# Introduction to Machine Learning

L. Amendola

SS2025

# Outline

What is ML?

A classification of ML methods

Algorithm 1 kNN
Algorithm 2 Decision Trees and Random Forest
Algorithm 3 Support Vector Machine
Algorithm 4 Neural Networks
Algorithm 5 Clustering (K-means etc.)

# Goals

The philosophy of ML
A first view of ML motivations and methods
Learn terminology
Familiarize with the main algorithms

**Algorithms, not theories!**

# The two cultures: Models vs. Data

*Statistical Science*
2001, Vol. 16, No. 3, 199–231

## Statistical Modeling: The Two Cultures

**Leo Breiman**

*Abstract.* There are two cultures in the use of statistical modeling to reach conclusions from data. One assumes that the data are generated by a given stochastic data model. The other uses algorithmic models and treats the data mechanism as unknown. The statistical community has been committed to the almost exclusive use of data models. This commitment has led to irrelevant theory, questionable conclusions, and has kept statisticians from working on a large range of interesting current problems. Algorithmic modeling, both in theory and practice, has developed rapidly in fields outside statistics. It can be used both on large complex data sets and as a more accurate and informative alternative to data modeling on smaller data sets. If our goal as a field is to use data to solve problems, then we need to move away from exclusive dependence on data models and adopt a more diverse set of tools.

# What is ML?

Let’s ask ChatGPT !

# What is ML?

Machine learning (ML) is a branch of artificial intelligence (AI) that focuses on building systems that can learn from data and improve their performance over time without being explicitly programmed. Instead of relying on rigid instructions, a machine learning model "learns" from patterns in data and adjusts its behavior based on experience.

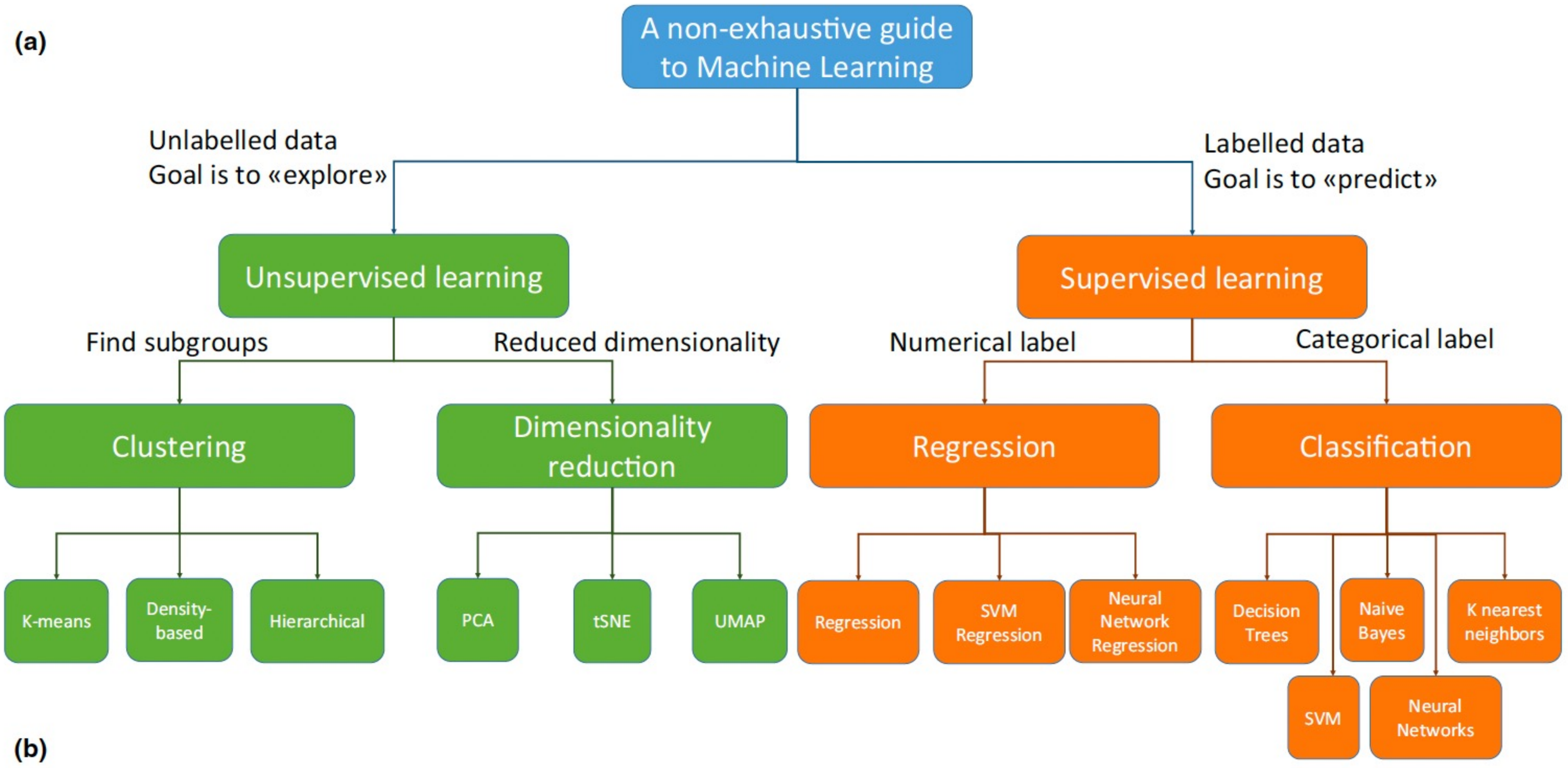
There are two main types of machine learning:

1. **Supervised learning**: In this type, the model is trained on a labeled dataset, meaning that the data includes both the input (features) and the correct output (labels). The model learns to map inputs to the correct outputs. An example is spam email detection, where the system learns to classify emails as "spam" or "not spam."
2. **Unsupervised learning**: Here, the model is given data without labels. The goal is to identify patterns or structures in the data, such as grouping similar items together (clustering) or reducing the dimensionality of the data (like with principal component analysis). An example is market segmentation, where customers are grouped based on purchasing behavior without predefined categories.

Machine learning is used in many fields today, like image recognition, language translation, recommendation systems (like Netflix or Spotify), self-driving cars, and more.

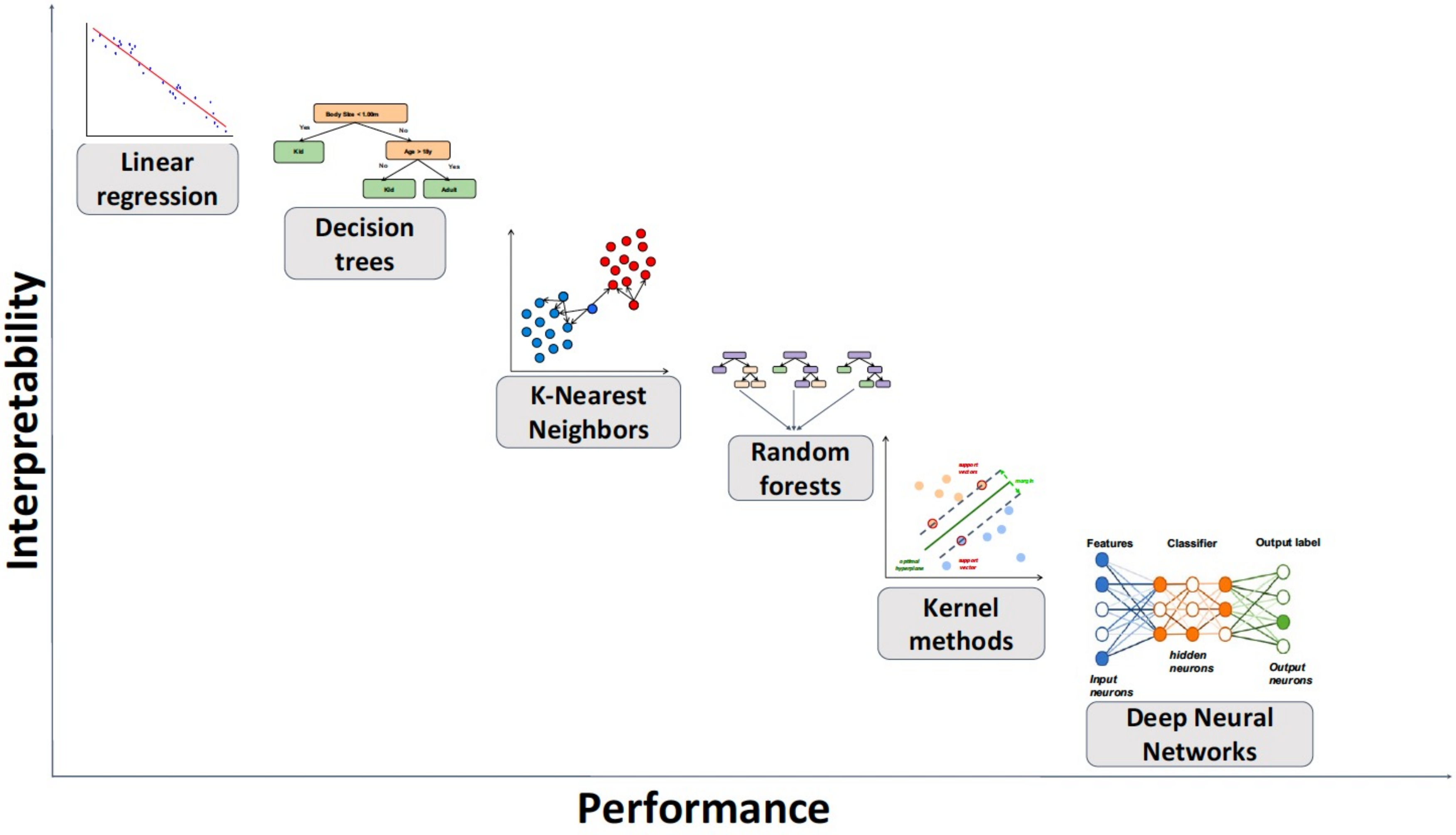
# How is different from standard statistical methods?

While both machine learning and statistical methods analyze data and uncover patterns, **machine learning focuses more on prediction and handling complex, large datasets**, and **statistical methods focus more on inference and understanding relationships** with a stronger emphasis on theory and interpretability.




Badillo et al.
CLINICAL PHARMACOLOGY & THERAPEUTICS | VOLUME 107 NUMBER 4 | April 2020

(b)


Interpretability
Performance
Linear regression
Decision trees
K-Nearest Neighbors
Random forests
Kernel methods
Deep Neural Networks
Features
Classifier
Output label
Input neurons
hidden neurons
Output neurons

# Typical ML problems

## Supervised problems

Recognize handwritten numbers/text

Speech recognition

Classify objects/images/etc into predefined classes

Predict if a patient will develop some medical condition

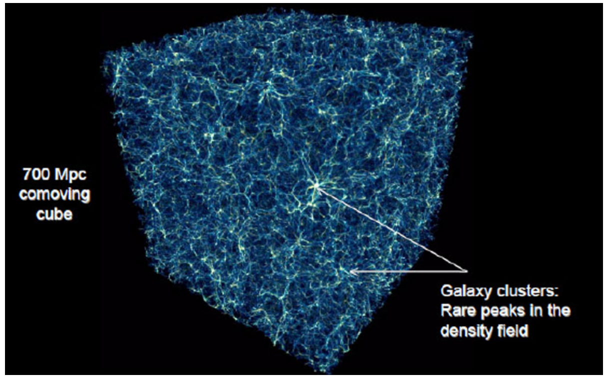
## Unsupervised problems

Find structure in data (i.e. clusters)

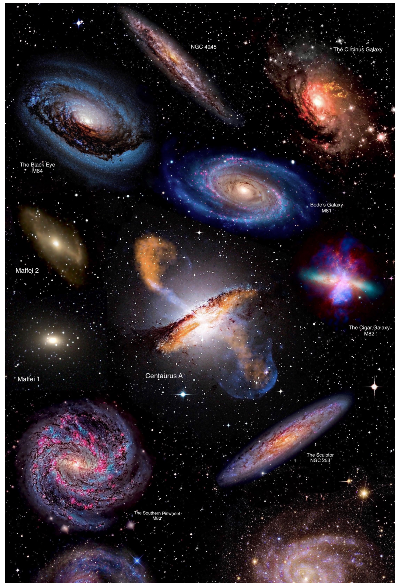
Detect outliers (anomalies)

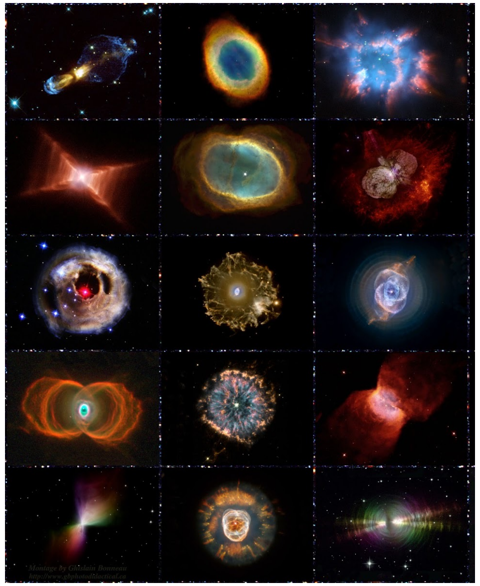
Reduce complexity (dimensionality) of a problem

# Applications to astrophysics

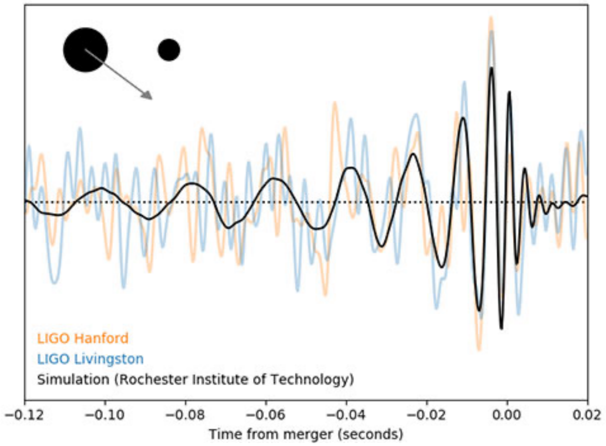



find cluster of galaxies in data;
collect DM particles in halos in sims





distinguish galaxies from
other diffuse objects




find outliers in GW signals

# From Statistics to ML

| Statistics | ML |
| --- | --- |
| model | network, graph |
| parameters | weights |
| fitting | learning |
| regression, classification | supervised learning |
| clustering | unsupervised learning |
| covariates (random variables) | features |
| likelihood | cost (or loss) function |

# From Statistics to ML

| Statistics | ML |
|---|---|
| model | network, graph |
| parameters | weights |
| fitting | learning |
| regression, classification | supervised learning |
| clustering | unsupervised learning |
| covariates (random variables) | features |
| likelihood | cost (or loss) function |

Example of regression:

If we measure a given temperature and volume in a gas, what is its pressure?

How much will inflation increase if the Central Bank lowers the interest rate?

Numerical answers

Example of classification:

If a patient has such & such features (categorical or ordinal) will he/she develop cancer?

If an unknown fruit is round, red, and crunchy, is it an apple?

Yes/no answers

# From Statistics to ML

| Statistics | ML |
|---|---|
| model | network, graph |
| parameters | weights |
| fitting | learning |
| regression, classification | supervised learning |
| clustering | unsupervised learning |
| covariates (random variables) | features |
| likelihood | cost (or loss) function |

Example of categorical (or nominal) features:

Patients can be females, with previous medical condition X, blood type Y, vaccinated, non smokers, etc

Example of ordinal (or numerical) features:

Patients can weigh *X* kg, have height *Y* cm, age *Z* etc

# ML algorithms

| Supervised |
|---|
| Linear regression |
| Decision Trees |
| Random Forests |
| k-NN |
| Support Vector Machines |
| Naïve Bayes |
| Neural Networks |
| Linear Discriminant Analysis |

# ML algorithms

| Unsupervised<br>Clustering |
|---|
| K-Means |
| Hierarchical clustering |
| Friends-of-friends |
| Gaussian Mixture Model |

| Unsupervised<br>Dimensional Reduction |
|---|
| Principal Component Analysis |
| Autoencoders |
| Generative Models |
| Linear Discriminant Analysis |

# Menu

Algorithm 1 k-Nearest Neighbors (8 min)
https://www.youtube.com/watch?v=b6uHw7QW_n4

Algorithm2: Decision Trees (14 min+14 min)
Decision trees https://www.youtube.com/watch?v=_WddbmNBCy0
Random forest https://www.youtube.com/watch?v=GOJg3EE-nDM

*Math interlude I*

Algorithm 3: Support Vector Machine (15 min)
https://www.youtube.com/watch?v=ny1iZ5A8ilA

*Math interlude II*

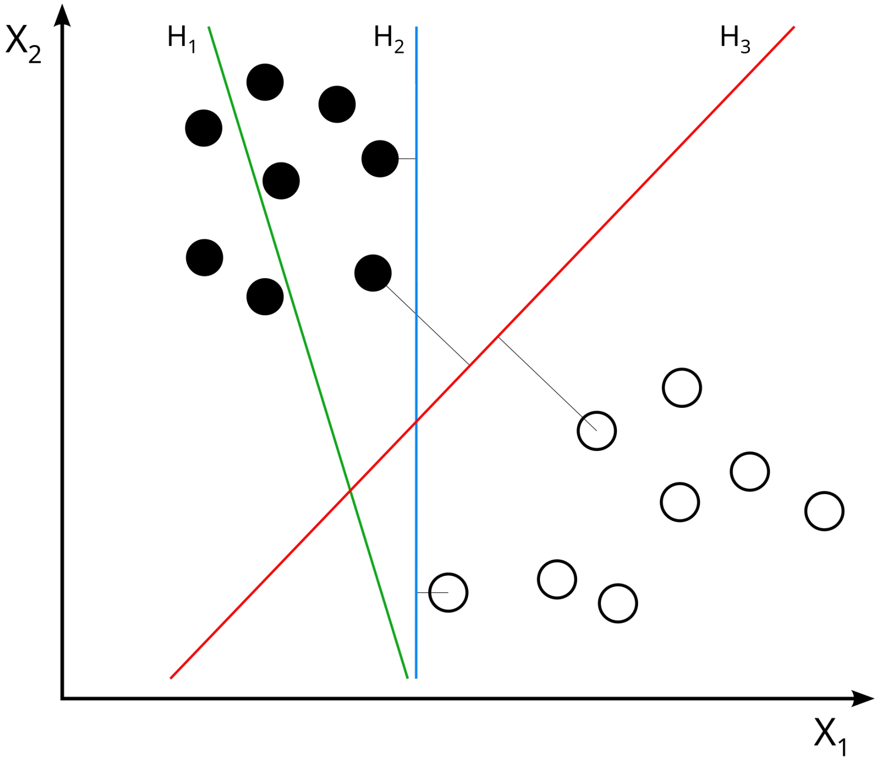
Algorithm 4: Neural Networks (18+20+12 min)
https://www.3blue1brown.com/topics/neural-networks

Algorithm 5: Unsupervised learning: clustering
k-means, hierarchical clustering, friends-of-friends

# *Math Interlude I*

# Support Vector Machines

linearly separable datsets




Problem: how to find the straight line that has the widest margin between the two sets?

# Side problem: Quadratic Optimization

general problem

$$\min_{x_1, x_2, \ldots x_n} \Phi(x_1, x_2, \ldots x_n)$$

$$g_i(\mathbf{x}) \leq 0$$

$$i = 1, 2, ..m$$

quadratic problem
(H: def pos)

$$\min_{x_1, x_2, \ldots x_n} \frac{1}{2} \mathbf{x} H \mathbf{x} - B \mathbf{x}$$

$$g_i(\mathbf{x}) \leq 0$$

$$i = 1, 2, ..m$$

# Quadratic Optimization

$$\min_{x_1, x_2, .. x_n} \frac{1}{2}\mathbf{x} H \mathbf{x} - B\mathbf{x}$$

$$g_i(\mathbf{x}) \leq 0$$

$$i = 1, 2, ..m$$

1D version

$$\begin{cases} \min_{x} \frac{1}{2} k x^2 \\ k > 0 \\ x \geq b \end{cases}$$

Dual problem:
Augmented Lagrangian
(Lagrangian multiplier)

$$L(x, \lambda) = \frac{1}{2} k x^2 + \lambda(b - x)$$

$$\lambda > 0$$

# Quadratic Optimization: b>0

$$L(x, \lambda) = \frac{1}{2} k x^2 + \lambda(b - x)$$
$$\lambda > 0$$

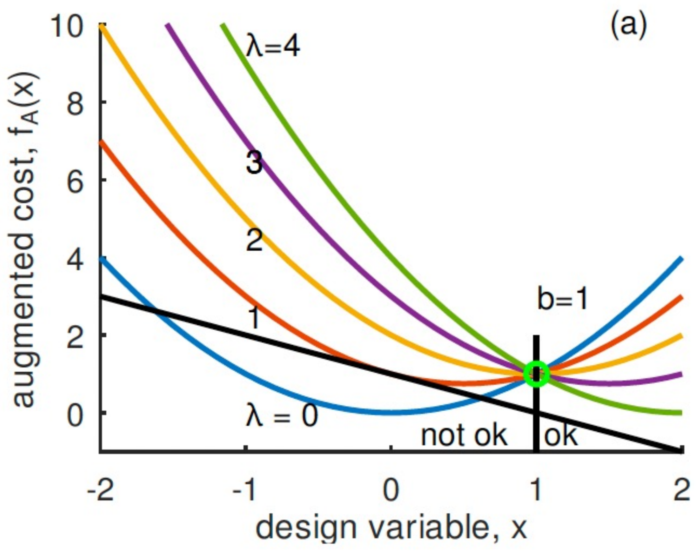
$$\frac{\partial L}{\partial x} = 0$$
$$\frac{\partial L}{\partial \lambda} = 0$$

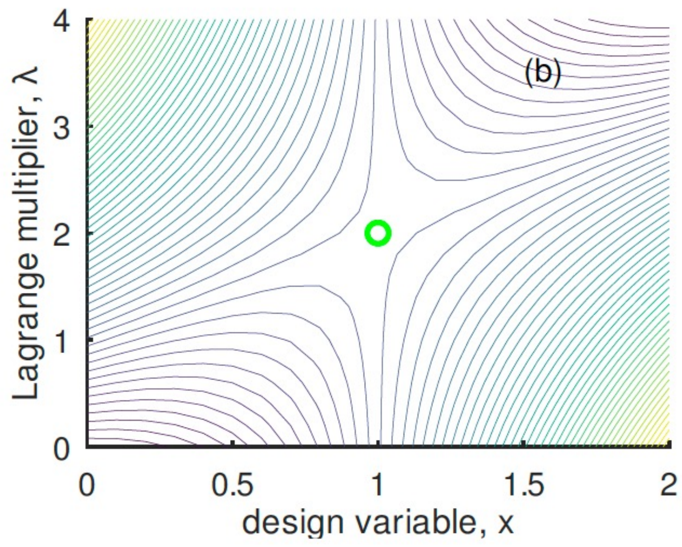
$$x^* = b$$
$$\lambda = k x^*$$

if *b>0*

*x=b, λ=kb*

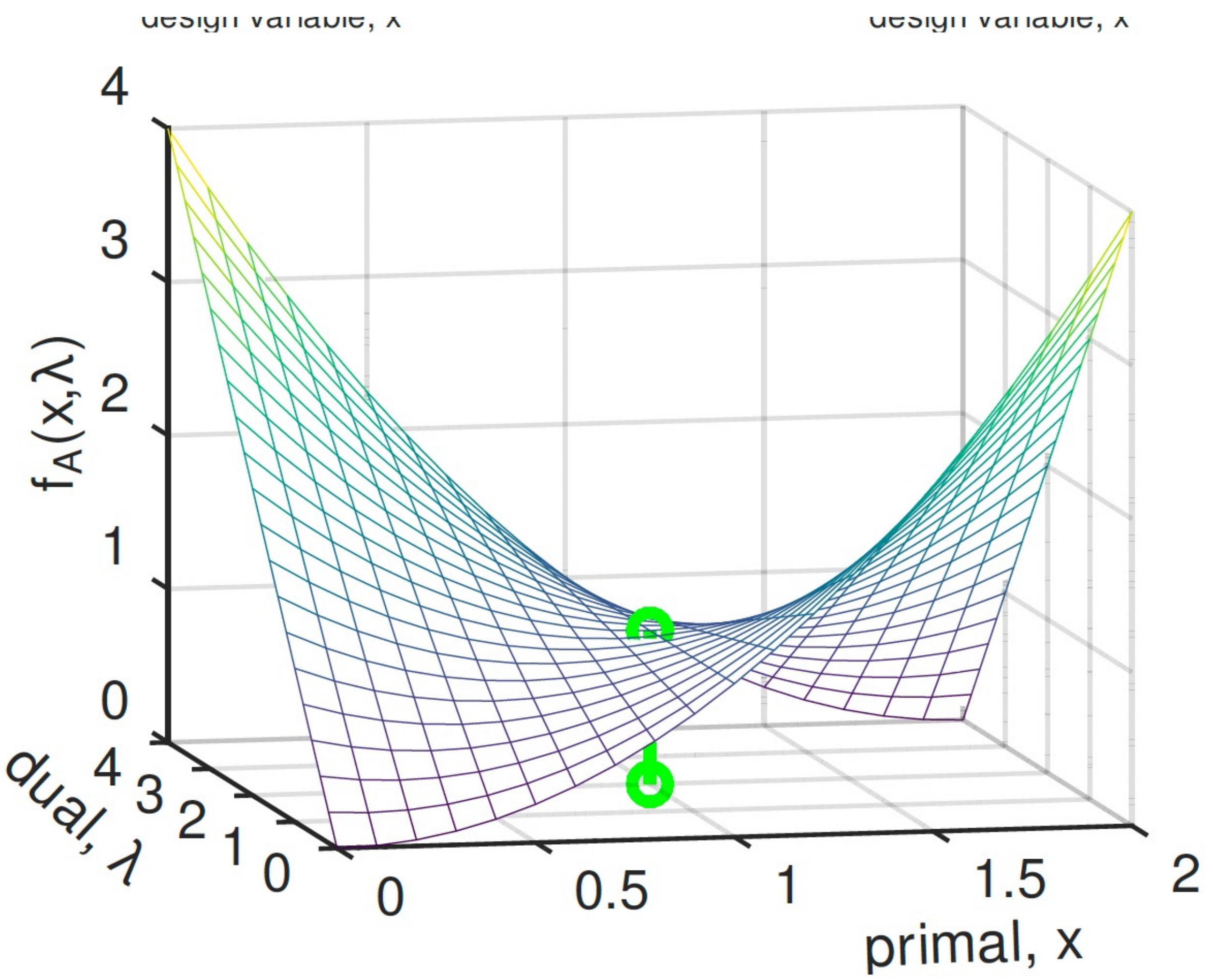
minimum in *x*, maximum (flat) in *λ*




*Quadratic Programming and Lagrange Multipliers*
*CEE 251L. Uncertainty, Design and Optimization*
*Department of Civil and Environmental Engineering*
*Duke University*
*Henri P. Gavin and Jeffrey T. Scruggs*

# Quadratic Optimization





Quadratic Programming and Lagrange Multipliers
CEE 251L. Uncertainty, Design and Optimization
Department of Civil and Environmental Engineering
Duke University
Henri P. Gavin and Jeffrey T. Scruggs

# Quadratic Optimization: b<0

$$L(x,\lambda) = \frac{1}{2} k x^2 + \lambda(b - x)$$
$$\lambda > 0$$

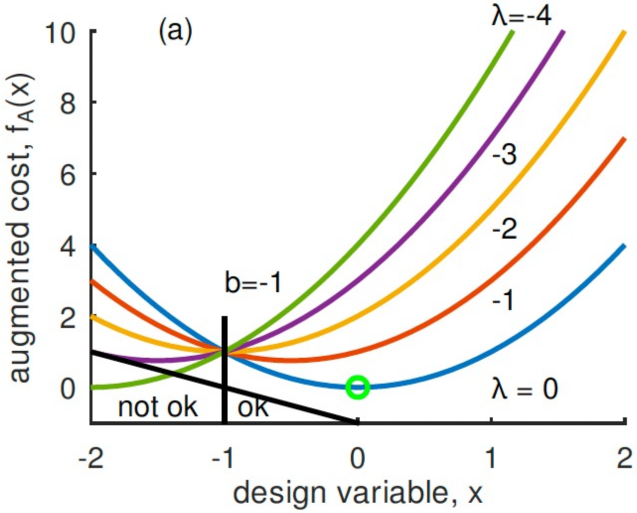
$$x^* = b < 0$$
$$\lambda = k x^* < 0$$

?

if *b<0*

*$\lambda$=0*

then $\lambda$ is irrelevant (*inactive*) and *x*=0*




*Quadratic Programming and Lagrange Multipliers*
*CEE 251L. Uncertainty, Design and Optimization*
*Department of Civil and Environmental Engineering*
*Duke University*
*Henri P. Gavin and Jeffrey T. Scruggs*

# Quadratic Optimization in nD

$$\min_{x_1, x_2, .. x_n} \frac{1}{2}\mathbf{x} H \mathbf{x} - B\mathbf{x}$$

$$g_i(\mathbf{x}) \leq 0$$

$$i = 1, 2, ..m$$

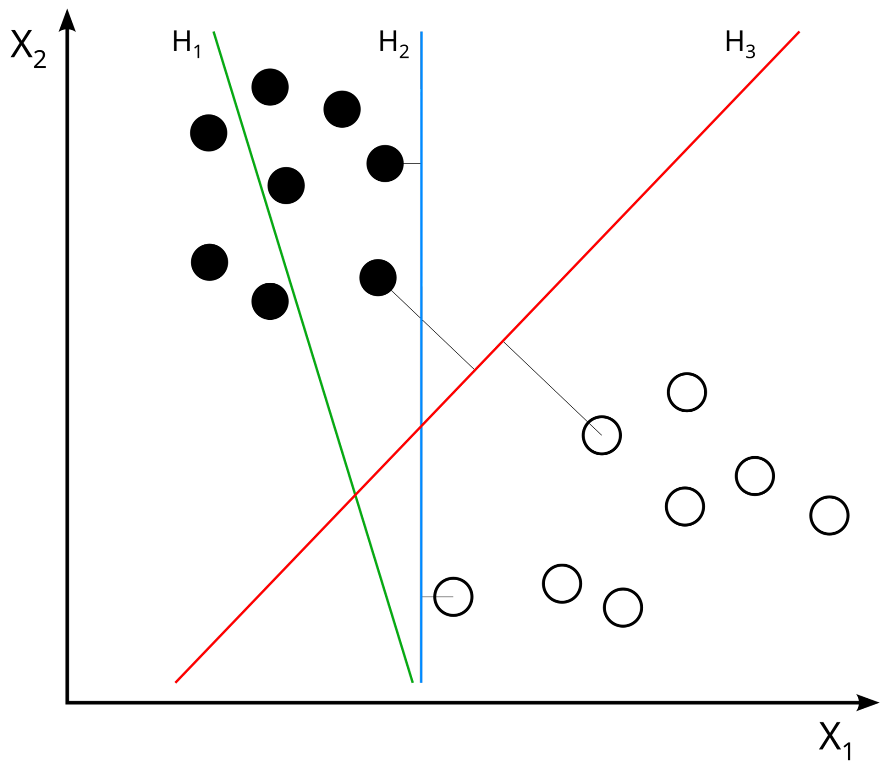
the main problem is to find
*active and inactive constraints*

method:

1) assume all constraints are inactive and find the minimum without constraints
2) assume all are active, find optimal ***x*** and ***λ***, see if they violate the constraint ***λ***>0
3) if they do, continue assuming all the possible combinations of active/inactive until all constraints are satisfied

# Support Vector Machines

linearly separable datsets






Problem: how to recast the problem into a Quadratic Optimization problem?

# Support Vector Machines

equation of a 2D line

$$y = ax + b$$

$$w_1 x_1 + w_2 x_2 - b = 0$$

equations for the two margins

$$w_1 x_1 + w_2 x_2 - b = \pm m$$

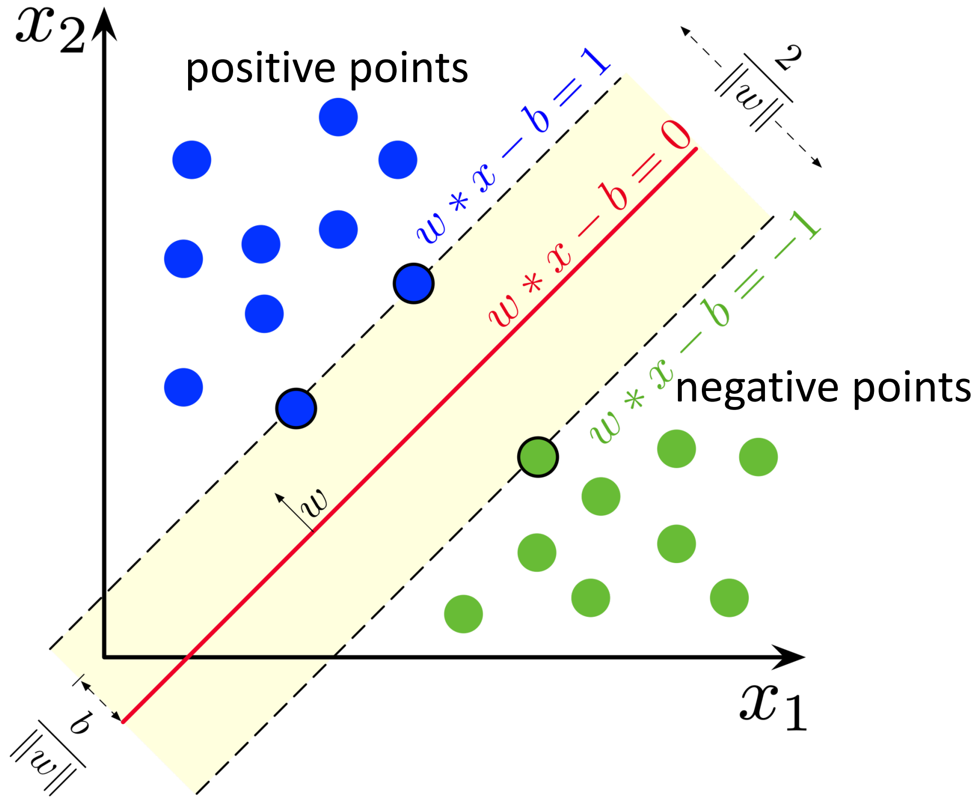
normalized form

$$\bar{w}_1 x_1 + \bar{w}_2 x_2 - \bar{b} = \pm 1$$

general equations in
D dimensions (hyperplanes)

$$\mathbf{w} \cdot \mathbf{x} - b = 0$$

$$\mathbf{w} \cdot \mathbf{x} - b = \pm 1$$




Problem: how to find the straight line that has the widest margin between the two sets?

# Support Vector Machines

general problem: given
*n* labeled data (vectors+label)

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2) \ldots (\mathbf{x}_n, y_n)$$
$$y_i = \pm 1$$

find the hyperplane ***w***,*b* such that
the margin is the widest

$$\mathbf{w}\cdot\mathbf{x} - b = 0$$
$$\mathbf{w}\cdot\mathbf{x} - b = \pm 1$$

distance between a point
and a line

$$d(\mathbf{x}_i) = \frac{\mathbf{w}\cdot\mathbf{x}_i - b}{|\mathbf{w}|}$$

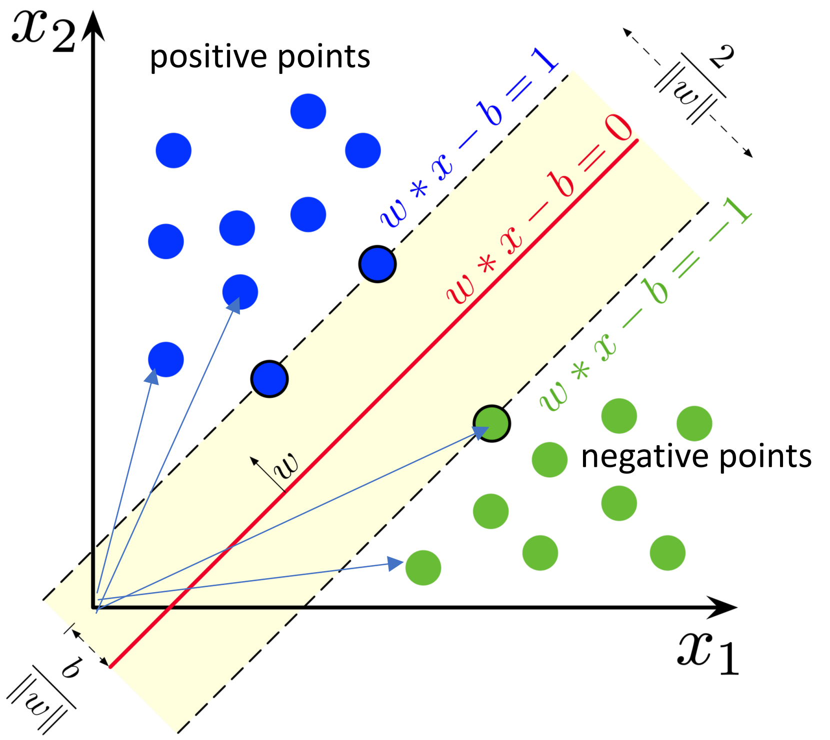
distance between a point
on the margin and the hyperplane

$$\frac{1}{|w|}$$

Total width of the margin

$$\frac{2}{|w|}$$

each point=vector



# Support Vector Machines

Optimization problem: minimize $|w|$

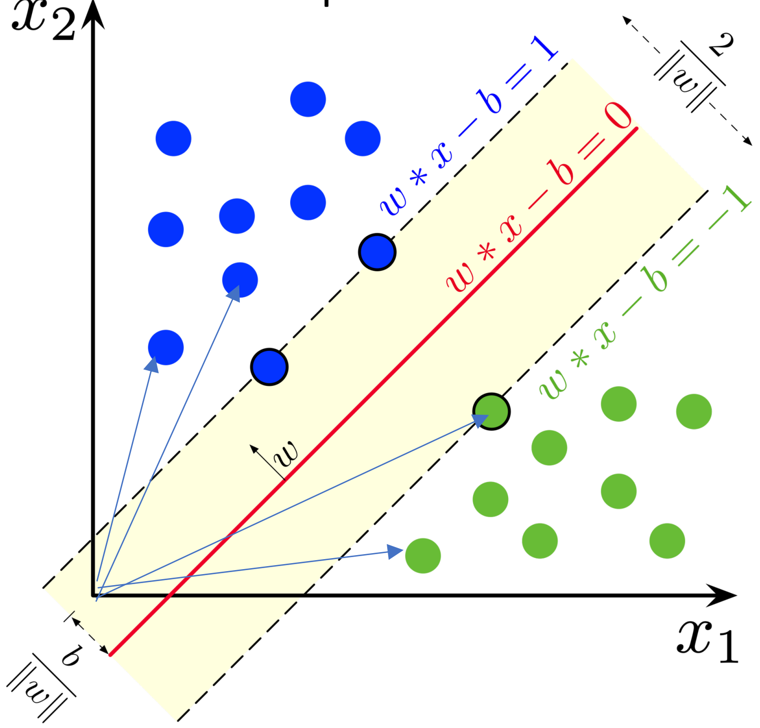
or equivalently $\frac{1}{2}|w|^2$

with the condition
that all the points are on
either side
(hard margin)

$$\mathbf{w}\cdot\mathbf{x}_i - b \geq 1 \text{ for } y_i = 1$$
$$\mathbf{w}\cdot\mathbf{x}_i - b \leq -1 \text{ for } y_i = -1$$

or simply $y_i(\mathbf{w}\cdot\mathbf{x}_i - b) \geq 1 \;\forall i$

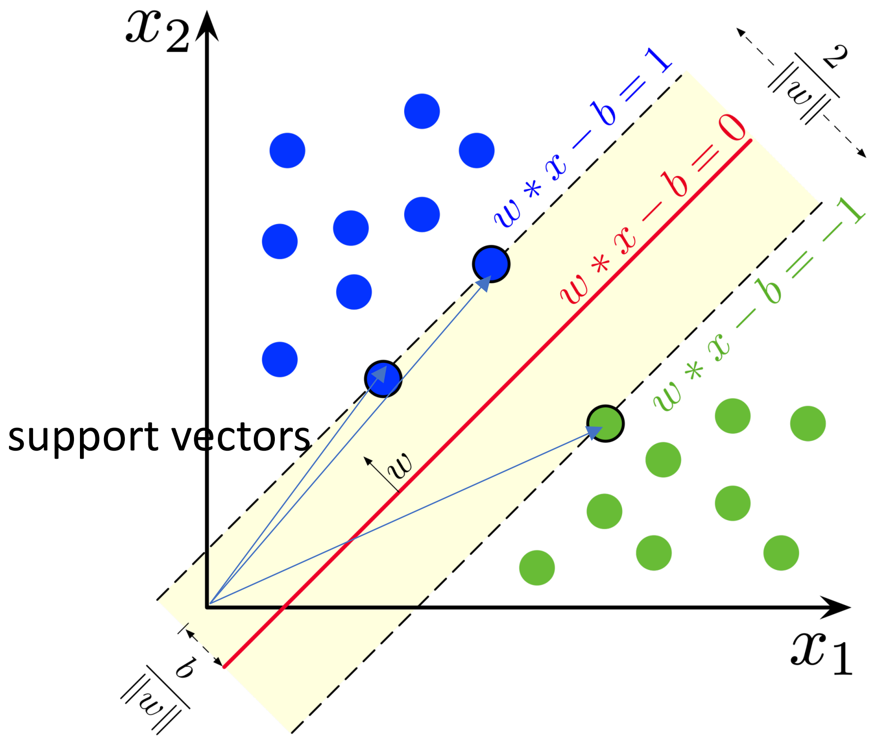
each point=vector



# Support Vector Machines: hard margins

find ***w***,*b* that minimize $\frac{1}{2}|w|^2$

with the condition $y_i(\mathbf{w}\cdot\mathbf{x}_i - b) - 1 \geq 0 \; \forall i$




Quadratic Optimization:
This problem can be solved by minimizing a Lagrangian with Lagrangian multipliers (dual problem)

$$L = \frac{1}{2}|w|^2 + \sum \alpha_i(1 - y_i(\mathbf{w}\cdot\mathbf{x}_i - b))$$

$$\alpha_i \geq 0$$

# Support Vector Machines: soft margins

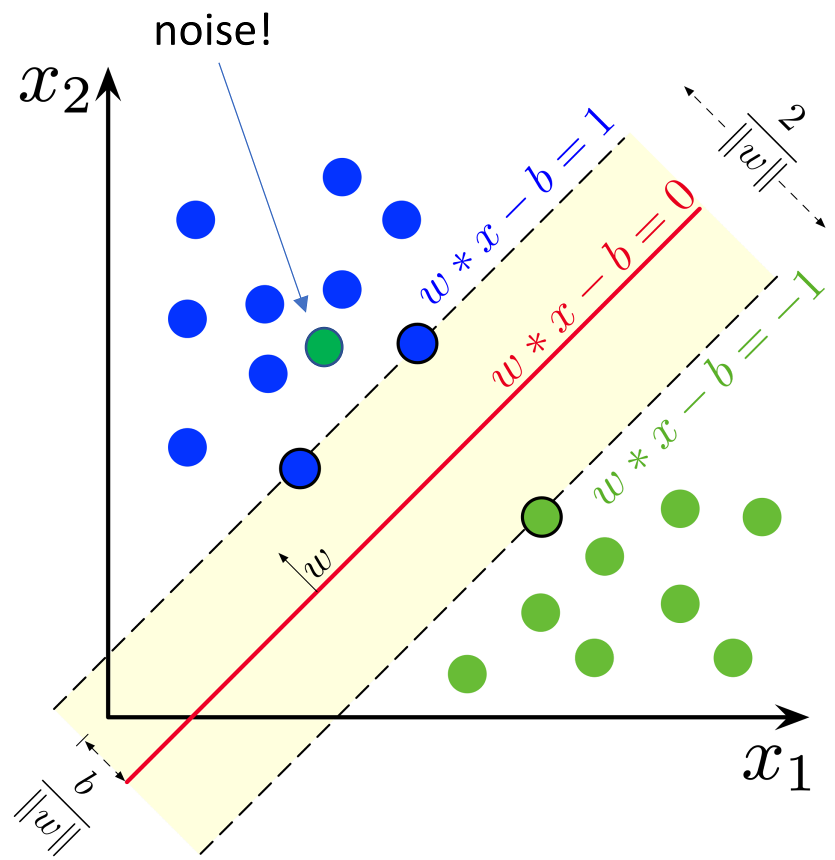
relax the constraint $y_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1 - \xi_i$

positive "slack" parameters

new Lagrangian with penalizer *C (hyperparameter)*

$$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \beta) = \frac{1}{2}|w|^2 + \underbrace{\frac{C}{n}\sum \xi_i}_{\text{penalizer}} + \underbrace{\sum \alpha_i(1 - \xi_i - y_i(\mathbf{w} \cdot \mathbf{x}_i - b))}_{\text{enforcing the margin}} - \underbrace{\sum \beta_i \xi_i}_{\text{positive slacks}}$$

$$\alpha_i, \beta_i \geq 0$$




if *C=infinity*, all the slack parameters must vanish: back to the hard margin!

# Extremizing the Lagrangian

minimum in $w$
maximum in $\alpha_i, \beta_i$

$$\boxed{L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \beta) = \frac{1}{2}|w|^2 + \frac{C}{n}\sum \xi_i + \sum \alpha_i(1 - \xi_i - y_i(\mathbf{w}\cdot\mathbf{x}_i - b)) - \sum \beta_i \xi_i}$$

derivatives

$$\left\{\begin{aligned} \frac{\partial L}{\partial \mathbf{w}} &= \mathbf{w} - \sum \alpha_i y_i \mathbf{x}_i = 0 \\ \frac{\partial L}{\partial b} &= -\sum \alpha_i y_i = 0 \\ \frac{\partial L}{\partial \xi_i} &= \frac{C}{n} - \alpha_i - \beta_i = 0 \end{aligned}\right.$$

Once we have the $\alpha_i$ we have the vector $w$ !

Inserting into the Lagrangian:

$$\begin{aligned} L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \beta) &= \frac{1}{2}\sum_{i,j} \alpha_i\alpha_j y_i y_j \mathbf{x}_i\cdot\mathbf{x}_j + \frac{C}{n}\sum \xi_i + \sum \alpha_i(1 - \xi_i - y_i(\sum \alpha_j y_j \mathbf{x}_j\cdot\mathbf{x}_i - b)) - \sum \beta_i\xi_i \\ &= -\frac{1}{2}\sum_{i,j} \alpha_i\alpha_j y_i y_j \mathbf{x}_i\cdot\mathbf{x}_j + \sum(\alpha_i + \beta_i)\xi_i + \sum \alpha_i(1 - \xi_i + y_i b)) - \sum \beta_i\xi_i \\ &= -\frac{1}{2}\sum_{i,j} \alpha_i\alpha_j y_i y_j \mathbf{x}_i\cdot\mathbf{x}_j + \sum \alpha_i + b\sum \alpha_i y_i \\ &= -\frac{1}{2}\sum_{i,j} \alpha_i\alpha_j y_i y_j \mathbf{x}_i\cdot\mathbf{x}_j + \sum \alpha_i \end{aligned}$$

Only depends on the scalar products $\boldsymbol{x}_i\boldsymbol{x}_j$

# Extremizing the Lagrangian

$$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \beta) = \frac{1}{2}|w|^2 + \frac{C}{n}\sum \xi_i + \sum \alpha_i(1 - \xi_i - y_i(\mathbf{w}\cdot\mathbf{x}_i - b)) - \sum \beta_i \xi_i$$

$$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \beta) = -\frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j + \sum \alpha_i$$

minimize wrt ***x***, maximize wrt $\boldsymbol{\alpha}$

With two constraints:

$$\frac{\partial L}{\partial \xi_i} = \frac{C}{n} - \alpha_i - \beta_i = 0 \quad \Rightarrow \quad 0 \leq \alpha_i \leq \frac{C}{n}$$

and $$\sum \alpha_i y_i = 0$$

# Extremizing the Lagrangian

$$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \beta) = -\frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j + \sum \alpha_i$$

$$0 \le \alpha_i \le \frac{C}{n}$$

$$\sum \alpha_i y_i = 0$$

This is a Quadratic Optimization problem:

vectors of 1s

Minimize $\frac{1}{2}\alpha^T H \alpha - \alpha^T e$

with constraints
$$\begin{cases} \alpha^T y = 0 \\ 0 \le \alpha_i \le \frac{C}{n} \end{cases}$$

Pos-def matrix $H_{ij} \equiv y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$

Several methods available to find the optimal $\alpha_i$ (not discussed here)

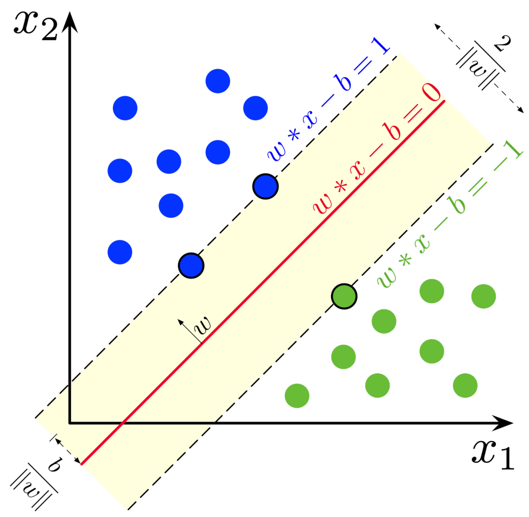
And since $\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum \alpha_i y_i \mathbf{x}_i = 0$ Once we have the vector $\boldsymbol{\alpha}$ we get the vector $\boldsymbol{w}$ !

# Extremizing the Lagrangian

$$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \beta) = \frac{1}{2}|w|^2 + \frac{C}{n}\sum \xi_i + \sum \alpha_i(1 - \xi_i - y_i(\mathbf{w}\cdot\mathbf{x}_i - b)) - \sum \beta_i \xi_i$$

$$0 \leq \alpha_i \leq \frac{C}{n}$$

Points such that $y_i(\mathbf{w}\cdot\mathbf{x}_i - b) > 1$ are "good" points
$L$ is maximized when $\alpha_i = 0$ and $\xi_i = 0$



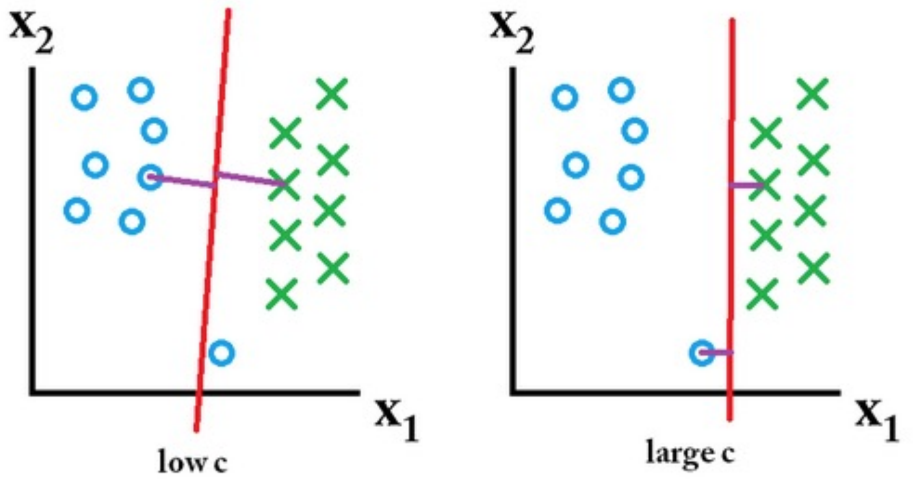

Points such that $y_i(\mathbf{w}\cdot\mathbf{x}_i - b) < 1$ are "bad" points
$L$ is maximized when $\alpha_i = C/n$ and $\xi_i > 0$

Points such that $y_i(\mathbf{w}\cdot\mathbf{x}_i - b) = 0$ are marginal points (*support vectors*)
$L$ is maximized when $0 < \alpha_i < \frac{C}{n}$ and $\xi_i = 0$

So the marginal points are those with $0 < \alpha_i < \frac{C}{n}$

Once we find the marginal points, $b = y_i - \mathbf{w}\cdot\mathbf{x}_i$ for any marginal point

# How to choose the hyperparameter C?




Large *C* produces a better separation, but small margin

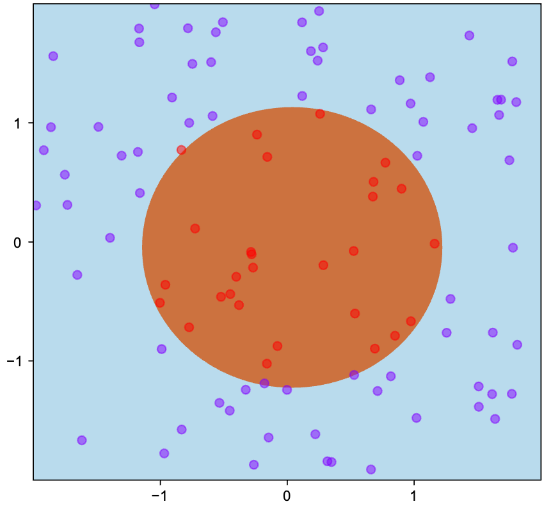
Small *C* produces some misclassification, but wider margins

As usual in ML, no hard rule!

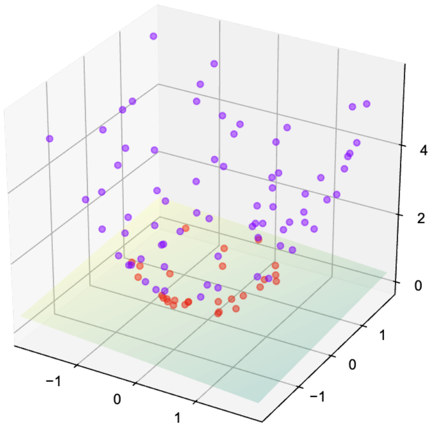
https://stats.stackexchange.com/questions/31066/what-is-the-influence-of-c-in-svms-with-linear-kernel

# Non-linearly separable datasets: The kernel trick

non-lin. separable



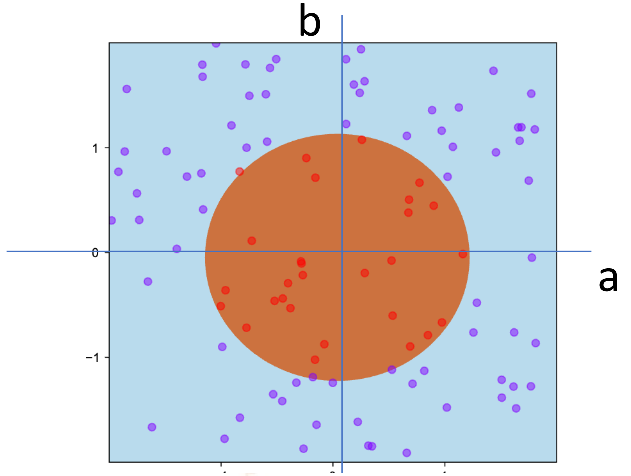

lin. separable

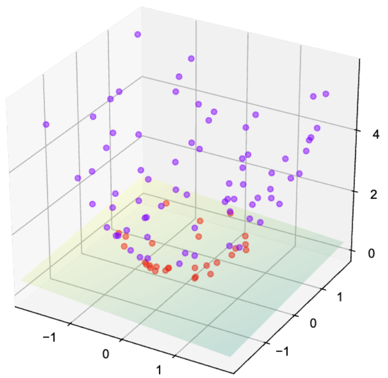



General idea:
find a trasformation of the original vectors into a higher-dimensional space such that the points become linearly separable

# Non-linearly separable datasets: The kernel trick

$$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \beta) = -\frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j + \sum \alpha_i$$




$$\mathbf{x} = \{a, b\} \rightarrow \phi(\mathbf{x}) = \{a, b, a^2 + b^2\}$$

$$K(\phi(\mathbf{x}_i)\cdot \phi(\mathbf{x}_j)) = a^2 + b^2 + (a^2 + b^2)(a^2 + b^2)$$

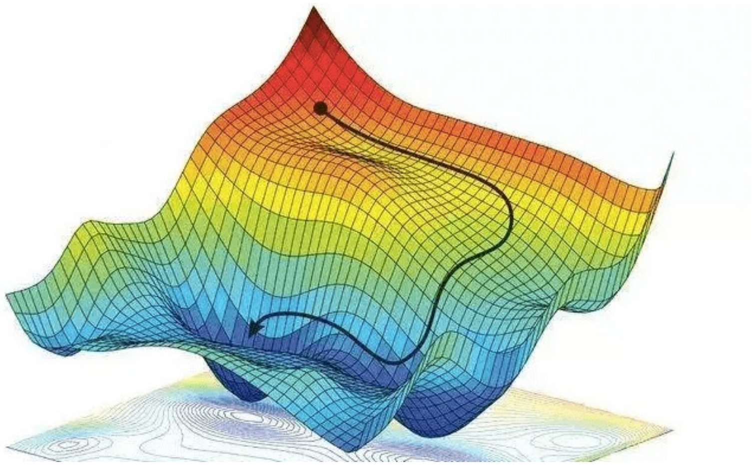
polynomial kernel $$K(\mathbf{x}_i \cdot \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j + |\mathbf{x}_i|^2 |\mathbf{x}_j|^2|$$

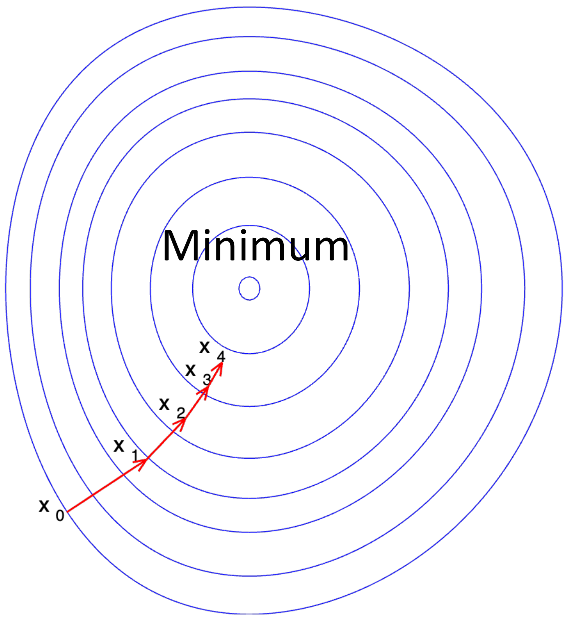
$$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \beta) = -\frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i \cdot \mathbf{x}_j) + \sum \alpha_i$$

# *Math Interlude II*

# Gradient descent

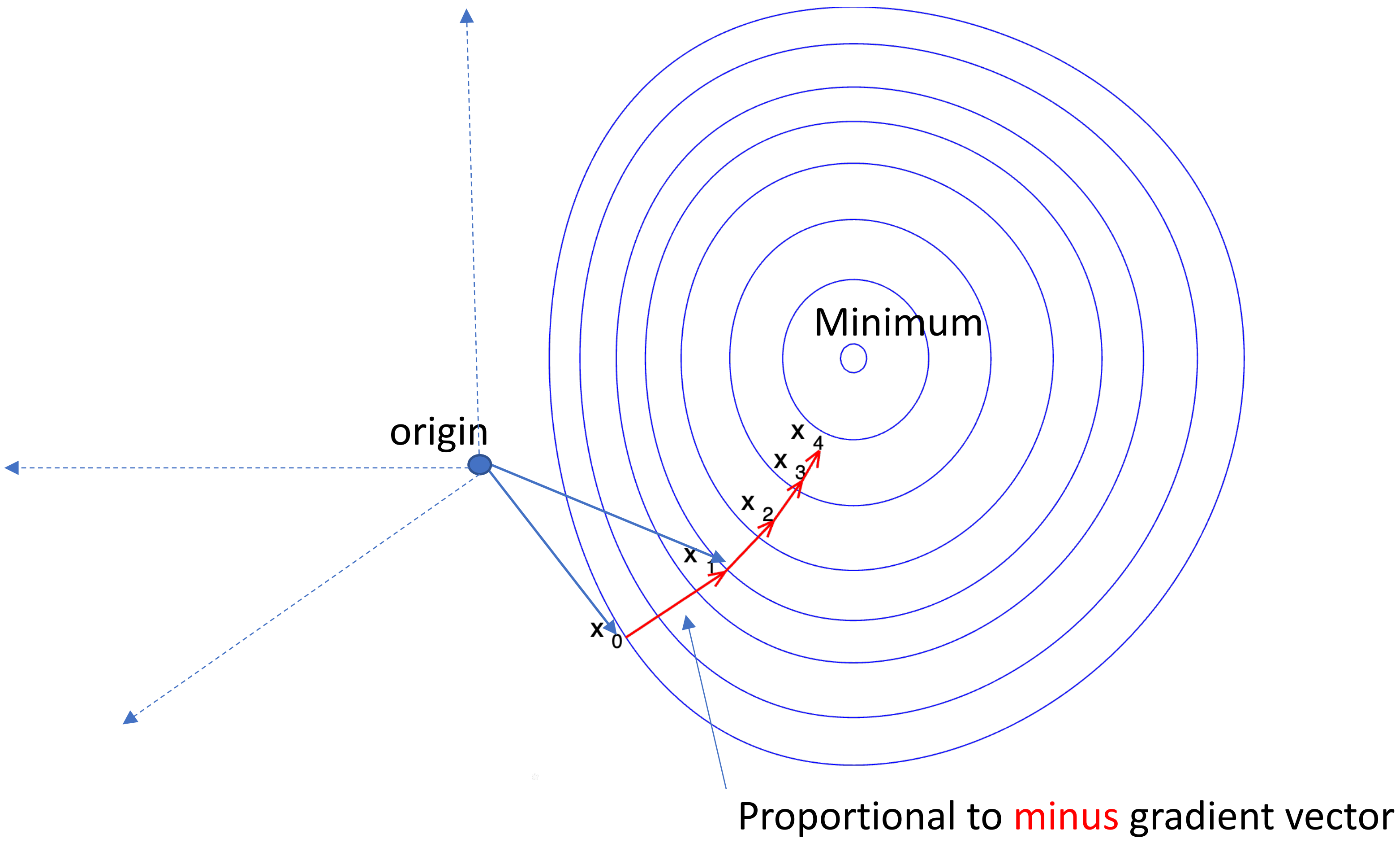
Theorem: the gradient vector is the direction of steepest ascent






By Gradient_descent.png: Zerodamage - This file was derived from:
Gradient descent.png:, Public Domain, https://commons.wikimedia.org/w/index.php?curid=20569355

# Gradient descent

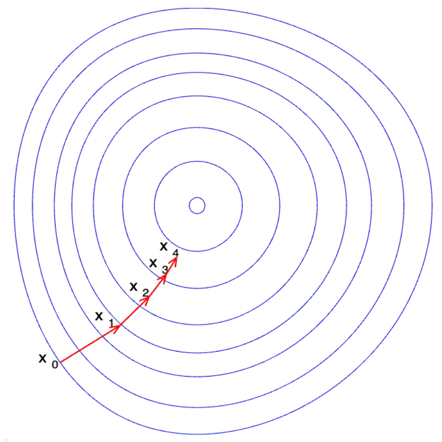



therefore $\mathbf{x}_{n+1} = \mathbf{x}_n - \gamma_n \nabla F(\mathbf{x}_n)$

# Gradient descent

How to find an optimal step?

There are many methods; if $F$ is differentiable, then:




Naïve form $$\mathbf{x}_{(n+1)} = \mathbf{x}_{(n)} - \gamma_{(n)} \nabla F(\mathbf{x}_{(n)})$$

Generalized form $$x_{k(n+1)} = x_{k(n)} - \gamma_{jk(n)} \partial_j F(\mathbf{x}_{(n)})$$

condition $$\partial_i F(x_{k(n+1)}) = \partial_i F(x_k - \gamma_{jk} \partial_j F(\mathbf{x})) \approx \partial_i F(\mathbf{x}) - \gamma_{jk} \partial_j F(\mathbf{x}) \partial_i \partial_k F(\mathbf{x}) = 0$$

$$\gamma_{jk} \partial_j F(\mathbf{x}) H_{ik} = \partial_i F(\mathbf{x})$$

solution $$\gamma_{jk} = H^{-1}_{jk}$$

therefore $$x_{k(n+1)} = x_{k(n)} - H^{-1}_{jk} \partial_j F(\mathbf{x}_{(n)})$$

Notice: it's no longer a steepest descent!
but still converges (mostly…)

# Comparing methods…

Comparing Supervised Learning Algorithms : Table

| Algorithm | Problem Type | Results interpretable by you? | Easy to explain algorithm to others? | Average predictive accuracy | Training speed | Prediction speed | Amount of parameter tuning needed (excluding feature selection) | Performs well with small number of observations? |
|---|---|---|---|---|---|---|---|---|
| KNN | Either | Yes | Yes | Lower | Fast | Depends on n | Minimal | No |
| Linear regression | Regression | Yes | Yes | Lower | Fast | Fast | None (excluding regularization) | Yes |
| Logistic regression | Classification | Somewhat | Somewhat | Lower | Fast | Fast | None (excluding regularization) | Yes |
| Naive Bayes | Classification | Somewhat | Somewhat | Lower | Fast (excluding feature extraction) | Fast | Some for feature extraction | Yes |
| Decision trees | Either | Somewhat | Somewhat | Lower | Fast | Fast | Some | No |
| Random Forests | Either | A little | No | Higher | Slow | Moderate | Some | No |
| AdaBoost | Either | A little | No | Higher | Slow | Fast | Some | No |
| Neural networks | Either | No | No | Higher | Slow | Fast | Lots | No |

Table

# More

- many nice videos about statistics https://www.tilestats.com/
- NN course https://www.3blue1brown.com/topics/neural-networks

Books:

- Acquaviva, Viviana: *Machine Learning for Physics and Astronomy*
- Alpaydın, Ethem: *Introduction to machine learning*
- Plaue, Matthias: *Data Science: an introduction to statistics and machine learning*