Statistical machine learning, high dimension and big data

S. Gaïffas

1CMAP - Ecole Polytechnique
1. Linear classification
   - Linear classification
   - Linearly separable data
   - Some geometry
   - Slacks
   - Hinge loss
   - Linear SVM

2. Beyond linear classification

3. Graphical Gaussian Model
   - Graphs
   - Graphical models
   - Gaussian graphical model
   - Graphical lasso
1. Linear classification
   - Linear classification
   - Linearly separable data
   - Some geometry
   - Slacks
   - Hinge loss
   - Linear SVM

2. Beyond linear classification

3. Graphical Gaussian Model
   - Graphs
   - Graphical models
   - Gaussian graphical model
   - Graphical lasso

- Polynomial features
- Features map and kernels
- Kernel trick
Linear classification

Classification

1. Using training data \((x_1, y_1), \ldots, (x_n, y_n)\), construct a decision rule \(f\) that predict the label \(\hat{y}\) of a new data point \(x\).

2. We saw that the logistic classifier is linear:

\[
\hat{y} = \text{sign}(\langle \hat{\theta}, x \rangle + \hat{b})
\]

where \(\text{sign} z = 1\) if \(z > 0\) and \(\text{sign} z = -1\) if \(z < 0\), and where \(\hat{\beta} \in \mathbb{R}^d\) and \(\hat{b} \in \mathbb{R}\) is the intercept.

3. The aim of linear classification is to find \(\hat{\theta}\) and \(\hat{b}\) using the training data.
Linearly separable data

- Data is **linearly separable** if we can find a classification rule that does not make any error on training data.
- Namely, can we find $\theta$ and $b$ such that

$$y_i(\langle \theta, x_i \rangle + b) \geq 0 \quad \text{for all } i = 1, \ldots, n$$

- **Strict linear separability** is when can we find $\theta$ such that

$$y_i(\langle \theta, x_i \rangle + b) > 0 \quad \text{for all } i = 1, \ldots, n$$

The hyperplane $\{x : \langle x, \theta \rangle + b = 0\}$ separates $-1$ and $+1$
An hyperplane $H = \{ x : \langle N, x \rangle + x_0 = 0 \}$ is a translation of a set of vectors orthogonal to $N$. Direction of the translation is given by $N$ and amount of translation by $x_0$. 
Some geometry

Given strict linear separability, we can rescale $\theta$ and $b$ and consider

$$y_i(\langle \theta, x_i \rangle + b) \geq 1 \quad \text{for all } \ i = 1, \ldots, n$$

Points with label 1 are contained in the half-space $\langle \theta, x \rangle + b \geq 1$
and those with label $-1$ are contained in half-space $\langle \theta, x \rangle + b \leq -1$
Some geometry

The distance between the 1’s hyperplane $\langle \theta, x \rangle + b = 1$ and the −1’s hyperplanes $\langle \theta, x \rangle + b = −1$ is equal to

$$\frac{2}{\|\theta\|_2}.$$ 

This is called the **margin**. The margin measures how much we can separate the data apart.

Badly or margin data are called the **support vectors**
The separability constraint

\[ y_i(\langle \theta, x_i \rangle + b) \geq 1 \quad \text{for all} \quad i = 1, \ldots, n \]

is too strong. We relax it a little bit by introducing “slacks”:

\[
\arg\min_{\theta \in \mathbb{R}^d, b \in \mathbb{R}, s_i \geq 0} \sum_{i=1}^{n} s_i \\
\text{subject to} \quad y_i(\langle \theta, x_i \rangle + b) \geq 1 - s_i \quad \text{for all} \quad i = 1, \ldots, n
\]
Slack rope
The hinge loss

The problem

argmin_{\theta \in \mathbb{R}^d, b \in \mathbb{R}, s_i \geq 0} \sum_{i=1}^{n} s_i

subject to \quad y_i(\langle \theta, x_i \rangle + b) \geq 1 - s_i \quad \text{for all} \quad i = 1, \ldots, n

is an optimization problem called “linear programming”. It can be interpreted differently:

argmin_{\theta} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \langle x_i, \theta \rangle + b)

where

\ell(y, z) = (1 - yz)_+ = \max(1 - yz, 0)

is the hinge loss.
The hinge loss

The hinge loss can be understood as a relaxation of the $0 - 1$ error (misclassification error)

$$\ell_{0-1}(y, z) = 1_{yz\leq 0}$$

Impossible to minimize the $0 - 1$ loss! Hinge loss gives an approximation to the number of errors made on the training set

Remark. Hinge is convex while $0 - 1$ loss is not
A solution to

$$\min_{\theta, b} \frac{1}{n} \sum_{i=1}^{n} \left(1 - y_i(\langle x_i, \theta \rangle + b)\right)_+$$

might not be unique, so we add a ridge penalization term:

$$\min_{\theta, b} \frac{1}{n} \sum_{i=1}^{n} \left(1 - y_i(\langle x_i, \theta \rangle + b)\right)_+ + \frac{\lambda}{2} \|\theta\|_2^2$$

where $\lambda > 0$ balances goodness-of-fit (measured by hinge loss) and energy of $\theta$. The solution is now unique.

This is the SVM (Support Vector Machine) algorithm. More precisely, it is the linear SVM.
The losses we’ve seen so far for classification

\[ \ell_0(y, z) = 1_{yz \leq 0} \]

\[ \ell_{\text{hinge}}(y, z) = (1 - yz)_+ \]

\[ \ell_{\text{logistic}}(y, z) = \log(1 + e^{-yz}). \]
Logistic regression vs Linear SVM

Grandmother’s recipes:
Grandmother’s recipes:

**Logistic regression**
- Logistic regression has a nice probabilistic interpretation
- Relies on the choice of the logit link function

**SVM**
- No model, only aims at separating points

No one is not better than the other in general. It depends on the data.

What is always important though is the choice of the features we work on
1. Linear classification
   - Linear classification
   - Linearly separable data
   - Some geometry
   - Slacks
   - Hinge loss
   - Linear SVM

2. Beyond linear classification
   - Polynomial features
   - Features map and kernels
   - Kernel trick

3. Graphical Gaussian Model
   - Graphs
   - Graphical models
   - Gaussian graphical model
   - Graphical lasso
Beyond linear classification

- Given features $x = [x_1, \ldots, x_d] \in \mathbb{R}^d$, we can construct many more features.
- For instance, we can add second order polynomials of the features $x_j^2, x_j x_k$ for any $1 \leq j, k \leq d$.
- It increases the number of features, hence dimension of the parameter $\theta$.
- Consider a **feature map** $\varphi(x)$ that adds all these new features.

A decision boundary $\langle \theta, \varphi(x) \rangle + b = 0$ is not an hyperplane anymore.
Beyond linear classification

- It is a common belief that we can always **increase the dimension** of the feature space to make data **almost linearly separable**

- A youtube video explains this:
  
  https://www.youtube.com/watch?v=3liCbRZPrZA
Beyond linear classification

You want to solve now

$$\min_{\theta, b} \frac{1}{n} \sum_{i=1}^{n} \left(1 - y_i(\langle \varphi(x_i), \theta \rangle + b) \right)_+ + \frac{\lambda}{2} ||\theta||^2$$

where $\theta$ is much larger and where $\varphi$ is a feature mapping
For the polynomial mapping $\varphi : \mathbb{R}^2 \to \mathbb{R}^3$

$$\varphi : x = (x_1, x_2) \mapsto (x_1^2, \sqrt{2}x_1x_2, x_2^2).$$

We have

$$\langle \varphi(x), \varphi(x') \rangle = x_1^2x_1'^2 + 2x_1x_2x_1'x_2' + x_2^2x_2'^2 = \langle x, x' \rangle^2$$

More generally we can define the “polynomial kernel”

$$K(x, x') = (1 + \langle x, x' \rangle)^d.$$ 

This kernel satisfies

$$K(x, x') = \langle \varphi(x), \varphi(x') \rangle$$

for some feature mapping $\varphi$. 
A **kernel** is a “new” inner product. It replaces $\langle x, x' \rangle$ by $\langle \varphi(x), \varphi(x') \rangle$ where $\varphi$ is a feature mapping.

A kernel $K$ is such that

$$K(x, x') = \langle \varphi(x), \varphi(x') \rangle$$

for some feature mapping $\varphi$.

Why is it important: the **kernel trick**. A theorem says that a solution $\hat{\theta}$ to

$$\min_{\theta, b} \frac{1}{n} \sum_{i=1}^{n} (1 - y_i (\langle \varphi(x_i), \theta \rangle + b))_+ + \frac{\lambda}{2} \|\theta\|_2^2$$

writes

$$\hat{\theta} = \sum_{i=1}^{n} \hat{u}_i \varphi(x_i) + \hat{b}$$

for some “dual” parameters $\hat{u}_1, \ldots, \hat{u}_n \in \mathbb{R}$. 
Actually computing a solution to

$$\min_{\theta, b} \frac{1}{n} \sum_{i=1}^{n} (1 - y_i(\langle \varphi(x_i), \theta \rangle + b))_+ + \frac{\lambda}{2} \|\theta\|^2$$

only requires computing the inner products $\langle \varphi(x_i), \varphi(x_i') \rangle$, hence only requires $K(x_i, x_i')$

Given a new $x \in \mathbb{R}^d$, the decision rule is $\langle \varphi(x), \hat{\theta} \rangle + \hat{b} \geq t$, which becomes

$$\sum_{i=1}^{n} \hat{u}_i \langle \varphi(x), \varphi(x_i) \rangle + \hat{b} \geq t$$

Only depends on $K$ again!
Kernels

- Actually we never need to know about the feature mapping \( \varphi \)!
- Only need to be able to know \( K \) in the computations
- This is the **kernel trick**

Popular kernels are:
- Gaussian or RBM (Radial Basis Function) kernel:

\[
K_\sigma(x, x') = \exp \left( - \frac{\|x - x'\|^2}{2\sigma^2} \right)
\]

- The polynomial kernel

\[
K(x, x') = (\langle x, x' \rangle + 1)^d
\]

Many other kernels in specific domains (text, image, video, genomics, etc.), that are adapted to the “geometry” of specific data.
Kernels

Decision rule with the RBF kernel: mixture of Gaussian functions

$$\mathbf{x} \mapsto \sum_{i=1}^{n} \hat{u}_i \exp \left( - \frac{\| \mathbf{x} - \mathbf{x}_i \|^2}{2 \sigma^2} \right) \geq t - \hat{b}$$
Decision rule with the RBF kernel: mixture of Gaussian functions

\[ x \mapsto \sum_{i=1}^{n} \hat{u}_i \exp \left( -\frac{\|x - x_i\|^2}{2\sigma^2} \right) \geq t - \hat{b} \]
Kernels

Decision rule with the RBF kernel: mixture of Gaussian functions

\[ x \mapsto \sum_{i=1}^{n} \hat{u}_i \exp \left( - \frac{\|x - x_i\|_2^2}{2\sigma^2} \right) \geq t - \hat{b} \]
1. Linear classification
   - Linear classification
   - Linearly separable data
   - Some geometry
   - Slacks
   - Hinge loss
   - Linear SVM

2. Beyond linear classification

3. Graphical Gaussian Model
   - Graphs
   - Graphical models
   - Gaussian graphical model
   - Graphical lasso

- Polynomial features
- Features map and kernels
- Kernel trick
$G = (V, E)$

\[
A = \begin{pmatrix}
1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 & 0 & 0 & 0 & 0
\end{pmatrix}
\]
Graphs
Graphs

Co-occurrence of words
Relation of artists in last.fm database
Graphs

Graph

- A graph $G$ consists of a set of vertices $V$ and a set of edges $E$
- We often note $G = (V, E)$
- $E$ is a subset of $V \times V$ containing ordered pairs of distinct vertices. An edge is directed from $j$ to $k$ if $(j, k) \in E$
- Undirected graphs, directed graphs
Graphical Model

- The set $V$ corresponds to a collection of random variables
- Denote $V = \{1, \ldots, p\}$ with $|V| = p$
  
  $$X = (X_1, \ldots, X_p) \sim \mathbb{P}$$

- The pair $(G, \mathbb{P})$ is a graphical model
Consider an undirected graph $G$ and a graphical model $(G, \mathbb{P})$

- We say that $\mathbb{P}$ satisfies the pairwise Markov property with respect to $G = (V, E)$ iif

\[
X_j \perp X_k \mid X_{V - \{j,k\}}
\]

for any $(j, k) \notin E, j \neq k$, namely $X_j$ and $X_k$ are conditionally independent given the all the other vertices

- A graphical model satisfying this property is called a conditional independence graph (CIG)
A Gaussian Graphical Model is a CIG with the assumption

\[ X = (X_1, \ldots, X_d) \sim N(0, \Sigma) \]

for a positive definite covariance matrix \( \Sigma \). Mean is zero to simplify notations.

A well-known result (Lauritzen (1996)):

\((j, k) \text{ and } (k, j) \in E \iff X_j \perp X_k \mid X_{V-\{j,k\}} \iff (\Sigma^{-1})_{j,k} = 0\]

[exerc.]

The edges can be read on the \textbf{precision} matrix \( K = \Sigma^{-1} \):

\((j, k) \in V \text{ and } (k, j) \in V \iff K_{j,k} \neq 0\)
The partial correlation $\rho_{j,k|V-\{j,k\}}$ between $X_j$ and $X_k$ conditional on $X_{V-\{j,k\}}$ is given by

$$\rho_{j,k|V-\{j,k\}} = -\frac{K_{j,k}}{\sqrt{K_{j,j}K_{k,k}}}$$

The partial correlation coefficients are regression coefficients: we can write

$$X_j = \beta_{j,k}X_k + \sum_{l\in V-\{j,k\}} \beta_{l,j}X_l + \varepsilon_j$$

where $\mathbb{E}[\varepsilon_j] = 0$ and $\varepsilon_j \perp X_{V-\{j\}}$, with

$$\beta_{j,k} = -\frac{K_{j,k}}{K_{j,j}} \text{ and } \beta_{k,j} = -\frac{K_{j,k}}{K_{k,k}}$$

[exerc.]
Suppose that we observe

\[ X_1, \ldots, X_n \text{ i.i.d. } N(0, \Sigma) \]

Put \( X \) the \( n \times p \) observation matrix with lines

\[ X_i = [X_{i,1} \cdots X_{i,p}] \]

Estimation of \( K = \Sigma^{-1} \) achieved by maximum likelihood estimation

\[
L(\Sigma; X) = \prod_{i=1}^{n} \frac{1}{(2\pi)^{p/2} \sqrt{\det \Sigma}} \exp\left( -\frac{1}{2} X_i^\top \Sigma^{-1} X_i \right)
\]

or

\[
L(K; X) = \prod_{i=1}^{n} \frac{\sqrt{\det(K)}}{(2\pi)^{p/2}} \exp\left( -\frac{1}{2} X_i^\top K X_i \right)
\]
Gaussian Graphical Models

- Minus log-likelihood is
  \[-\ell(K; X) = -\log \det K + \langle \hat{\Sigma}, K \rangle + c\]
  where $c$ does not depend on $K$ and where $\langle A, B \rangle = \text{tr}(A^T B)$

- Prior assumption: each vertex isn’t connected to all others: there is only few edges in the graph

- Use $\ell_1$-penalization on $K$ to obtain a sparse solution

**Graphical Lasso** [Friedman et al (2007), Banerjee et al (2008)]

$$\hat{K} \in \arg\min_{K:K>0} \left\{ -\log \det K + \langle \hat{\Sigma}, K \rangle + \lambda \sum_{1\leq j<k\leq p} |K_{j,k}| \right\}$$
Sparse Gaussian Graphical Model

Empirical covariance | Ledoit-Wolf covariance | GraphLasso covariance | True covariance

Empirical precision | Ledoit-Wolf precision | GraphLasso precision | True precision
How to solve

\[ \hat{K} \in \arg\min_{K > 0} \left\{ -\log \det K + \langle \hat{\Sigma}, K \rangle + \lambda \|K\|_1 \right\} \]

- It is a convex minimization problem: $-\log \det$ is convex
- $-\log \det$ differentiable, with $\nabla \log \det(X) = X^{-1}$
- Recall that $\max_{\|X\|_\infty \leq 1} \langle X, Y \rangle = \|K\|_1$
- Dual problem is

\[
\max_{\|X\|_\infty \leq \lambda} \left\{ \log \det(\hat{\Sigma} + X) + p \right\}
\]

and primal and dual variable related by $K = (\hat{\Sigma} + X)^{-1}$

- Duality gap is

\[ \langle K, \hat{\Sigma} \rangle - p + \lambda \|K\|_1 \]

[Exerc.]
Sparse Gaussian Graphical Model

- Rewrite dual problem

\[- \min_{\|X\|_\infty \leq \lambda} \left\{ - \log \det(\hat{\Sigma} + X) - p \right\}\]

as

\[\min_{\|X - \hat{\Sigma}\|_\infty \leq \lambda} - \log \det(X)\]

- This will be optimized recursively by updating over a single row and column of $K$ at a time
Sparse Gaussian Graphical Model

- Let \( X_{-j,-k} \) be the matrix with removed \( j \)-th line and \( k \)-th column and \( X_{-j} \) the \( j \)-th column with removed \( j \)-th entry.
- Recall the Schur complement formula

\[
\det \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \det(A) \det(D - CA^{-1}B)
\]

Namely

\[
\det \begin{bmatrix} K_{-p,-p} & k_p \\ k_p^\top & k_{p,p} \end{bmatrix} = \det(K_{-p,-p}) \det(k_{p,p} - k_p^\top K_{-p,-p}^{-1} k_p)
\]

- If we are at iteration \( k \), update the \( p \)-th row and column by \( k_p^{(k)} \) solution of

\[
\min_{\|y - \hat{\Sigma}_j\|_\infty \leq \lambda} y^\top (K_{-j,-j}^{(k-1)})^{-1} y
\]
The dual problem

$$\begin{align*}
\min_{y} & \quad y^\top (K_{-j,-j}^{(k-1)})^{-1} y \\
\text{subject to} & \quad \|y - \hat{\Sigma}_j\|_\infty \leq \lambda
\end{align*}$$

is a box-constrained quadratic program

Its dual is

$$\min_{x} x^\top K_{-j,-j}^{(k-1)} x - \langle \hat{\Sigma}_j, x \rangle + \lambda\|x\|_1 = \min_{x} \|Ax - b\|_2^2 + \lambda\|x\|_1$$

with $A = (K_{-j,-j}^{(k-1)})^{1/2}$ and $b = \frac{1}{2}(K_{-j,-j}^{(k-1)})^{-1/2}\hat{\Sigma}_j$

Several Lasso problem at each iteration
Algorithm for graphical Lasso [Block coordinate descent]

- Initialize $\hat{K}^{(0)} = K^{(0)} = \hat{\Sigma} + \lambda I$
- For $k \geq 0$ repeat
  - for $j = 1, \ldots, p$
    - solve
      \[
      \hat{x} \in \arg\min_x \left\| (K_{-j,-j}^{(j-1)})^{1/2} x - \frac{1}{2} (K_{-j,-j}^{(j-1)})^{-1/2} \hat{\Sigma}_j \right\|^2_2 + \lambda \|x\|_1
      \]
  - Obtain $K^{(j)}$ by replacing $j$-th row and column of $K^{(j-1)}$ by $\hat{x}$
  - Put $\hat{K}^{(k)} = K^{(p)}$ and $K^{(0)} = \hat{K}^{(k)}$
  - If
    \[
    \langle \hat{K}^{(k)}, \hat{\Sigma} \rangle - p + \lambda \|\hat{K}^{(k)}\|_1 \leq \varepsilon
    \]
    stop and return $\hat{K}^{(k)}$