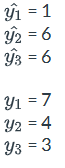
1. Select all of the statements below that correctly characterize the relationship between machine learning and optimization. Selected Answer,

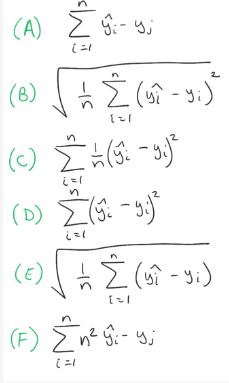
* They are two distinct 清楚的 fields with a high degree of overlap.
* They are essentially the same because they are designed to solve the same problems.两者不同，只是有交集
* Optimization is intended to do function approximation while machine learning performs function minimization. optimization 是 function minimization / maximization , ML 是 minimize the classification error
* Optimization methods are frequently incorporated into ML methods.

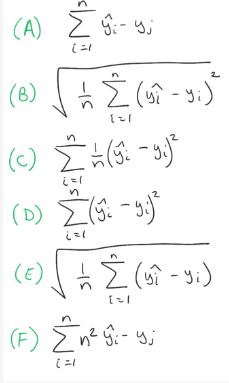
1. The goal of optimization is function minimization / minimization to find the set of inputs to a function that result in the extrema of function
2. Suppose a true mapping function exists f:A→ B along with historical pairs of observations from the domain and range. Which of the following describes the aim of machine learning in this scenario?

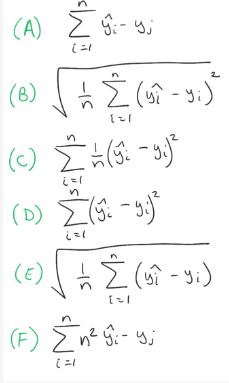
* to approximate f
* to find the parameters of f
* to make predictions in B
* to minimize f

1. True - Every machine learning method makes use of some kind of optimization.
2. Compute the [RSME](https://www.statology.org/rmse-calculator/) of the following point. 上面是observed ， 下面是predicted

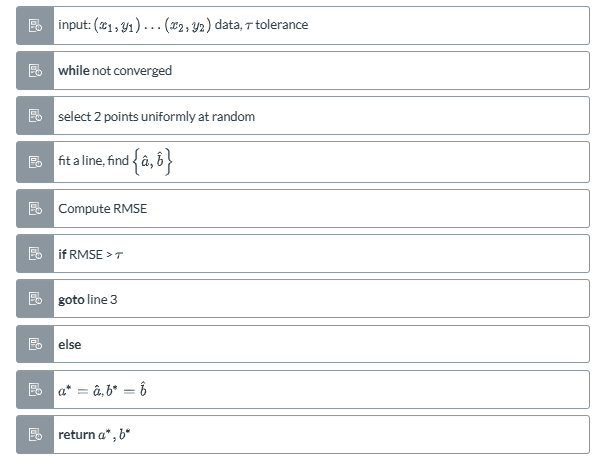


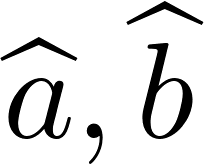
1. Identify the formulas  
   residual 

RMSE

MSE 

1. simplified RANSAC algorithm



1. In Lecture Notes 3.2, when we use simplified RANSAC to find an estimate {[](https://www.codecogs.com/eqnedit.php?latex=%7B%5Cwidehat%7Ba%7D%2C%5Cwidehat%7Bb%7D%7D#0) }, what have we actually accomplished?

* We fitted the model parameters using RANSAC.
* We found a solution.
* We found an instance of a model.
* We found the optimal parameters.

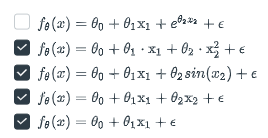
1. form an expression for the square loss



1. If you encounter non-zero losses when the hypothesis space is limited to linear functions,

* the choice of linear functions must be inappropriate.
* means that even the best hypothesis ℎ∗ in may not fit the data.
* the underlying function must be non-linear.
* this is to be expected, as there is nearly always some noise in the data.

1. which of the following functions are linear models that can be solved with linear regression.

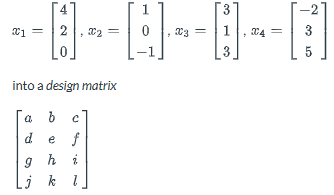


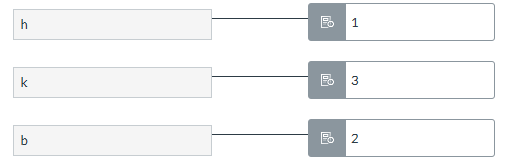
1. an expression that minimizes the mean square error (MSE) via the model parameters.



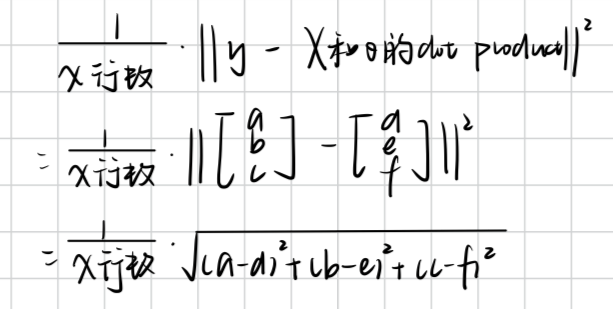
1. Arrange the four input vectors given below

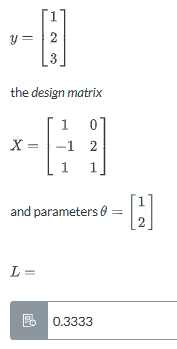
将vector按顺序横过来，然后对着填就可以



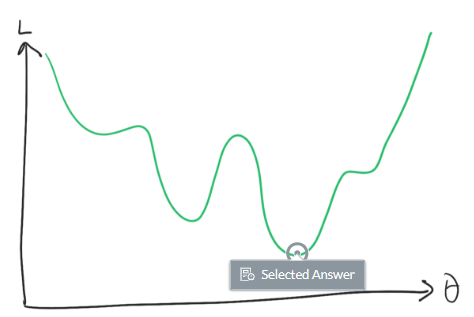


1. Compute the expected loss / empirical risk using Mean Squared Error given the following

公式   [dotproduct generator](https://keisan.casio.com/exec/system/1311593076)

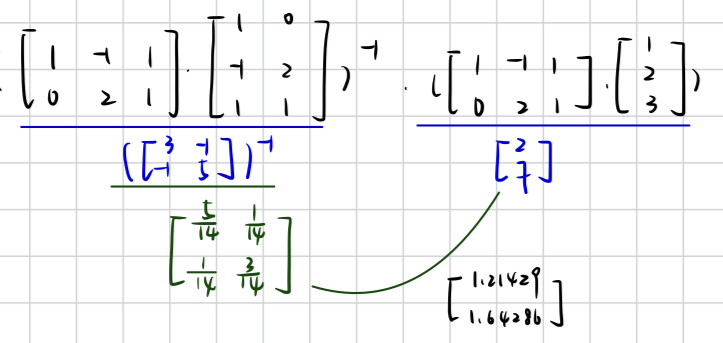


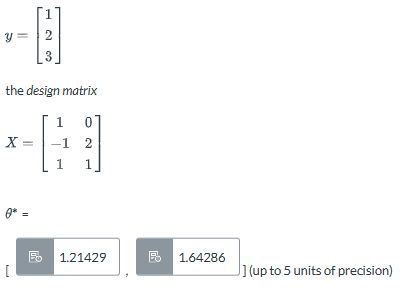
1. If we know that the expected loss function is convex , then we can get an expression for the optimal parameters θ\* by setting the gradient to zero
2. True - you can test for convexity by connecting any two points of a function with a line, then check to see if the function lays below the line. If this is true for all pairs of points, the function is convex.
3. Click on the location in the plot where lays.



1. Compute the optimal parameters for the following linear regression problem using the algebraic method from Lesson 3.2

公式

[dot product generator](https://keisan.casio.com/exec/system/1311593076) , [inverse generator](https://matrix.reshish.com/inverse.php)

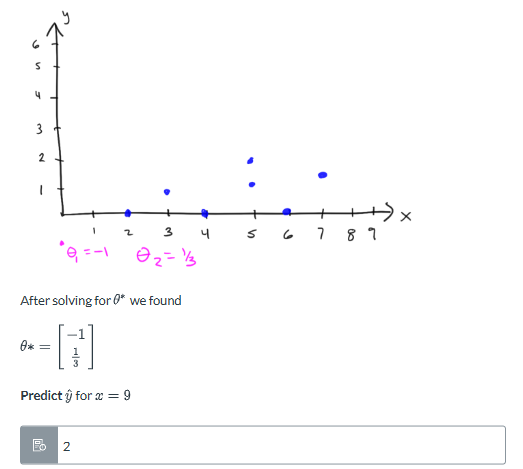


1. Under what conditions can we find the exact solution to a linear regression problem.

* If is well-behaved
* If is invertible
* If is full rank
* If the determinant of the gram matrix of is non-zero

1. 公式

重点 ： 就很有可能的方向是反的，要看图！

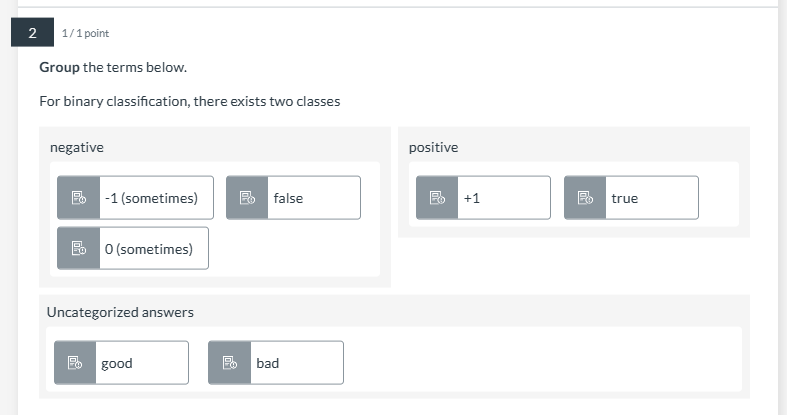


1. False - points where always correspond to a minima or local minima.

2.3

1. For the six cases depicted above, categorize each as a regression problem or classification problem.

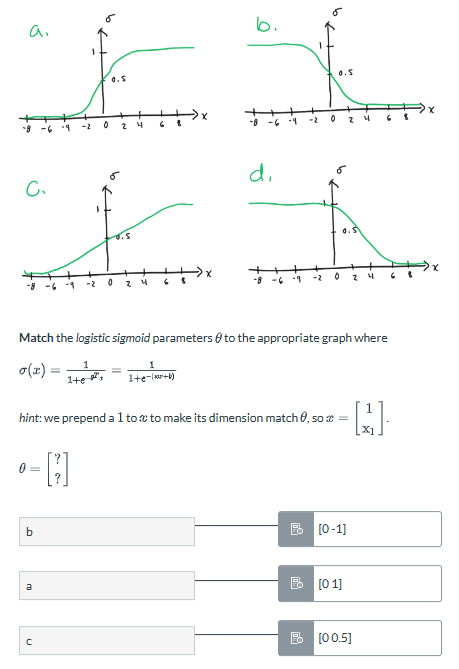
| classification A或B，A和B分开的问题 | regression 预测 |
| --- | --- |
|  |  |

1. binary classification 
2. In logistic regression we **estimate** the parameter **p** from the Bernoulli distribution. To  **decide** upon a label  **ŷ** given an example **x** , we estimate **p̂** and apply a threshold, commonly ≥ 0.5 for

**ŷ = 1** and < 0.5 for **ŷ = 0**

1. What are the reasons we introduce the *logistic sigmoid* function into the logistic regression problem?

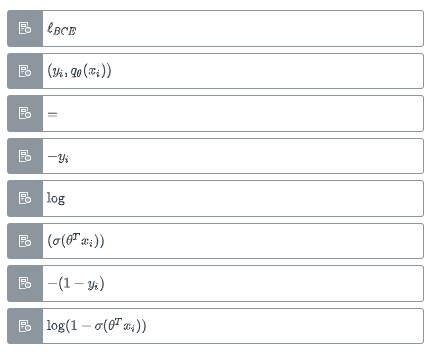
* the range of the logistic sigmoid is bounded to [0,1]
* the logistic sigmoid returns ŷ =1 at x=0
* the steepness of the transition can be controlled with parameters
* the logistic sigmoid has a smooth transition between 0 and 1



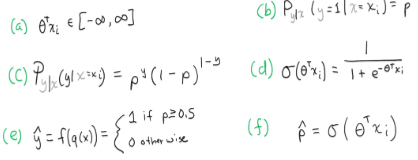
1. **Mark** all of the correct statements regarding *logistic regression*.

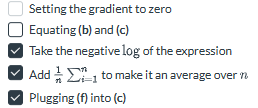
* It is a linear model ( is linear in
* Its domain is bounded to [0,1]
* It *regresses* the probabilities of class membership.
* It cannot perform classification without a decision rule.
* We can add a third class y= −1to logistic regression without any modifications.

1. **Arrange** the following terms to arrive at the expression for the binary cross entropy loss

****

1. Which of the following are steps necessary to derive the average binary cross entropy loss used in logistic regression?

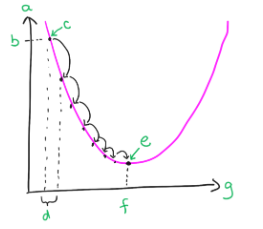


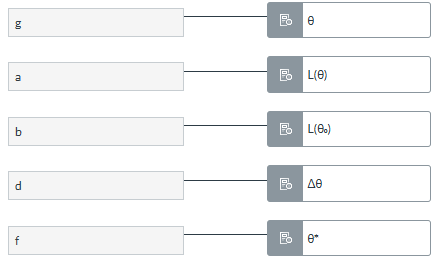


1. Why do we use gradient descent to optimize the parameters for logistic regression?

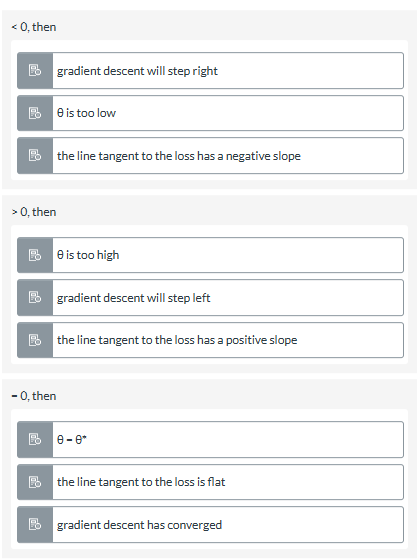
* The closed form solution is not necessarily optimal
* There is no known closed form expression to minimize in the average BCE loss
* Gradient descent converges to a better solution
* Gradient descent is always faster

1. Match the following to correctly annotate the diagram of gradient descent.



>

1. When performing gradient descent on a 1D convex loss, if the gradient of the average loss

****

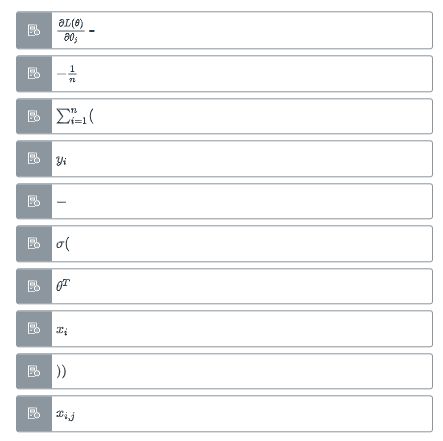
1. When performing gradient descent with a fixed learning rate

* the step size at each location is proportional to the gradient of the average loss at that location
* the step size is determined by the gradient at
* the step size will be constant
* the step size is proportional to the learning rate

1. 

* the term after the minus corresponds to the size
* this is the definition of the update step in gradient descent
* larger values of decrease the step size
* the minus (−) in the rightmost expression ensures increases when the gradient is positive
* must be set by hand (or some other scheme)
* in this context is a hyperparameter

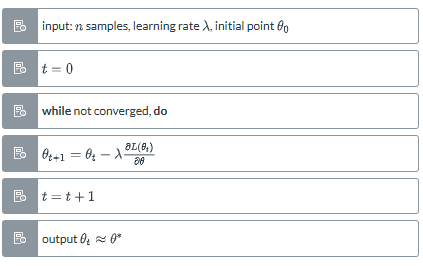
1. **Arrange** the following to form the expression for the partial derivative of the average loss w.r.t. elements of used in the gradient descent update step



1. Which of the following conditions are necessary to apply *gradient descent* (and for it to converge)?

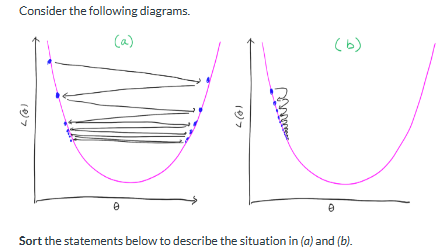
* The objective function that we are minimizing must be differentiable.
* The objective function must not have discontinuities.
* A reasonable learning rate must be chosen.
* The objective function we are minimizing must be convex.
* The design matrix must have full rank.

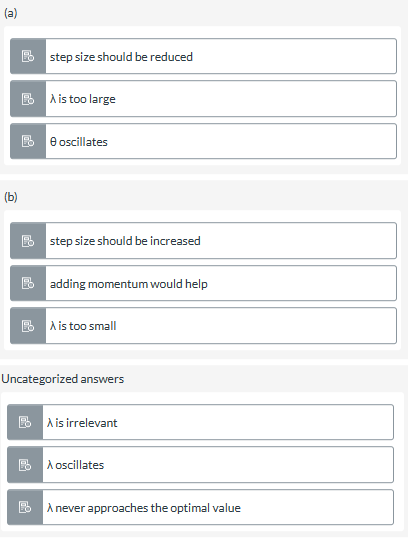
1. **Arrange** the following steps for *gradient descent* in the correct order



1. **Fill in** the following statement about *convergence* of gradient descent.

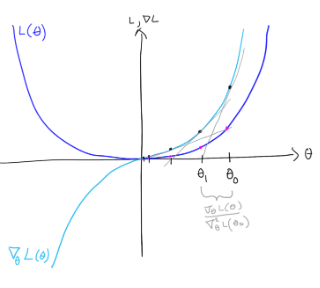
In theory, to check when gradient descent has converged, we can check to see if the derivative is **zero**. In practice, this may take an unreasonably long time due to a small step size, possible if  **λ** is poorly chosen. But **θₜ** may be close enough to  **θ\*** for a good solution. We can check this by checking **Δθ** to see if it is zero, or nearly so. We can also check the change in **L(θ)** to see if it stopped decreasing. Another option that is often used is to iterate for a **fixed** number of steps.





1. **Mark** all of the correct statements about *2nd order* optimization methods.

* Newton's method is a pure 2nd order that does not need to tune a learning rate
* Quasi-Newton methods approximate
* 1st order methods typically converge faster by taking cheaper but less effective steps than 2nd order methods
* 2nd order methods always converge in fewer steps than 1st order methods
* Newton's method is impractical to use in practice because of the cost to compute in the update equation.

1. 

**Complete** the following statement about Newton's method.

In each iteration of Newton's Methods, the parameters are updated by

**-∇L (θₜ)/∇² L(θₜ).** In essence, Newton's method is seeking the **zero** of the

gradient ∇L (θₜ). We can visualize the gradient of the gradient of L (θₜ) as a

**gray** line tangent to the gradient (the **cyan** curve). The next step, θ₁, should be

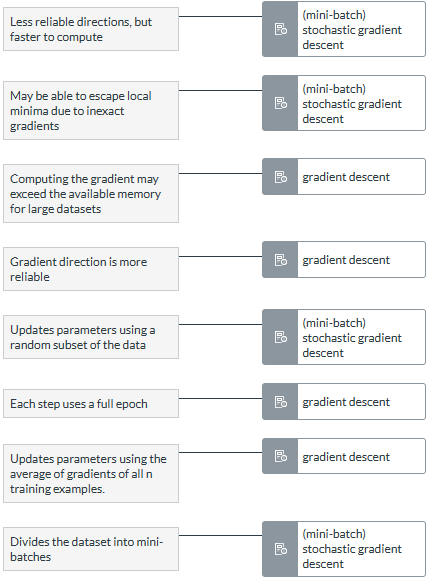
where this tangent line's value is **0**. If we treat θ₁ as the origin, we can find the

step size as the "run" or horizontal component of a line pointing towards ∇L

(θ₀) where the slope is "rise over run". The vertical component, or rise, is ∇L

(θ₀). Dividing this by the slope **∇L (θ₀)/∇² L(θ₀)** gives us the step size (the "run").

1. **Match** the following statements about *gradient descent*, *mini-batch (stochastic) gradient descent*, and *stochastic gradient descent.*

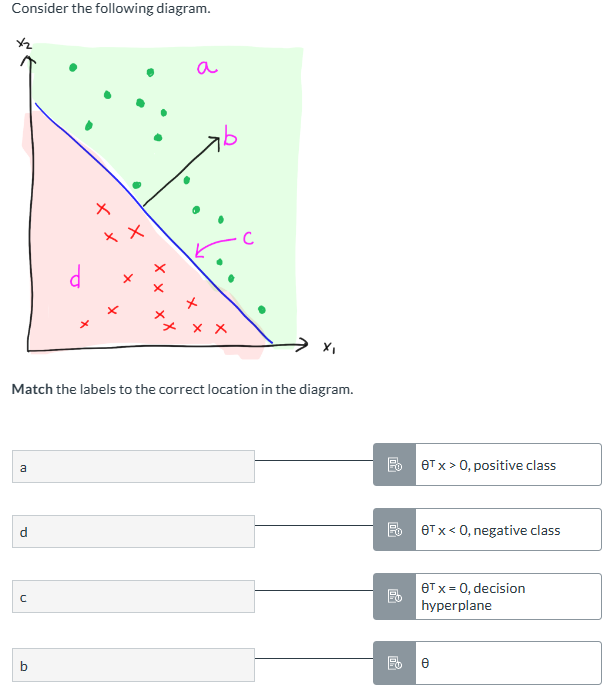
**

1. ***Fill in*** *the missing terms to complete the following statement.*

*Gradient descent can converge slowly on* ***smooth****, flat curvatures. On* ***pathological***  *shapes, often described as valleys or ravines, gradient descent can* ***oscillate*** *or approach the minimum with* ***tiny steps****.A method to address this issue is to add* ***momentum*** *into the optimizer, which builds speed if it goes in* ***the same direction***

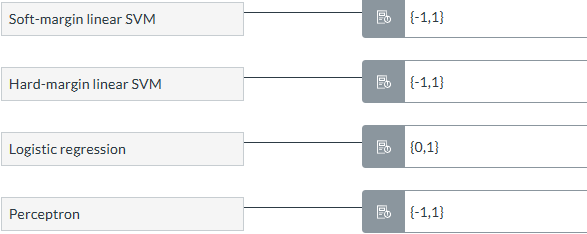
2,4

1. False - The word 'hyperplane' in *separating hyperplane* comes from the fact that the separation boundary is actually many planes.



1. In the Lecture Notes (and in general), the convention for binary classification is not always consistent. In some formulations it is more natural to set y={0,1}

where 0 corresponds to the *negative* class, and 1 corresponds to the *positive* class while in others it is typical to set y={−1,1} where −1corresponds to the *negative* class, and 1 corresponds to the *positive* class   
**Match** which methods follow which convention correctly.



1. **Compute** the output of a perceptron, given

= [-1.3 -1.4 2.3]

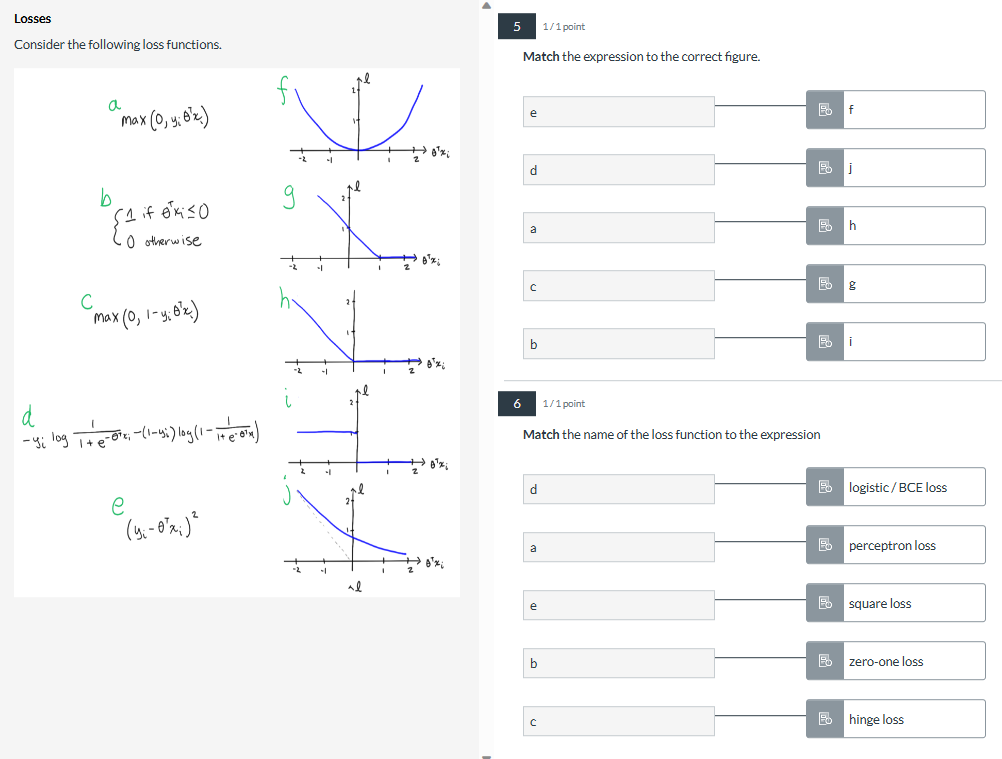
= [1 8 7]

Note: if by chance you arrive at a solution = 0, Canvas logic cannot handle this case. In this case enter 1 as your solution.

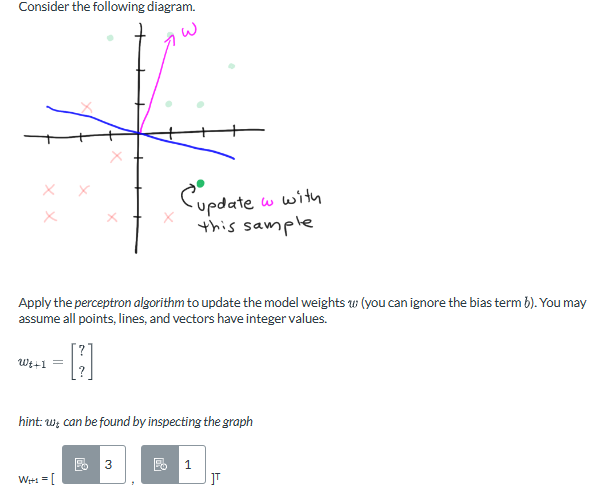
公式 : output= (,如果output > 0 , 结果就取1， 其余都是0



1. 和 6.

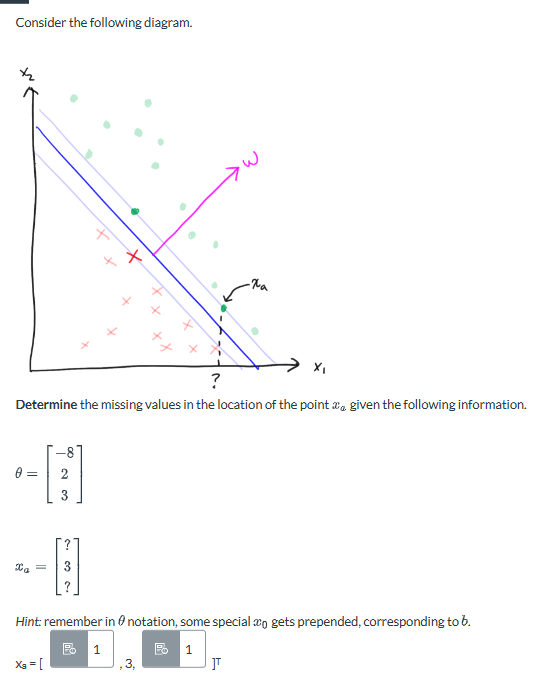


7.perceptron algorithm update model

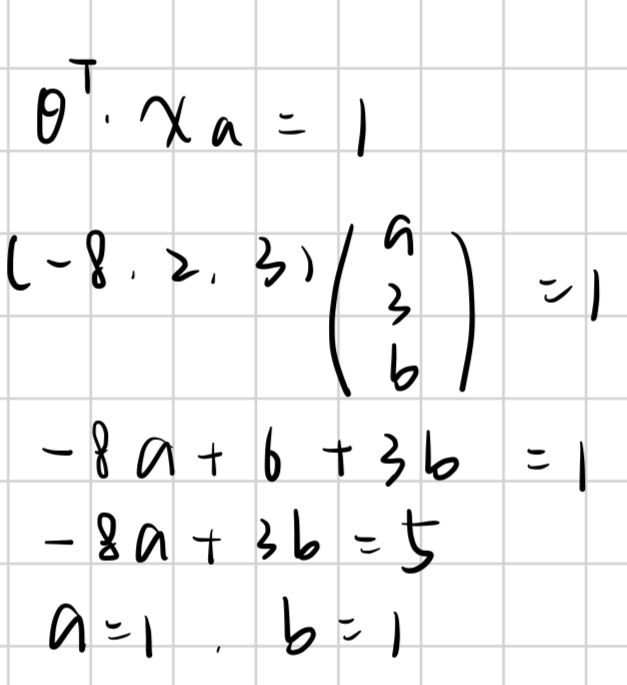


w = [1,3],

**8.------**



，结果只要满足A和B的关系式就可



9. An SVM classifier

* defines the margin in terms of the weights w
* Chooses support vectors randomly from the positive class
* Has, as one goal, to minimize the classification error
* Has, as one goal, to maximize the width of the margin

10. To formulate the **hard margin** classifier, we stipulate that a **margin**

that separates the **positive** class from the negative class should be maximized. Thus, the expression to be **maximized** is the width **2 / ||w||**. We also want to ensure that on the positive side of the margin **wᵀxᵢ + b ≥ 1**, and on the **negative** side of the margin

**wᵀxᵢ + b ≤-1**. These expressions can be arranged into a **constrained optimization** problem to be **minimized**.

11. Arrange the following expressions into the constrained optimization formulation for a hard margin classifier.



12. We formulate the optimization problem for a hard margin SVM as

* The equality constraints in the canonical form are implicit in our minimization term.
* corresponds to in the canonical (standard) constrained optimization form.
* To get our inequality constraints into the canonical form, we must set the right-hand side to 0
* The feasible region of the solution to the problem we've set up is a hyperplane.

13.True - a crucial drawback to the *hard margin SVM* is that it does not have any way to account for samples that lay on the wrong side of the margin.

14. The **feasible region** is the space in which the solution is **constrained** to. The optimal point **does not necessarily fall** within the feasible region. Equality constraints restrict the feasible region to a **hyperplane**. Inequality constraints restrict the feasible region to a

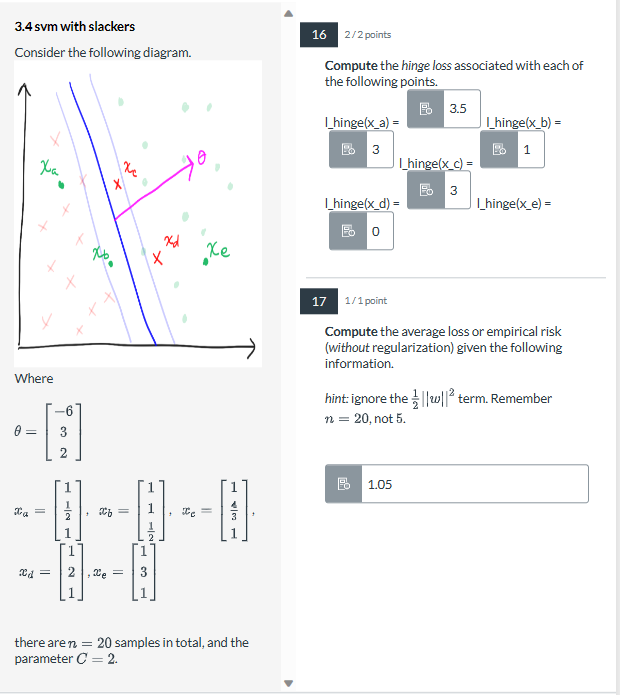
**polytope**.

15. SVM and perceptron can classify perfectly

技巧 ： perfectly的就是可以拉一条线直接将绿红完全分开

not perfect就是拉一条线之后，绿的那边会有红点，红的那边有绿点

16. 和 17

****

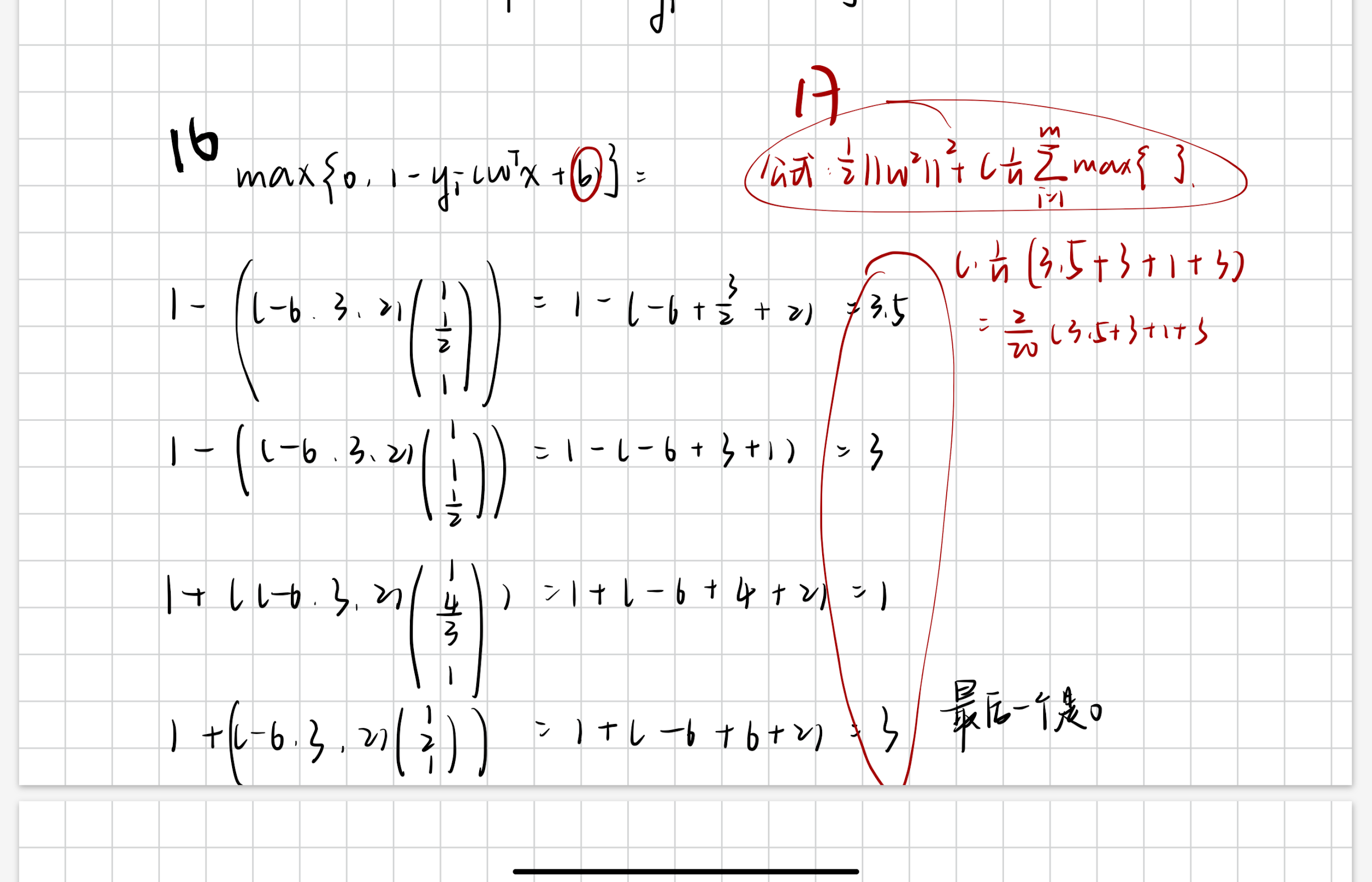
16. 每个sample的hinge loss 公式是 ： 3D，代表，所以就是用后一个公式，

其中

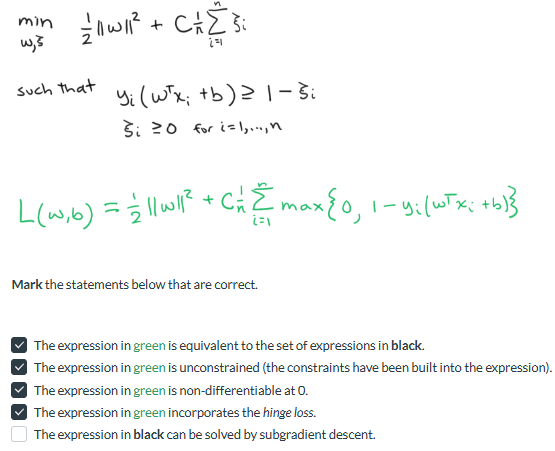
计算完了之后和0比大小

注意：绿点在绿色的区域，loss就是0，红点在红区域，loss也为0，也可以算保险一下，但是应该为0，此时的算数和n和C无关

17. 公式 ， 此时要看算不算在里面， 后面就是将上面所有的sample loss加起来除



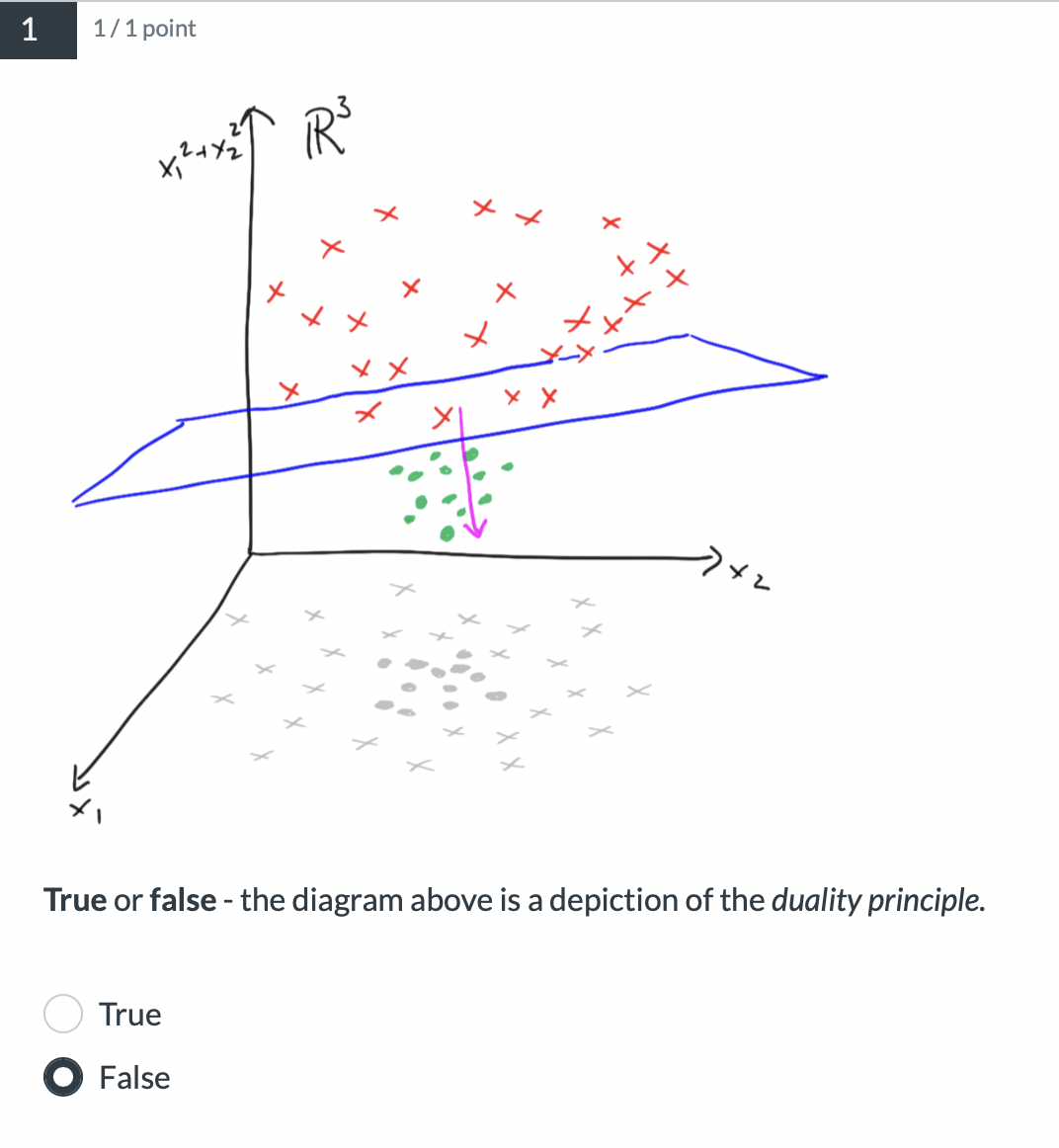
18.

****

19. The constant C in an SVM

* is not important for the performance
* is a hyperparameter of the model
* controls how tolerant the classifier is for samples that violate the margin
* controls the relative strength of the regularization term and the average loss term

Part 2.5



**2. Mark** all of the following statements that are correct.

❌The dual formulation is necessary to apply the *kernel trick.*

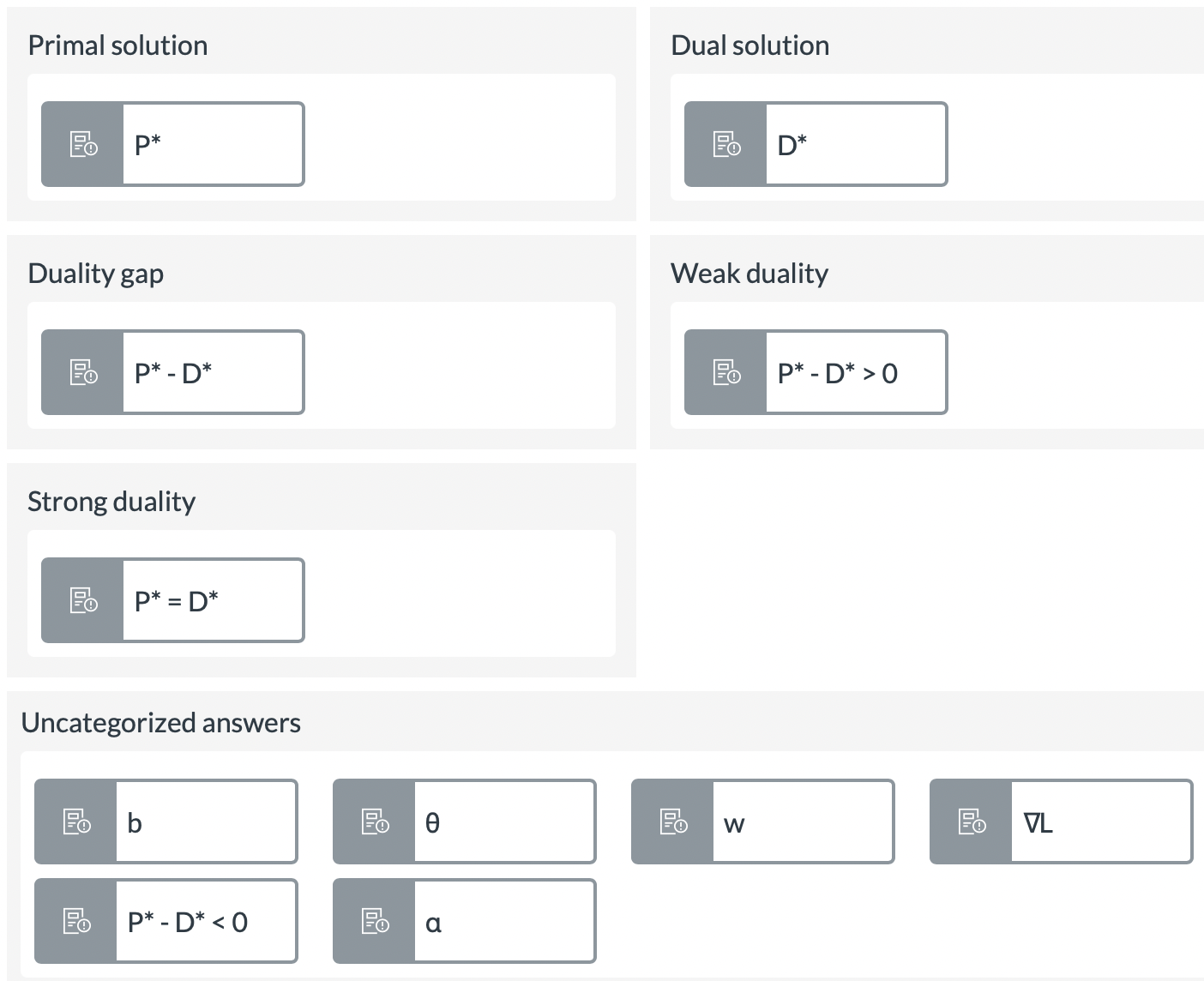
✅*The optimization variables in the dual form are Lagrange Multipliers.*

✅*The duality principle states that a constrained optimization problem, the primaI, may be reformulated from a second perspective, the dual.*

❌*The solution to the primal can always be found by solving the dual.*

✅*The dual is always concave.*

3.**Group** the terms into the correct categories.



**4.Fill in** the missing terms in the statement below.

The Lagrangian is a method to reformulate the multiple relations in a constrained optimization problem into a single expression. The underlying idea is that the common solution lays at a point where the function to minimize f(x) meets its constraint

function g(x) , and at that point the gradients of the two functions will point in the same direction and be proportional. The Lagrangian takes the form L(x, α) = f(x) + αg(x) because when you apply ∇L(x, α), a solution exists where the scaled gradients of the objective and constraint functions sum to 0.

**5. Arrange** the following terms to formulate the *Lagrangian* of the following constrained optimization problem.

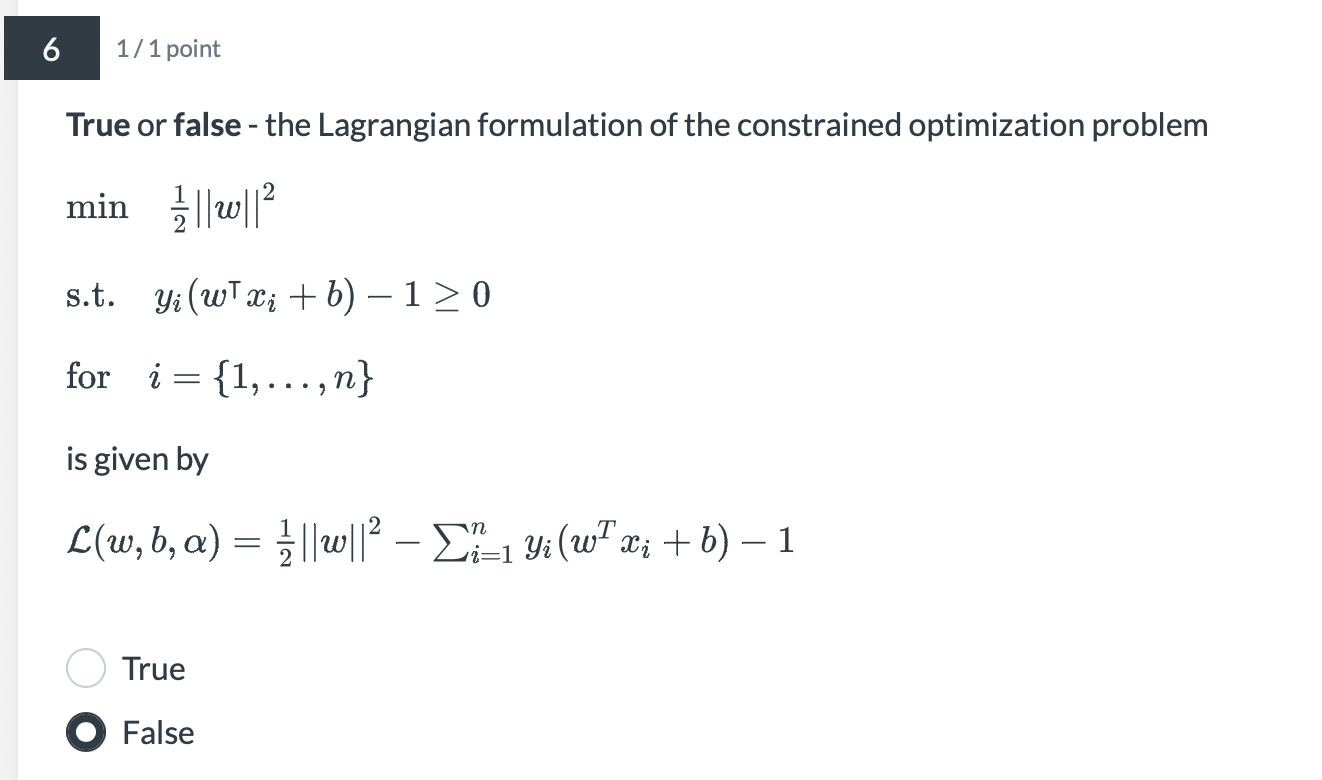
minx^2+z^2

s.t.2x^2−z=b

Try fullscreen mode if you have trouble arranging the terms



**6. True** or **false** - the Lagrangian formulation of the constrained optimization problem



**7. Mark** all of the following statements that are correct.

✅The dual objective is concave.

❌The solution to the dual D\* is always equal to the solution of the primal P\*

✅The *primal objective* is given by 

✅The *dual objective* is always less-than-or-equal-to the optimal value of the dual solution.

✅The *dual objective* is given by 

**8.** The *min-max inequality* implies that D\*=P\*

**True** or **False**?

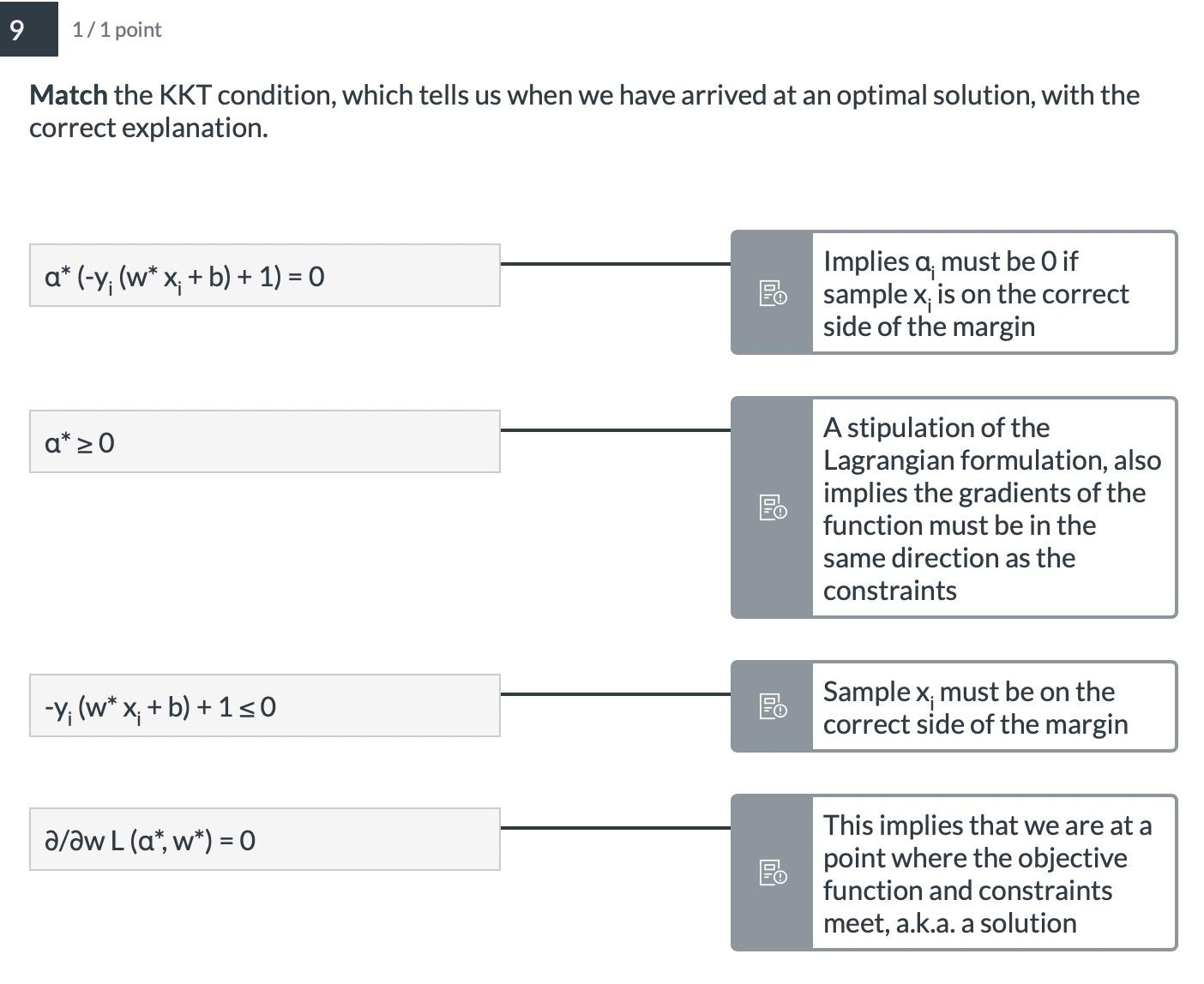
❌True

✅False

**9. Fill in** the following statement correctly.

The primal-dual formulation told us that the dual is a lower bound of the solution of the primal, or D\* ≤ P\*. For the SVM formulation, we have strong duality which states that D\* = P\*, and we can use the Karush Kuhn Tucker (KKT) conditions to check if the current solution is optimal (or nearly so).

**10.Match** the KKT condition, which tells us when we have arrived at an optimal solution, with the correct explanation.

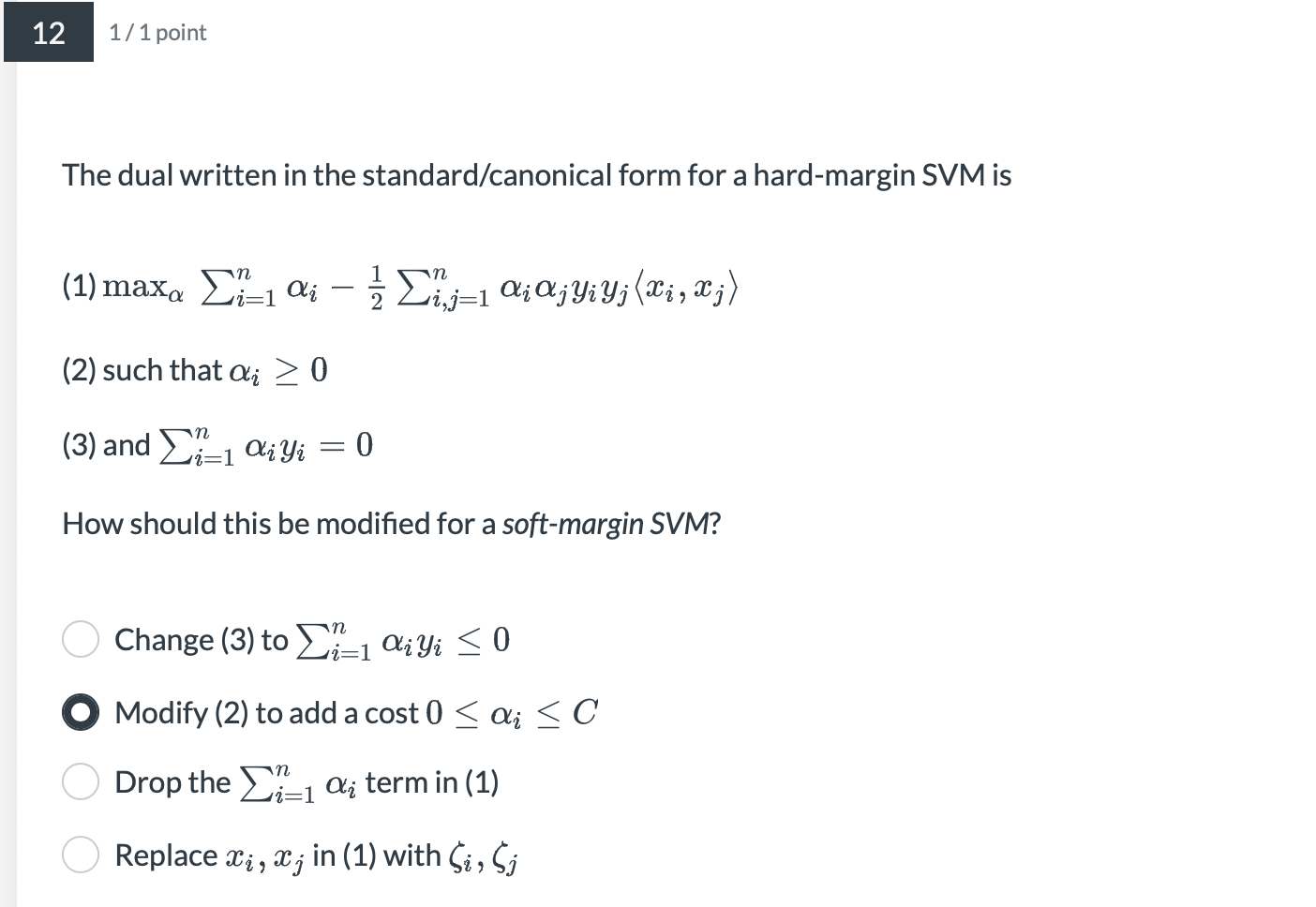


**11. True** or **false**? A soft-margin linear SVM can be optimized without using the Lagrangian dual formulation.

✅True

❌False

**12.** The dual written in the standard/canonical form for a hard-margin SVM is



**13.** To solve the dual problem, we applied *Sequential Minimal Optimization (SMO),* a type of *coordinate descent.*

**Mark** the following statements below that are correct.

❌To solve the dual, we can use SMO to freely traverse in any αi without any constraints

✅Coordinate descent methods break the problem down to individual dimensions and improve one dimension at at time

✅SMO searches for two dimensions {αi,αj} at each iteration

✅Because the dual is a concave function, we are actually performing coordinate ascent

**14.** Assume we have solved the dual problem with strong duality P\*=D\*. How do we find the values of the primal variables θ's?

**Mark** all of the following statements that are correct.

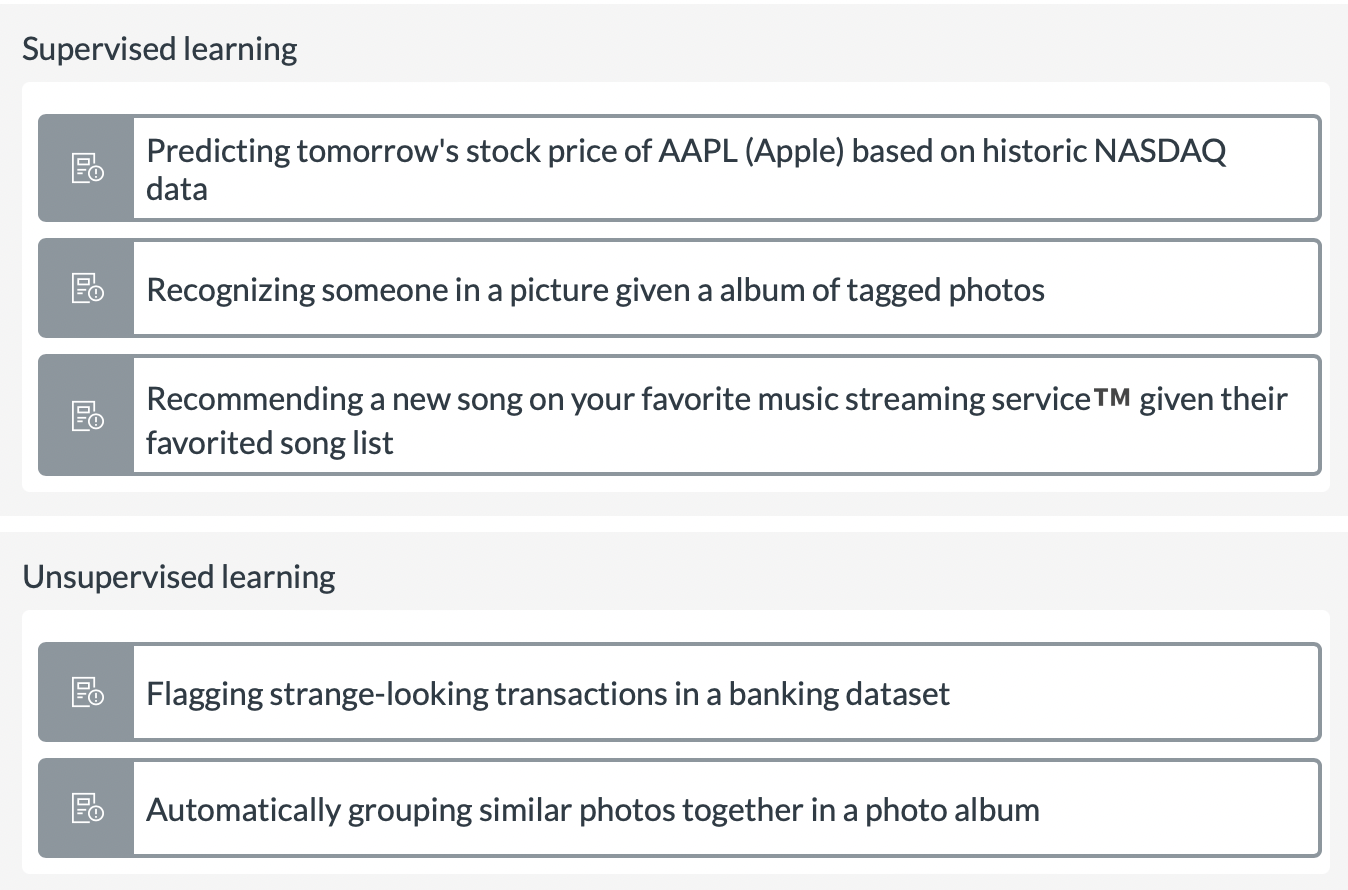
❌In ML, we only care about finding the value of P\*, so finding the θ's is unnecessary

✅The relationship between primal and dual variables θ's and α's can be found using the KKT conditions

❌At the solution α's and θ's will be the same

Part 2.6

**1. Group** the following examples into the most correct category.



**2. Fill in** the following statement with the correct terms.

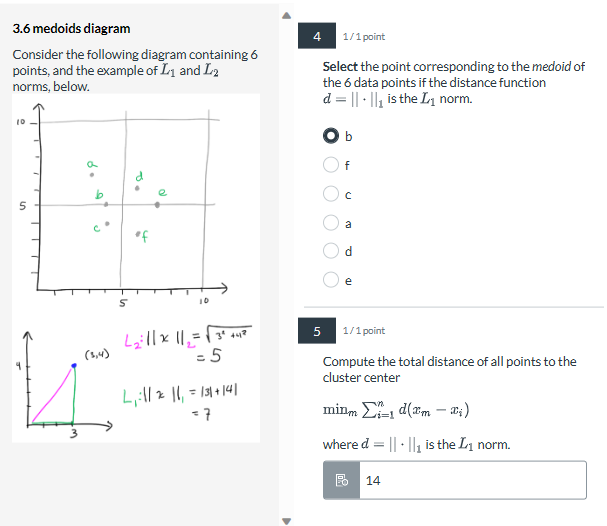
**Clustering** is an **unsupervised learning** task where the objective is to **group** a set of samples such that the samples within each group are **more similar** to one another than to the other groups. **k-medoids** is a ML model to solve such tasks.

**3.** Clustering is not a true *unsupervised learning* approach because, like *supervised* approaches, it predicts labels zi.

❌True

✅False

4和5

****

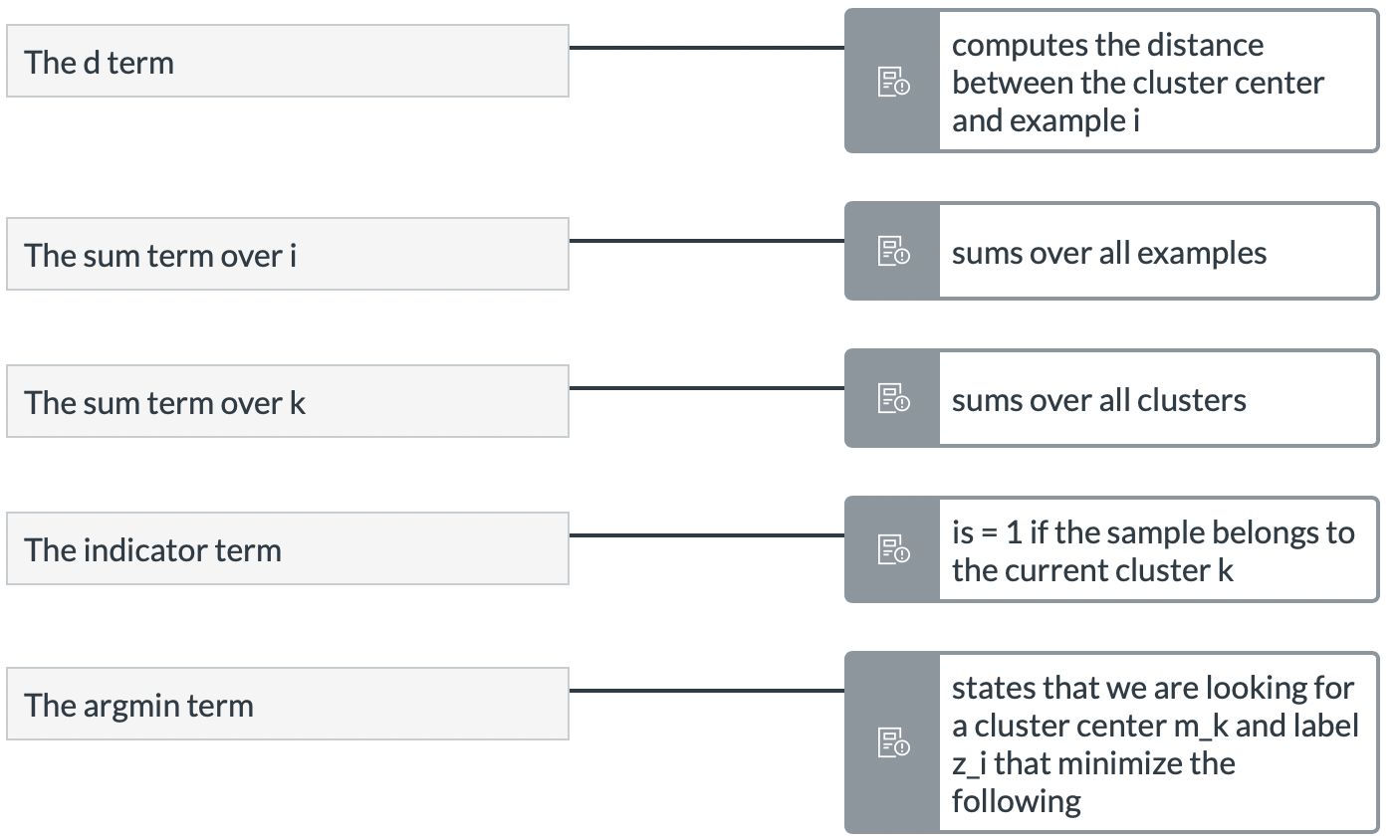
4. 找一圈点里中间的那个点

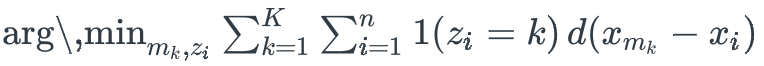
5. 首先要看是还是。

如果是就是每个点到中心点的x和y的距离加在一起，例如a到b的距离是竖2 +横1 =3

如果是就是每个点到中心点的距离的平方和在开方再加在一起， 例如a到b的距离是

**6.**The expression for the objective of *k-medoids* can be written as follows 

**Match** the following correctly.

**7.** The optimization problem for *k-medoids* clustering can be expressed as

**Mark** the following correct statements.

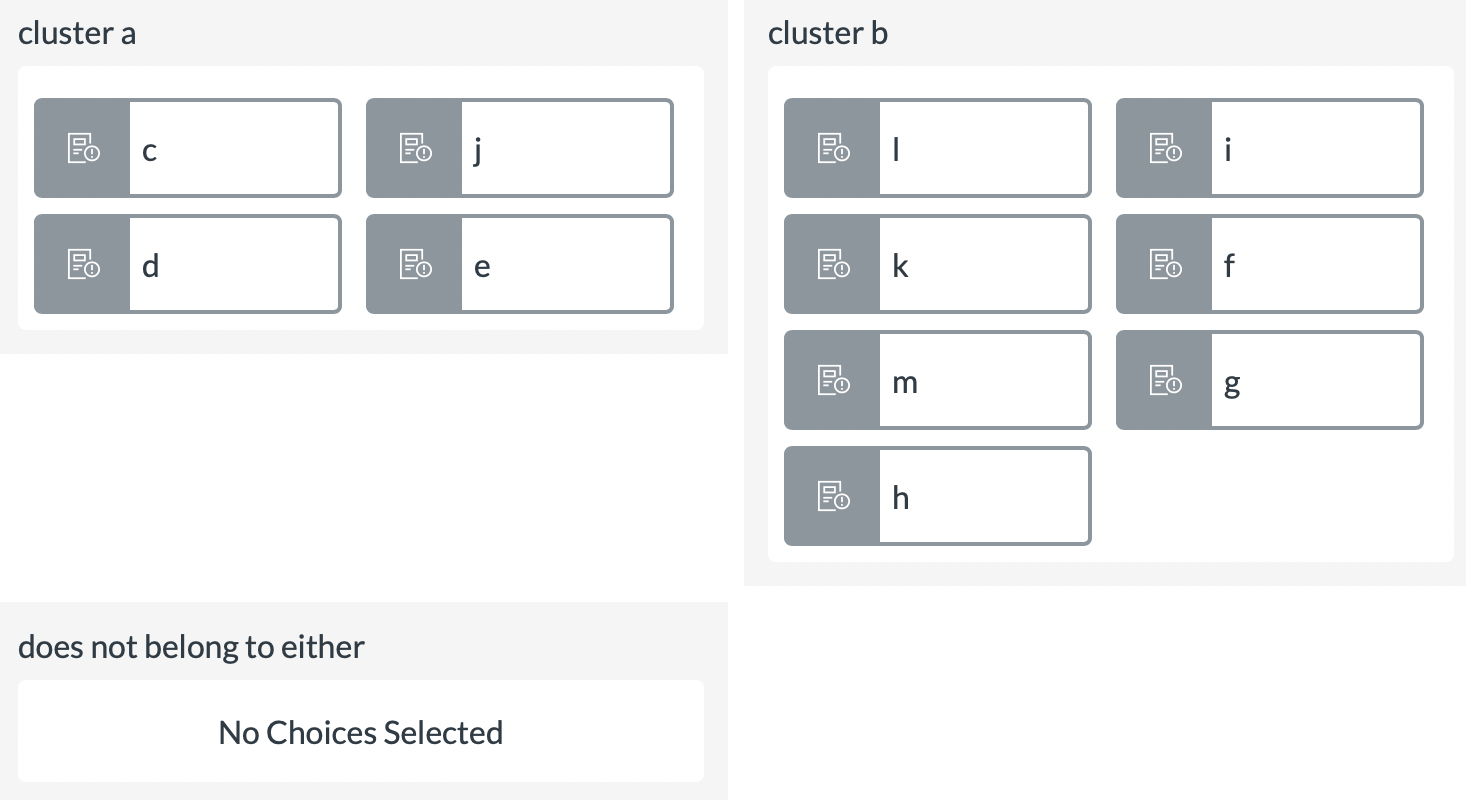
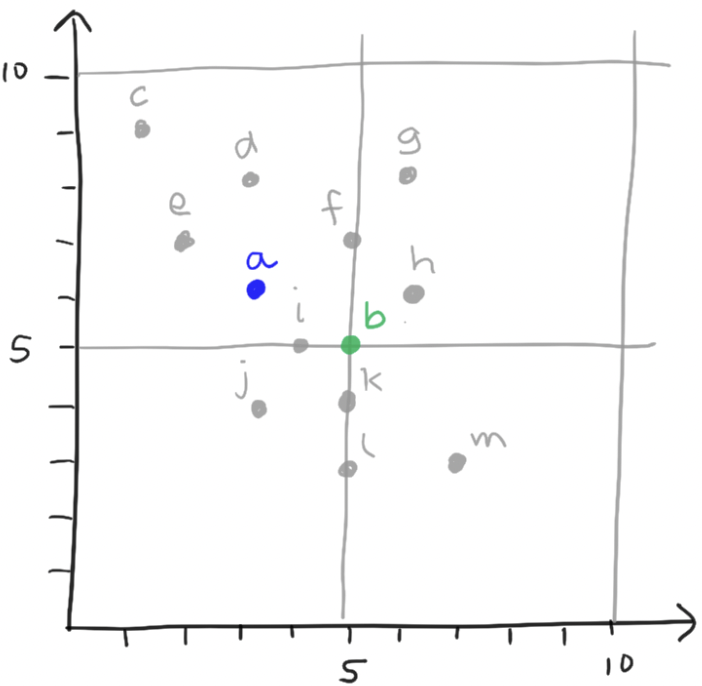
✅zi is discrete

✅mk is discrete

❌We cannot apply gradient descent, but we could use a subgradient method

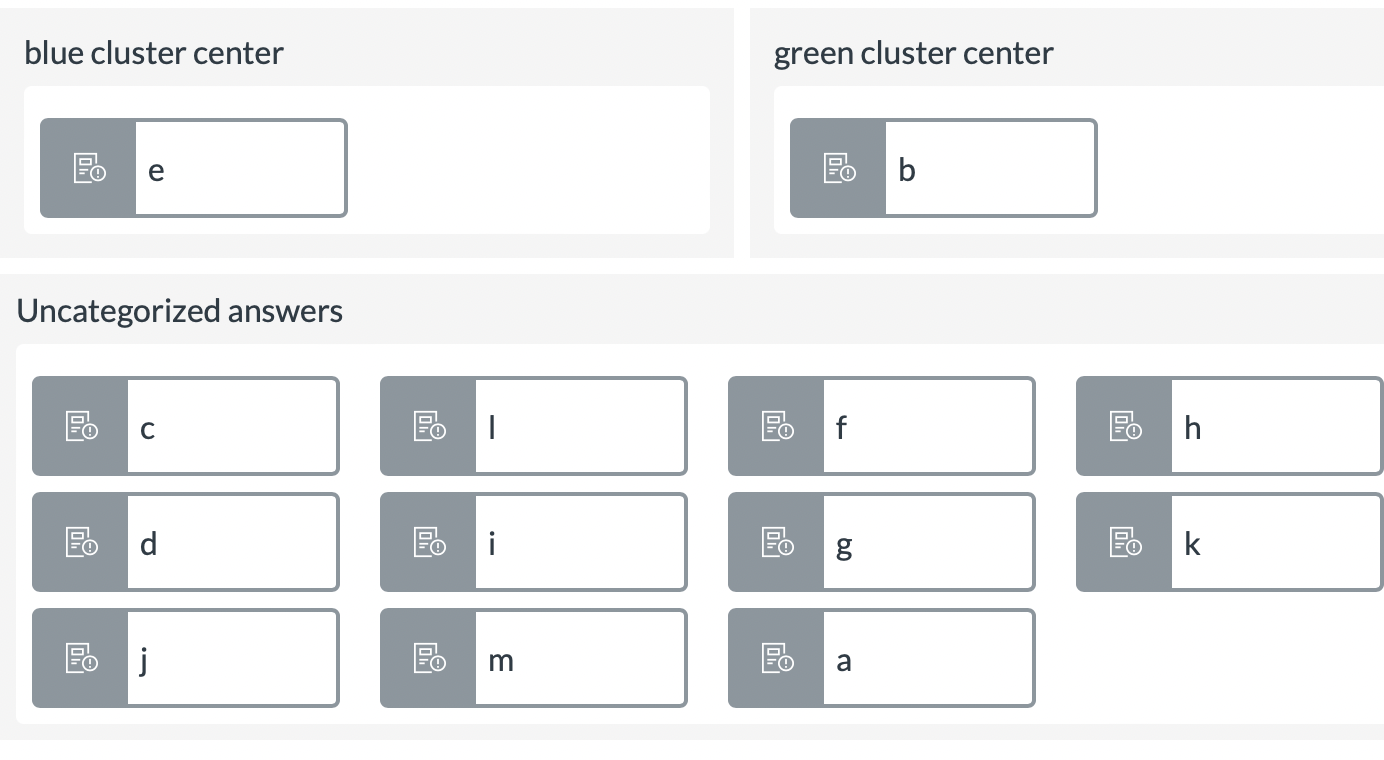
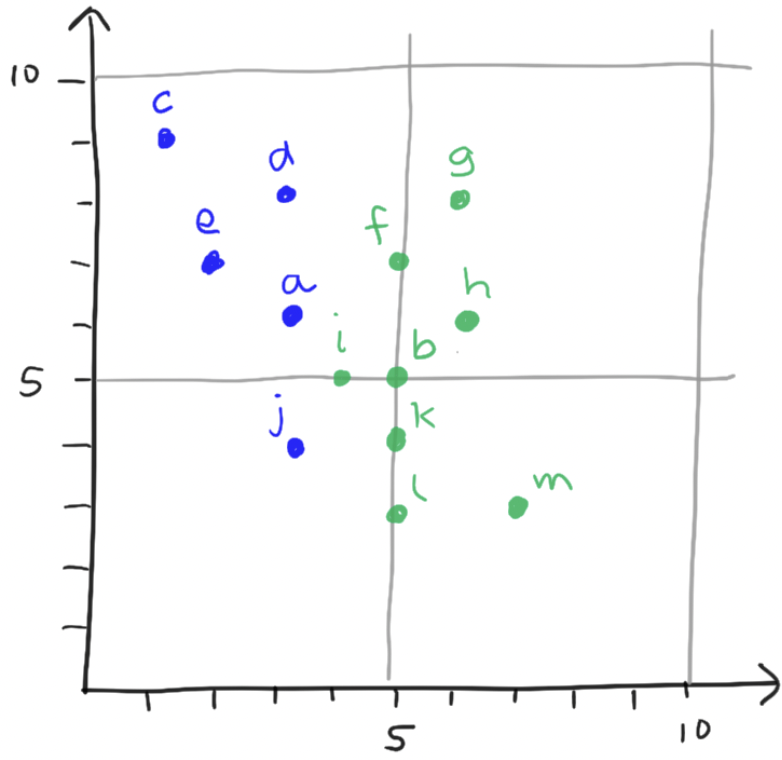
✅We can alternate between optimizing for mk and zi to find a solution for both

**8. Assign** each data point to the correct cluster center (k-medoids), using d=||⋅||1, the L1 norm.

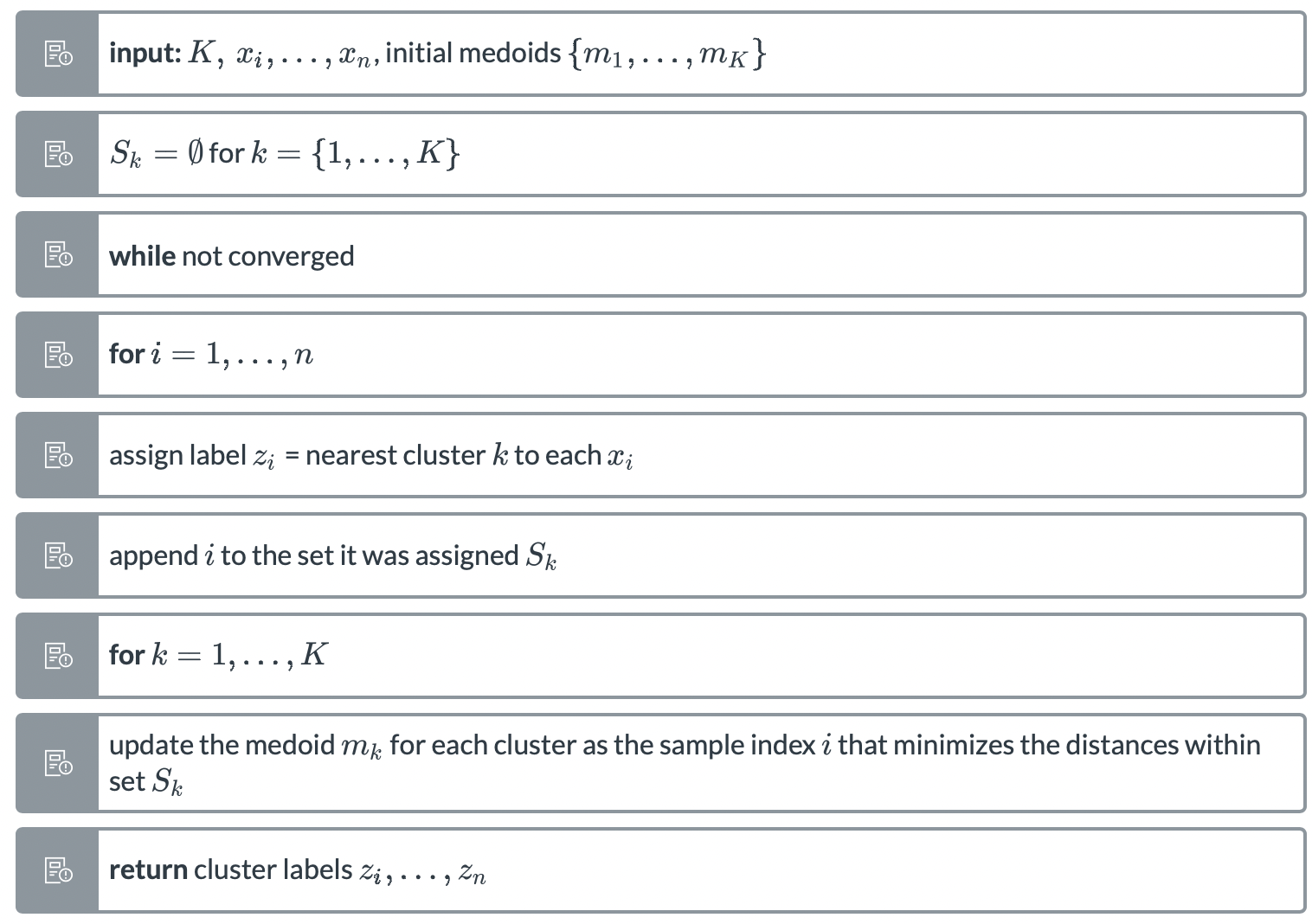


这里就是找离两个中心点距离最近的几个点

**9.Update** the cluster centers (k-medoids) given the assignments from the previous step in k-medoids. Use d=||⋅||1, the L1 norm, as the distance function.



在每个区域内找最中心的点

**10. Arrange** the following into the correct order for the *voronoi method for k-medoids*

**11.** Assume we have trained k-medoids on some dataset with k clusters. Training has stopped, and we now have predicted labels z1,…,zn .

✅k is a hyperparameter that we must choose

✅computing the sum of distances within each cluster is one valid way of measuring the quality of the clusters

❌z1,…,zn always represent the optimal solution

**12.***k-medoids* optimizes by alternating between two objectives using *update* and *assignment* steps. Several other ML methods use a similar alternating approach, notably *k-means* and *Expectation Maximization (EM).*

**True** or **false**?

✅True

❌False

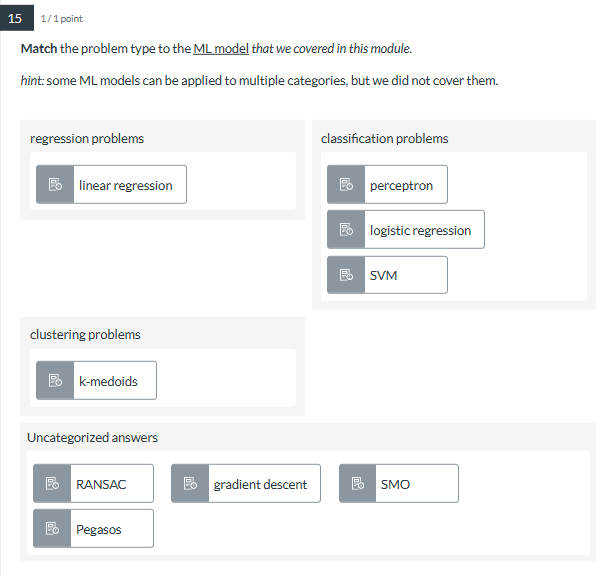
13. **Mark** all of the following correct statements.

* *maximum bipartite matching* is an example of a discrete optimization problem
* discrete optimization is the problem of finding a minimum/maximum of an objective function where the optimization variables are *discrete*
* *k-medoids* is a discrete optimization technique
* Alternating or partitional optimization methods like k-medoids and EM do not have the same convergence guarantees that gradient-based methods have
* discrete optimization problems do not often appear in ML
* *k-medoids* is similar in spirit to *coordinate ascent/descent* in that it partitions the problem and iteratively solves parts of the problem.

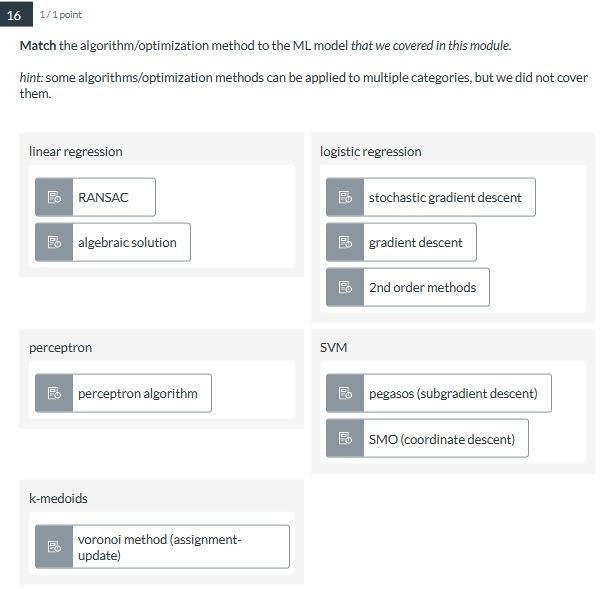
**14. *Fill in*** *the following statement correctly.*

From the perspective of this module, a real-world problem can be formulated as a function **h : x → y** which we aim to **approximate** with an ML model. The ML model defines a cost or **average loss** function **L** which quantifies the model's **mistakes**. The role of **optimization** is to search for the **parameters θ\*** that minimize L in a process that is often referred to as **training.**

*15.Match the problem type to the ML model that we covered in this module.*

**

*16.****Match*** *the algorithm/optimization method to the ML model that we covered in this module.*

**

**17**. Consider a linear regression problem with a loss function given bywhere the gram matrix is invertible, so we solve it algebraically.

**Mark** the following statements that correctly characterize this ML model.

✅The optimization problem it poses is continuous

✅The solution is deterministic given the data

❌The optimizer relies on 1st or 2nd order gradients

✅It is a linear model

✅The optimization problem is convex

✅Its predictions are linear w.r.t. the input

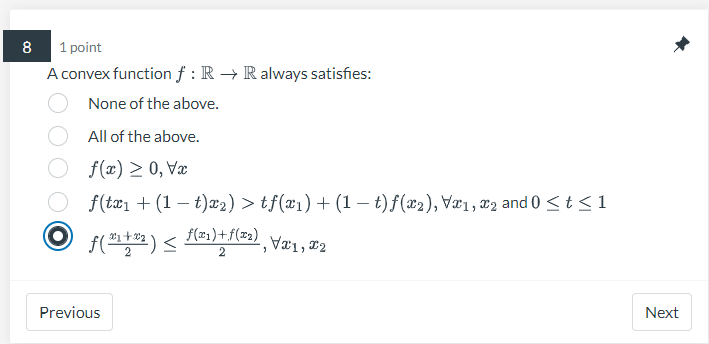
❌The solution is constrained

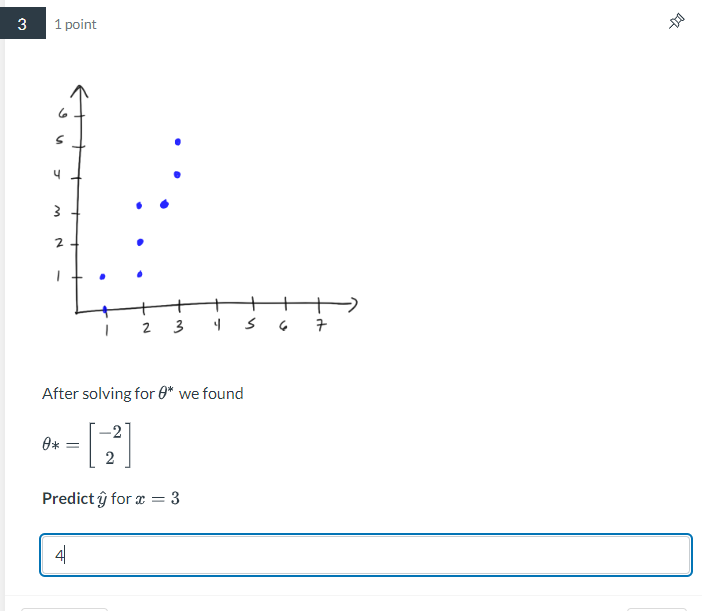
*18. Consider a linear soft-margin SVM problem with a loss function given by*

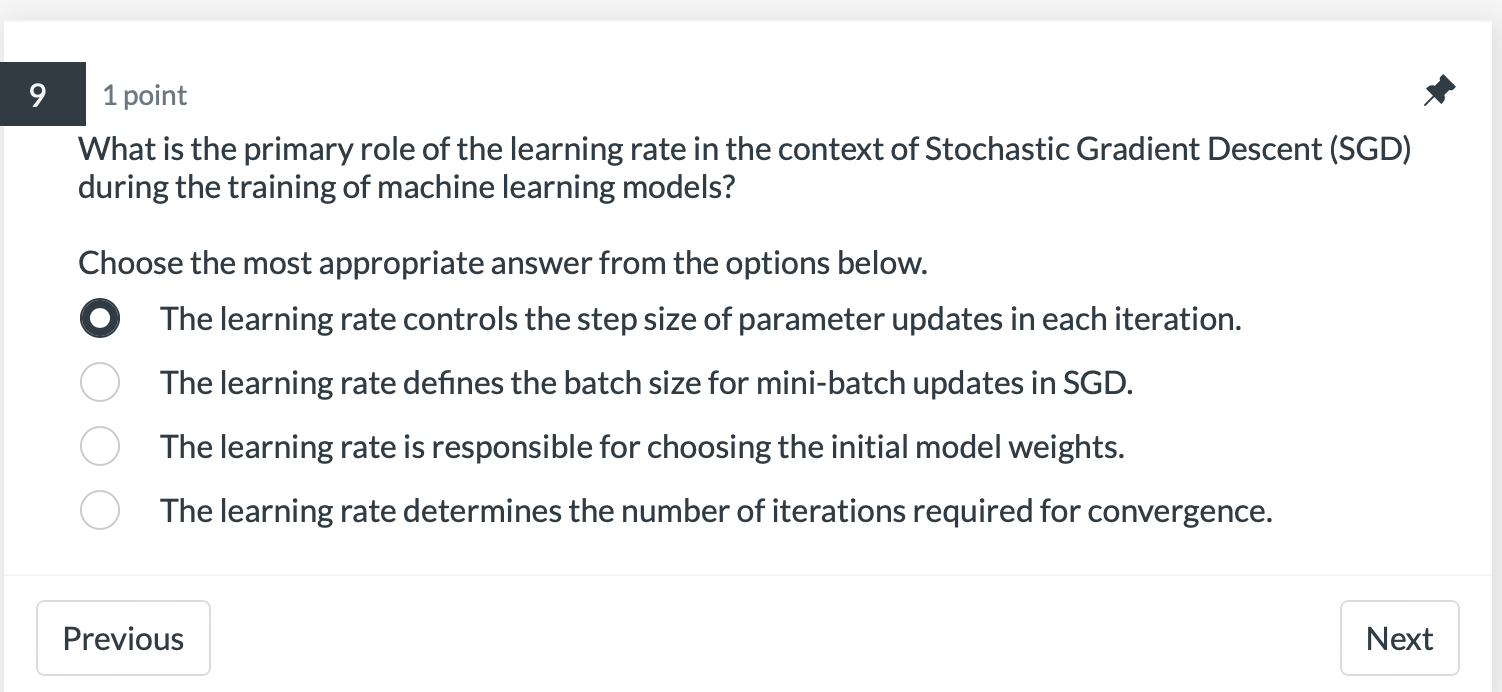
*where we apply the Pegasos subgradient descent method to train the model.****Mark*** *the following statements that correctly characterize this ML model.*

* The optimizer relies on 1st or 2nd order gradients
* The solution is constrained
* The optimization problem it poses is continuous
* It relies on a convexity assumption
* It is a linear model
* Its predictions are linear w.r.t. the input

—------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------从这里往下贴\_—----------







meixuan bu zhidao 