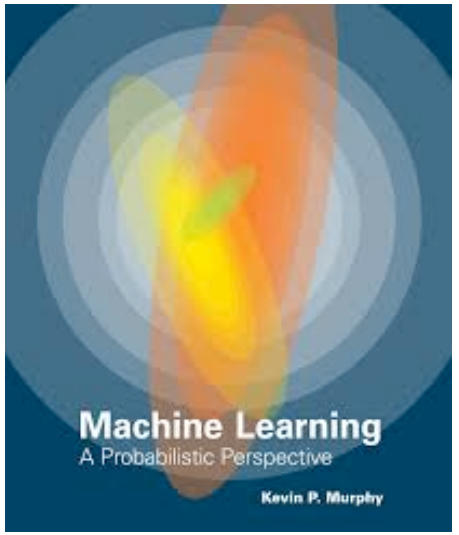
Machine Learning

"a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to **predict** future data, or perform other kinds of decision making" – Murphy 2012

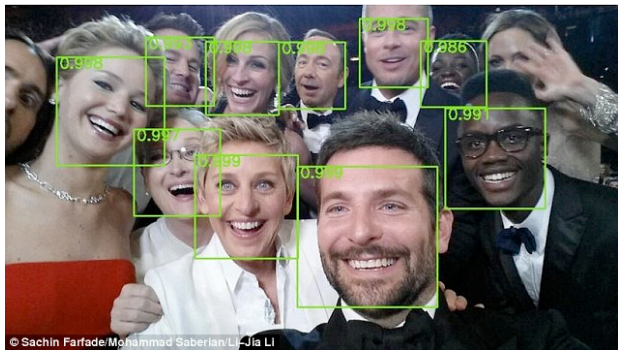


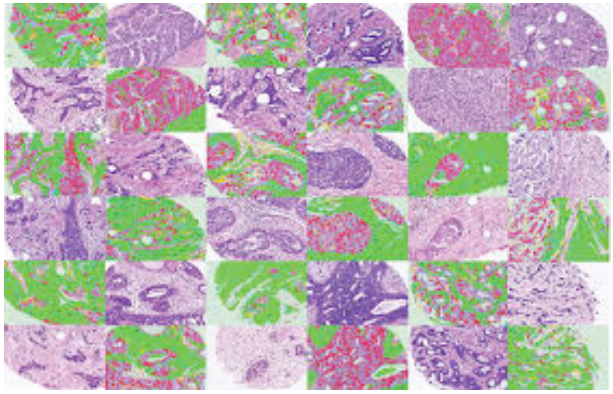
Classification
(one of the simplest types of prediction problems)

Goal: Learn to classify examples

E.g.,

- Images (face recognition)
- Emails (spam filtering)
- Biological samples (tumor classification)



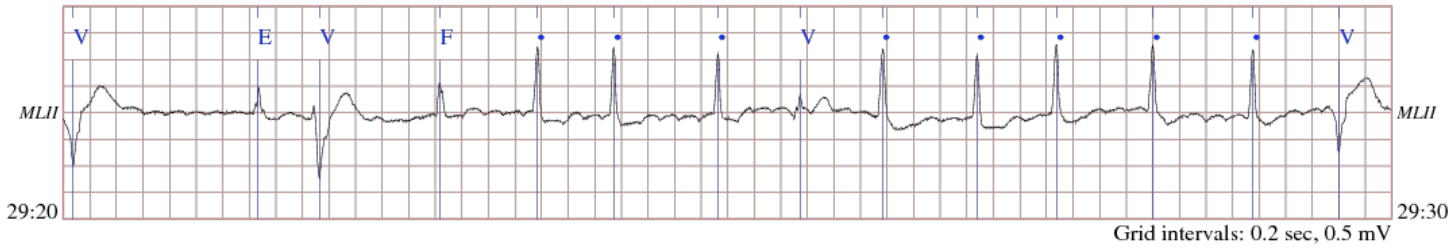


Heartbeat Classification

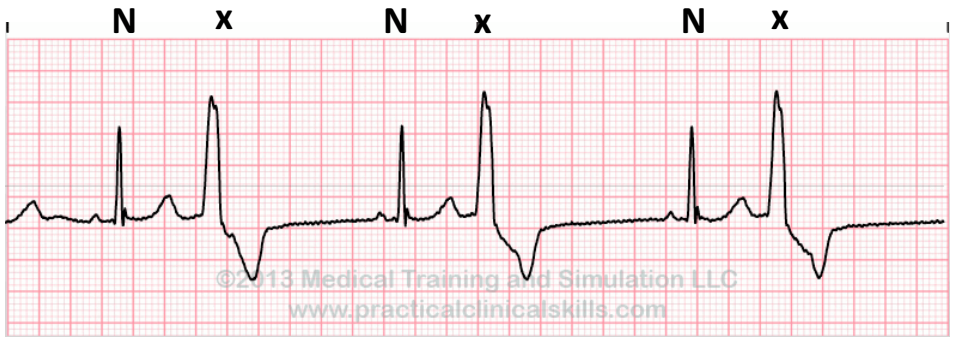
Motivation:

- The analysis of long-term ECG recordings can help physicians understand a patient's health.
- Labeling heartbeats can be an important step in this task; provides a level of abstraction.
- >100 000 beats in just 24 hours, so needs to be automated.

ECG Recording

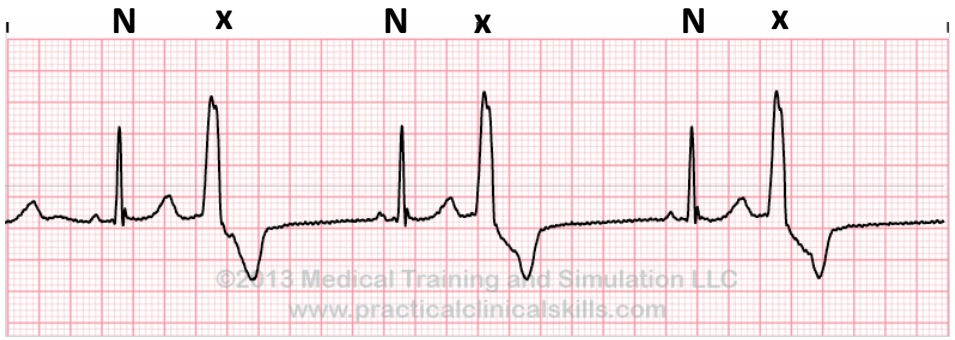


Example: Heartbeat Classification



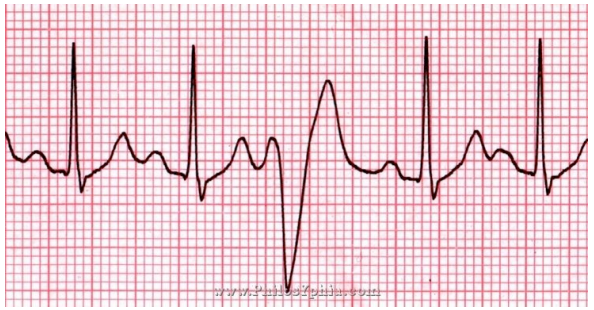
Labeled Training Data

Example: Heartbeat Classification



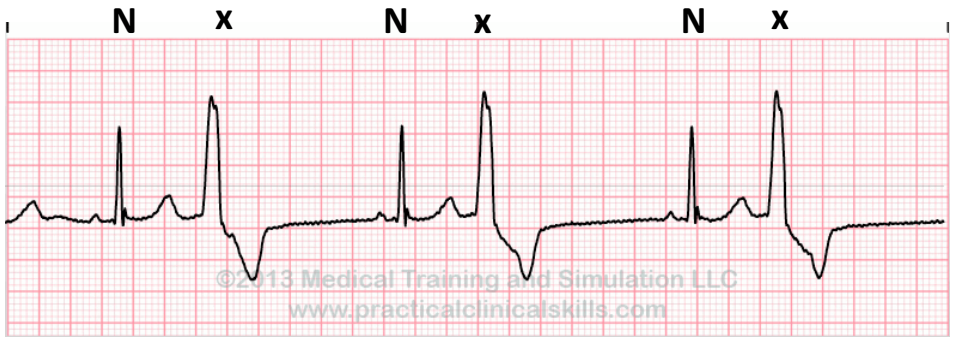
Labeled Training Data

New Patient Recording



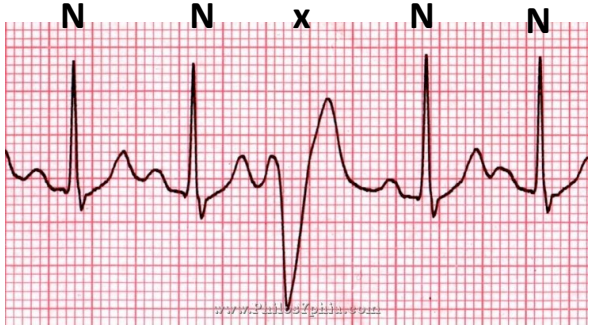
Task: Interpret data & make classifications

Example: Heartbeat Classification



Labeled Training Data

New Patient Recording



Task:
Interpret data & make classifications

Example: Heartbeat Classification

Labeled Feature Vector Representation

d-dimensional feature vector

binary label

$\rightarrow \{\overline{\mathbf{x}}^{(1)}, -1\}$

$\rightarrow \{\overline{\mathbf{x}}^{(2)}, +1\}$

Example: Heartbeat Classification

Labeled Feature Vector Representation

***d*-dimensional feature vector**

binary label

$\rightarrow \{\overline{\mathbf{x}}^{(1)}, -1\}$

$\rightarrow \{\overline{\mathbf{x}}^{(2)}, +1\}$

Possible information to encode as features:

-amplitude

-timing info/context

-frequency information

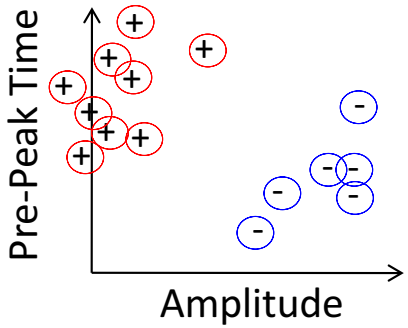
Example: Heartbeat Classification

Labeled Feature Vector Representation

$\rightarrow \{\overline{\mathbf{x}}^{(1)}, -1\}$

$\rightarrow \{\overline{\mathbf{x}}^{(2)}, +1\}$

FEATURE SPACE

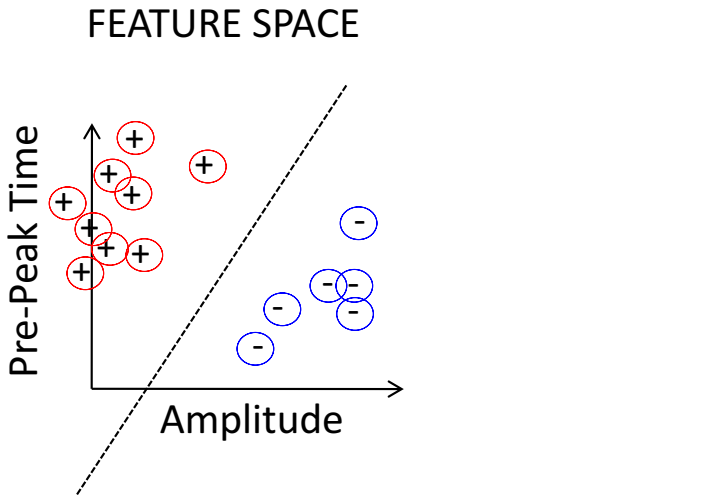


Example: Heartbeat Classification

Labeled Feature Vector Representation

$\rightarrow \{\overline{\mathbf{x}}^{(1)}, -1\}$

$\rightarrow \{\overline{\mathbf{x}}^{(2)}, +1\}$



Learning goal: identify the decision boundary that separates the positive from negative examples.

Supervised Learning

$$S_n = \{\bar{x}^{(i)}, y^{(i)}\}_{i=1}^{n} \quad \bar{x} \in \mathcal{X} \quad y \in \mathcal{Y}$$

Goal: learn a mapping from $\mathcal{X} \rightarrow \mathcal{Y}$ that generalizes to yet unseen data.

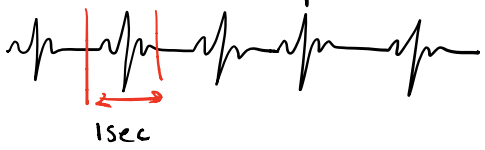
Lecture #1: Classification as an ML problem

- Feature vector $\bar{x} = [x_1, x_2, x_3, \ldots, x_d]^T$ $\quad \bar{x} \in \mathbb{R}^d$
- Labels $y \in \{-1, +1\}$ (binary)
- Training set of examples $S_n = \{\bar{x}^{(i)}, y^{(i)}\}_{i=1}^{n}$
- Classifier $h: \mathbb{R}^d \to \{-1, 1\}$

Goal: Select the best h from a set of possible classifiers $\mathcal{H}$ that would have the best chance of correctly classifying new examples

The problem of select h from $\mathcal{H}$ → solved by a learning algorithm typically an optimization problem with respect to S_n

Example: ECG data sampled at 360Hz



$\bar{x} = [x_1, x_2, x_3, \ldots, x_{360}]$, $\quad x_k \in \{1, \ldots, 2048\}$

- training examples $n = 50$ $\quad \{\bar{x}^{(i)}, y^{(i)}\}_{i=1}^{50}$ $\quad y \in \{-1, +1\}$
- x_3 is different in every sample

Given the small # of training examples, we can trivially come up with a solution that maps each beat to the correct label just based on a look up table that use x_3.

But is this a good classifier? no, it overfits

Generalization → works well on unseen examples

Problem: too many choices in $\mathcal{H}$, so many that we may end up choosing a classifier that does well on the specific training set but fails applied to new data

Solution: constrain $\mathcal{H}$, but... can't be too small either or we may end up unable to classify S_n 'underfit'

model selection: finding the right blalance

Linear Classification :

$\mathcal{H}$: thresholded linear mappings from feature vectors to labels

$$h(\bar{x};\theta) = \begin{cases} +1 & \bar{\theta}\cdot\bar{x} > 0 \\ -1 & \bar{\theta}\cdot x < 0 \end{cases} \quad \text{where } \bar{\theta}\in\mathbb{R}^d \quad \bar{\theta} = [\theta_1, \theta_2, \theta_3, \ldots, \theta_d]$$

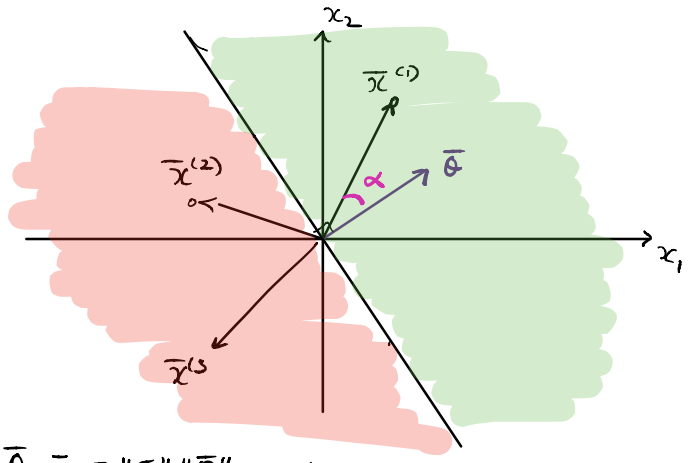
$$\bar{\theta}\cdot\bar{x} = \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \cdots + \theta_d x_d \leftarrow \text{a } \underline{\text{linear}} \text{ combination of the input features}$$

model parameters.

different $\bar{\theta}$s produce different labelings for some $\bar{x}$

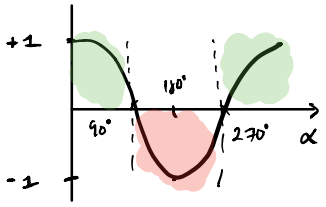
Geometrically

let $d = 2$



$$\bar{\theta}\cdot\bar{x} = \|\bar{x}\|\,\|\bar{\theta}\|\cos\alpha$$

recall: $\|\bar{x}\| = \sqrt{x_1^2 + x_2^2 + \cdots + x_d^2}$ "L2-norm" always ≥ 0



can be < 0, so α determines the sign of $\bar{\theta}\cdot\bar{x}$

What if a point lies on the decision boundary?

$$\bar{x}\cdot\bar{\theta} = \|\bar{x}\|\|\bar{\theta}\|\cos 90^\circ = 0$$

∴ the decision boundary can be defined as $\bar{x}\cdot\bar{\theta} = 0$

in 2-dimensions $\theta_1 x_1 + \theta_2 x_2 = 0 \Rightarrow x_2 = -\frac{\theta_1}{\theta_2} x_1$ ← $\bar{\theta}$ defines the slope of decision boundary.

How do we select $\bar{\theta}$?

Approach: find $\bar{\theta}$ that works well on the training data

$$S_n = \{\bar{x}^{(i)}, y^{(i)}\}$$

Minimize Training Error: fraction of examples for which the classifier predicts the wrong label.

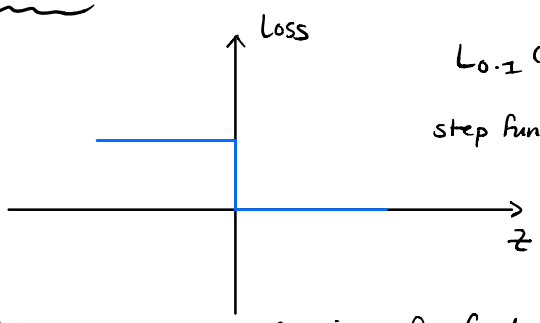
$$\mathcal{E}(\bar{\theta}) = \frac{1}{n}\sum_{i=1}^{n} [[y^{(i)} \neq h(\bar{x}^{(i)}; \theta)]]$$

$$= \frac{1}{n}\sum_{i=1}^{n} [[y^{(i)}(\bar{\theta}\cdot\bar{x}^{(i)}) \leq 0]]$$

by convention pts on the decision boundary are misclassified

$$= \frac{1}{n}\sum_{i=1}^{n} \text{Loss}_{0\text{-}1}(y^{(i)}(\bar{\theta}\cdot\bar{x}^{(i)}))$$

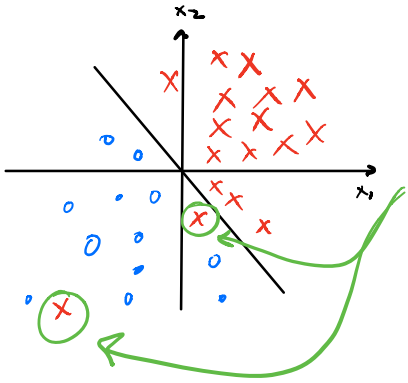
"zero-one loss"



$$L_{0\text{-}1}(z) = [[z \leq 0]] = \begin{cases} 1 & \text{if } z \leq 0 \\ 0 & \text{otherwise} \end{cases}$$

step function

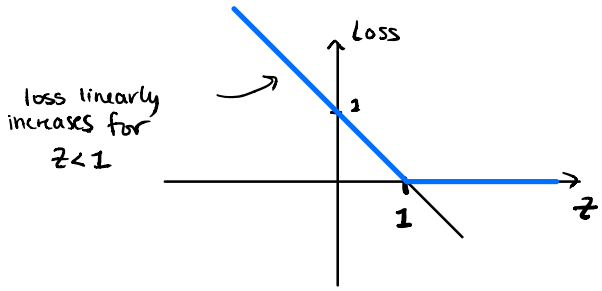
Unfortunately, a reasonable algorithm for finding $\bar{\theta}$ that minimizes training error is not easy to solve in general. So we will consider algorithms that approximately minimize training error.



Let's instead consider a loss function that treats these two errors differently.

"Hinge-Loss"

$$\text{Loss}_h(z) = \max\{1-z, 0\}$$



$z = y^{(i)}\,\bar{\theta}\cdot\bar{x}^{(i)}$

so if $y^{(i)}\,\bar{\theta}\cdot\bar{x}^{(i)} < 1$ we incur a cost

Advantage: forces predictions to be more than just correct $y^{(i)}\,\bar{\theta}\cdot\bar{x}^{(i)} \geq 1$

Approximate training error with hinge loss

$$R_n(\bar{\theta}) = \frac{1}{n}\sum_{i=1}^{n} \max\{1 - y^{(i)}\,\bar{\theta}\cdot\bar{x}^{(i)}, 0\}$$

Idea: by minimizing "empirical risk" we can obtain a classifier that generalizes well

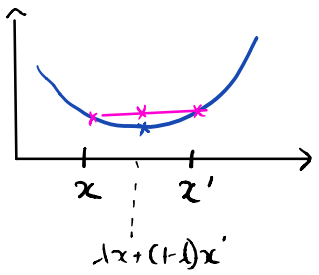
We can easily minimize $R_n(\bar{\theta})$ since it is a convex function, convexity guarantees there's a simple algorithm that will find the minimum

Recall: a convex function $f(x)$ is any function that

$f(\lambda x + (1-\lambda)x') \leq \lambda f(x) + (1-\lambda)f(x')$ for any x, x' $\lambda \in [0,1]$

- every chord lies above the function

e.g.



*characteristic bowl shape.

A simple algorithm for finding the minimum?

gradient descent, stochastic gradient descent

Gradient Descent

We will use gradient descent to minimize $R_n(\bar{\theta})$ with hinge loss.

* gradient points in the direction $R_n(\bar{\theta})$ increases

$$\nabla_{\bar{\theta}} R_n(\bar{\theta}) = \left[\frac{\partial R_n(\bar{\theta})}{\partial \theta_1}, \frac{\partial R_n(\bar{\theta})}{\partial \theta_2}, \frac{\partial R_n(\bar{\theta})}{\partial \theta_3}, \dots, \frac{\partial R_n(\bar{\theta})}{\partial \theta_d} \right]$$

Idea: take a small step in the opposite direction

$$\bar{\theta}^{(k+1)} = \bar{\theta}^{(k)} - \eta \nabla_{\bar{\theta}} R_n(\bar{\theta}) \Big|_{\bar{\theta} = \bar{\theta}^{(k)}}$$

(η: step size or learning rate)

recall $R_n(\bar{\theta}) = \frac{1}{n} \sum_{i=1}^{n} \max \{ 1 - y^{(i)} \bar{\theta} \cdot \bar{x}^{(i)}, 0 \}$

(summation involved in calculating gradient makes gradient descent slow)

Stochastic Gradient Descent

Idea: update $\bar{\theta}$ based on a small batch or a single point

$\bar{\theta}^{(0)} = \bar{0}$, $k = 0$

while convergence criteria not met:

randomly select $i \in \{1, \dots, n\}$

$$\bar{\theta}^{(k+1)} = \bar{\theta}^{(k)} - \eta \nabla_{\bar{\theta}} \text{Loss}_H \left(y^{(i)} \bar{\theta} \cdot \bar{x}^{(i)} \right) \Big|_{\bar{\theta} = \bar{\theta}^{(k)}}$$

Technicality: $R_n(\bar{\theta})$ with hinge loss is piecewise linear

what do we do?

when differentiable → no problem
when subdifferentiable → choose any gradient around the kink

Note: if $y^{(i)} \bar{\theta} \cdot \bar{x}^{(i)} > 1 \rightarrow$ loss is zero & no update

if $y^{(i)} \bar{\theta} \cdot \bar{x}^{(i)} \leq 1$

then $\nabla_{\bar{\theta}} \text{Loss}\left(y^{(i)} \bar{\theta} \cdot \bar{x}^{(i)}\right)$

$= \nabla_{\bar{\theta}} \left(1 - y^{(i)} (\bar{\theta} \cdot \bar{x}^{(i)})\right)$

$= -y^{(i)} \bar{x}^{(i)}$

$\therefore$ update rule: if $y^{(i)} \theta \cdot \bar{x}^{(i)} \leq 1$:

$$\bar{\theta}^{(k+1)} = \bar{\theta}^{(k)} + \eta\, y^{(i)} \bar{x}^{(i)}$$

Notes:

- (stochastic) gradient descent is a general algorithm that can be applied to non-convex functions
- SGD often gets closer to the solution more quickly than GD
- with appropriate learning rate if $R_n(\bar{\theta})$ is convex will almost surely converge to minimum