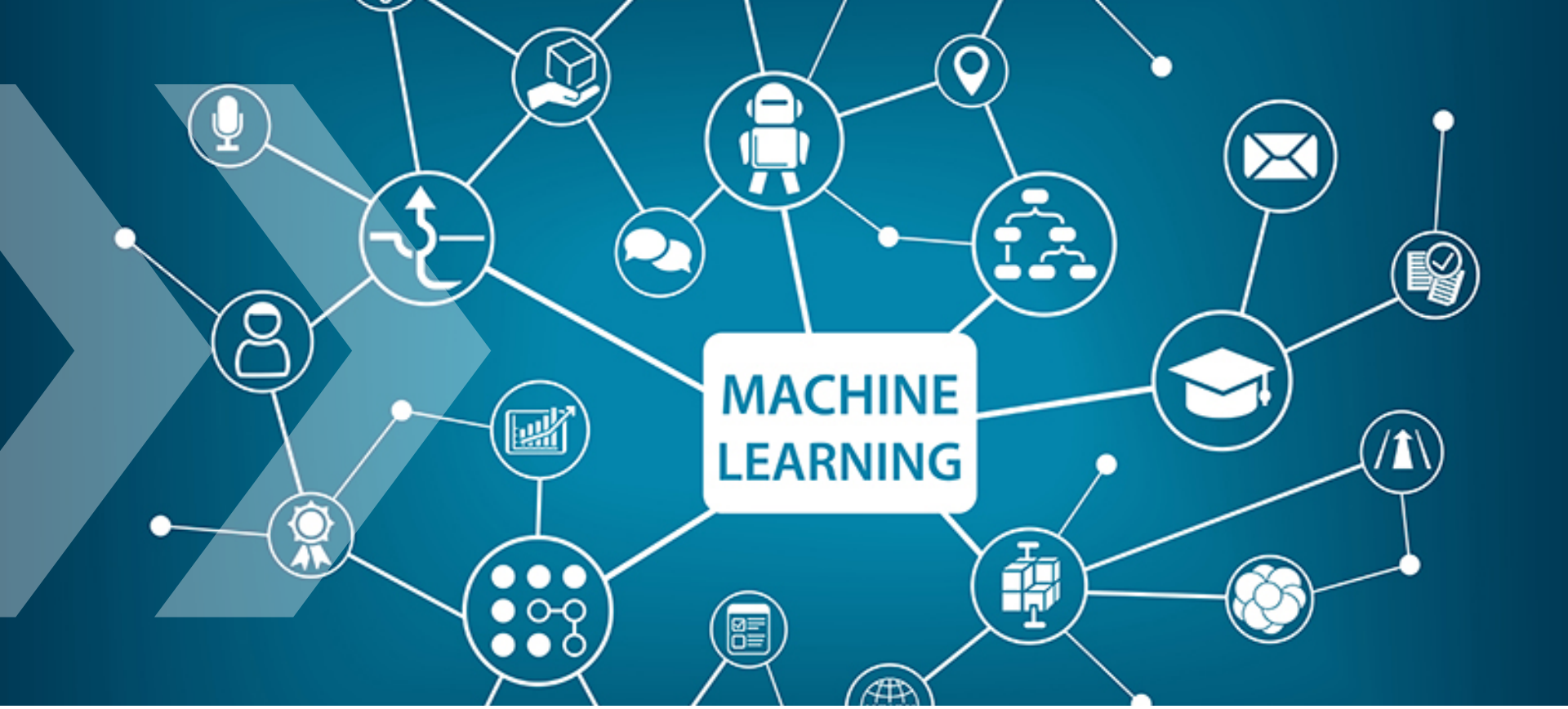


CS489/698: Intro to ML

Lecture 04: Logistic Regression

UNIVERSITY OF WATERLOO

Outline

- **Announcements**
- Baseline
- Learning “Machine Learning” Pyramid
- Regression or Classification (that’s it!)
- History of Classification
- History of Solvers (Analytical to Convex to “Non-Convex but smooth”)
- Convexity
- SGD
- Perceptron Review
- Bernoulli model / Logistic Regression
- Tensorflow Playground / Demo code
- Multiclass

UNIVERSITY OF WATERLOO

Announcements

- Assignment 1 due next Tuesday

UNIVERSITY OF WATERLOO

Outline

- Announcements
- **Baseline**
- Learning “Machine Learning” Pyramid
- Regression or Classification (that’s it!)
- History of Classification
- History of Solvers (Analytical to Convex to “Non-Convex but smooth”)
- Convexity
- SGD
- Perceptron Review
- Bernoulli model / Logistic Regression
- Tensorflow Playground / Demo code
- Multiclass

UNIVERSITY OF WATERLOO

Baseline

- Assuming Lin Alg Basics:

$$Ax = b$$

$$A = \begin{bmatrix} 4 & -5 \\ -2 & 3 \end{bmatrix}, \quad b = \begin{bmatrix} -13 \\ 9 \end{bmatrix}.$$

$$C = AB \in \mathbb{R}^{m \times p},$$

$$C_{ij} = \sum_{k=1}^{n} A_{ik} B_{kj}.$$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$(A^T)^T = A$$

$$(AB)^T = B^T A^T$$

$$(A + B)^T = A^T + B^T$$

$$\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}.$$

$$x^T Ax = \sum_{i=1}^{n} x_i (Ax)_i = \sum_{i=1}^{n} x_i \left(\sum_{j=1}^{n} A_{ij} x_j \right) = \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij} x_i x_j \quad .$$

$$x^T Ax = (x^T Ax)^T = x^T A^T x = x^T \left(\frac{1}{2} A + \frac{1}{2} A^T \right) x,$$

$$\nabla_A f(A) \in \mathbb{R}^{m \times n} = \begin{bmatrix} \frac{\partial f(A)}{\partial A_{11}} & \frac{\partial f(A)}{\partial A_{12}} & \cdots & \frac{\partial f(A)}{\partial A_{1n}} \\ \frac{\partial f(A)}{\partial A_{21}} & \frac{\partial f(A)}{\partial A_{22}} & \cdots & \frac{\partial f(A)}{\partial A_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f(A)}{\partial A_{m1}} & \frac{\partial f(A)}{\partial A_{m2}} & \cdots & \frac{\partial f(A)}{\partial A_{mn}} \end{bmatrix}$$

$$\nabla_x^2 f(x) \in \mathbb{R}^{n \times n} = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_2^2} & \cdots & \frac{\partial^2 f(x)}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \frac{\partial^2 f(x)}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{bmatrix}.$$

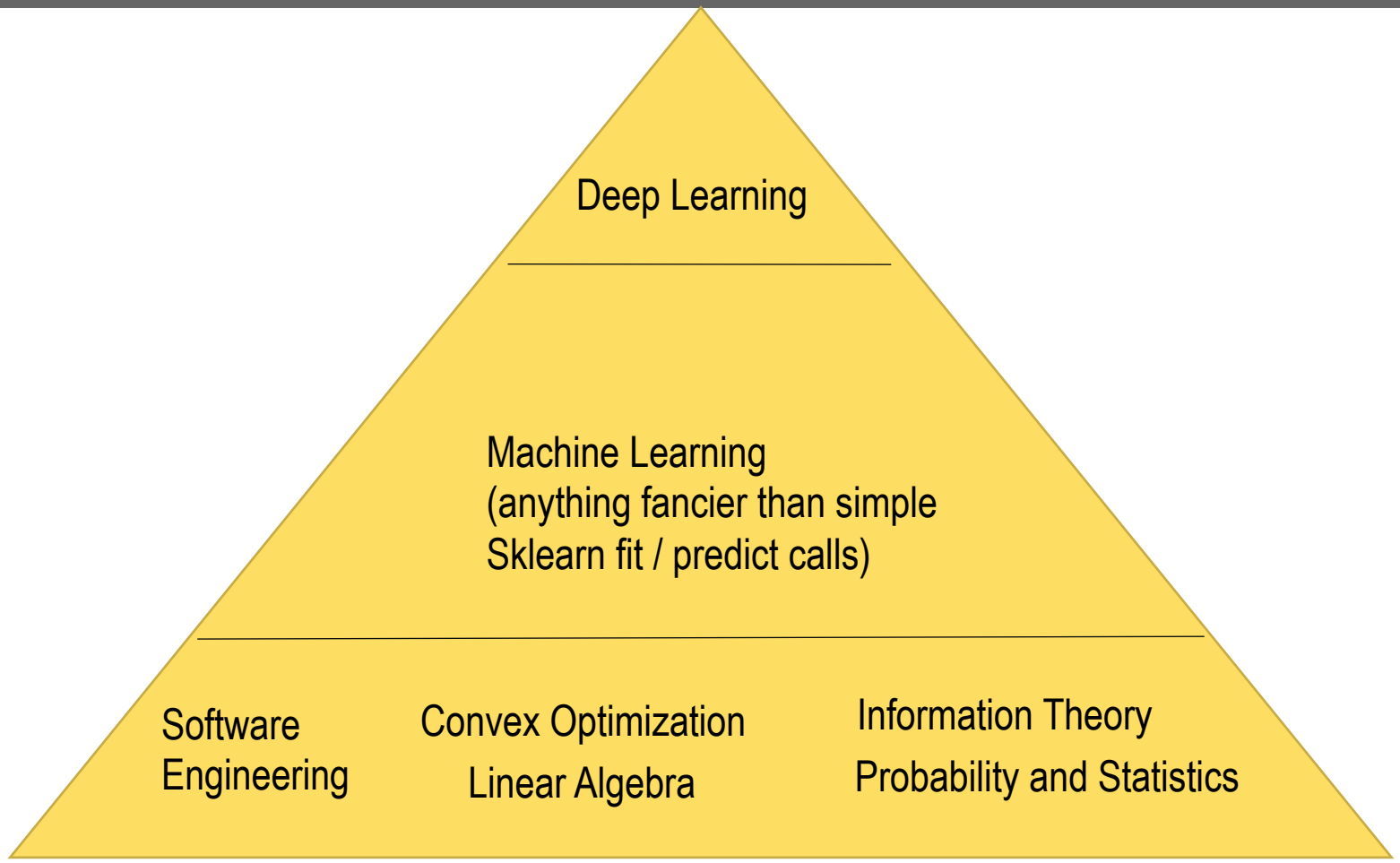
focal Systems

Outline

- Announcements
- Baseline
- **Learning "Machine Learning" Pyramid**
- Regression or Classification (that's it!)
- History of Classification
- History of Solvers (Analytical to Convex to "Non-Convex but smooth")
- Convexity
- SGD
- Perceptron Review
- Bernoulli model / Logistic Regression
- Tensorflow Playground / Demo code
- Multiclass

UNIVERSITY OF WATERLOO

ML Pyramid



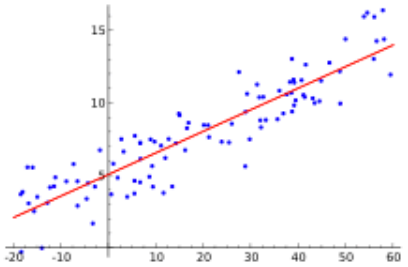
focal Systems

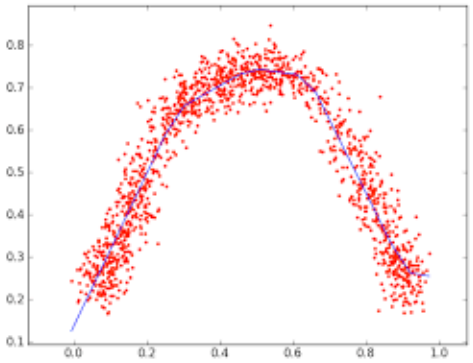
Outline

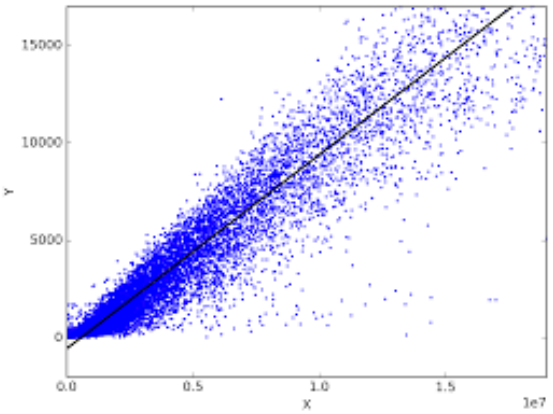
- Announcements
- Baseline
- Learning “Machine Learning” Pyramid
- **Regression or Classification (that’s it!)**
- History of Classification
- History of Solvers (Analytical to Convex to “Non-Convex but smooth”)
- Convexity
- SGD
- Perceptron Review
- Bernoulli model / Logistic Regression
- Tensorflow Playground / Demo code
- Multiclass

UNIVERSITY OF WATERLOO

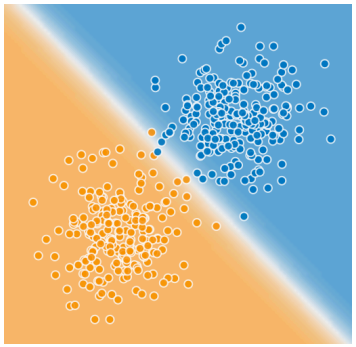
Regression or Classification

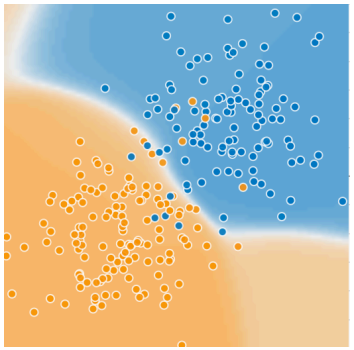


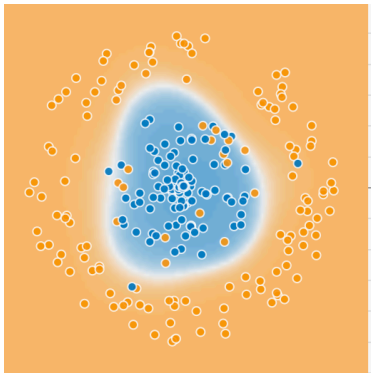


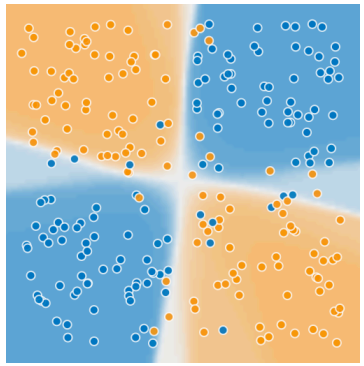


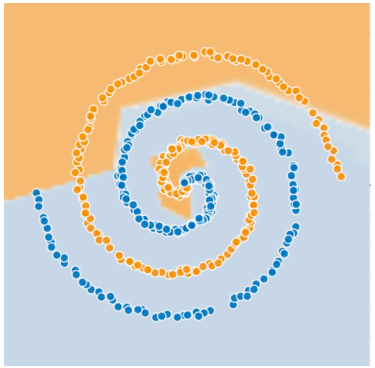
or











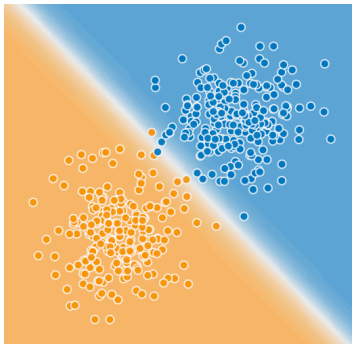
focal Systems

Outline

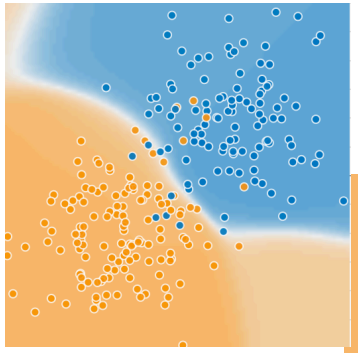
- Announcements
- Baseline
- Learning “Machine Learning” Pyramid
- Regression or Classification (that’s it!)
- **History of Classification**
- History of Solvers (Analytical to Convex to “Non-Convex but smooth”)
- Convexity
- SGD
- Perceptron Review
- Bernoulli model / Logistic Regression
- Tensorflow Playground / Demo code
- Multiclass

UNIVERSITY OF WATERLOO

Classification

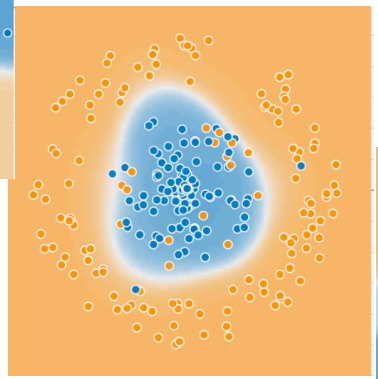


Perceptron



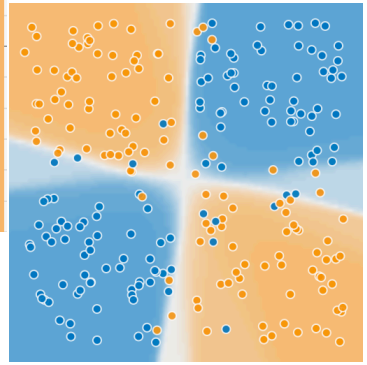
!Perceptron

Logistic Regression

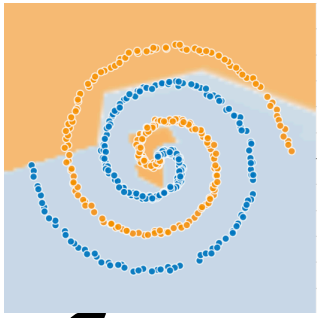


!Logistic Regression

MLP
Feature Eng
etc



etc



?

Higher “complexity” datasets need higher ”capacity” models
(Formally, higher VC-Dimensionality)

focal Systems

Outline

- Announcements
- Baseline
- Learning “Machine Learning” Pyramid
- Regression or Classification (that’s it!)
- History of Classification
- **History of Solvers (Analytical to Convex to “Non-Convex but smooth”)**
- Convexity
- SGD
- Perceptron Review
- Bernoulli model / Logistic Regression
- Tensorflow Playground / Demo code
- Multiclass

UNIVERSITY OF WATERLOO

History of Solvers

Closed Form <1950s

Least Squares

etc

Normal equation

$$\Theta = (X^T X)^{-1} X^T y$$

Iterative Methods+Convex 1950-2012

Interior Point Methods

Log Barrier

etc

Optimization problem in standard form

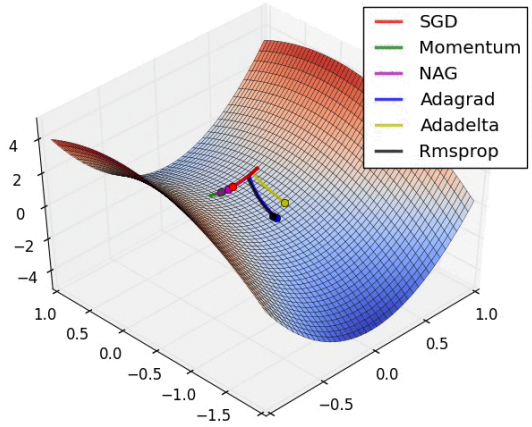
$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \\ & h_i(x) = 0, \quad i = 1, \ldots, p \end{array}$$

- $x \in \mathbf{R}^n$ is the optimization variable
- $f_0 : \mathbf{R}^n \to \mathbf{R}$ is the objective or cost function
- $f_i : \mathbf{R}^n \to \mathbf{R}$, $i = 1, \ldots, m$, are the inequality constraint functions
- $h_i : \mathbf{R}^n \to \mathbf{R}$ are the equality constraint functions

Iterative Methods+Smooth 2012+

Deep Learning

etc



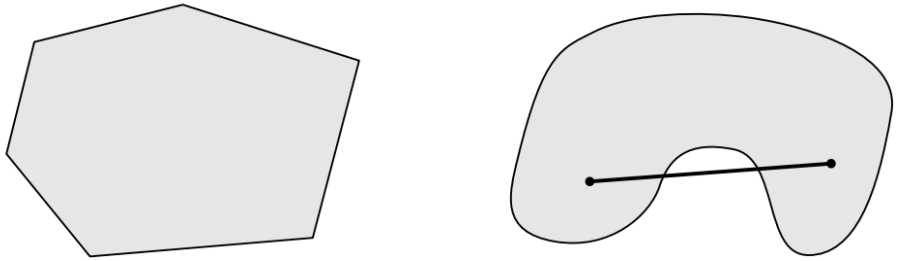
focal Systems

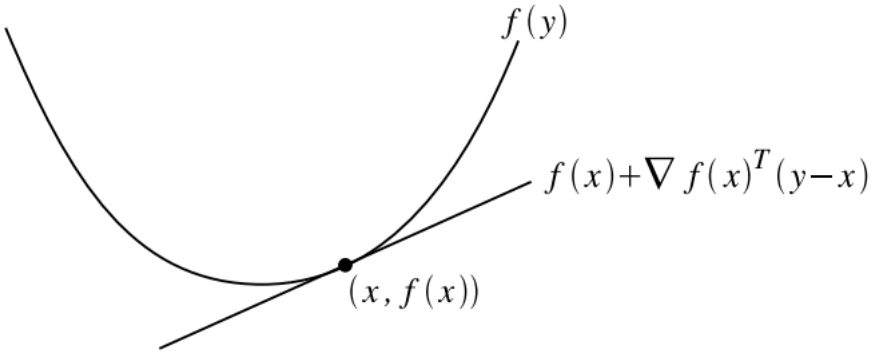
Outline

- Announcements
- Baseline
- Learning “Machine Learning” Pyramid
- Regression or Classification (that’s it!)
- History of Classification
- History of Solvers (Analytical to Convex to “Non-Convex but smooth”)
- **Convexity**
- SGD
- Perceptron Review
- Bernoulli model / Logistic Regression
- Tensorflow Playground / Demo code
- Multiclass

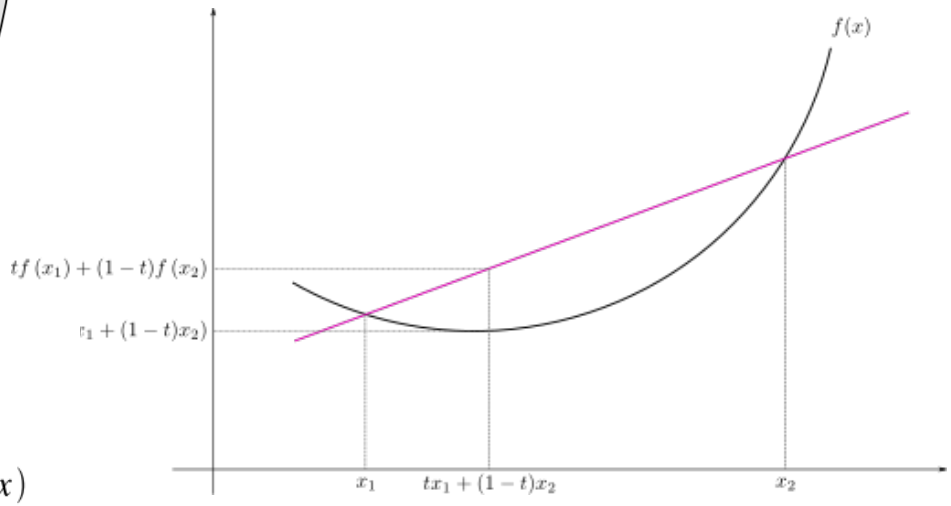
UNIVERSITY OF WATERLOO

Convexity





Jensen's Inequality



$$f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2).$$

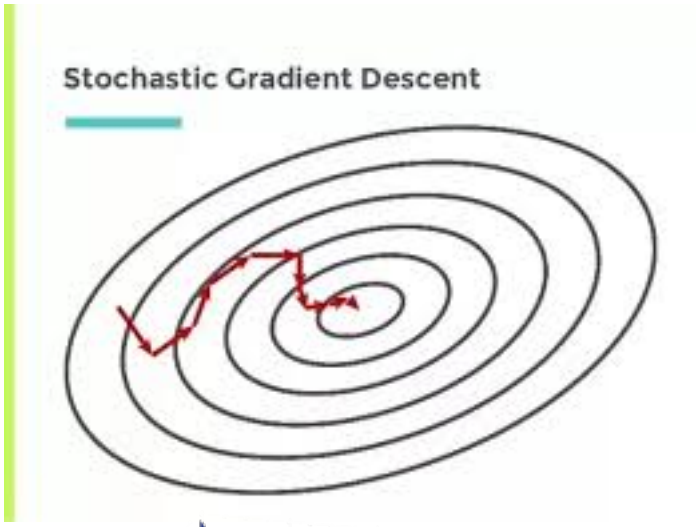
focal Systems

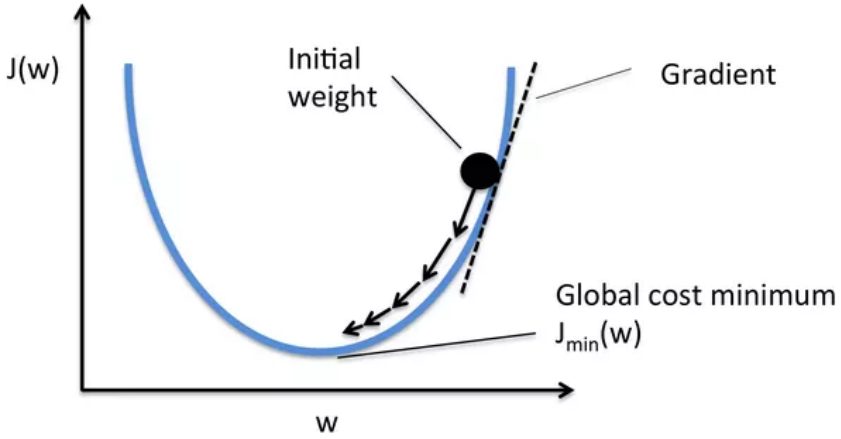
Outline

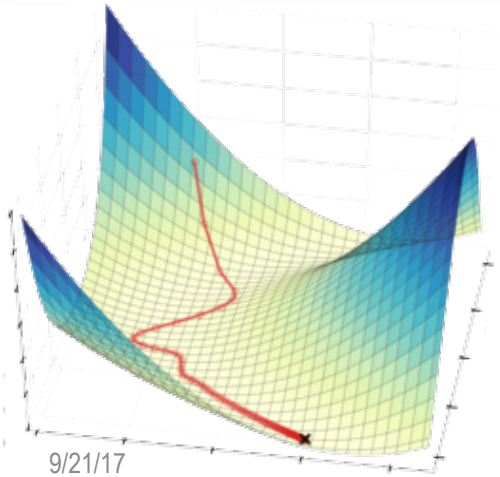
- Announcements
- Baseline
- Learning “Machine Learning” Pyramid
- Regression or Classification (that’s it!)
- History of Classification
- History of Solvers (Analytical to Convex to “Non-Convex but smooth”)
- Convexity
- **SGD**
- Perceptron Review
- Bernoulli model / Logistic Regression
- Tensorflow Playground / Demo code
- Multiclass

UNIVERSITY OF WATERLOO

SGD







$$w := w - \eta \nabla Q_i(w).$$

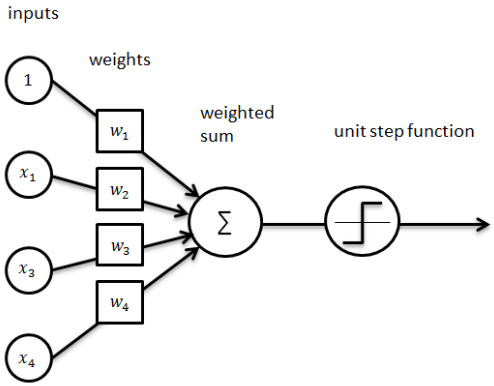
focal Systems

Outline

- Announcements
- Baseline
- Learning “Machine Learning” Pyramid
- Regression or Classification (that’s it!)
- History of Classification
- History of Solvers (Analytical to Convex to “Non-Convex but smooth”)
- Convexity
- SGD
- **Perceptron Review**
- Bernoulli model / Logistic Regression
- Tensorflow Playground / Demo code
- Multiclass

UNIVERSITY OF WATERLOO

Perceptron

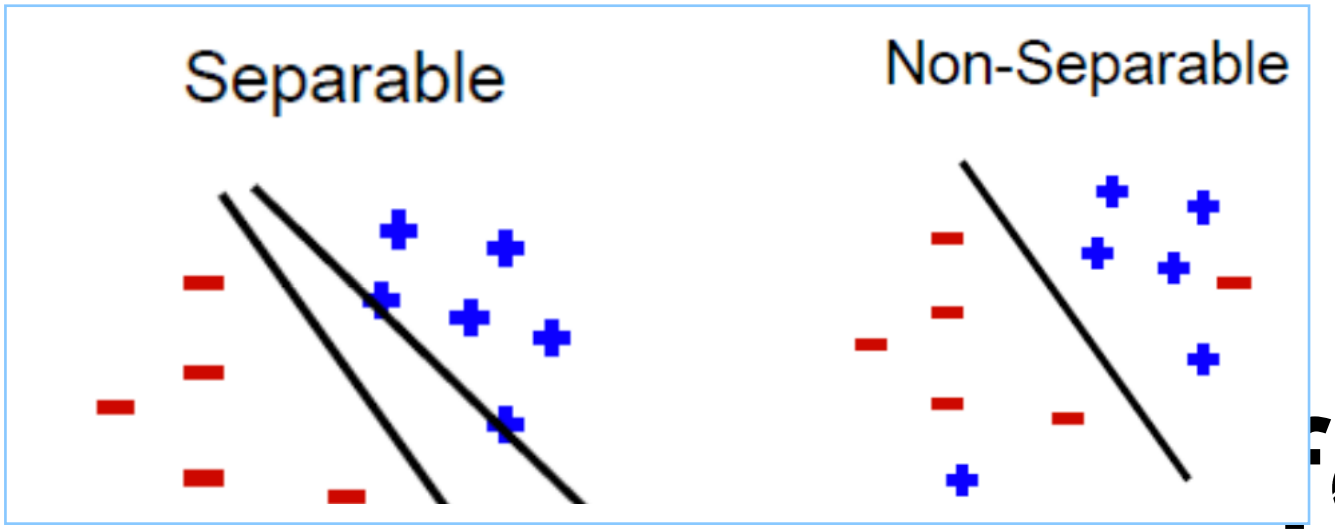


Algorithm 1: The perceptron algorithm (Rosenblatt 1958)

Input: $A \in \mathbb{R}^{(d+1)\times n}$, threshold $\delta \geq 0$, initialize $\mathbf{w}_0 \in \mathbb{R}^{d+1}$ arbitrarily

```
repeat
    select some column a of A;
    if ⟨a, w_t⟩ ≤ δ then
        w_{t+1} = w_t + a ;                    // up
        t ← t + 1
until convergence;
```

Issues:



focal Systems

Outline

- Announcements
- Baseline
- Learning “Machine Learning” Pyramid
- Regression or Classification (that’s it!)
- History of Classification
- History of Solvers (Analytical to Convex to “Non-Convex but smooth”)
- Convexity
- SGD
- Perceptron Review
- **Bernoulli model / Logistic Regression**
- Tensorflow Playground / Demo code
- Multiclass

UNIVERSITY OF WATERLOO

Bernoulli model

- Let P(Y=1 | X=**x**) = p(**x**; **w**), parameterized by **w**

- Conditional likelihood on {($\mathbf{x}_1$, y_1), … ($\mathbf{x}_n$, y_n)}:

$$\mathbf{P}(Y_1 = y_1, \ldots, Y_n = y_n | X_1 = \mathbf{x_1}, \ldots, X_n = \mathbf{x}_n)$$

  - simplifies if independence holds

$$\prod_{i=1}^{n} \mathbf{P}(Y_i = y_i | X_i = \mathbf{x}_i) = \prod_{i=1}^{n} p(\mathbf{x}_i; \mathbf{w})^{y_i} (1 - p(\mathbf{x}_i; \mathbf{w}))^{1-y_i}$$

  - Assuming y_i is {0,1}-valued

UNIVERSITY OF WATERLOO

Naïve solution

$$\prod_{i=1}^{n} p(\mathbf{x}_i; \mathbf{w})^{y_i} (1 - p(\mathbf{x}_i; \mathbf{w}))^{1-y_i}$$

- Find **w** to maximize conditional likelihood
- What is the solution if p(**x**; **w**) does not depend on **x**?
- What is the solution if p(**x**; **w**) does not depend on ?

UNIVERSITY OF WATERLOO

Generalized linear models (GLM)

- $y \sim$ Bernoulli(p); $p = p(\mathbf{x}; \mathbf{w})$ natural parameter
  - Logistic regression
- $y \sim$ Normal($\boldsymbol{\mu}, \sigma^2$); $\boldsymbol{\mu} = \boldsymbol{\mu}(\mathbf{x}; \mathbf{w})$
  - (weighted) least-squares regression
- GLM: $y \sim \exp(\theta\, \varphi(y) - A(\theta))$

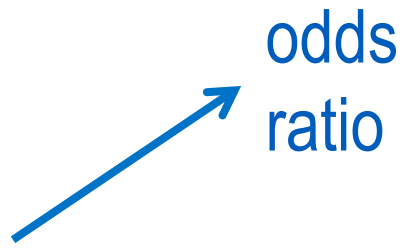
log-partition function

sufficient statistics

UNIVERSITY OF WATERLOO

Logit transform

- $p(\mathbf{x}; \mathbf{w}) = \mathbf{w}^\top \mathbf{x}$? p >=0 not guaranteed…
- $\log p(\mathbf{x}; \mathbf{w}) = \mathbf{w}^\top \mathbf{x}$? better!
  - LHS negative, RHS real-valued…
- Logit transform $\log \frac{p(\mathbf{x}; \mathbf{w})}{1 - p(\mathbf{x}; \mathbf{w})} = \mathbf{w}^\top \mathbf{x}$



- Or equivalently $p(\mathbf{x}; \mathbf{w}) = \frac{1}{1 + \exp(-\mathbf{w}^\top \mathbf{x})}$

UNIVERSITY OF WATERLOO

Prediction with confidence

$$p(\mathbf{x}; \mathbf{w}) = \frac{1}{1 + \exp(-\mathbf{w}^\top \mathbf{x})}$$

- $\hat{y} = 1$ if $p = P(Y=1 \mid X=x) > ½$ iff $\mathbf{w}^T\mathbf{x} > 0$
- Decision boundary $\mathbf{w}^T\mathbf{x} = 0$
- $\hat{y} = \text{sign}(\mathbf{w}^T\mathbf{x})$ as before, but with confidence $p(x; w)$

UNIVERSITY OF WATERLOO

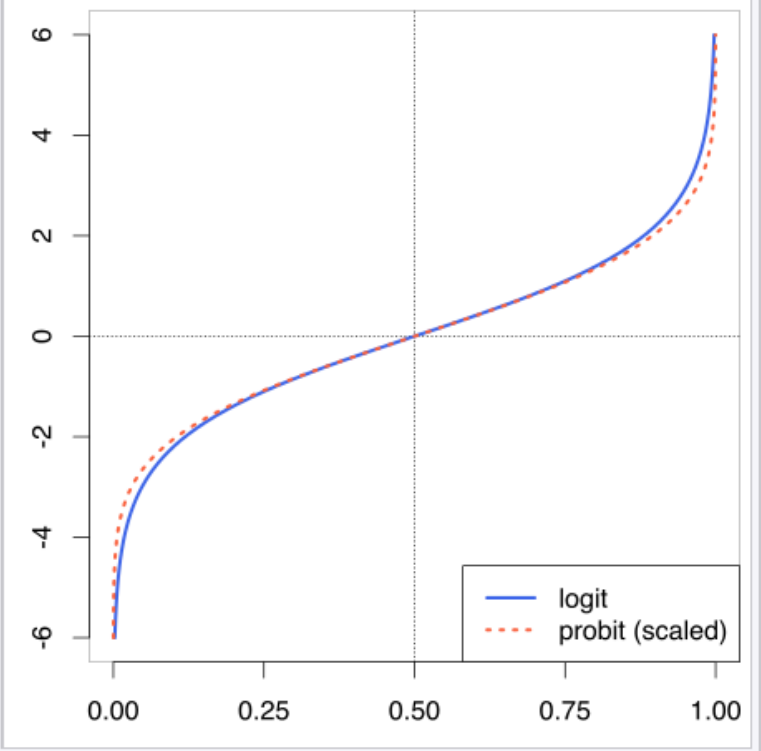
Not just a classification algorithm

- Logistic regression does more than classification
  - it estimates conditional probabilities
  - under the logit transform assumption
- Having confidence in prediction is nice
  - the price is an assumption that may or may not hold
- If classification is the sole goal, then doing extra work
  - as shall see, SVM only estimates decision boundary

UNIVERSITY OF WATERLOO

More than logistic regression

- F(p) transforms p from [0,1] to R
- Then, equating F(p) to a linear function $\mathbf{w}^T\mathbf{x}$
- But, there are many other choices for F!
  - precisely the inverse of any distribution function!



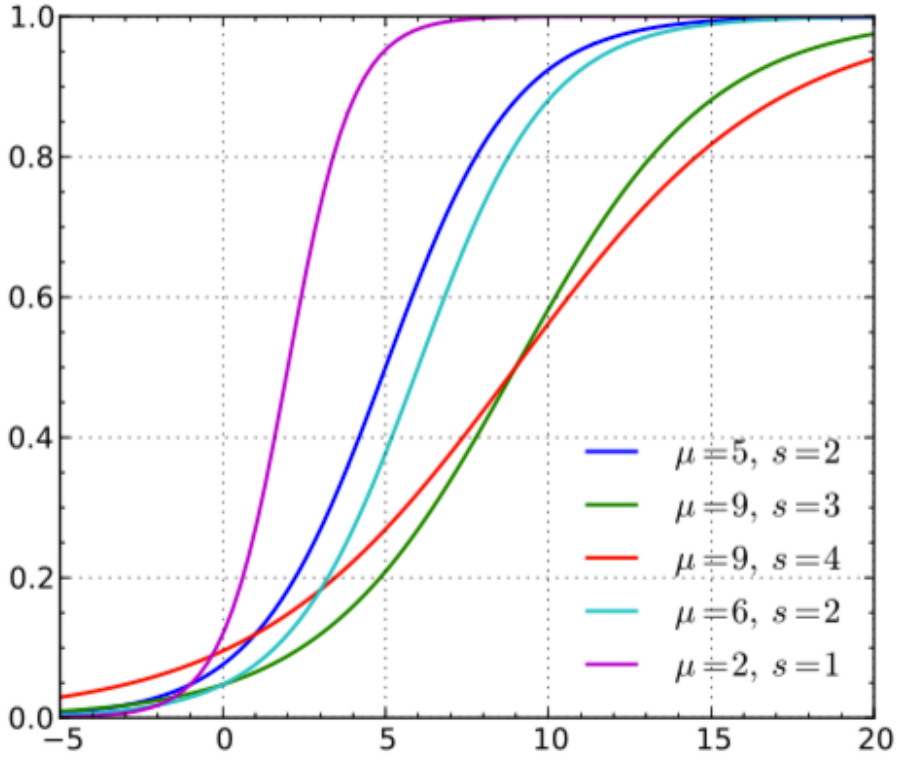
Comparison of the logit function with a scaled probit (i.e. the inverse CDF of the normal distribution), comparing $\operatorname{logit}(x)$ vs. $\Phi^{-1}(x)/\sqrt{\frac{\pi}{8}}$, which makes the slopes the same at the origin.

Logistic distribution

- Cumulative Distribution Function

$$F(x;\mu,s) = \frac{1}{1+\exp(-\frac{x-\mu}{s})}$$

- Mean mu, variance $s^2\pi^2/3$

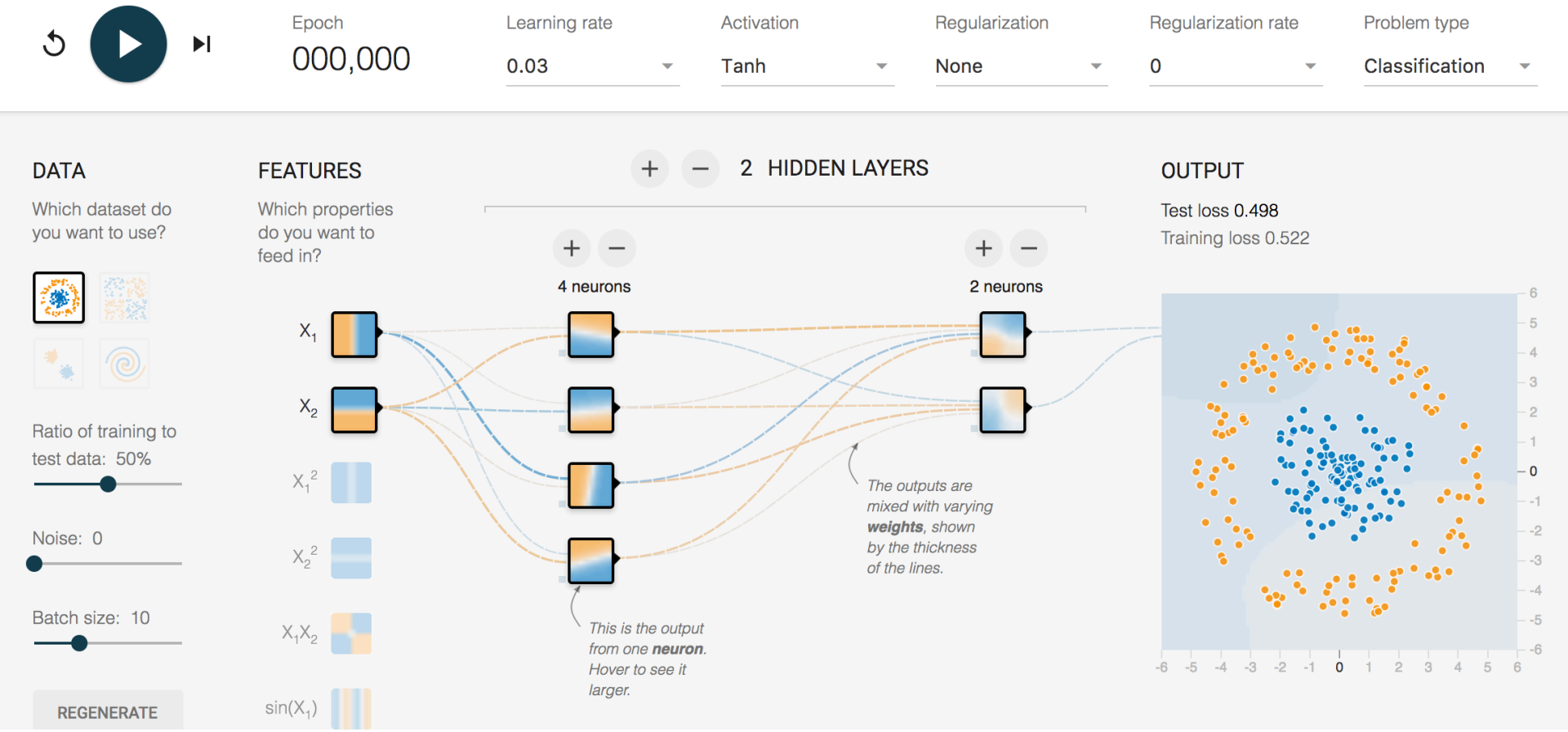


Outline

- Announcements
- Baseline
- Learning “Machine Learning” Pyramid
- Regression or Classification (that’s it!)
- History of Classification
- History of Solvers (Analytical to Convex to “Non-Convex but smooth”)
- Convexity
- SGD
- Perceptron Review
- Bernoulli model / Logistic Regression
- **Tensorflow Playground / Demo code**
- Multiclass

UNIVERSITY OF WATERLOO

Playground



focal Systems

Outline

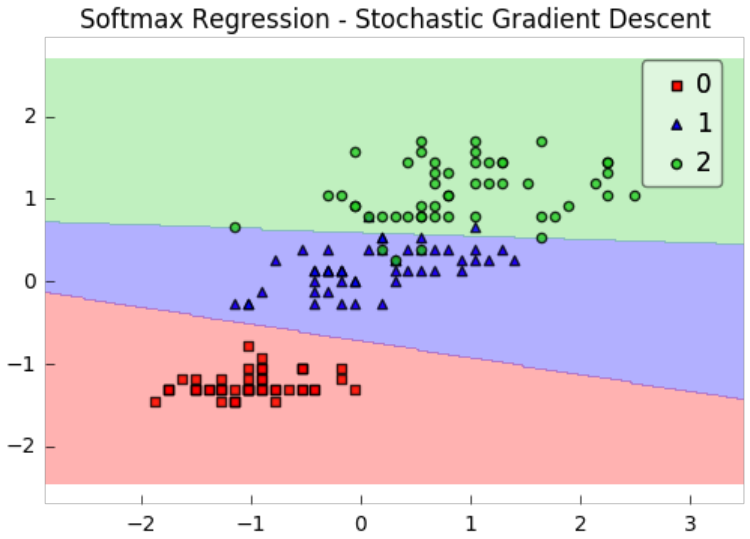
- Announcements
- Baseline
- Learning “Machine Learning” Pyramid
- Regression or Classification (that’s it!)
- History of Classification
- History of Solvers (Analytical to Convex to “Non-Convex but smooth”)
- Convexity
- SGD
- Perceptron Review
- Bernoulli model / Logistic Regression
- Tensorflow Playground / Demo code
- **Multiclass**

UNIVERSITY OF WATERLOO

More than 2 classes

- Softmax

$$\mathbf{P}(Y = c|\mathbf{x}, W) = \frac{\exp(\mathbf{w}_c^\top \mathbf{x})}{\sum_{q=1}^{k} \exp(\mathbf{w}_q^\top \mathbf{x})}$$



UNIVERSITY OF WATERLOO

More than 2 classes

- Softmax

$$\mathbf{P}(Y = c|\mathbf{x}, W) = \frac{\exp(\mathbf{w}_c^\top \mathbf{x})}{\sum_{q=1}^{k} \exp(\mathbf{w}_q^\top \mathbf{x})}$$

- Again, nonnegative and sum to 1

- Negative log-likelihood (y is one-hot)

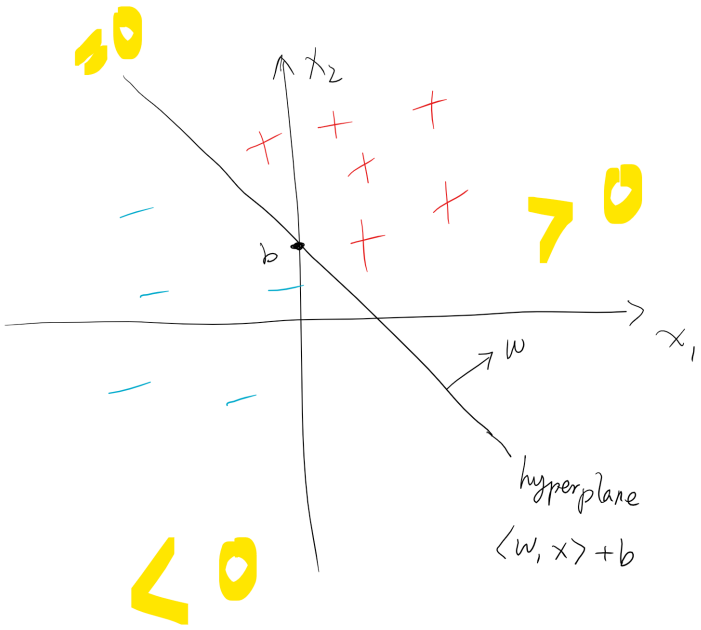
$$-\log \prod_{i=1}^{n} \prod_{c=1}^{k} p_{ic}^{y_{ic}} = -\sum_{i=1}^{n} \sum_{c=1}^{k} y_{ic} \log p_{ic}$$

UNIVERSITY OF WATERLOO

Questions?

Classification revisited

- $\hat{y} = \text{sign}(\mathbf{x}^T\mathbf{w} + b)$
- How confident we are about $\hat{y}$?
- $|\mathbf{x}^T\mathbf{w} + b|$ seems a good indicator
  - real-valued; hard to interpret
  - ways to transform into [0,1]
- Better(?) idea: learn confidence directly



UNIVERSITY OF WATERLOO

Conditional probability

- P(Y=1 | X=**x**): conditional on seeing **x**, what is the chance of this instance being positive, i.e., Y=1?
  - obviously, value in [0,1]

- P(Y=0 | X=**x**) = 1 – P(Y=1 | X=**x**), if two classes
  - more generally, sum to 1

Notation (Simplex). $\Delta_{k-1} := \{ \mathbf{p} \text{ in } R^k : \mathbf{p} \geq 0, \Sigma_i p_i = 1 \}$

UNIVERSITY OF WATERLOO

Reduction to a harder problem

- $P(Y=1 \mid X=\mathbf{x}) = E(1_{Y=1} \mid X=\mathbf{x})$

$$1_A = \begin{cases} 1, & A \text{ is true} \\ 0, & A \text{ is false} \end{cases}$$

- Let $Z = 1_{Y=1}$, then regression function for (X, Z)
  - use linear regression for binary Z?
- Exploit structure!
  - conditional probabilities are in a simplex
- Never reduce to unnecessarily harder problem

UNIVERSITY OF WATERLOO

Maximum likelihood

$$\prod_{i=1}^{n} p(\mathbf{x}_i; \mathbf{w})^{y_i} (1 - p(\mathbf{x}_i; \mathbf{w}))^{1-y_i}$$

$$p(\mathbf{x}; \mathbf{w}) = \frac{1}{1 + \exp(-\mathbf{w}^\top \mathbf{x})}$$

- Minimize negative log-likelihood

$$\sum_i \log(e^{(1-y_i)\mathbf{w}^\top \mathbf{x}_i} + e^{-y_i \mathbf{w}^\top \mathbf{x}_i}) \equiv \sum_i \log(1 + e^{-\tilde{y}_i \mathbf{w}^\top \mathbf{x}_i})$$

UNIVERSITY OF WATERLOO

Newton's algorithm

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \eta_t [\nabla^2 f(\mathbf{w}_t)]^{-1} \cdot \nabla f(\mathbf{w}_t)$$

$$\nabla f(\mathbf{w}_t) = X^\top (\mathbf{p} - \mathbf{y})$$

$$\nabla^2 f(\mathbf{w}_t) = \sum_i p_i (1 - p_i) \mathbf{x}_i \mathbf{x}_i^\top$$

PSD

$$p_i = \frac{1}{1 + e^{-\mathbf{w}_t^\top \mathbf{x}_i}}$$

Uncertain predictions get bigger weight

- η = 1: iterative weighted least-squares

UNIVERSITY OF WATERLOO

A word about implementation

- Numerically computing exponential can be tricky
  - easily underflows or overflows
- The usual trick
  - estimate the range of the exponents
  - shift the mean of the exponents to 0

UNIVERSITY OF WATERLOO

Robustness

$$\ell(t) = \log(1 + e^t) \quad L(\hat{y}, y) = \ell(-\hat{y}y) \quad \hat{y} = \mathbf{w}^\top \mathbf{x}$$

- Bounded derivative

$$\ell'(t) = \frac{e^t}{1 + e^t} = \frac{1}{1 + e^{-t}}$$

- Variational exponential

Larger exp loss gets smaller weights

$$\log(1 + e^t) = \min_{0 \le \eta \le 1} \eta e^t - \log(\eta) + \eta - 1$$

UNIVERSITY OF WATERLOO