CS 360: Machine Learning

Prof. Sara Mathieson

Fall 2019

HAVERFORD COLLEGE

Admin

- Office hours **today 12-1pm** (in lab)
- Lab 7 due Sunday Nov 10
- No class Tuesday Nov 12 (away at conference)
  - time for project proposal!
- Project proposal due Wednesday Nov 13
- Lab 8 due Sunday Nov 17
- Midterm 2: Nov 21 (in-class) + take home due Tues Nov 26
- Nov 28-29: Thanksgiving break!

Outline for November 5

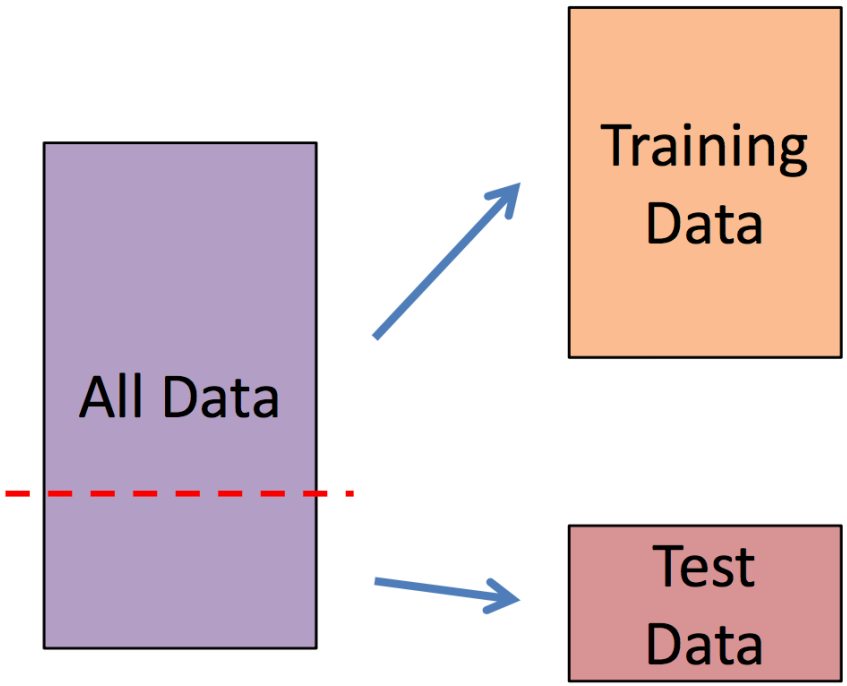
- Cross-validation
- Recap Lagrange multipliers
- Lagrangian for SVMs
- Extensions of SVMs
- Solving the SVM optimization problem

- Lab check in Thursday! (either problem set or coding complete)
- Reading quiz Thursday: on material from today

Outline for November 5

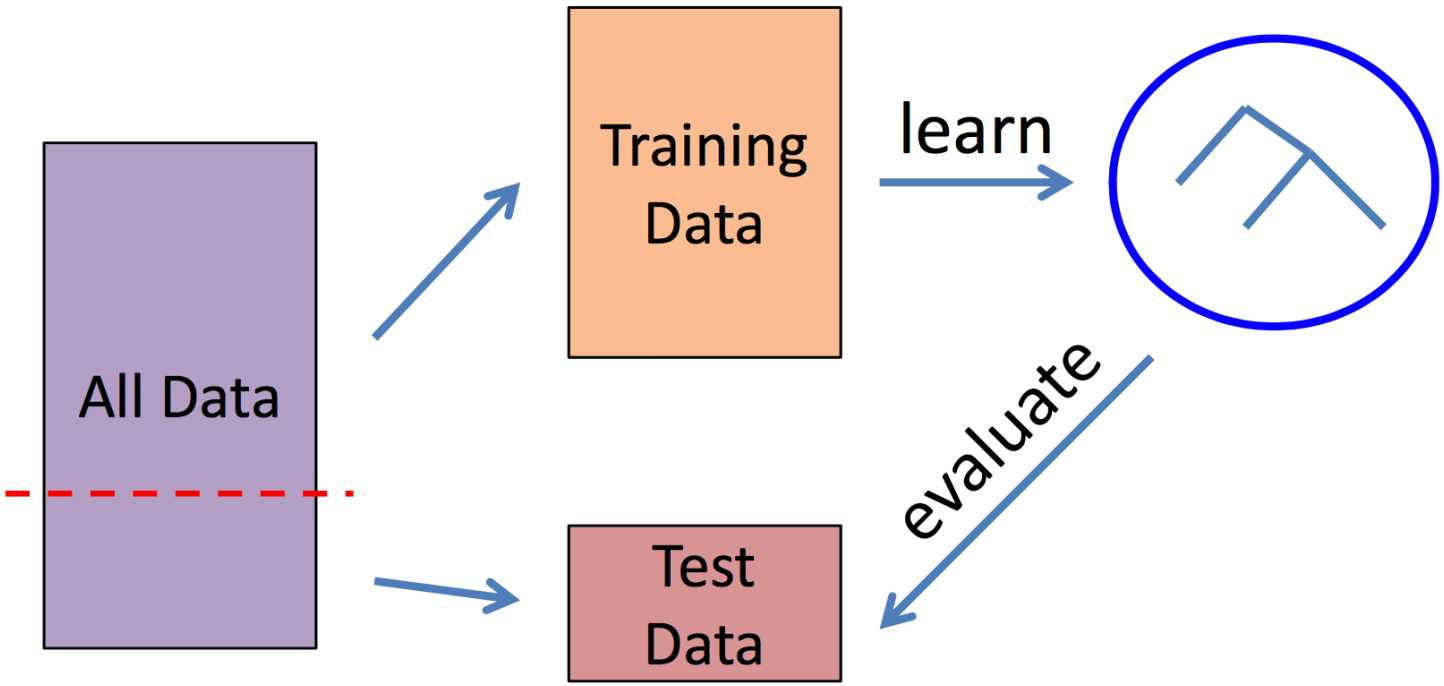
- Cross-validation
- Recap Lagrange multipliers
- Lagrangian for SVMs
- Extensions of SVMs
- Solving the SVM optimization problem

Evaluation in Practice



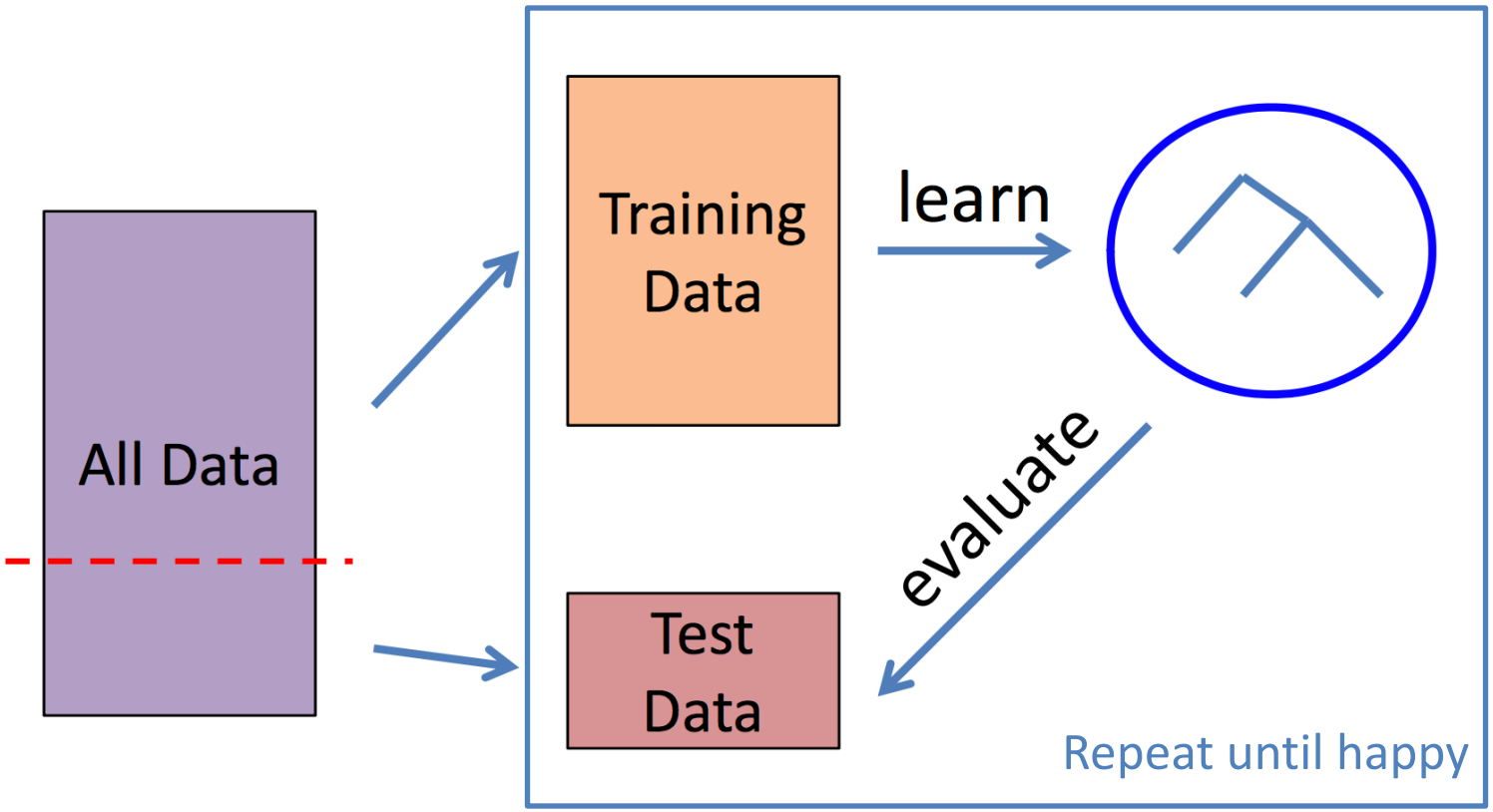
Modified from Jessica Wu

Evaluation in Practice



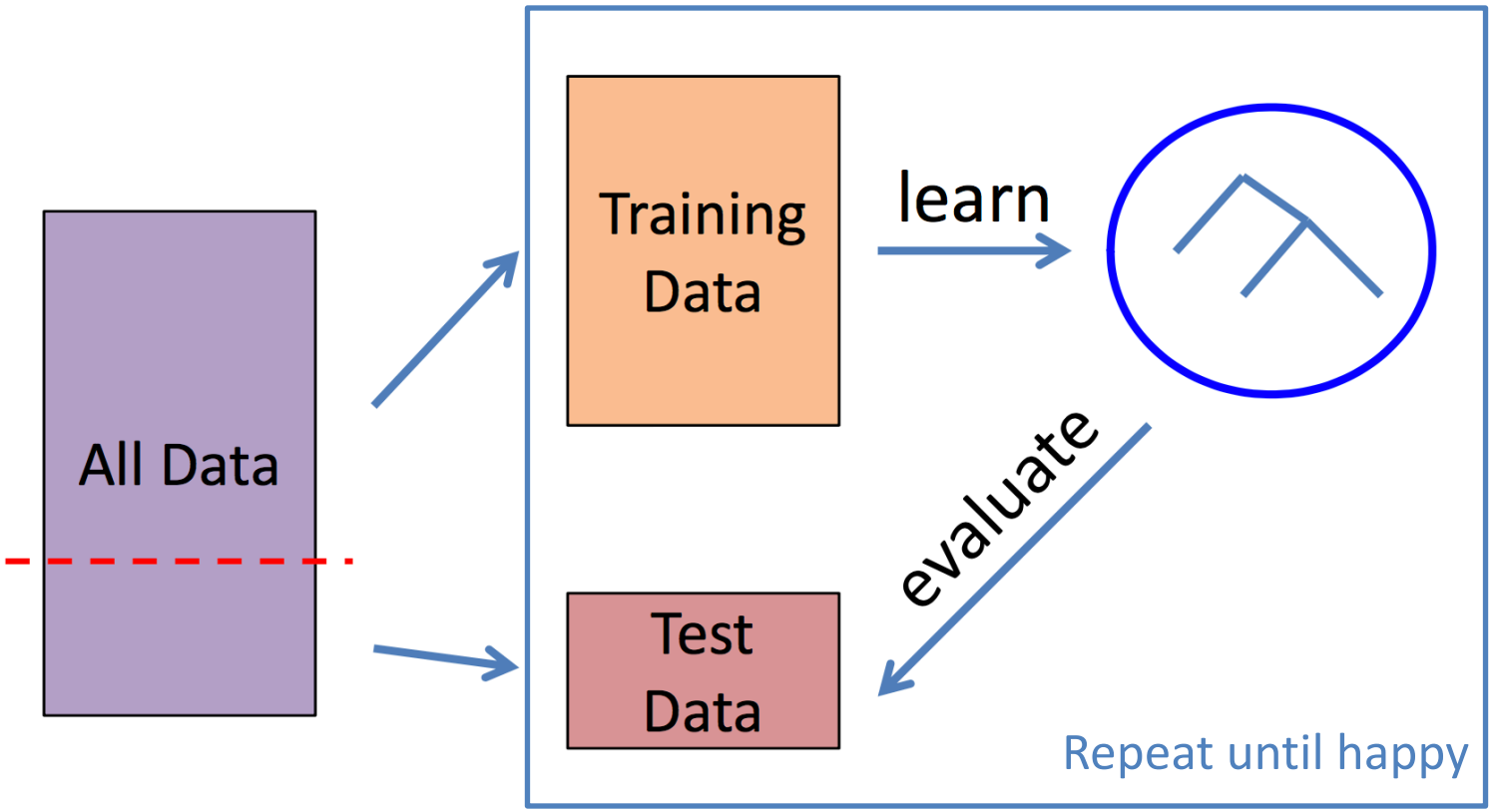
Modified from Jessica Wu

Evaluation in Practice



Modified from Jessica Wu

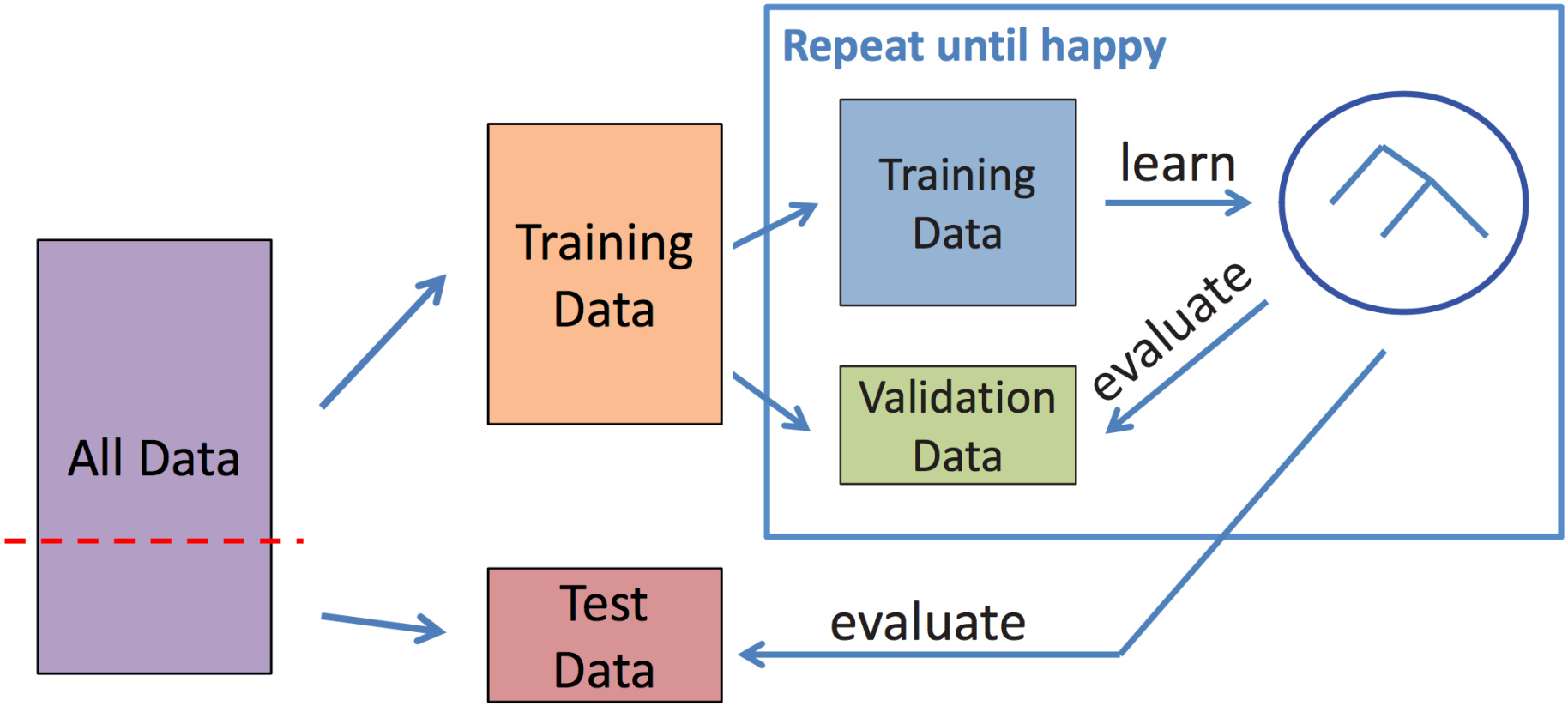
Evaluation in Practice



NO! Using test data as part of the model selection process

Modified from Jessica Wu

Better: use a *validation* dataset



Modified from Jessica Wu

k-fold Cross Validation

- Why just choose one particular "split" of data?
  - in principle, we should do this multiple times since performance may be different for each split

Slide: Jessica Wu

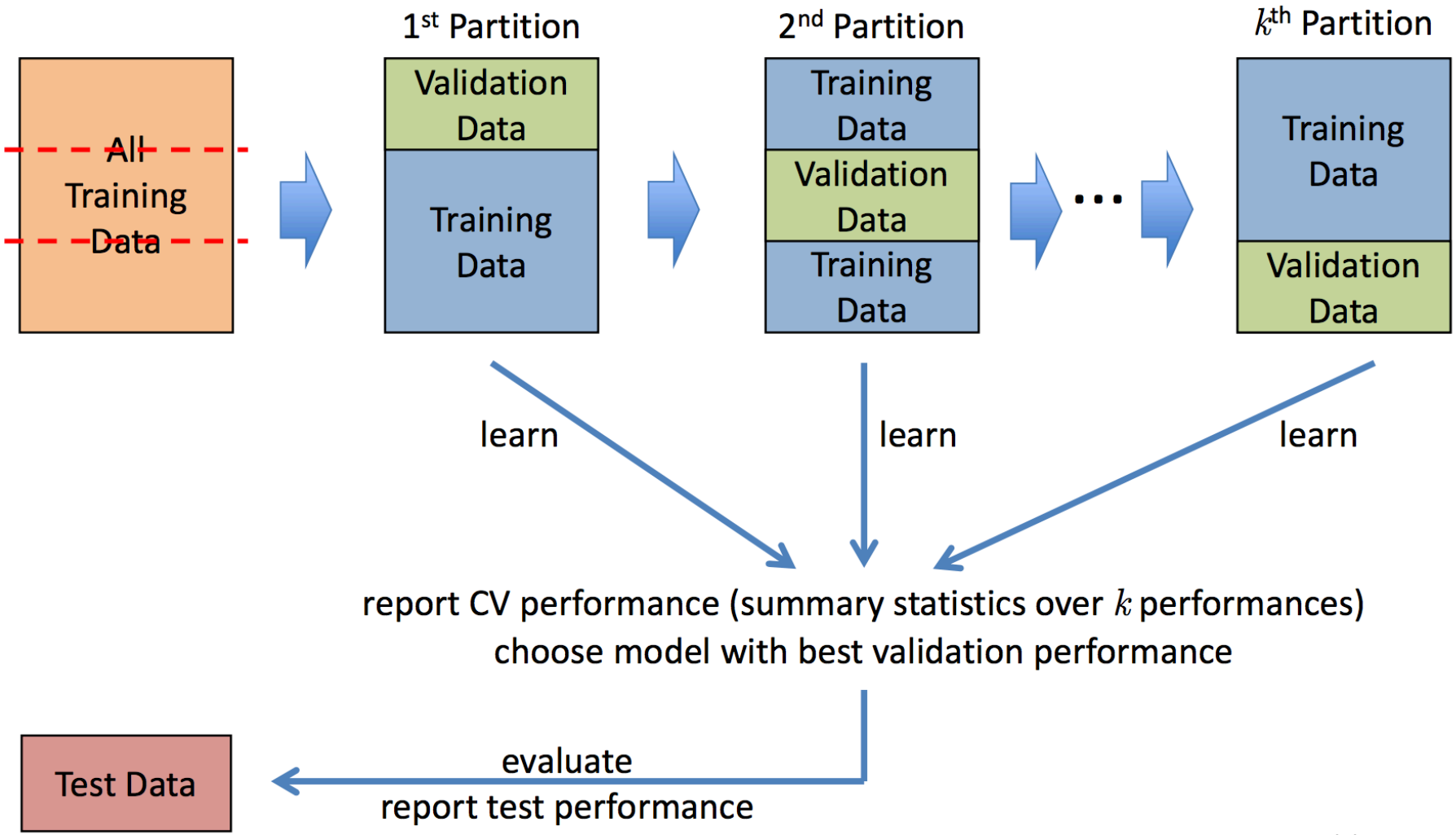
k-fold Cross Validation

- Why just choose one particular "split" of data?
  - in principle, we should do this multiple times since performance may be different for each split

- k-Fold Cross-Validation (e.g., $k = 10$)
  - randomly partition full data set of n instances into **k disjoint subsets** (each roughly of size n/k)
  - choose each fold in turn as validation set; train model on the other $k-1$ folds and evaluate
  - compute statistics over k test performances, or choose best of k models

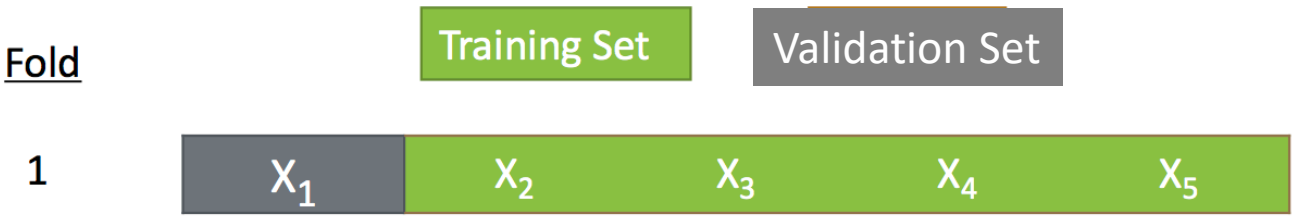
Slide: Jessica Wu

k-fold Cross Validation



Slide: Jessica Wu

k-fold Cross Validation



Evaluation

Accuracy(P_1) = 11/20

Slide: Ameet Soni

k-fold Cross Validation

Evaluation

Accuracy(P_1) = 11/20

Accuracy(P_2) = 17/20

Accuracy(P_3) = 16/20

Accuracy(P_4) = 13/20

Accuracy(P_5) = 16/20

Slide: Ameet Soni

k-fold Cross Validation

Evaluation

Accuracy(P_1) = 11/20

Accuracy(P_2) = 17/20

Accuracy(P_3) = 16/20

Accuracy(P_4) = 13/20

Accuracy(P_5) = 16/20

Generalization: average accuracy across all folds = 73/100 = 73%

Slide: Ameet Soni

Discussion

1) What are the costs of *k*-fold cross validation?

2) Pros and cons of no longer having one model?

3) How to choose *k*?

Discussion

1) What are the costs of *k*-fold cross validation?

    - Computational, especially if training takes a long time

2) Pros and cons of no longer having one model?

    - Con: might be hard to interpret
    - Pro: might be able to average results

3) How to choose *k*?

    - Large *k* can be good for small datasets (i.e. where *n* is small)
    - Tradeoff between computation and reducing variance
    - Many choose *k*=10 in practice :)

Cross Validation: other considerations

- Can use cross-validation to choose hyper parameters

- Leave-one-out cross validation (LOOCV)
  - Special case of $k=n$
  - Train using $n-1$ examples, evaluate on remaining
  - Repeat n times

- Can do multiple trials of CV

The Short Way
(that Many People Actually Use)

- Split into only training data + validation data
- Train on training data, evaluate on validation data
- Report cross-validation performance
  - possibly also training performance

- Why is this used?
  - might not be enough data to create held-out test set
  - you cannot trust that authors did not peek at test data anyway =P

Slide: Jessica Wu

Outline for November 5

- Cross-validation
- Recap Lagrange multipliers
- Lagrangian for SVMs
- Extensions of SVMs
- Solving the SVM optimization problem

Functional and Geometric Margins

SVM classifier:
(same as perceptron)

$$h(\vec{x}) = \text{sign}\left(\vec{w} \cdot \vec{x} + b\right)$$

Functional and Geometric Margins

SVM classifier:
(same as perceptron)

$$h(\vec{x}) = \text{sign}\left(\vec{w} \cdot \vec{x} + b\right)$$

Functional Margin:

$$\hat{\gamma}_i = y_i(\vec{w} \cdot \vec{x}_i + b)$$

Functional and Geometric Margins

SVM classifier:
(same as perceptron)

$$h(\vec{x}) = \text{sign}\left(\vec{w} \cdot \vec{x} + b\right)$$

Functional Margin:

$$\hat{\gamma}_i = y_i(\vec{w} \cdot \vec{x}_i + b)$$

Geometric Margin:
(distance between example and hyperplane)

$$\gamma_i = y_i \left(\frac{\vec{w}}{\|\vec{w}\|} \cdot \vec{x}_i + \frac{b}{\|\vec{w}\|} \right)$$

Functional and Geometric Margins

SVM classifier:
(same as perceptron)

$$h(\vec{x}) = \text{sign}\left(\vec{w} \cdot \vec{x} + b\right)$$

Functional Margin:

$$\hat{\gamma}_i = y_i(\vec{w} \cdot \vec{x}_i + b)$$

Geometric Margin:
(distance between example and hyperplane)

$$\gamma_i = y_i \left(\frac{\vec{w}}{\|\vec{w}\|} \cdot \vec{x}_i + \frac{b}{\|\vec{w}\|} \right)$$

Note:

$$\boxed{\gamma_i = \frac{\hat{\gamma}_i}{\|\vec{w}\|}}$$

Optimization Problem: try 3

Idea: put arbitrary constraint on functional margin

$$\boxed{\hat{\gamma} = 1}$$

$$\min_{\vec{w},b} \quad \frac{1}{2}\|\vec{w}\|^2$$

$$\text{s.t.} \quad y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1, \quad i = 1, \cdots, n$$

Optimization Problem: try 3

Idea: put arbitrary constraint on functional margin

$$\boxed{\hat{\gamma} = 1}$$

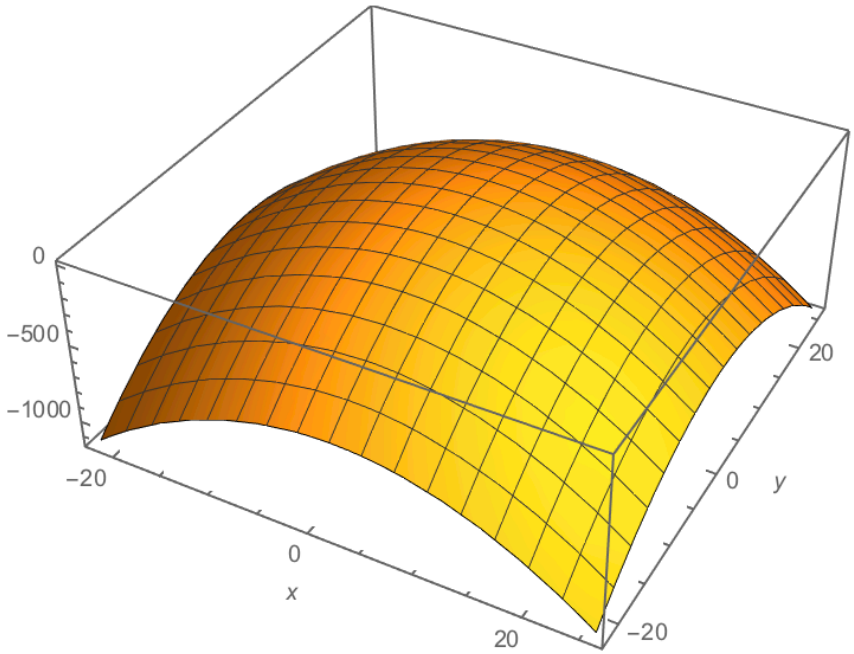
$$\min_{\vec{w},b} \quad \frac{1}{2}\|\vec{w}\|^2$$

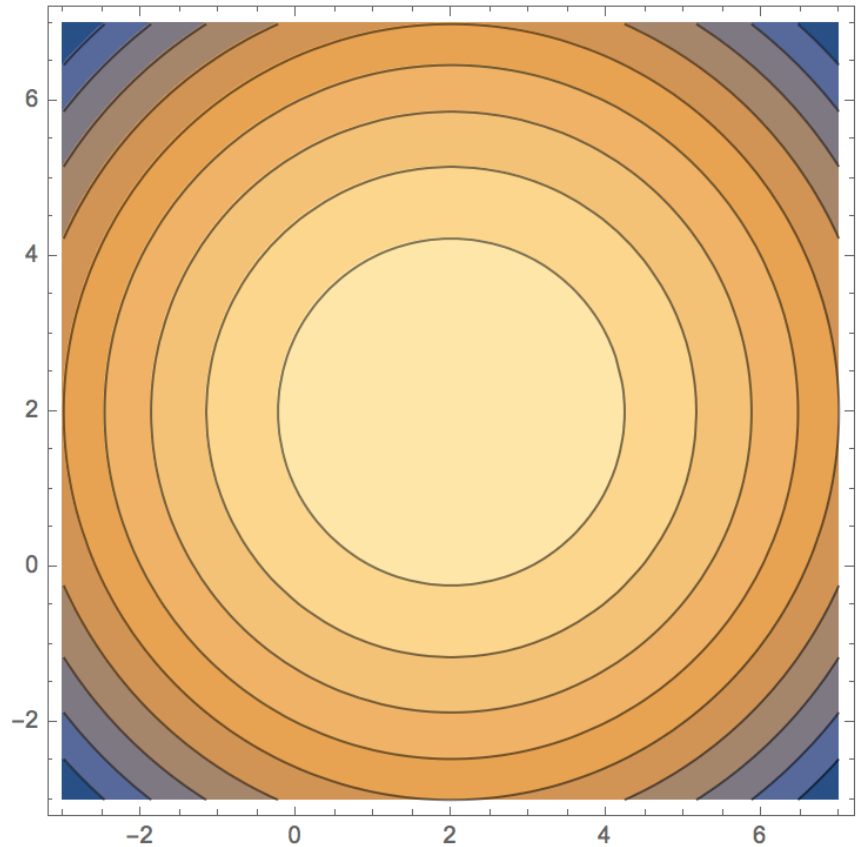
$$\text{s.t.} \quad y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1, \quad i = 1, \cdots, n$$

$$\boxed{\begin{aligned} &\min_{\vec{w},b} \quad \frac{1}{2}\|\vec{w}\|^2 \\ &\text{s.t.} \qquad y_i(\vec{w} \cdot \vec{x}_i + b) - 1 \geq 0, \quad i = 1, \cdots, n \end{aligned}}$$

Lagrange multipliers example 1

$$f(x,y) = 5 - (x-2)^2 - (y-2)^2$$





Contour plot of *f*(*x*,*y*)

$$\text{maximize}_{x,y} \quad f(x,y)$$

$$s.t. \quad g(x,y) = 0$$

$$g(x,y) = -5 + x + y$$

SVMs so far

$\min_{\vec{w},b}$	$\frac{1}{2}\|\vec{w}\|^2$
s.t.	$y_i(\vec{w}\cdot\vec{x}_i+b)-1\geq 0$

$i=1\cdots n$

"$f(x,y)$" function (regularization)

points away from margin

"$g(x,y)=0$"

$$\max f(x,y) = 5-(x-2)^2-(y-2)^2$$

$$\text{s.t. } g(x,y) = -5+x+y = 0$$

Lagrangian

Lagrange multiplier

$$\max L(x,y,\lambda) = f(x,y) - \lambda g(x,y)$$

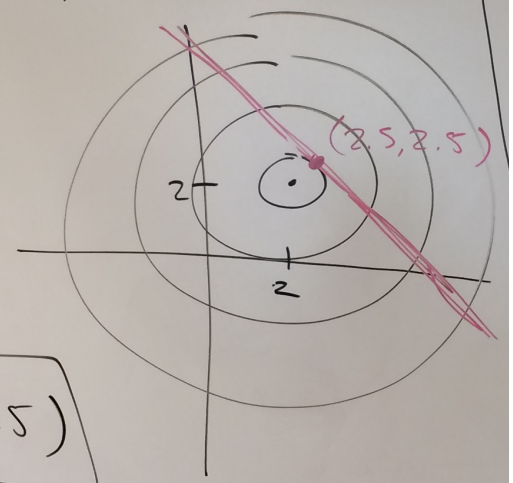
$$\nabla L(x,y,\lambda) = \begin{bmatrix} \frac{\partial f}{\partial x} - \lambda \frac{\partial g}{\partial x} \\ \frac{\partial f}{\partial y} - \lambda \frac{\partial g}{\partial y} \\ g(x,y) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\nabla f(x,y) = \lambda \nabla g(x,y)$$

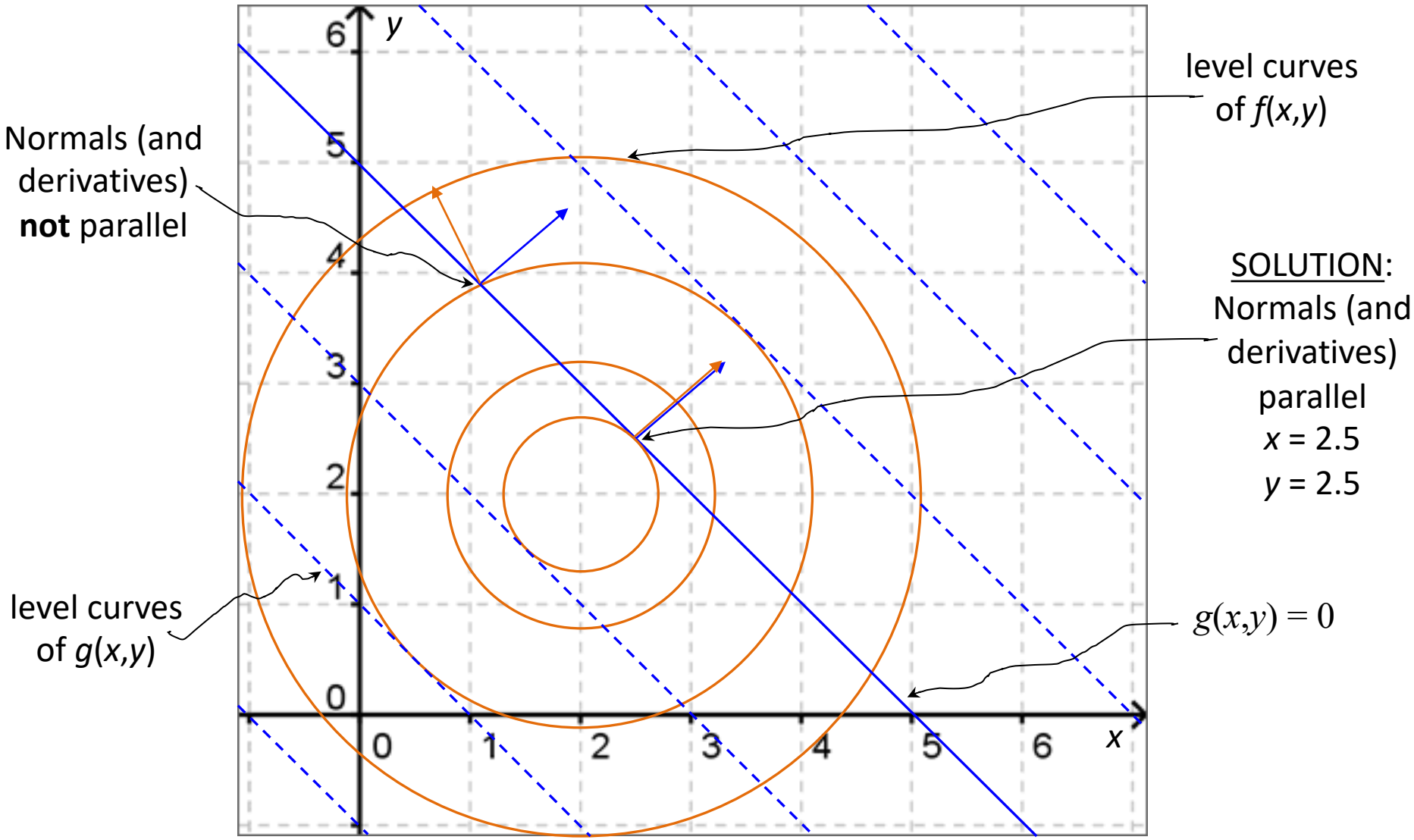
3 equations & $\underbrace{3 \text{ unknowns}}_{x, y, \lambda}$!

$$-2(x-2) - \lambda = 0$$
$$-2(y-2) - \lambda = 0$$
$$-5 + x + y = 0$$
$$x = 5 - y$$

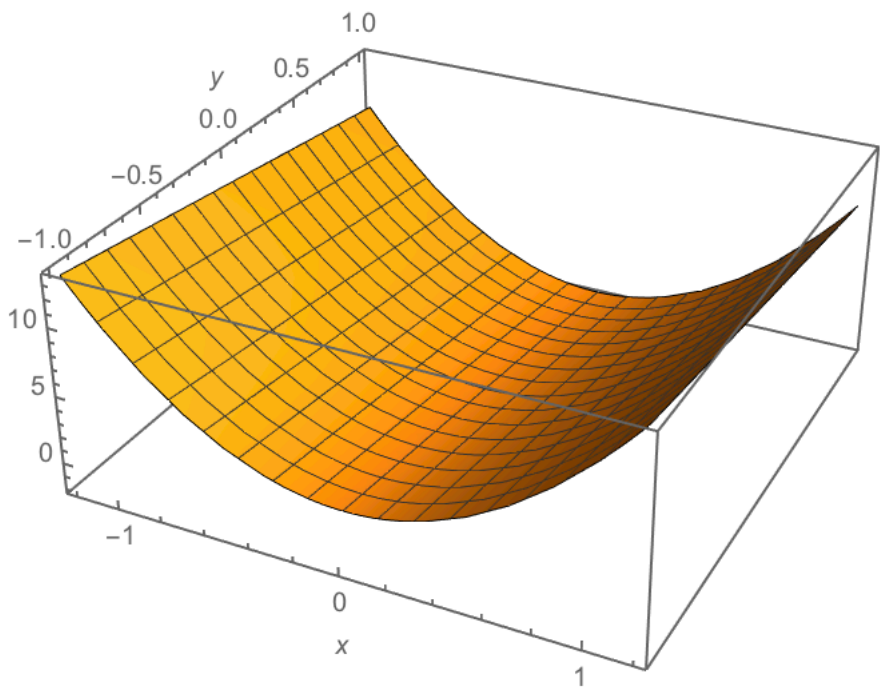
$$\Rightarrow \boxed{\begin{array}{l} (x, y) = (2.5, 2.5) \\ \lambda = -1 \end{array}}$$

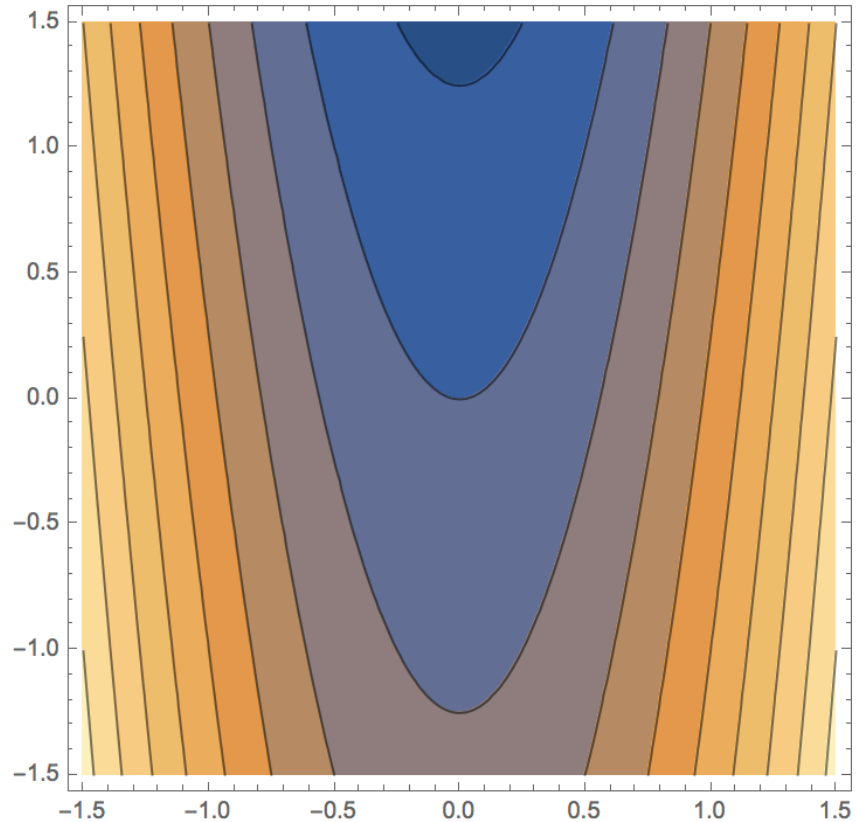


Lagrange multipliers example 1



Lagrange multipliers example 2



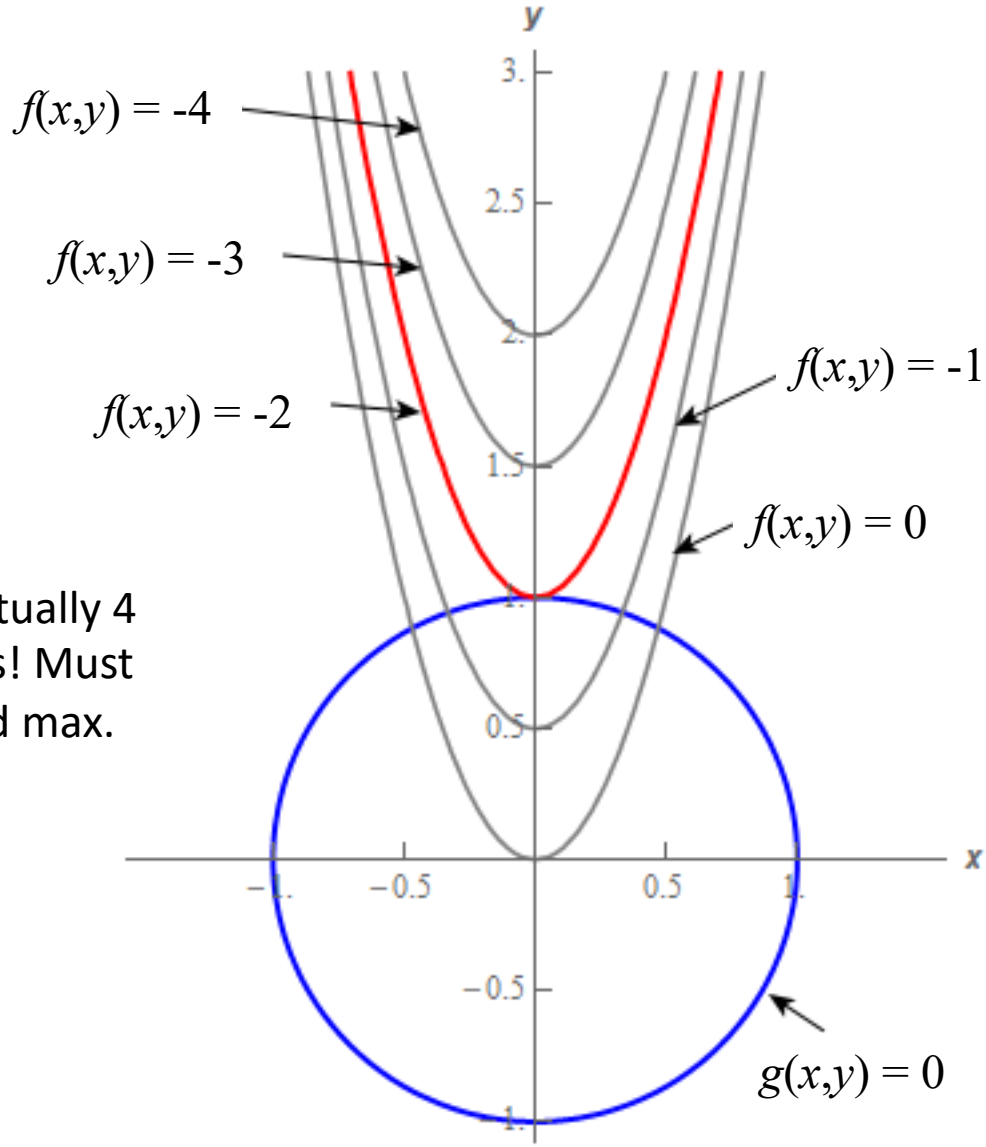


Contour plot of f(x,y)

$$\text{maximize}_{x,y} \quad f(x,y)$$

$$s.t. \quad g(x,y) = 0$$

Lagrange multipliers example 2



Note: there are actually 4 potential solutions! Must plug in to *f* to find max.

Outline for November 5

- Cross-validation
- Recap Lagrange multipliers
- Lagrangian for SVMs
- Extensions of SVMs
- Solving the SVM optimization problem

Back to SVMs

$$h(\underbrace{\vec{w}, b}_{X \,\&\, Y}, \underbrace{\vec{\alpha}}_{\lambda}) = \frac{1}{2}\|\vec{w}\|^2 - \sum_{i=1}^{n} \alpha_i \Big[\underbrace{y_i(\vec{w} \cdot \vec{x}_i +}_{\text{positive}}$$

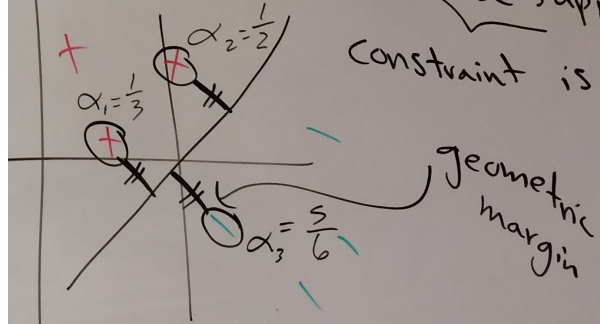
minimize → maximize

Lagrange multipliers

$\alpha_i = 0$ if x_i is not a support vector.

$\alpha_i > 0$ if x_i is a support vector

constraint is $\boxed{\text{active}}$!



geometric margin

b)-1]

$$\nabla_{\vec{w}} L(\vec{w}, b, \vec{\alpha}) = \vec{w} - \sum_{i=1}^{n} \alpha_i y_i \vec{x}_i = 0$$

min wrt $\vec{w}$ + b

$$\Rightarrow \boxed{\vec{w} = \sum_{i=1}^{n} \alpha_i y_i \vec{x}_i}$$

only depends on S.V..

$$\frac{\partial L(\vec{w}, b, \vec{\alpha})}{\partial b} = \boxed{\sum_{i=1}^{n} \alpha_i y_i = 0} \Rightarrow \boxed{\sum_{i: y_i = 1} \alpha_i = \sum_{i: y_i = -1} \alpha_i}$$

"dual"

no $\vec{w}$ or b

$$\max W(\vec{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j (\vec{x}_i \cdot \vec{x}_j)$$

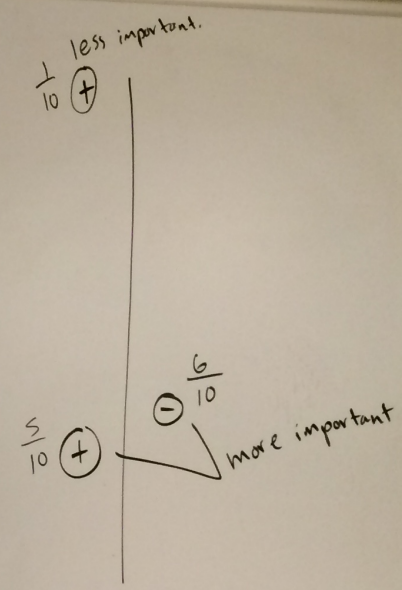
optimize using coordinate ascent.

Whatever I want

s.t. $\alpha_i \geq 0 \ \forall i$

& $\sum_{i=1}^{n} \alpha_i y_i = 0$

exercise



Outline for November 5

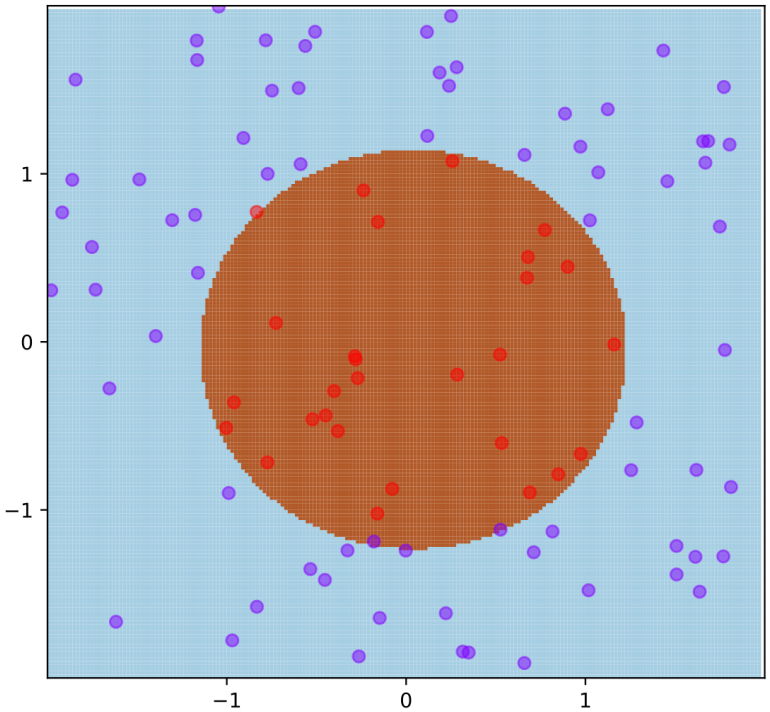
- Cross-validation
- Recap Lagrange multipliers
- Lagrangian for SVMs
- Extensions of SVMs
- Solving the SVM optimization problem

Kernel Idea

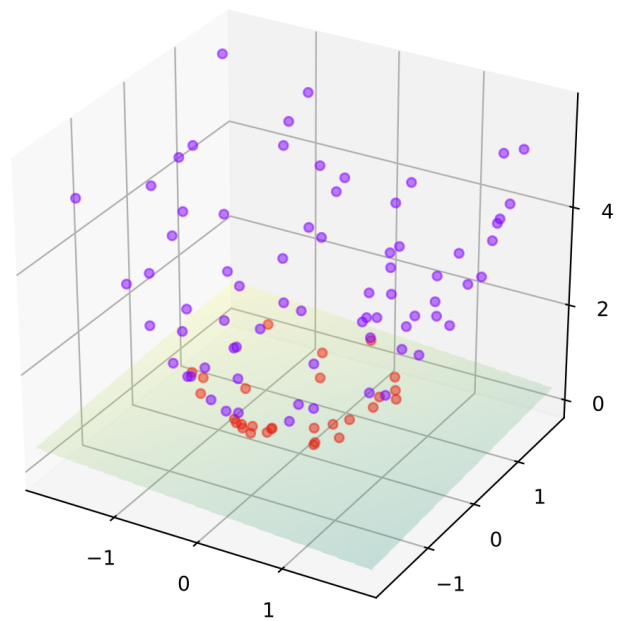
- By solving the dual form of the problem, we have seen how all computations can be done in terms of inner products between examples
- One example of an inner product is the dot product, which is the linear version of SVMs
- But there are many others!
- Intuition: if points are close together, their kernel function will have a large value (measure of similarity)

Kernel Trick example

Feature mapping: $\varphi(\boldsymbol{x}) = (x_1,\ x_2,\ x_1^2 + x_2^2)$



Original feature space



Mapping after applying kernel
(can now find a hyperplane)

Kernel function: $K(\boldsymbol{x},\ \boldsymbol{z}) = \boldsymbol{x} \bullet \boldsymbol{z} + ||\boldsymbol{x}||^2\ ||\boldsymbol{z}||^2$

Image: Shiyu Ji (wikipedia)

Gaussian Kernel

- Gaussian kernel is near 0 when points are far apart and near 1 when they are similar
- Also called Radial Basis Function (RBF) kernel

$$K(\vec{x}, \vec{z}) = \exp\left(-\frac{\|\vec{x} - \vec{z}\|^2}{2\sigma^2}\right)$$

Gaussian Kernel

- Gaussian kernel is near 0 when points are far apart and near 1 when they are similar
- Also called Radial Basis Function (RBF) kernel

$$K(\vec{x}, \vec{z}) = \exp\left(-\frac{\|\vec{x} - \vec{z}\|^2}{2\sigma^2}\right)$$

Often re-parametrized by gamma (different gamma!)

$$\boxed{\gamma = \frac{1}{2\sigma^2}}$$

$$K(\vec{x}, \vec{z}) = \exp\left(-\gamma\|\vec{x} - \vec{z}\|^2\right)$$

We will tune gamma as part of Lab 7

Soft-margin SVMs (non-separable case)

- Idea: we will use regularization to add a cost for each point being incorrectly classified by the hyperplane

- Hopefully many costs will be 0, but we can accommodate a few outliers

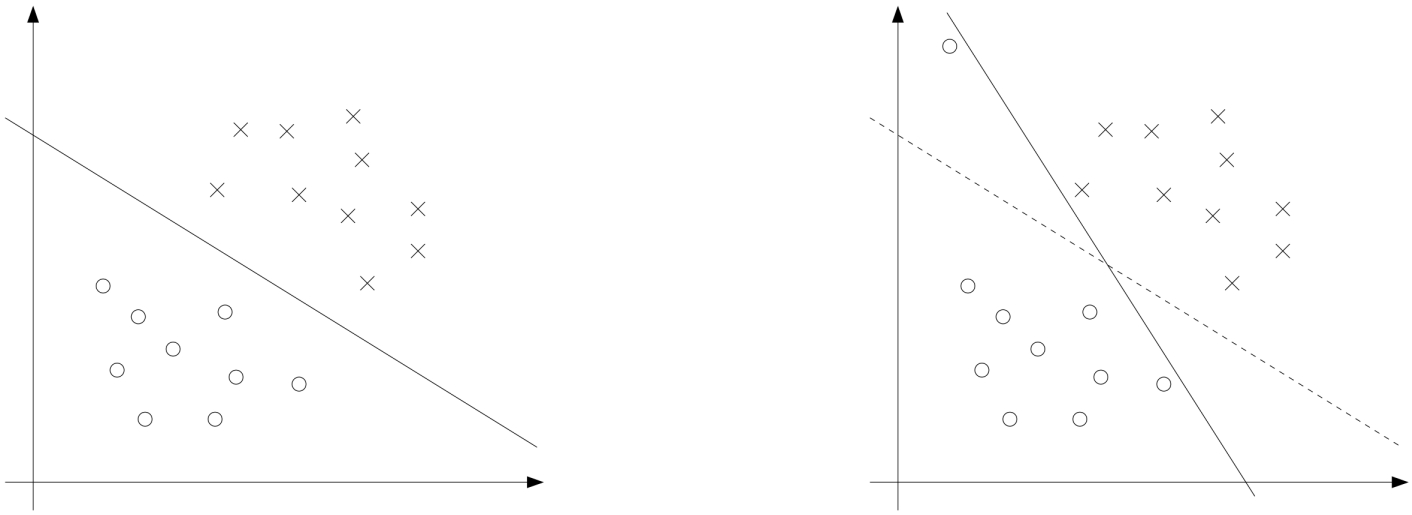


Figure: Andrew Ng

Soft-margin SVMs (non-separable case)

- New optimization problem with regularization
- We will tune the *C* parameter as part of Lab 7

$$\min_{\xi,\vec{w},b} \quad \frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$

"flexible margin"

$$\text{s.t.} \quad y_i(\vec{w}\cdot\vec{x}_i + b) \geq 1 - \xi_i, \quad i = 1, \cdots, n$$

$$\text{and} \quad \xi_i \geq 0, \quad i = 1, \cdots, n$$

Outline for November 5

- Cross-validation
- Recap Lagrange multipliers
- Lagrangian for SVMs
- Extensions of SVMs
- Solving the SVM optimization problem

Meta-optimization process

- Incremental SVM optimization algorithm

Meta-optimization process

- Incremental SVM optimization algorithm

- Choose a subset S of examples and run optimization to get alpha values

Meta-optimization process

- Incremental SVM optimization algorithm
- Choose a subset S of examples and run optimization to get alpha values
- Identify which alpha values are 0 => these cannot be support vectors in final solution!

Meta-optimization process

- Incremental SVM optimization algorithm
- Choose a subset S of examples and run optimization to get alpha values
- Identify which alpha values are 0 => these cannot be support vectors in final solution!
- Discard these points and add new ones; repeat

if 3 s.v.

(a) connect 2 of same class with line ℓ

(b) drop $\perp$ from 3rd pt to ℓ, creating (m) ← line segment

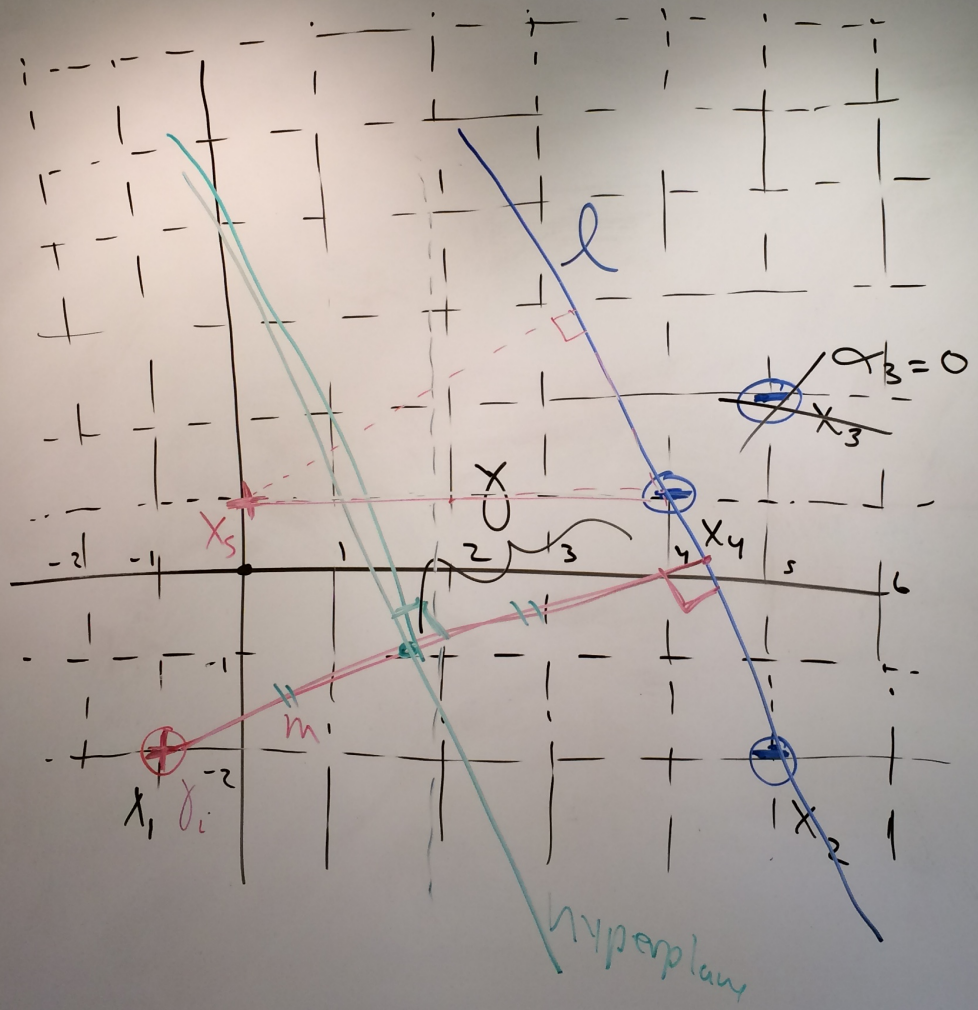
(c) hyperplane is $\perp$ bisector of m

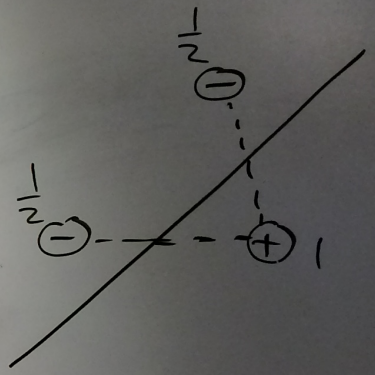
Round 1

$K = 4$

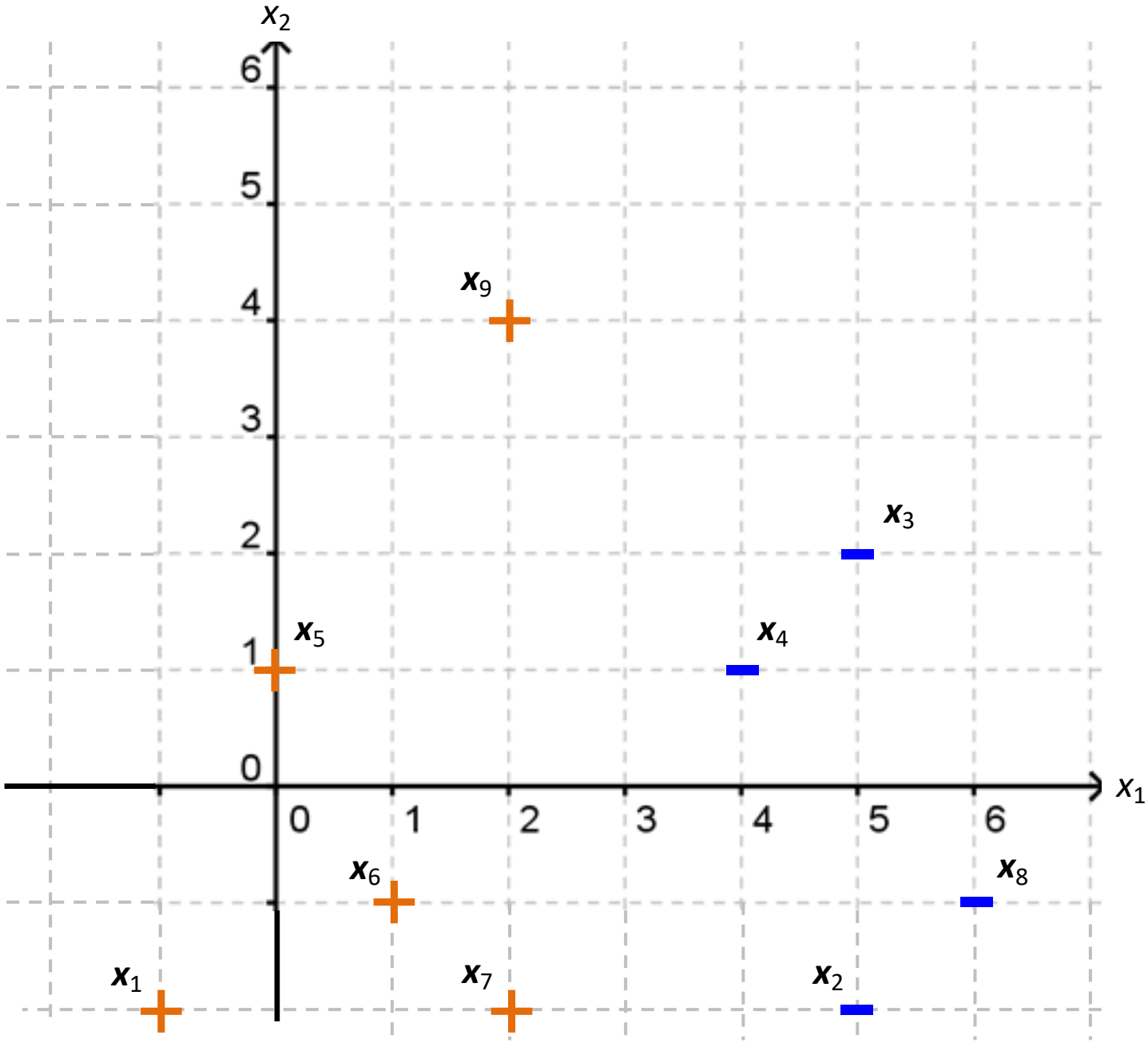
$S = \{x_1, x_2, x_3, x_4\}$

$S.V. = \{x_1, x_2, x_4\}$

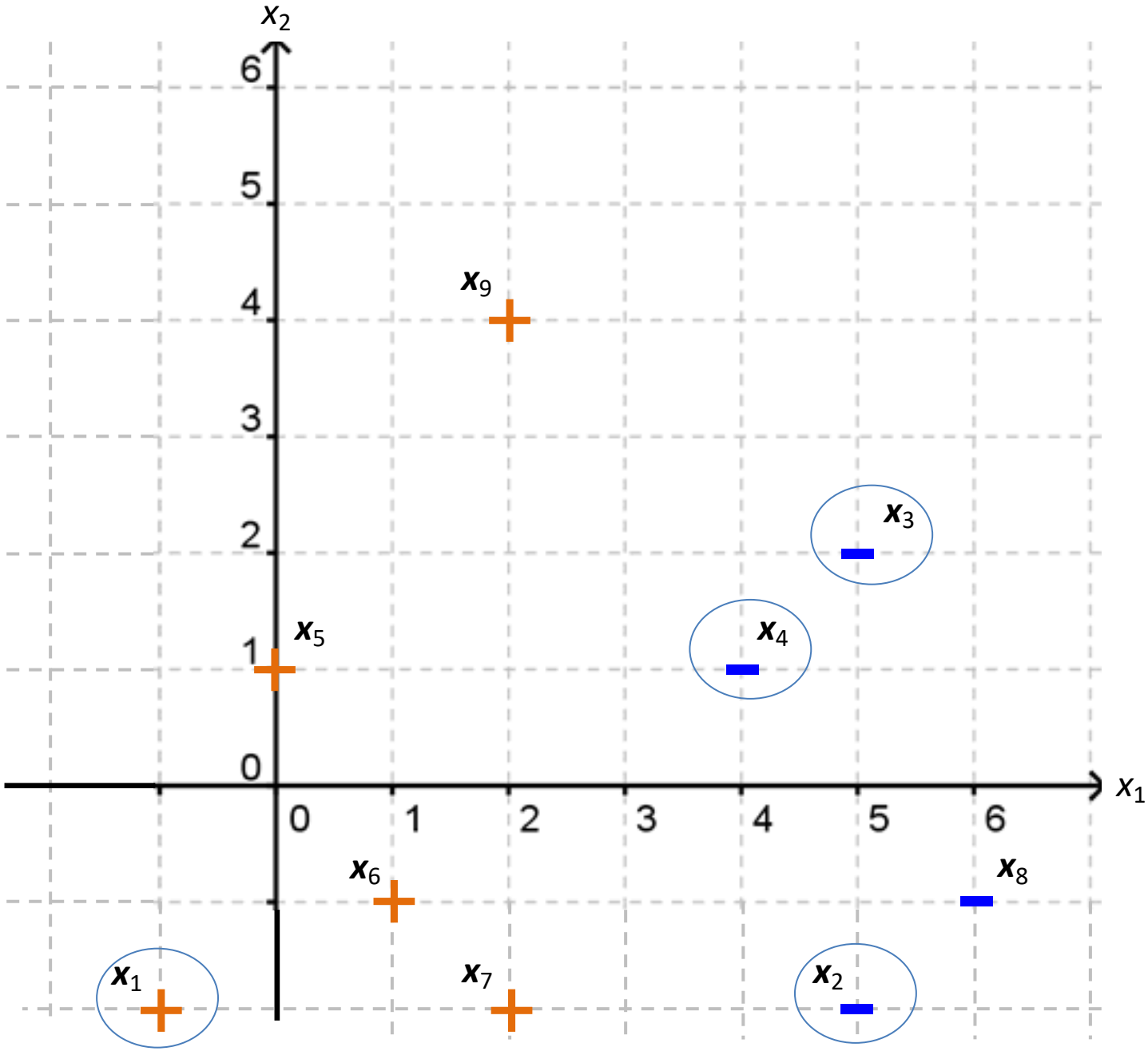

ℓ
$\alpha_3 = 0$
x_3
x
x_5
x_4
m
x_1
γ_i
x_2
hyperplane


$\frac{1}{2}$
$\frac{1}{2}$
1

Meta-optimization: example



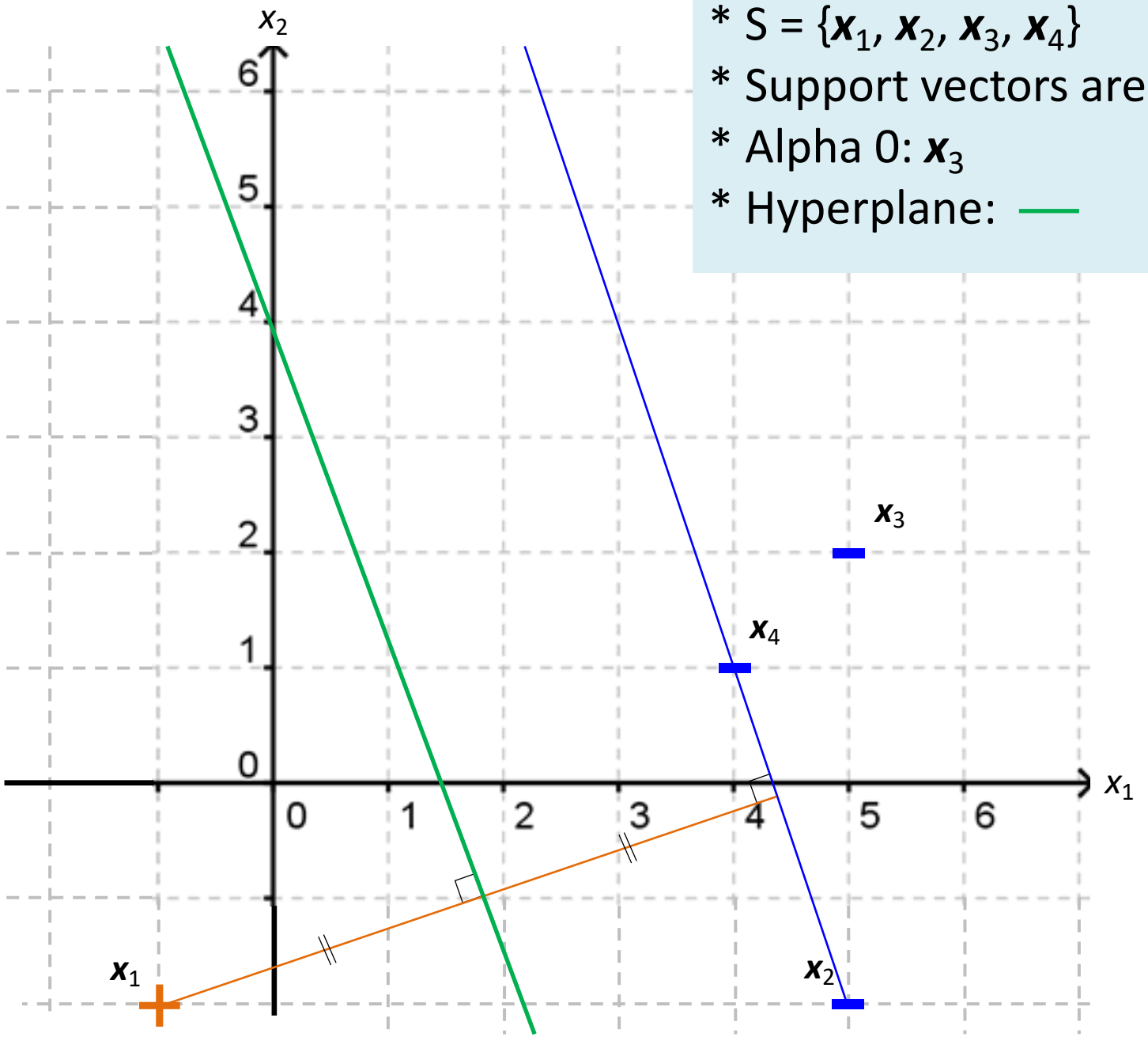
Meta-optimization: example



$K = 4$

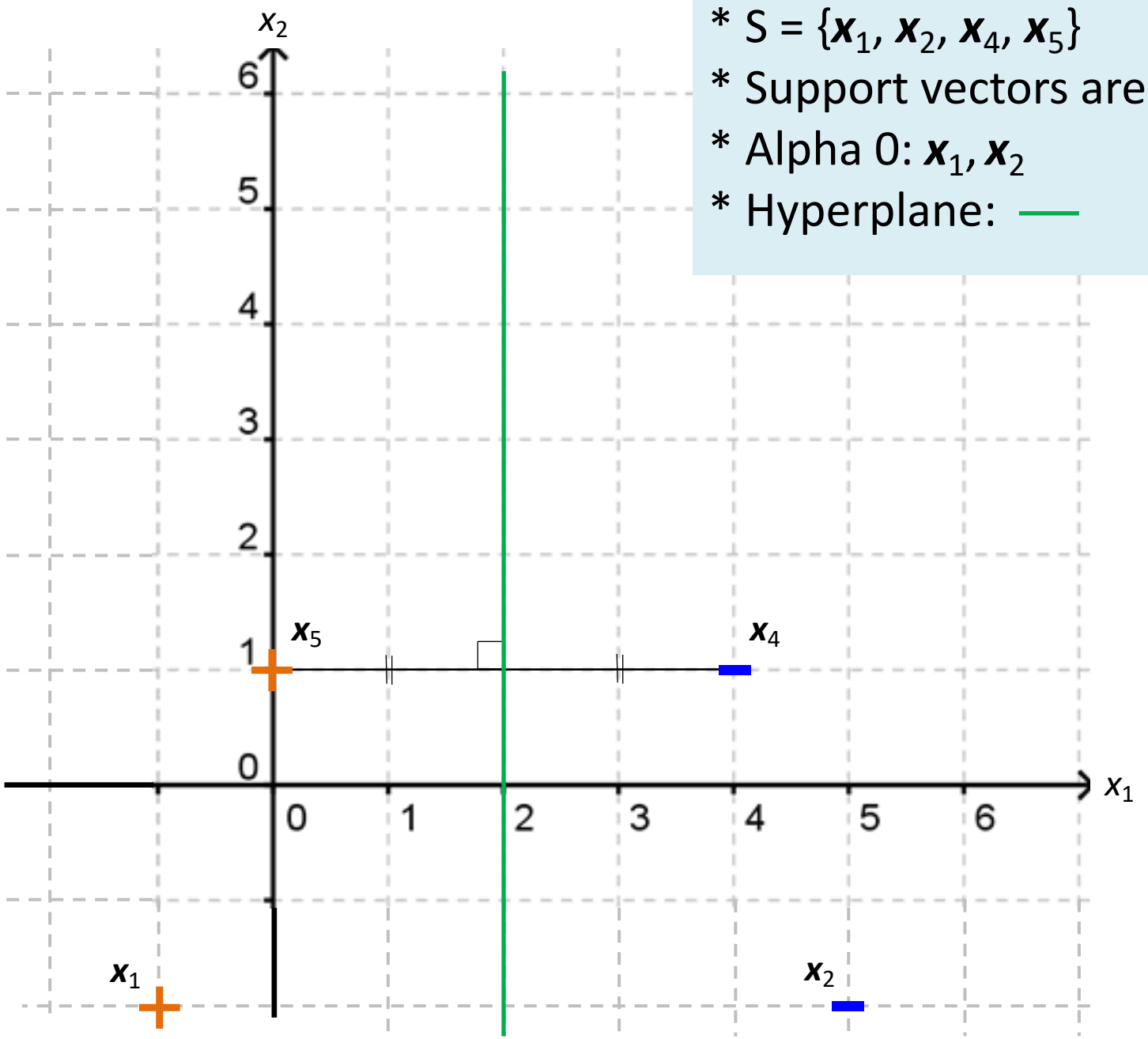
Round 1:

* S = {$\boldsymbol{x}_1$, $\boldsymbol{x}_2$, $\boldsymbol{x}_3$, $\boldsymbol{x}_4$}
* Support vectors are: $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, $\boldsymbol{x}_4$
* Alpha 0: $\boldsymbol{x}_3$
* Hyperplane:



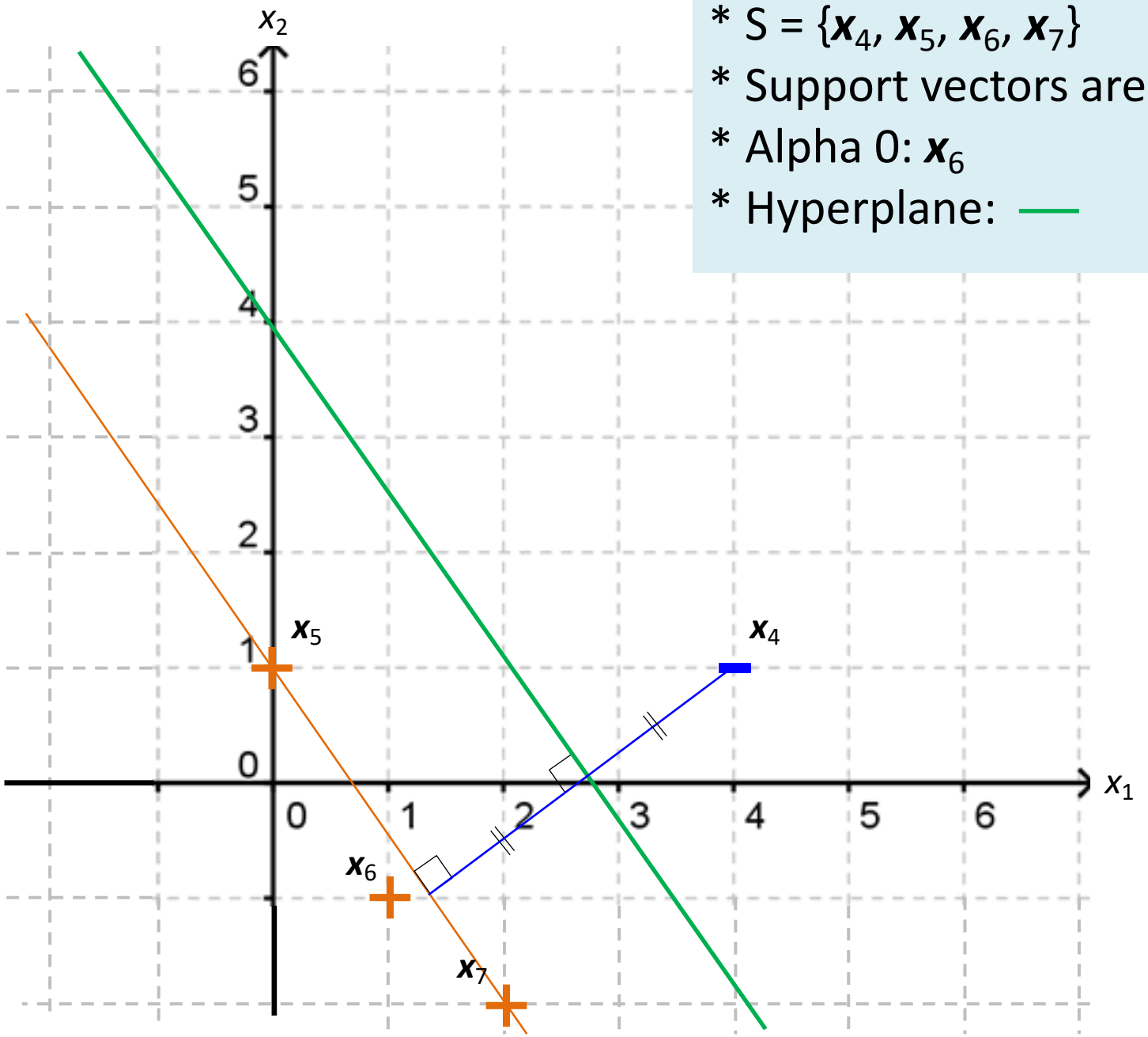
Round 1:

* S = $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_4, \boldsymbol{x}_5\}$
* Support vectors are: $\boldsymbol{x}_4, \boldsymbol{x}_5$
* Alpha 0: $\boldsymbol{x}_1, \boldsymbol{x}_2$
* Hyperplane:



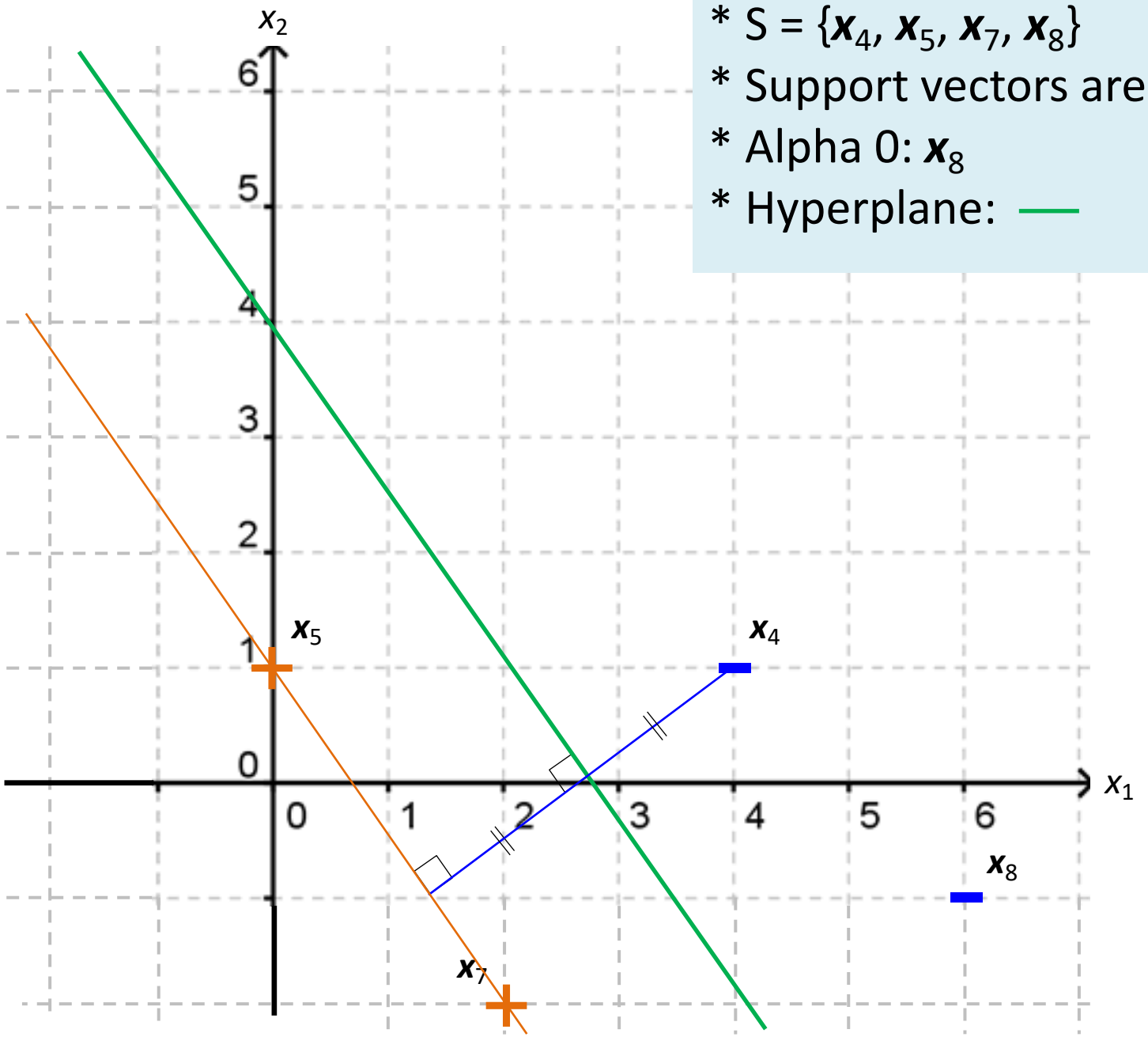
Round 3:

* S = {$\boldsymbol{x}_4$, $\boldsymbol{x}_5$, $\boldsymbol{x}_6$, $\boldsymbol{x}_7$}
* Support vectors are: $\boldsymbol{x}_4$, $\boldsymbol{x}_5$, $\boldsymbol{x}_7$
* Alpha 0: $\boldsymbol{x}_6$
* Hyperplane:



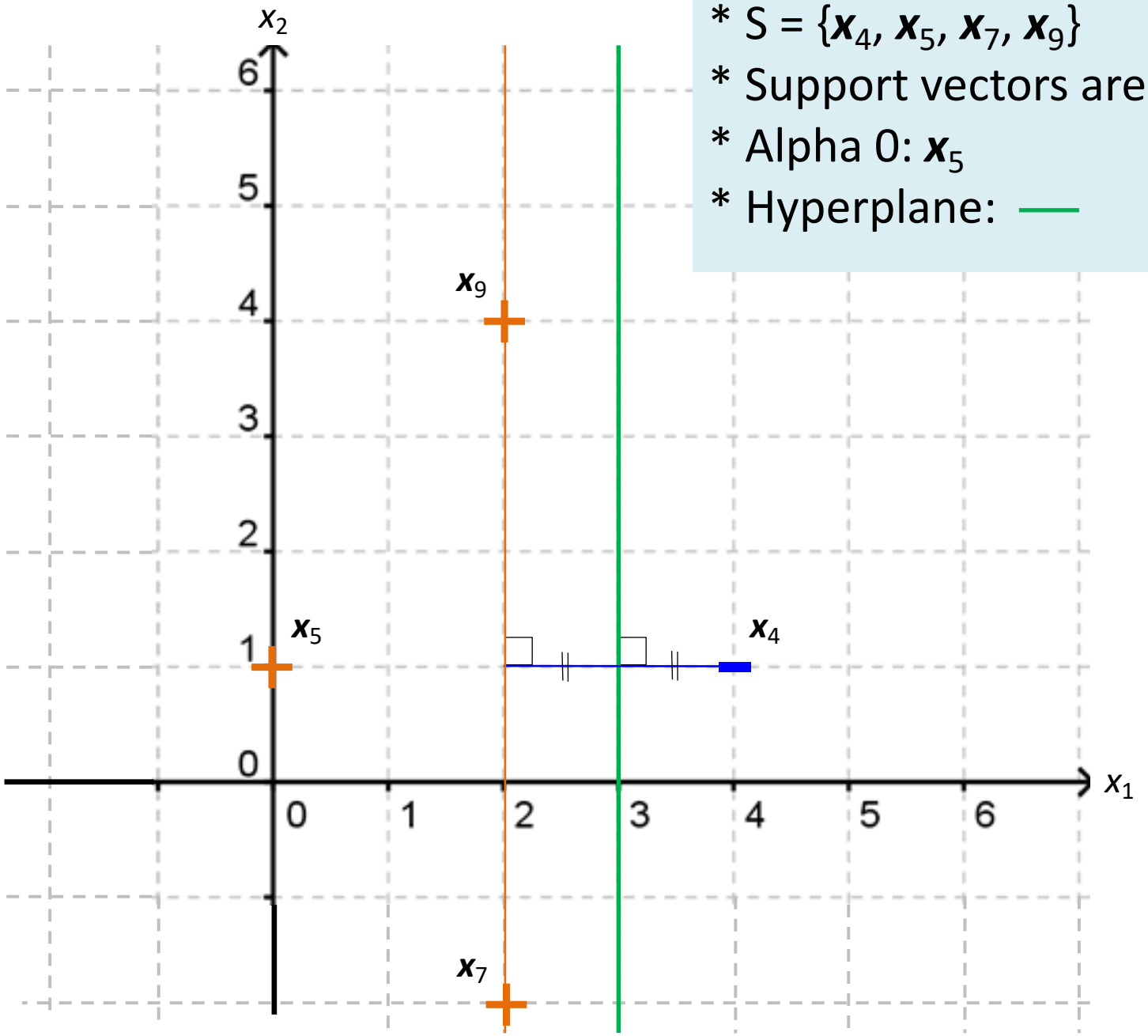
Round 4:

* S = {$\boldsymbol{x}_4$, $\boldsymbol{x}_5$, $\boldsymbol{x}_7$, $\boldsymbol{x}_8$}
* Support vectors are: $\boldsymbol{x}_4$, $\boldsymbol{x}_5$, $\boldsymbol{x}_7$
* Alpha 0: $\boldsymbol{x}_8$
* Hyperplane:



Round 5:

* S = {$\boldsymbol{x}_4$, $\boldsymbol{x}_5$, $\boldsymbol{x}_7$, $\boldsymbol{x}_9$}
* Support vectors are: $\boldsymbol{x}_4$, $\boldsymbol{x}_7$, $\boldsymbol{x}_9$
* Alpha 0: $\boldsymbol{x}_5$
* Hyperplane:



Handout 17, Final Solution