi(\mathbf{x})$, such that $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^\top\phi(\mathbf{x}')$. Let $\mathbf{x}^T = (x_i, ..., x_d)$.

   (a) $(1, \sqrt{2}x_1, ..., \sqrt{2}x_d, x_1x_1, x_1x_2, ...x_ix_j...)$

   (b) $(1 + x_1, ...1 + x_i, ...1 + x_d)$

(c) $(1, -\sqrt{2}x_1, ..., -\sqrt{2}x_d, -x_1x_1, -x_1x_2, ... - x_ix_j...)$

(d) $(1, \frac{1}{\sqrt{2}}x_1, ..., \frac{1}{\sqrt{2}}x_d, x_1x_1, x_1x_2, ...x_ix_j..., \frac{1}{\sqrt{2}}x_1, ..., \frac{1}{\sqrt{2}}x_d)$

Solution:
(a) and (c) are correct answers.

$$(\mathbf{x}^T\mathbf{x}' + 1)^2 = (\Sigma_i x_i x_i^{'} + 1)^2 = 1 + 2\Sigma_i x_i x_i^{'} + \Sigma_i\Sigma_j(x_ix_j)(x_i^{'}x_j^{'})$$

(a) $(1, \sqrt{2}x_1, ..., \sqrt{2}x_d, x_1x_1, x_1x_2, ...x_ix_j...)^T(1, \sqrt{2}x_1, ..., \sqrt{2}x_d, x_1x_1, x_1x_2, ...x_ix_j...)$
$= 1 + 2\Sigma_i x_i x_i^{'} + \Sigma_i\Sigma_j(x_ix_j)(x_i^{'}x_j^{'})$.

(b) $(1 + x_1, ...1 + x_i, ...1 + x_d)^T(1 + x_1, ...1 + x_i, ...1 + x_d)$
$= \Sigma_i(1 + x_i)^2 = \Sigma_i(1 + 2x_i + x_i^2) \neq 1 + 2\Sigma_i x_i x_i^{'} + \Sigma_i\Sigma_j(x_ix_j)(x_i^{'}x_j^{'})$.

(c) $(1, -\sqrt{2}x_1, ..., -\sqrt{2}x_d, -x_1x_1, -x_1x_2, ...-x_ix_j...)^T(1, -\sqrt{2}x_1, ..., -\sqrt{2}x_d, -x_1x_1, -x_1x_2, ...-x_ix_j...) =$
$1 + 2\Sigma_i x_i x_i^{'} + \Sigma_i\Sigma_j(x_ix_j)(x_i^{'}x_j^{'})$.

(d) $(1, \frac{1}{\sqrt{2}}x_1, ..., \frac{1}{\sqrt{2}}x_d, x_1x_1, ...x_ix_j..., \frac{1}{\sqrt{2}}x_1, ..., \frac{1}{\sqrt{2}}x_d)^T(1, \frac{1}{\sqrt{2}}x_1, ..., \frac{1}{\sqrt{2}}x_d, x_1x_1, ...x_ix_j..., \frac{1}{\sqrt{2}}x_1, ..., \frac{1}{\sqrt{2}}x_d) =$
$1+2\Sigma_i\frac{1}{2}x_ix_i^{'}+\Sigma_i\Sigma_j(x_ix_j)(x_i^{'}x_j^{'}) = 1+\Sigma_ix_ix_i^{'}+\Sigma_i\Sigma_j(x_ix_j)(x_i^{'}x_j^{'}) \neq 1+2\Sigma_ix_ix_i^{'}+\Sigma_i\Sigma_j(x_ix_j)(x_i^{'}x_j^{'})$

6. For the dataset $X = \{\mathbf{x}_i\}_{i=1,2} = \{(-3, 4), (1, 0)\}$ and the feature map $\phi(\mathbf{x}) = [x^{(1)}, x^{(2)}, \|\mathbf{x}\|]$, calculate the Gram matrix (for a vector $\mathbf{x} \in \mathbb{R}^2$ we denote by $x^{(1)}, x^{(2)}$ its components).

   (a) $\begin{pmatrix} 50 & 2 \\ 2 & 2 \end{pmatrix}$

   (b) $\begin{pmatrix} 50 & 4 \\ 4 & 4 \end{pmatrix}$

   (c) $\begin{pmatrix} -50 & 2 \\ 2 & 2 \end{pmatrix}$

   (d) $\begin{pmatrix} 50 & 2 \\ 4 & 4 \end{pmatrix}$

   Solution:
   The correct answer is (a).
   First, we get $\phi(x)$ for each x.

   (a) $\phi([-3, 4]) = (-3, 4, 5)$
   (b) $\phi([1, 0]) = (1, 0, 1)$

   Now we get the inner products:

   (a) $\phi([-3, 4])^T\phi([-3, 4]) = 50$
   (b) $\phi([-3, 4])^T\phi([1, 0]) = 2$
   (c) $\phi([1, 0])^T\phi([1, 0]) = 2$

   And now the Gram matrix ϕ is simply given by $\phi_{i,j} = \phi(x_i)^T\phi(x_j)$; using the above:

$$\begin{pmatrix} 50 & 2 \\ 2 & 2 \end{pmatrix}$$