Introduction to machine learning

Maksim Kretov

Lecture 4: Examples of methods

5vision, 2017

Course information

Course

10 lectures + 2 seminars; February-May 2017.

Schedule and up-to-date syllabus

https://goo.gl/xExEuL

Contact information and discussion

Maksim Kretov (kretovmk@gmail.com)

Slack group: https://miptmlcourse.slack.com

to get an invite, send e-mail to kretovmk@gmail.com.

Plan of the course

Math and basics of ML	(1-2)		*Theoretical tasks*
Some of ML methods	(3)		
Seminar on ML basics	(4)	← **Today**	
Basics of neural networks	(5)	← **Start playing with NNs**	*Practical tasks*
Deep learning overview	(6)		
Training deep networks	(7)		
DL for Computer Vision	(8-9)	**Solving more complex ML tasks using NNs**	
DL for time series prediction	(10-11)		
Concluding seminar	(12)		

Plan for the lecture

A. Previous lecture

B. Classification task

C. Discriminative models

   1. Logistic regression

D. Generative models

   1. Linear Discriminant Analysis

E. Conclusions

F. Homework

A. Previous lecture

Machine learning tasks: Supervised learning

Given: $D = \{(\mathbf{x}_i, y_i), i = 1, .. N\}$

Desired output: policy $\delta: D \rightarrow A$

Bayesian approach
Parameters θ of $p(\mathbf{x}, y|\theta)$ are treated as random variables.

Frequentist approach
Parameters θ of $p(\mathbf{x}, y|\theta)$ are unknown but fixed values.

High-level steps:

1. Select model M parameterized by parameters θ => $p(\mathbf{x}, y|M, \theta)$
2. Infer best θ that explains given dataset D OR calculate posterior distribution
3. Specify loss function $L(y, a)$
4. Design decision procedure δ

A. Previous lecture

Bayesian decision theory

Use accuracy hereinafter:

$$L(y,a) = \begin{cases} 0 \;\; if \;\; y = a \\ 1 \;\; if \;\; y \neq a \end{cases}$$

Corresponding policy:

$\delta(\mathbf{x}) = \mathrm{argmax}_{y \in Y} p(y|\mathbf{x}, \theta)$ Bayesian classification rule

Point estimates (MAP, MLE): $p(y|\mathbf{x}, \theta)$

Posterior distribution: $p(y|\mathbf{x}, D)$

B. Classification task

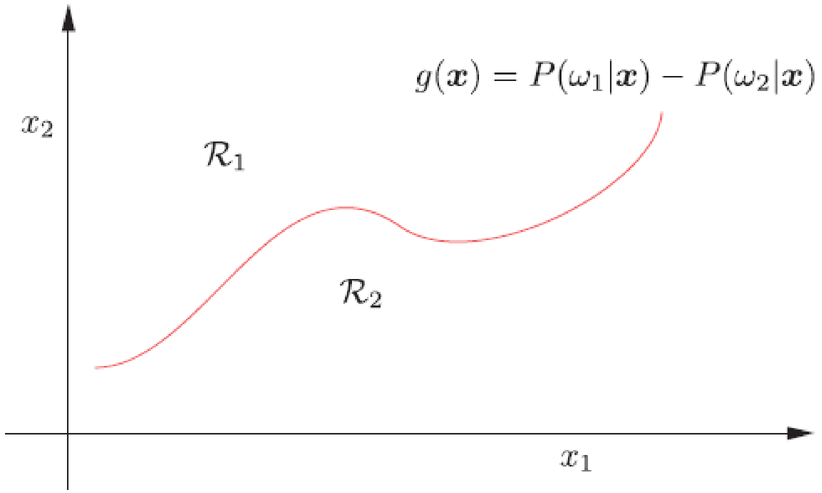
Binary classification task

Classifier parts feature space into regions by partition function $g\colon \mathrm{R}^l \to \mathrm{R}$

$g(x) = p(c = 1|\mathrm{x}) - p(c = 0|\mathrm{x})$

Class 1: $g(x) > 0$

Class 2: $g(x) < 0$



Discriminative and generative classifiers

$p(\mathbf{x}, y|\tilde{\theta}) = p(y|\mathbf{x}, \theta)p(\mathbf{x}|\pi)$ discriminative

$p(\mathbf{x}, y|\tilde{\theta}) = p(\mathbf{x}|y, \theta)p(y|\pi)$ generative

* Image from [5]

B.1 Discriminative models: Logistic regression

Model specification for binary classification

Discriminative classifier => directly models $p(y|\mathbf{x}, \theta)$:

$p(y|\mathbf{x}, \theta) = \mathrm{Ber}(y|\sigma(\theta^T\mathbf{x}))$ $\sigma(x) = \frac{1}{1+\exp(-x)}$ sigmoid function

$p(y = 1|\mathbf{x}, \theta) = \sigma(\theta^T\mathbf{x})$ => $\ln\frac{p(y=1|\mathbf{x},\theta)}{p(y=0|\mathbf{x},\theta)} = \theta^T\mathbf{x}$ ratio of posteriors (log odds)

Advantages

- Relatively easy to fit
- Easy to extend to multi-class tasks
- Clear interpretation (using log odds)

B.1 Discriminative models: Logistic regression

Model fitting: MLE

Consider negative log likelihood:

$$\mathrm{NLL}(\theta) = -\sum_i \log[(\sigma(\theta^T \mathbf{x}_i))^{y_i}(1-\sigma(\theta^T \mathbf{x}_i))^{1-y_i}] =$$

$$= -\sum_i y_i \log s_i + (1-y_i)\log(1-s_i)$$

binary cross-entropy

$$s_i \triangleq \sigma(\theta^T \mathbf{x}_i)$$

Note: That means we can use cross-entropy as a loss function and receive the same algorithm in ERM framework.

=> Minimization of NLL w.r.t. θ is performed iteratively, for example by gradient descent (steepest descent).

B.1 Discriminative models: Logistic regression

Optimization of parameters: gradient descent

Gradient: $g = \frac{\partial \mathrm{NLL}(\theta)}{\partial \theta} = \mathbf{X}^T(s - y)$

Hessian: $H = \frac{\partial g^T}{\partial \theta} = \mathbf{X}^T R \mathbf{X}$ $\qquad R = diag\{s_i(1 - s_i)\}$

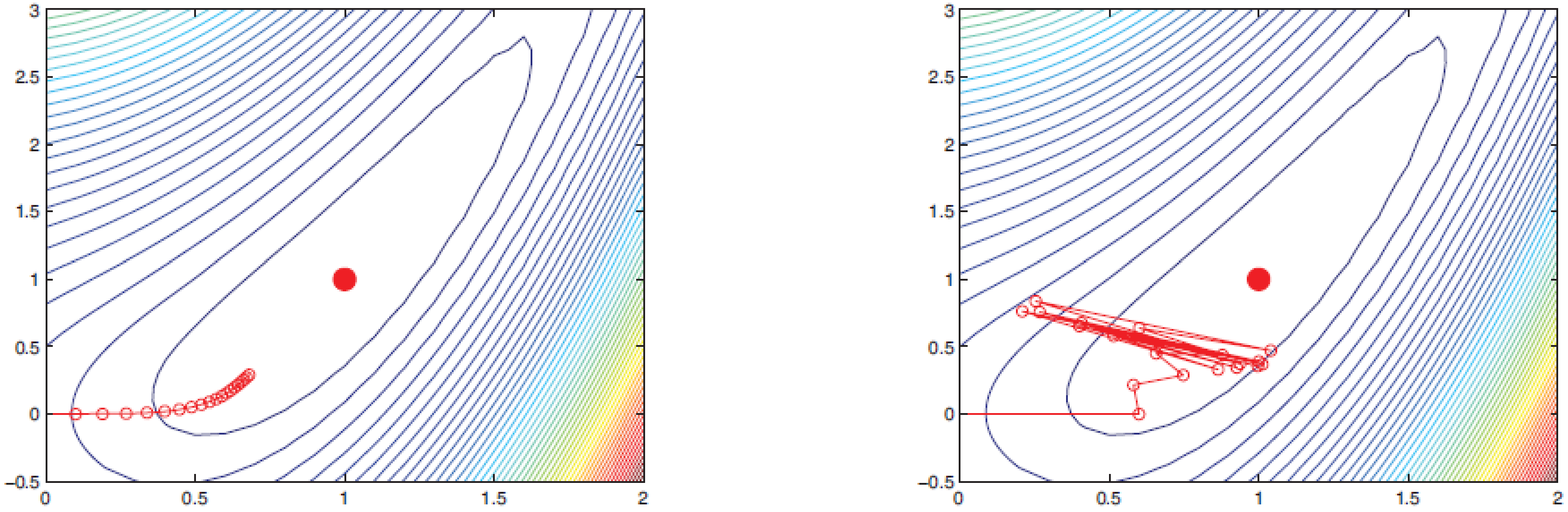
H is positive semi-definite, so $\mathrm{NLL}(\theta)$ has **unique minimum**.

=> Gradient descent: $\theta^{i+1} = \theta^i - \alpha^i g^i = \theta^i - \alpha^i \mathbf{X}^T\left(s^i - y\right)$

Exercise: *Prove that Hessian is positive semi-definite (just use definition).*

B.1 Discriminative models: Logistic regression

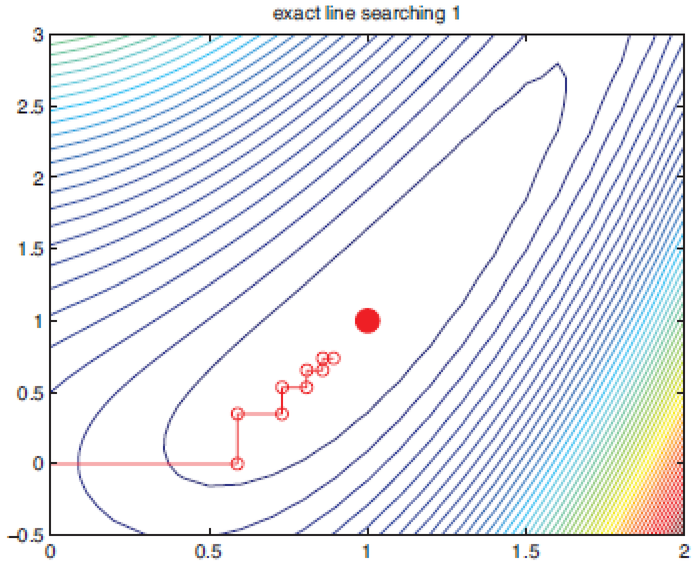
Gradient descent: learning rate



* Image from [3]

B.1 Discriminative models: Logistic regression

Gradient descent: line search for learning rate



Taylor series expansion:

$f(\theta + \alpha d) \approx f(\theta) + \alpha g^T d$

In case of gradient descent:

$d = -g$

=> If we choose α small enough, then

$f(\theta + \alpha d) < f(\theta)$

We can pick such α to minimize $f\left(\theta^i + \alpha d^i\right)$

=> Line search in direction d^i

* Image from [3]

B.1 Discriminative models: Logistic regression

Other 1st order methods

Momentum: $\theta^{i+1} = \theta^i - \alpha^i g^i + \beta_k(\theta^i - \theta^{i-1})$

Second order methods

Newton's algorithm:

$\theta^{i+1} = \theta^i - \alpha^i H_i^{-1} g^i$

=> Quasi-Newton methods (for example, BFGS).

***Exercise:** Prove formula for Newton's algorithm.*

Decision policy

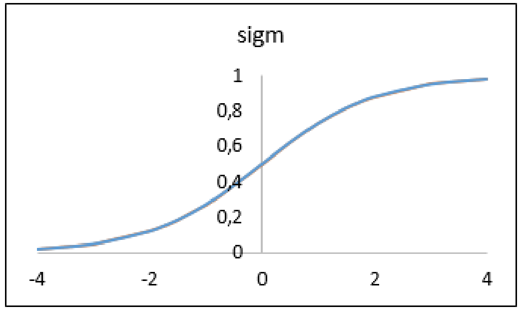
Follows directly from Bayesian classification rule: select $\mathrm{argmax}_{y \in Y} p(y|\mathbf{x}, \theta)$

* Image from [3]

B.1 Discriminative models: Logistic regression

Regularization

If classes are linearly separable then <u>sigmoid</u> → <u>step function</u> according to MLE, and $||\theta|| \rightarrow \infty$ => w/o regularization this results in overfitting.



$$\sigma(x) = \frac{1}{1+\exp(-x)}$$

Multi-class logistic regression

$$p(y = c|\mathbf{x}, \theta) = \frac{\exp \theta_c^T \mathbf{x}}{\sum_{c'} \exp \theta_{c'}^T \mathbf{x}} \quad \varphi_{ic} = p(y = c|\mathbf{x}_i, \theta)$$

$$\mathrm{NLL}(\theta_1, \theta_2, .. \theta_M) = -\sum_i \sum_c y_{ic} \log \varphi_{ic} \Rightarrow g_c = \frac{\partial \mathrm{NLL}}{\partial \theta_c} = \sum_i (\varphi_{ic} - y_{ic}) \mathbf{x}_i$$

<u>Exercise:</u> *<u>Prove formulas multiclass regression.</u>*

* Image from [3]

10 minute break..

C.1 Generative models: LDA

Multivariate normal distribution (MVN)

$$N(\mathbf{x}|\mu,\Sigma) \triangleq \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right]$$

Model specification: Discriminant analysis

Generative classifier => $p\left(\mathbf{x},y|\tilde{\theta}\right) = p(\mathbf{x}|y,\theta)p(y|\theta')$

$$p(\mathbf{x}|y=c,\theta_c\ ) = N(\mathbf{x}|\mu_c,\Sigma_c)$$

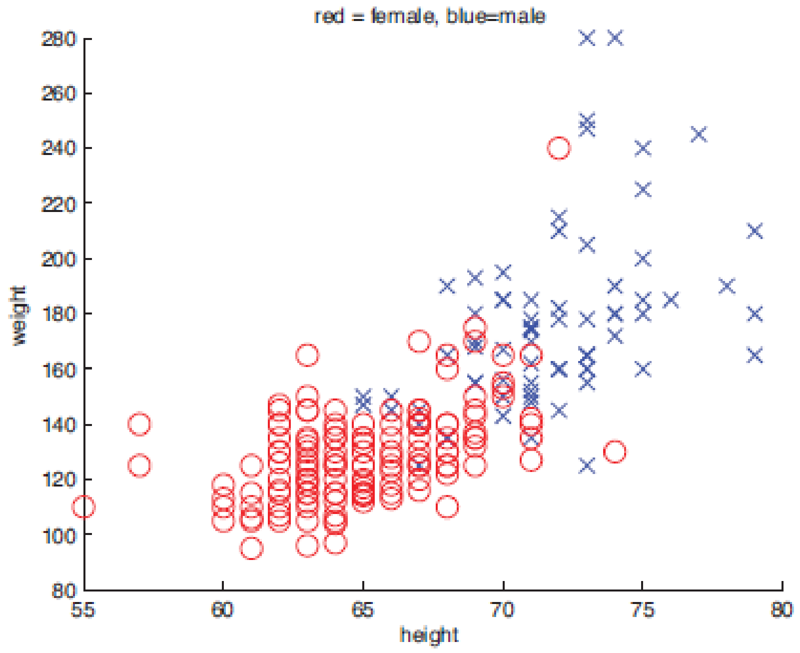
MLE for parameters of MVN

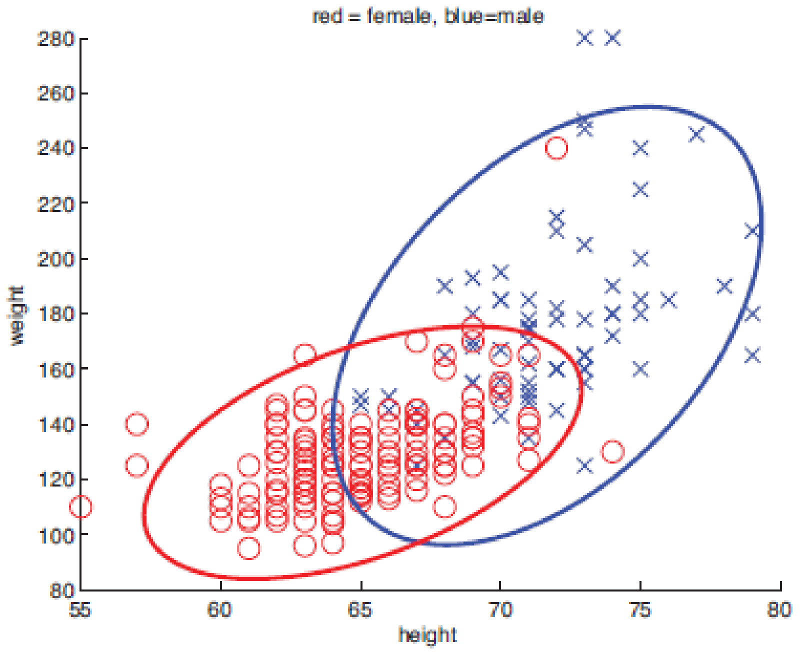
$$\hat{\mu} = \frac{1}{N}\sum_n x_n = \bar{x}$$

$$\hat{\Sigma} = \frac{1}{N}\sum_n (x_i-\bar{x})(x_i-\bar{x})^T$$

C.1 Generative models: LDA

Example: binary classification task





* Image from [3]

C.1 Generative models: LDA

Model fitting: MLE

Need $p\left(y|\mathbf{x},\tilde{\theta}\ \right)$ for Bayesian classification rule:

$$p\left(y|\mathbf{x},\tilde{\theta}\ \right) = \frac{p(\mathbf{x}|y,\theta)p(y|\pi)}{\sum_y p(\mathbf{x}|y,\theta)p(y|\pi)} \sim p(\mathbf{x}|y,\theta)p(y|\pi)$$

Parameters π in $p(y|\pi)$ may be estimated as empirical counts (MLE):

$$\hat{\pi}_c = N_c/N$$

Parameters θ may be estimated using MLE for MVN:

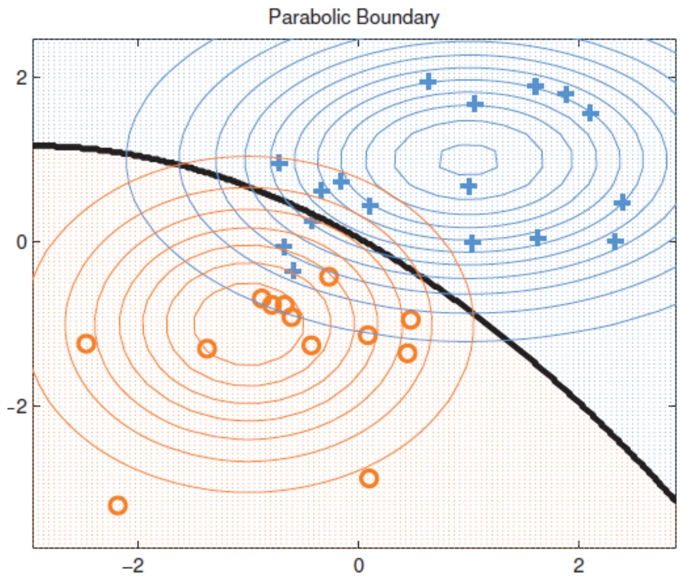
$$\theta = \{\mu, \Sigma\}$$

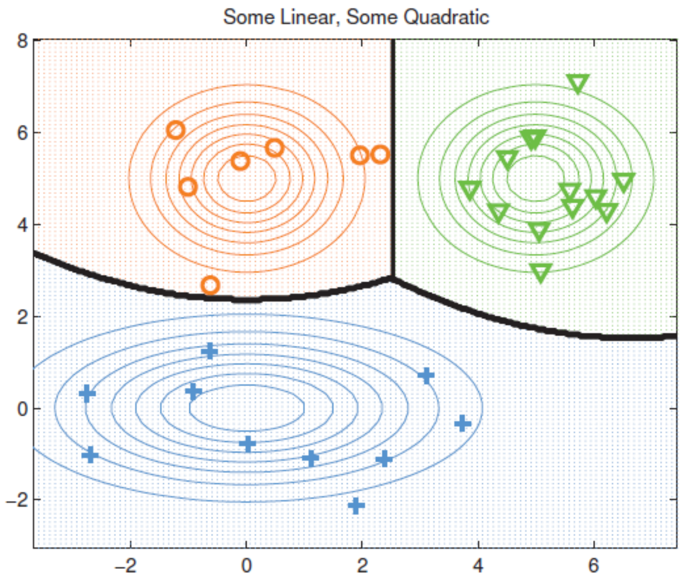
$$\hat{\mu} = \frac{1}{N}\sum_n x_n = \bar{x} \qquad \hat{\Sigma} = \frac{1}{N}\sum_n (x_i - \bar{x})(x_i - \bar{x})^T$$

C.1 Generative models: LDA

Model specification : Quadratic DA (QDA)

$$p(y|\mathbf{x},\theta) = \frac{\pi_c|\Sigma_c|^{-1/2}\exp\left[-\frac{1}{2}(x-\mu_c)^T\Sigma_c^{-1}(x-\mu_c)\right]}{\sum_c \pi_c|\Sigma_c|^{-1/2}\exp\left[-\frac{1}{2}(x-\mu_c)^T\Sigma_c^{-1}(x-\mu_c)\right]}$$





***Exercise:** Prove decision line is quadratic.*

* Image from [3]

C.1 Generative models: LDA

Model specification : Linear DA (LDA)

$$p(y=c|\mathbf{x},\theta)=\frac{\pi_c|\Sigma_c|^{-1/2}\exp\left[-\frac{1}{2}(x-\mu_c)^T\Sigma_c^{-1}(x-\mu_c)\right]}{\sum_c \pi_c|\Sigma_c|^{-1/2}\exp\left[-\frac{1}{2}(x-\mu_c)^T\Sigma_c^{-1}(x-\mu_c)\right]}$$

$$-\frac{1}{2}(x-\mu_c)^T\Sigma_c^{-1}(x-\mu_c)=-\frac{1}{2}x^T\Sigma_c^{-1}x+\mu_c^T\Sigma_c^{-1}x+-\frac{1}{2}\mu_c^T\Sigma_c^{-1}\mu_c$$

Additional assumption: $\Sigma_c=\Sigma$ => $x^T\Sigma^{-1}x$ cancels out:

$$p(y=c|\mathbf{x},\theta)=\frac{\exp(\beta_c^T x+\gamma_c)}{\sum_{c\prime}\exp(\beta_{c\prime}^T x+\gamma_{c\prime})}$$

softmax function

$$\beta_c \quad =\Sigma^{-1}\mu_c$$

$$\gamma_c=-\frac{1}{2}\mu_c^T\Sigma^{-1}\mu_c+\log\pi_c$$

C.1 Generative models: LDA

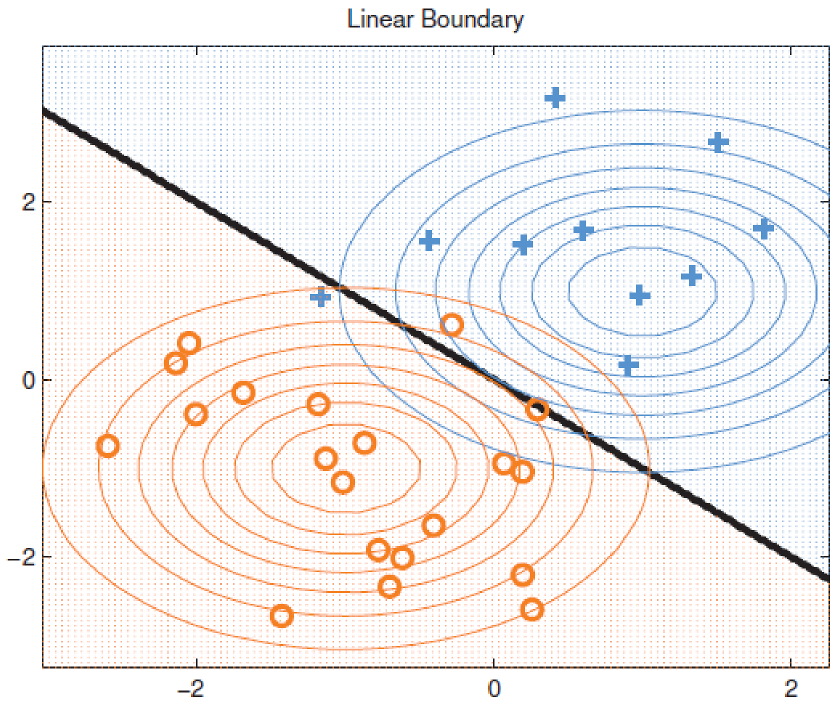
Decision surface for LDA

By definition on decision surface: $p(y = c|\mathbf{x}, \tilde{\theta}\ ) = p(y = c'|\mathbf{x}, \tilde{\theta}\ )$

$\beta_c^T x + \gamma_c = \beta_{c'}^T x + \gamma_{c'}$

$x^T(\beta_{c'} - \beta_c\ ) = \gamma_c - \gamma_{c'}$

=> Decision boundary is straight line.



* Image from [3]

C.1 Generative models: LDA

LDA

$$p(y = c|\mathbf{x}, \theta) = \frac{\exp(\beta_c^T x + \gamma_c)}{\sum_{c'} \exp(\beta_{c'}^T x + \gamma_{c'})}$$

$$\beta_c \ = \Sigma^{-1}\mu_c$$

$$\gamma_c = -\frac{1}{2}\mu_c^T\Sigma^{-1}\mu_c + \log \pi_c$$

Logistic Regression

$$p(y = c|\mathbf{x}, \theta) = \frac{\exp \theta_c^T x}{\sum_{c'} \theta_{c'}^T x}$$

(shift included in x)

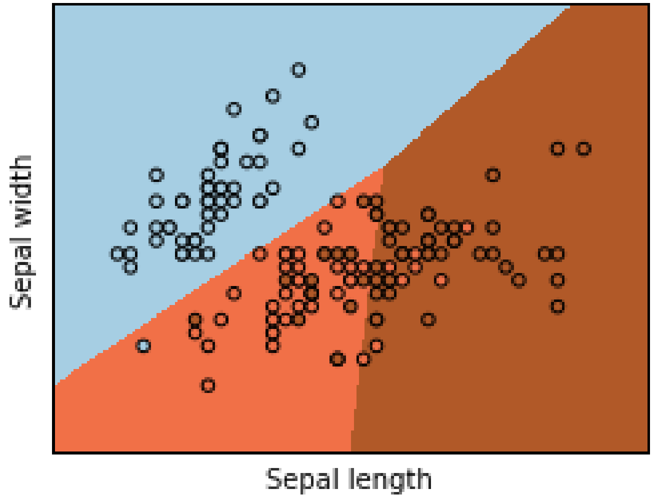
Distribution over class labels has the same form => linear decision boundary in both cases. **BUT**:

1. Other generative models can result in the same form of distribution over class labels => LDA makes stronger assumptions.

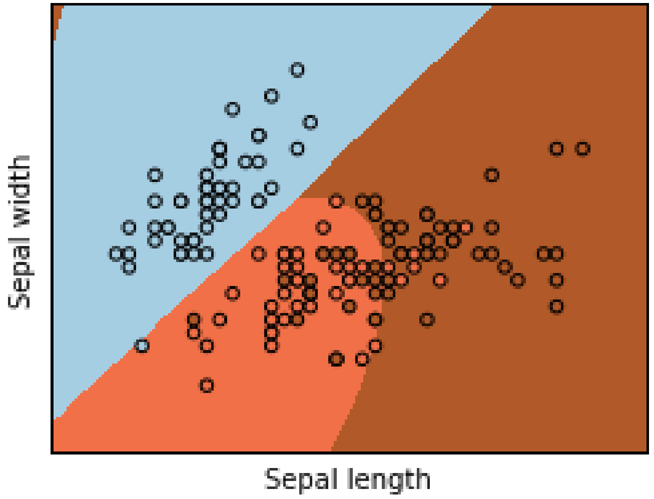
2. Different training process (optimization objective).

* Image from [3]

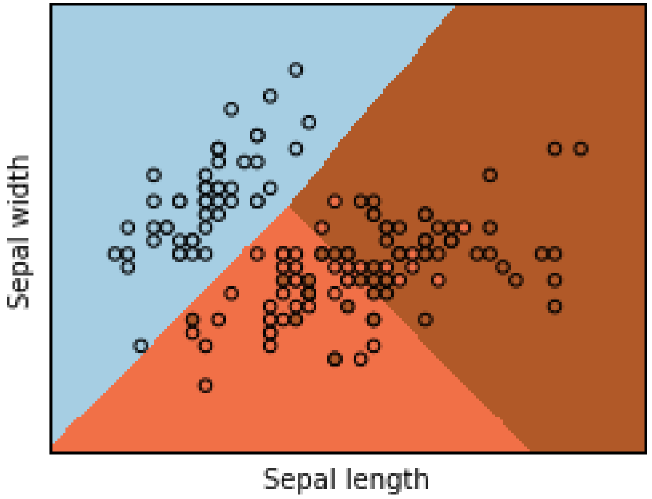
C.1 Generative models: LDA



LDA



QDA



LogReg

D. Conclusions

Generative models

handle missing features

easier to fit

can fit classes separately

Discriminative models

feature preprocessing

weaker assumptions

D. Conclusions

Methods covered

1. Bayesian approach: Bayesian inference, MAP.
2. Frequentist approach: MLE, ERM.

In the next lectures we will focus on ERM for neural networks:

$$\theta_{opt} = \operatorname{argmin}_\theta \frac{1}{N} \sum_{n=1}^{N} L\big(y_n, f(\mathbf{x}_{\boldsymbol{n}}|\theta)\big) + \lambda P(\theta)$$

All our preferences are quantified through loss function.

Next week

Starting main part of the course: Neural Networks

1. Basic neural networks
    a) Universal approximation theorem
    b) Types of neural networks
2. Training techniques
    a) Backpropagation
    b) Other (genetic algorithms)

E. Homework

For all:

1. ML: [3] ch. 4, 8 or [5] ch. 7.

2. Exercises from presentation.

Refs

1. Thorough review of relevant math topics:
http://info.usherbrooke.ca/hlarochelle/ift725/review.pdf

2. Ian Goodfellow, Yoshua Bengio and Aaron Courville, Deep Learning.

3. Kevin P. Murphy, Machine Learning: A probabilistic perspective.

4. David Barber, Bayesian Reasoning and Machine Learning.

5. Sergios Theodoridis, Machine Learning: A Bayesian and optimization perspective.