Optimisation et apprentissage.

Alexandre d’Aspremont, CNRS - ENS.
Introduction

Today...

- Focus on **convexity** and its impact on complexity.
- Convex approximations, duality.
- Applications in learning.
In optimization.

Twenty years ago. . .

- Solve realistic large-scale problems using naive algorithms.
- Solve small, naive problems using serious algorithms.

Twenty years later. . .

- Solve realistic problems in e.g. statistics, signal processing, using efficient algorithms with explicit complexity bounds.
- Statisticians have started to care about complexity.
- Optimizers have started to care about statistics.
Key message from **complexity theory**: as the problem dimension gets large

- all **convex** problems are easy,
- most nonconvex problems are hard.
Convex problem.

\[
\begin{align*}
\text{minimize} & \quad f_0(x) \\
\text{subject to} & \quad f_i(x) \leq 0, \quad i = 1, \ldots, m \\
& \quad a_i^T x = b_i, \quad i = 1, \ldots, p
\end{align*}
\]

\(f_0, f_1, \ldots, f_m\) are convex functions, the equality constraints are all affine.

- Strong assumption, yet \textit{surprisingly expressive}.
- Good convex approximations of nonconvex problems.
**First-order condition.** Differentiable $f$ with convex domain is convex iff

$$ f(y) \geq f(x) + \nabla f(x)^T(y - x) \quad \text{for all } x, y \in \text{dom } f $$

First-order approximation of $f$ is global underestimator.
Ellipsoid method

Ellipsoid method. Developed in 70s by Shor, Nemirovski and Yudin.

- Function $f : \mathbb{R}^n \to \mathbb{R}$ convex (and for now, differentiable)
- **problem:** minimize $f$
- **oracle model:** for any $x$ we can evaluate $f$ and $\nabla f(x)$ (at some cost)

By evaluating $\nabla f$ we rule out a halfspace in our search for $x^*$.
Ellipsoid method

Suppose we have evaluated $\nabla f(x_1), \ldots, \nabla f(x_k)$, on the basis of $\nabla f(x_1), \ldots, \nabla f(x_k)$, we have localized $x^*$ to a polyhedron.

**Question:** what is a ‘good’ point $x_{k+1}$ at which to evaluate $\nabla f$?
Ellipsoid algorithm

Idea: localize $x^*$ in an ellipsoid instead of a polyhedron.

Compared to cutting-plane method:

- localization set doesn’t grow more complicated
- easy to compute query point
- but, we add unnecessary points in step 4
Ellipsoid Method

Ellipsoid method:

- Simple formula for $\mathcal{E}^{(k+1)}$ given $\mathcal{E}^{(k)}$
- $\text{vol}(\mathcal{E}^{(k+1)}) < e^{-\frac{1}{2n}} \text{vol}(\mathcal{E}^{(k)})$
Ellipsoid Method: example
A linear program (LP) is written

\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad Ax = b \\
& \quad x \geq 0
\end{align*}

where $x \geq 0$ means that the coefficients of the vector $x$ are nonnegative.

- Starts with Dantzig’s simplex algorithm in the late 40s.
- First efficient algorithm with polynomial complexity derived by Karmarkar [1984], using interior point methods.
Duality. The two linear programs

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad Ax = b \\
& \quad x \geq 0
\end{align*}
\]

\[
\begin{align*}
\text{maximize} & \quad y^T b \\
\text{subject to} & \quad c - A^T y \geq 0
\end{align*}
\]

have the same optimal values.

- Similar results hold for most \textbf{convex} problems.
- Usually both primal and dual have a natural interpretation.
- Many algorithms solve both problems simultaneously.
Support Vector Machines
Support Vector Machines

Simplest version. . .

- **Input**: A set of *points* (in 2D here) and *labels* (black & white).
- **Output**: A linear classifier separating the two groups.
Linear Classification

The **linear separation** problem.

**Inputs:**

- Data **points** \( x_j \in \mathbb{R}^n, \quad j = 1, \ldots, m. \)
- Binary **Labels** \( y_j \in \{-1, 1\}, \quad j = 1, \ldots, m. \)

**Problem:**

\[
\text{find} \quad w \in \mathbb{R}^n \\
\text{such that} \quad \langle w, x_j \rangle \geq 1 \quad \text{for all } j \text{ such that } y_j = 1 \\
\quad \langle w, x_j \rangle \leq -1 \quad \text{for all } j \text{ such that } y_j = -1
\]

**Output:**

- The classifier vector \( w \).
Nonlinear classification.

The problem:

\[
\begin{align*}
\text{find} & \quad w \\
\text{such that} & \quad \langle w, x_j \rangle \geq 1 \quad \text{for all } j \text{ such that } y_j = 1 \\
& \quad \langle w, x_j \rangle \leq -1 \quad \text{for all } j \text{ such that } y_j = -1
\end{align*}
\]

is linear in the variable \( w \). Solving it amounts to solving a linear program.

Suppose we want to add quadratic terms in \( x \):

\[
\begin{align*}
\text{find} & \quad w \\
\text{such that} & \quad \langle w, (x_j, x_j^2) \rangle \geq 1 \quad \text{for all } j \text{ such that } y_j = 1 \\
& \quad \langle w, (x_j, x_j^2) \rangle \leq -1 \quad \text{for all } j \text{ such that } y_j = -1
\end{align*}
\]

this is still a (larger) linear program in the variable \( w \).

Nonlinear classification is as easy as linear classification.
This trick means that we are not limited to linear classifiers:

Separation by ellipsoid  
Separation by 4th degree polynomial

Both are equivalent to linear classification. . . just increase the dimension.
Suppose the two sets are not \textit{separable}. We solve instead

\[
\text{minimize} \quad 1^T u + 1^T v \\
\text{subject to} \quad \langle w, x_j \rangle \geq 1 - u_j \quad \text{for all } j \text{ such that } y_j = 1 \\
\langle w, x_j \rangle < -(1 - v_j) \quad \text{for all } j \text{ such that } y_j = -1 \\
u \succeq 0, \quad v \succeq 0
\]

Can be interpreted as a heuristic for minimizing the number of misclassified points.
Robust linear discrimination

Suppose instead that the two data sets are well \textit{separated}.

(Euclidean) distance between hyperplanes

\begin{align*}
\mathcal{H}_1 &= \{z \mid a^T z + b = 1\} \\
\mathcal{H}_2 &= \{z \mid a^T z + b = -1\}
\end{align*}

is $\text{dist}(\mathcal{H}_1, \mathcal{H}_2) = 2/\|a\|_2$

to separate two sets of points by maximum margin,

\begin{align*}
\text{minimize} & \quad (1/2)\|a\|_2 \\
\text{subject to} & \quad a^T x_i + b \geq 1, \quad i = 1, \ldots, N \\
& \quad a^T y_i + b \leq -1, \quad i = 1, \ldots, M
\end{align*}

(1)

(after squaring objective) a QP in $a, b$
Classification

In practice . . .

- The data has very high dimension.
- The classifier is highly nonlinear.
- **Overfitting is a problem:** tradeoff between error and margin.
Support Vector Machines: Duality

Given \( m \) data points \( x_i \in \mathbb{R}^n \) with labels \( y_i \in \{-1, 1\} \).

- The \textbf{maximum margin classification} SVM problem can be written

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \|w\|_2^2 + C \mathbf{1}^T z \\
\text{subject to} & \quad y_i (w^T x_i) \geq 1 - z_i, \quad i = 1, \ldots, m \\
& \quad z \geq 0
\end{align*}
\]

in the variables \( w, z \in \mathbb{R}^n \), with parameter \( C > 0 \).

- The Lagrangian is written

\[
L(w, z, \alpha) = \frac{1}{2} \|w\|_2^2 + C \mathbf{1}^T z + \sum_{i=1}^{m} \alpha_i (1 - z_i - y_i w^T x_i)
\]

with dual variable \( \alpha \in \mathbb{R}_+^m \).
Support Vector Machines: Duality

- The Lagrangian can be rewritten

\[
L(w, z, \alpha) = \frac{1}{2} \left( \left\| w - \sum_{i=1}^{m} \alpha_i y_i x_i \right\|_2^2 - \left\| \sum_{i=1}^{m} \alpha_i y_i x_i \right\|_2^2 \right) + (C \mathbf{1} - \alpha)^T z + \mathbf{1}^T \alpha
\]

with dual variable \( \alpha \in \mathbb{R}^n_+ \).

- Minimizing in \((w, z)\) we form the dual problem

\[
\begin{align*}
\text{maximize} & \quad -\frac{1}{2} \left\| \sum_{i=1}^{m} \alpha_i y_i x_i \right\|_2^2 + \mathbf{1}^T \alpha \\
\text{subject to} & \quad 0 \leq \alpha \leq C
\end{align*}
\]

- At the optimum, we must have

\[
w = \sum_{i=1}^{m} \alpha_i y_i x_i \quad \text{and} \quad \alpha_i = C \text{ if } z_i > 0
\]

(this is the representer theorem).
If we write $X$ the data matrix with columns $x_i$, the dual can be rewritten

$$\begin{align*}
\text{maximize} & \quad -\frac{1}{2} \alpha^T \text{diag}(y) X^T X \text{diag}(y) \alpha + 1^T \alpha \\
\text{subject to} & \quad 0 \leq \alpha \leq C
\end{align*}$$

This means that the data only appears in the dual through the gram matrix

$$K = X^T X$$

which is called the kernel matrix.

In particular, the original dimension $n$ does not appear in the dual.

SVM complexity only grows with the number of samples, typically $O(m^{1.5})$. 
Support Vector Machines: the kernel trick

Kernels.

- All matrices written \( K = X^T X \) can be kernel matrices.
- Easy to construct from highly diverse data types.

Examples.

- Kernels for **voice recognition**

![Voice excerpt](image)

- Kernels for **gene sequence alignment**

<table>
<thead>
<tr>
<th>Sequence 1</th>
<th>Sequence 2</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAB24882</td>
<td>TYHMCQFHCYVNNHSGEKLYPECNERSKAFSCPQHLQCHRQIQGEKTHEHNQCGKAFPT</td>
<td>60</td>
</tr>
<tr>
<td>AAB24881</td>
<td>-------------------YECNQCGKAFACQHSSLKCHYRTHIGEKPYECNQCGKAFSK</td>
<td>40</td>
</tr>
<tr>
<td>AAB24882</td>
<td>PSHLQYHERHTHTGEKPYECHQCGQAFKSCSLQRHHRTEHTGEKPYE-CNQCGKAFQ-</td>
<td>116</td>
</tr>
<tr>
<td>AAB24881</td>
<td>HSHLQCHRHTHTGEKPYECNQCGKAFSGQHLQQRHRTEHTGEKPYMNVINMVKPLHNS</td>
<td>98</td>
</tr>
</tbody>
</table>
Support Vector Machines: the kernel trick

- Kernels for **images**

- Kernels for **text classification**

*Ryanair Q3 profit up 30%, stronger than expected. (From Reuters.)*

**DUBLIN, Feb 5 (Reuters) - Ryanair (RYA.I: Quote, Profile, Research) posted a 30 pct **jump** in third-quarter net **profit** on Monday, confounding analyst **expectations** for a **fall**, and **ramped up** its full-year **profit** goal while predicting big fuel-cost **savings** for the following year (...).**

<table>
<thead>
<tr>
<th>profit</th>
<th>loss</th>
<th>up</th>
<th>down</th>
<th>jump</th>
<th>fall</th>
<th>below</th>
<th>expectations</th>
<th>ramped up</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Compressed Sensing
Consider the following underdetermined linear system

\[
A \ x = b
\]

where \( A \in \mathbb{R}^{m \times n} \), with \( n \gg m \).

Can we find the \textit{sparsest} solution?
Compressed Sensing

- **Signal processing:** We make a few measurements of a high dimensional signal, which admits a sparse representation in a well chosen basis (e.g. Fourier, wavelet). Can we reconstruct the signal exactly?

- **Coding:** Suppose we transmit a message which is corrupted by a few errors. How many errors does it take to start losing the signal?

- **Statistics:** Variable selection in regression (LASSO, etc).
Why **sparsity**?

- Sparsity is a proxy for **power laws**. Most results stated here on sparse vectors apply to vectors with a power law decay in coefficient magnitude.
- Power laws appear everywhere. . .
  - Zipf law: word frequencies in natural language follow a power law.
  - Ranking: pagerank coefficients follow a power law.
  - Signal processing: $1/f$ signals
  - Social networks: node degrees follow a power law.
  - Earthquakes: Gutenberg-Richter power laws
  - River systems, cities, net worth, etc.
Compressed Sensing

Frequency vs. word in Wikipedia (from Wikipedia).

A. d'Aspremont

INRIA, Apr. 2014 31/52
Frequency vs. magnitude for earthquakes worldwide. [Christensen et al., 2002]
Figure 3. Log-log plot of the PageRank distribution of the Brown domain (*.brown.edu). A vast majority of the pages (except those with very low PageRank) follow a power law with exponent close to 2.1. The plot almost flattens out for pages with very low PageRank.

Figure 4. Log-log plot of the PageRank distribution of the WT10g corpus. The slope is close to 2.1. Note that the plot looks much sharper than the corresponding plot for the Brown web. Also, the tapering at the top is much less pronounced.

Pages vs. Pagerank on web sample. [Pandurangan et al., 2006]
Cumulative degree distribution in networks. [Newman, 2003]
Compressed Sensing

Getting the sparsest solution means solving

\[
\begin{align*}
\text{minimize} & \quad \text{Card}(x) \\
\text{subject to} & \quad Ax = b
\end{align*}
\]

which is a (hard) combinatorial problem in \( x \in \mathbb{R}^n \).

A classic heuristic is to solve instead

\[
\begin{align*}
\text{minimize} & \quad \|x\|_1 \\
\text{subject to} & \quad Ax = b
\end{align*}
\]

which is equivalent to an (easy) linear program.
Compressed Sensing

Example: we fix $A$ and draw many sparse signals $e$. Plot the probability of perfectly recovering $e$ by solving

$$\begin{align*}
\text{minimize} & \quad \|x\|_1 \\
\text{subject to} & \quad Ax = Ae
\end{align*}$$

in $x \in \mathbb{R}^n$, with $n = 50$ and $m = 30$. 
Compressed Sensing

- For some matrices $A$, when the solution $e$ is sparse enough, the solution of the linear program problem is also the sparsest solution to $Ax = Ae$. [Donoho and Tanner, 2005, Candès and Tao, 2005]

- Let $k = \text{Card}(e)$, this happens even when $k = O(m)$ asymptotically, which is provably optimal.
Semidefinite Programming
A linear program (LP) is written

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad Ax = b \\
& \quad x \geq 0
\end{align*}
\]

where \( x \geq 0 \) means that the coefficients of the vector \( x \) are nonnegative.
A **semidefinite program** (SDP) is written

\[
\begin{align*}
\text{minimize} & \quad \text{Tr}(CX) \\
\text{subject to} & \quad \text{Tr}(A_iX) = b_i, \quad i = 1, \ldots, m \\
& \quad X \succeq 0
\end{align*}
\]

where \(X \succeq 0\) means that the matrix variable \(X \in S_n\) is **positive semidefinite**.

- Nesterov and Nemirovskii [1994] showed that the **interior point algorithms** used for linear programs could be extended to semidefinite programs.
- Key result: **self-concordance** analysis of Newton’s method (affine invariant smoothness bounds on the Hessian).
Semidefinite Programming

■ Modeling
- Linear programming started as a toy problem in the 40s, many applications followed.
- Semidefinite programming has much stronger expressive power, many new applications being investigated today (cf. this talk).
- Similar conic duality theory.

■ Algorithms
- Robust solvers for solving large-scale linear programs are available today (e.g. MOSEK, CPLEX, GLPK).
- Not (yet) true for semidefinite programs. Very active work now on first-order methods, motivated by applications in statistical learning (matrix completion, NETFLIX, structured MLE, . . . ).
The NETFLIX challenge
Video On Demand and DVD by mail service in the United States, Canada, Latin America, the Caribbean, United Kingdom, Ireland, Sweden, Denmark, Norway, Finland.

- About 25 million users and 60,000 films.

- Unlimited streaming, DVD mailing, cheaper than CANAL+ :)

- Online movie recommendation engine.
Users assign **ratings** to a certain number of movies:

<table>
<thead>
<tr>
<th>Users</th>
<th>Movies</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>

Objective: make recommendations for other movies. . .
Top 10 for alexandre

Popular on Netflix
Collaborative prediction

Infer user preferences and movie features from user ratings.

- A linear prediction model

\[ \text{rating}_{ij} = u_i^T v_j \]

where \( u_i \) represents user characteristics and \( v_j \) movie features.

- This makes collaborative prediction a matrix factorization problem. We look for a linear model by factorizing \( M \in \mathbb{R}^{n \times m} \) as:

\[ M = U^T V \]

where \( U \in \mathbb{R}^{n \times k} \) represents user characteristics and \( V \in \mathbb{R}^{k \times m} \) movie features.

- Overcomplete representation. . . We want \( k \) to be as small as possible, i.e. we seek a low rank approximation of \( M \).
We would like to solve

\[
\text{minimize } \operatorname{Rank}(X) + c \sum_{(i,j) \in S} \max(0, 1 - X_{ij} M_{ij})
\]

non-convex and numerically hard.

Relaxation result in Fazel et al. [2001]: replace \( \operatorname{Rank}(X) \) by its convex envelope on the spectahedron to solve:

\[
\text{minimize } \|X\|_* + c \sum_{(i,j) \in S} \max(0, 1 - X_{ij} M_{ij})
\]

where \( \|X\|_* \) is the nuclear norm, i.e. sum of the singular values of \( X \).

This is a convex semidefinite program in \( X \).
Collaborative prediction

NETFLIX challenge.

- NETFLIX offered $1 million to the team who could improve the quality of its ratings by 10%, and $50,000 to the first team to improve them by 1%.
- It took two weeks to beat the 1% mark, and three years to reach 10%.
- Very large number of scientists, students, postdocs, etc. working on this.
- The story could end here. But all this work had surprising outcomes...
Phase Recovery

Molecular imaging

(from [Candes et al., 2011])

- CCD sensors only record the magnitude of diffracted rays, and loose the phase
- **Fraunhofer diffraction:** phase is required to invert the 2D Fourier transform
Phase Recovery

Focus on the **phase retrieval** problem, i.e.

\[
\text{find } \quad x \\
\text{such that } \quad |\langle a_i, x \rangle|^2 = b_i^2, \quad i = 1, \ldots, n
\]

in the variable \( x \in \mathbb{C}^p \).


\[
|\langle a_i, x \rangle|^2 = b_i^2 \iff \text{Tr}(a_ia_i^*xx^*) = b_i^2
\]

- [Chai et al., 2011] and [Candes et al., 2013] formulate phase recovery as a **matrix completion** problem

\[
\text{Minimize } \quad \text{Rank}(X) \\
\text{such that } \quad \text{Tr}(a_ia_i^*X) = b_i^2, \quad i = 1, \ldots, n \\
X \succeq 0
\]
[Recht et al., 2007, Candes and Recht, 2008, Candes and Tao, 2010] show that under certain conditions on $A$ and $x_0$, it suffices to solve

$$\begin{align*}
\text{Minimize} & \quad \text{Tr}(X) \\
\text{such that} & \quad \text{Tr}(a_ia_i^*X) = b_i^2, \quad i = 1, \ldots, n \\
& \quad X \succeq 0
\end{align*}$$

which is a (convex) semidefinite program in $X \in H_p$.

- Solving the convex semidefinite program yields a solution to the combinatorial, hard reconstruction problem.
- Apply results from collaborative filtering (NETFLIX) to molecular imaging.
Merci!
References


