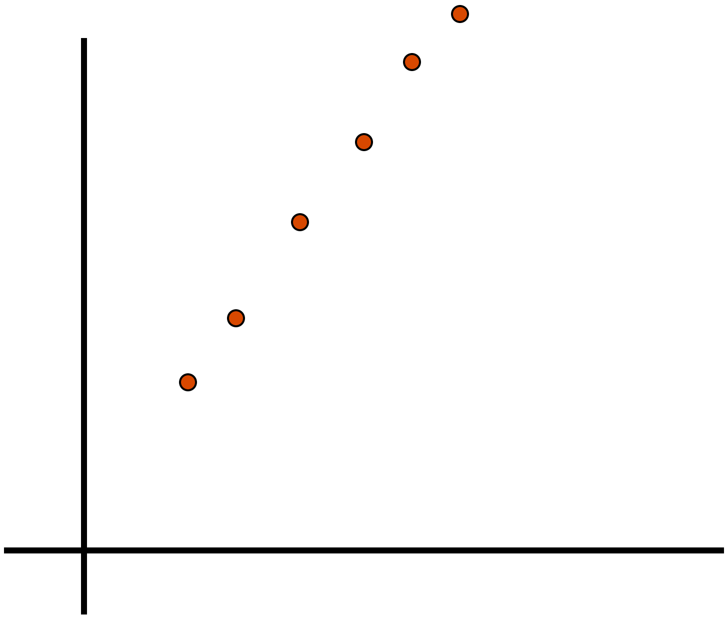
Lecture 20 – Matrix optimization in ML

Principal component analysis

Let us take three points in $\mathbf{R}^2$:

$$\begin{aligned} x_1 &= (2,1) \\ x_2 &= (4,2) \\ x_3 &= (6,3) \end{aligned}$$

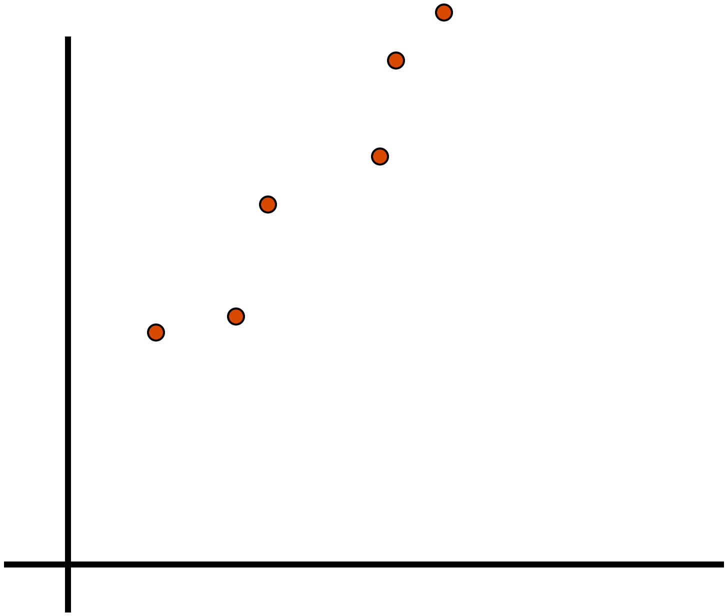


$$Y = \frac{1}{3}\sum_{i=1}^{3} x_i {x_i}^\top = \frac{1}{3}\left(\begin{bmatrix} 4 & 2 \\ 2 & 1 \end{bmatrix} + \begin{bmatrix} 16 & 8 \\ 8 & 4 \end{bmatrix} + \begin{bmatrix} 36 & 18 \\ 18 & 9 \end{bmatrix}\right)$$

$$Y = \frac{1}{3}\sum_{i=1}^{3} x_i {x_i}^\top = \frac{1}{3}\begin{bmatrix} 56 & 28 \\ 28 & 14 \end{bmatrix} = \frac{14}{3}\begin{bmatrix} 2 \\ 1 \end{bmatrix}\begin{bmatrix} 2 & 1 \end{bmatrix}$$

Principal component analysis

Find direction of largest variance



Let us take three points in $\mathbf{R}^2$:

$$\begin{aligned} y_1 &= (2,1) + (O(\epsilon), O(\epsilon)) \\ y_2 &= (4,2) + (O(\epsilon), O(\epsilon)) \\ y_3 &= (6,3) + (O(\epsilon), O(\epsilon)) \end{aligned}$$

$$A = \frac{14}{\sqrt{3}} \left[\begin{array}{c} 2/\sqrt{3} \\ 1/\sqrt{3} \end{array} \right] \left[\begin{array}{cc} 2/\sqrt{3} & 1/\sqrt{3} \end{array} \right] + \left[\begin{array}{cc} O(\epsilon) & O(\epsilon) \\ O(\epsilon) & O(\epsilon) \end{array} \right]$$

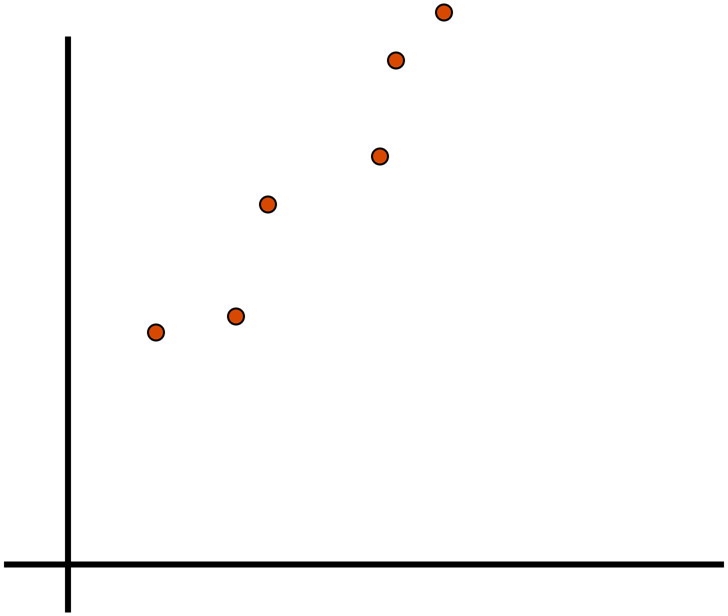
$$\left[\begin{array}{c} 2/\sqrt{3} \\ 1/\sqrt{3} \end{array} \right] = \mathrm{argmax}_{x \in \mathbf{R}^2,\ \|x\|=1} x^\top A x$$

Sparse principal component analysis

Find a **sparse** direction of largest variance

Let us take three points in $\mathbf{R}^2$:

$$\begin{aligned} y_1 &= (2,1) + (O(\epsilon), O(\epsilon)) \\ y_2 &= (4,2) + (O(\epsilon), O(\epsilon)) \\ y_3 &= (6,3) + (O(\epsilon), O(\epsilon)) \end{aligned}$$



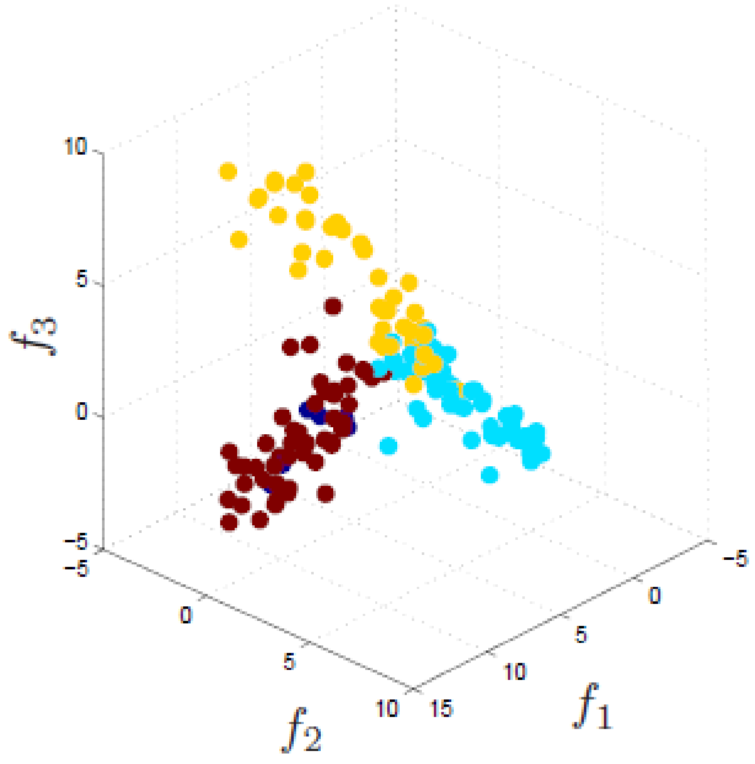
$$A = \frac{1}{3}\left[\begin{array}{cc} 56+O(\epsilon) & 28+O(\epsilon) \\ 28+O(\epsilon) & 14+O(\epsilon) \end{array}\right] \approx \frac{56}{3}\left[\begin{array}{c} 1 \\ 0 \end{array}\right]\left[\begin{array}{cc} 1 & 0 \end{array}\right] + \left[\begin{array}{cc} O(\tilde{\epsilon}) & O(\tilde{\epsilon}) \\ O(\tilde{\epsilon}) & O(\tilde{\epsilon}) \end{array}\right]$$

$$\left[\begin{array}{c} 1 \\ 0 \end{array}\right] = \operatorname{argmax}_{x\in\mathbf{R}^2,\ \|x\|=1,\ \mathrm{card}(x)=1} x^\top A x$$

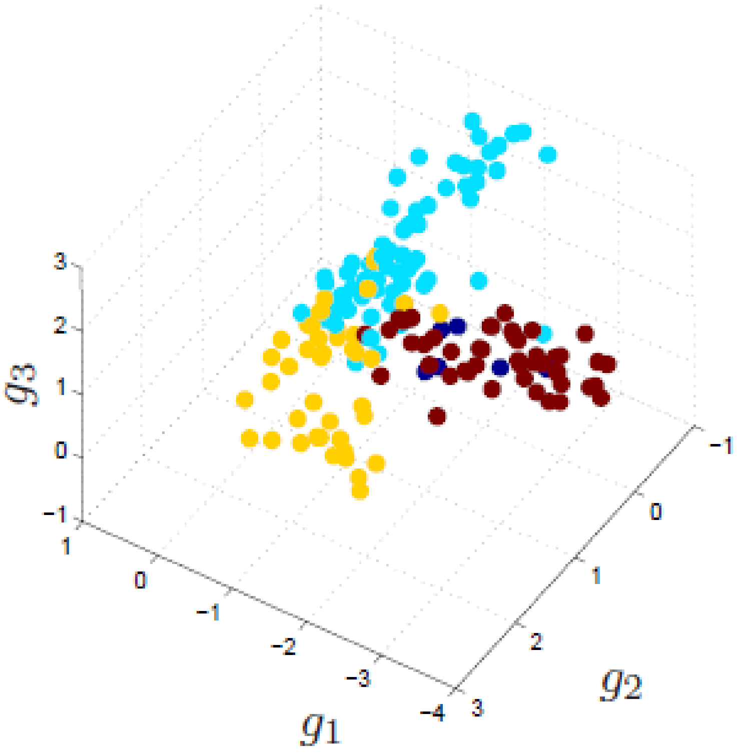
Introduction

Clustering of gene expression data in PCA versus sparse PCA, on 500 genes.

PCA



Sparse PCA



The PCA factors f_i on the left are dense and each use all 500 genes.
The sparse factors $g_1,\ g_2$ and g_3 on the right involve 6, 4 and 4 genes respectively.

Sparse PCA

Given a set $Y \in \mathbf{R}^{m \times n}$ compute empirical covariance matrix $A = \frac{1}{m} Y^\top Y$

Principal component analysis

Maximize the variance explained by factor x

$$\begin{array}{ll} \max\limits_{x \in \mathbf{R}^n} & x^\top A x \\ \text{s.t.} & \|x\|_2 = 1 \end{array}$$

Sparse principal component analysis

Maximize the variance explained by a factor x with bounded cardinality

$$\begin{array}{ll} \max_{x \in \mathbf{R}^n} & x^\top A x \\ \text{s.t.} & card(x) = k \\ & \|x\|_2 = 1 \end{array}$$

Semidefinite relaxation

Start from:

$$\begin{array}{ll} \text{maximize} & x^TAx \\ \text{subject to} & \|x\|_2 = 1 \\ & \mathbf{Card}(x) \leq k, \end{array}$$

where $x \in \mathbf{R}^n$. Let $X = xx^T$ and write everything in terms of the matrix X:

$$\begin{array}{ll} \text{maximize} & \mathbf{Tr}(AX) \\ \text{subject to} & \mathbf{Tr}(X) = 1 \\ & \mathbf{Card}(X) \leq k^2 \\ & X = xx^T, \end{array}$$

Replace $X = xx^T$ by the equivalent $X \succeq 0,\ \mathbf{Rank}(X) = 1$:

$$\begin{array}{ll} \text{maximize} & \mathbf{Tr}(AX) \\ \text{subject to} & \mathbf{Tr}(X) = 1 \\ & \mathbf{Card}(X) \leq k^2 \\ & X \succeq 0,\ \mathbf{Rank}(X) = 1, \end{array}$$

again, this is the same problem.

Semidefinite relaxation

We have made **some progress**:

- The objective $\mathbf{Tr}(AX)$ is now **linear** in X
- The (non-convex) constraint $\|x\|_2 = 1$ became a **linear** constraint $\mathbf{Tr}(X) = 1$.

But this is still a hard problem:

- The $\mathbf{Card}(X) \leq k^2$ is still non-convex.
- So is the constraint $\mathbf{Rank}(X) = 1$.

We still need to relax the two non-convex constraints above:

- If $u \in \mathbf{R}^p$, $\mathbf{Card}(u) = q$ implies $\|u\|_1 \leq \sqrt{q}\|u\|_2$. So we can replace $\mathbf{Card}(X) \leq k^2$ by the weaker (but **convex**): $\mathbf{1}^T|X|\mathbf{1} \leq k$.
- We simply drop the rank constraint

Semidefinite Programming

Semidefinite relaxation:

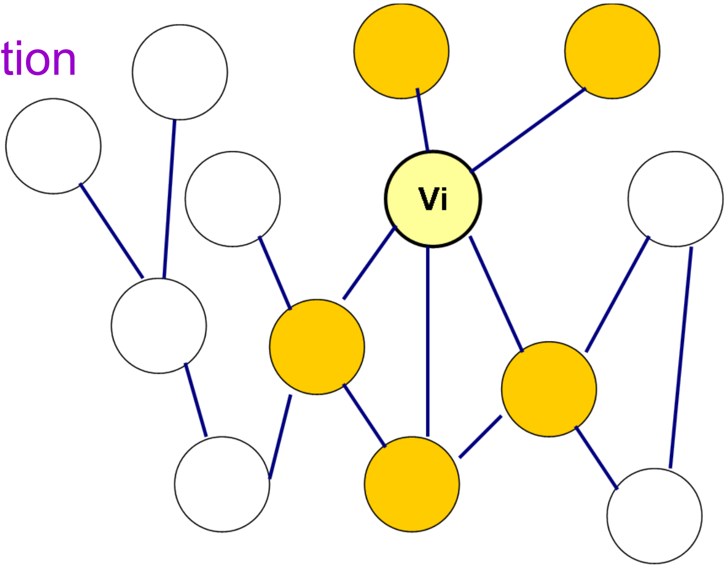
$$\begin{array}{ll} \text{maximize} & x^T A x \\ \text{subject to} & \|x\|_2 = 1 \\ & \mathbf{Card}(x) \leq k, \end{array} \qquad \textbf{becomes} \qquad \begin{array}{ll} \text{maximize} & \mathbf{Tr}(AX) \\ \text{subject to} & \mathbf{Tr}(X) = 1 \\ & \mathbf{1}^T |X| \mathbf{1} \leq k \\ & X \succeq 0, \end{array}$$

- This is a **semidefinite program** in the variable $X \in \mathbf{S}^n$...
- Solve small problems (a few hundred variables) using IP solvers, etc.
- Dimensionality reduction apps: solve very large instances.

Solution: use first order algorithm...

Sparse inverse covariance selection

Sparse inverse covariance selection



p random varibles

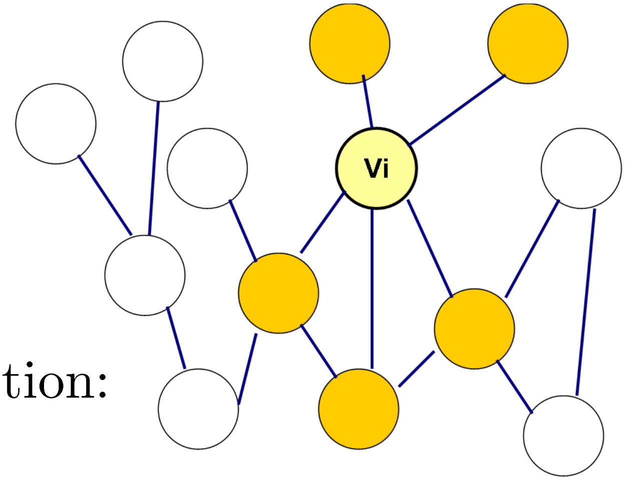
$\boldsymbol{x} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$

Multivariate Gaussian probability density function:

$$P(\mathbf{x}) = (2\pi)^{-\frac{n}{2}} \det(\Sigma)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu})\right)$$

- $\Sigma \in R^{n \times n}$ - covariance matrix
- Zeros in Σ^{-1} : conditional independence
- Sparsity of Σ^{-1} : better interpretability

Sparse inverse covariance selection



Multivariate Gaussian probability density function:

$$P(\mathbf{x}) = (2\pi)^{-\frac{n}{2}} \det(\Sigma)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x}-\boldsymbol{\mu})\right)$$

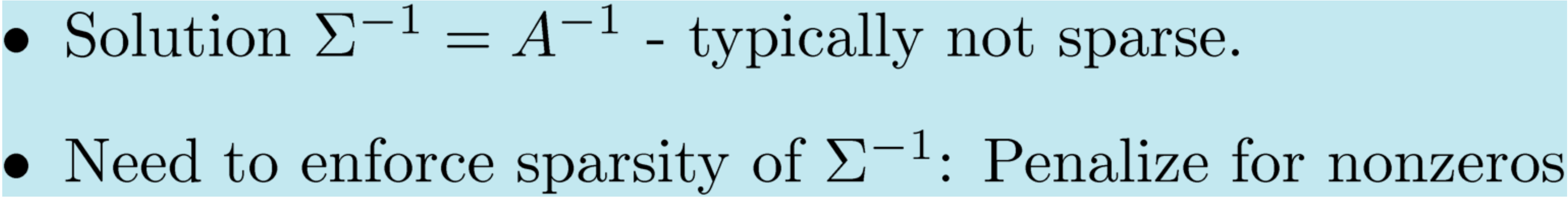
- Given X - m realizations of $\boldsymbol{x}$, $(\mu = 0)$
- $\max_{\Sigma} \log(P(X)) = \max_{\Sigma} \frac{m}{2} \log(\det(\Sigma^{-1})) - \frac{1}{2} Tr((XX^\top)\Sigma^{-1})$
- Can compute Σ^{-1} maximing log-likelihood

Optimizing log likelihood

- $\max_\Sigma \log(P(X)) = \max_\Sigma \frac{m}{2}\log(\det(\Sigma^{-1})) - \frac{1}{2}Tr((XX^\top)\Sigma^{-1})$

- Let $A = \frac{1}{m}XX^\top$

- $\Sigma^{-1} = \arg\max_C \frac{m}{2}(\log\det C - Tr(AC))$



- Solution $\Sigma^{-1} = A^{-1}$ - typically not sparse.

- Need to enforce sparsity of Σ^{-1}: Penalize for nonzeros

Enforcing sparsity

- NP-hard formulation

$$\Sigma^{-1} = \arg\max_C \left(\tfrac{m}{2}(\log\det C - Tr(AC)) - \lambda Card(C) \right)$$

- Convex relaxation

$$\Sigma^{-1} = \arg\max_C \tfrac{m}{2}(\log\det C - Tr(AC)) - \lambda ||C||_1$$

$$(\|C\|_1 = \textstyle\sum_{ij} |C_{ij}|)$$

- Convex optimization problem with unique solution for each λ

Primal-dual pair of problems

Primal problem

$$\max_{C \succ 0} \frac{m}{2}(\ln\det(C) - Tr(AC)) - \lambda \|C\|_1$$

Reformulate using constraints

$$\begin{array}{ll} \max\limits_{C',C''} & \frac{m}{2}[\ln\det(C'-C'') - Tr(A(C'-C''))] - \lambda Tr(E(C'+C'')), \\ \text{s. t.} & C' \geq 0,\ C'' \geq 0,\ C'-C'' \succ 0 \end{array}$$

Lagrangian

$$L(C', C'', U, V) =$$

$$\frac{m}{2}[\ln\det(C'-C'') - Tr(A(C'-C''))] - \lambda Tr(E(C'+C'')) + U.*C' + V.*C''$$

$$U, V, C', C'' \geq 0$$

Deriving the dual

$$\nabla_{C'} L(C', C'', U, V) = \frac{m}{2}[(C' - C'')^{-1} - A] - \lambda E + U = 0$$

$$U \geq 0$$

$$\nabla_{C''} L(C', C'', U, V) = \frac{m}{2}[-(C' - C'')^{-1} + A] - \lambda E + V = 0$$

$$V \geq 0$$

$$\Downarrow$$

$$W = (C' - C'')^{-1}$$

$$-\lambda E + V = \frac{m}{2} W - A = \lambda E - U$$

$$U, V \geq 0$$

$$\Downarrow$$

$$\tfrac{m}{2} \|W - A\|_\infty \leq \lambda$$

Primal-dual pair of problems

Primal problem

$$\max_{C\succ 0} \tfrac{m}{2}(\text{lndet}(C) - Tr(AC)) - \lambda\|C\|_1$$

Dual problem

$$\max_{W\succ 0}\{\tfrac{m}{2}\ln(\det(W)) - mp/2 : \text{ s.t. } \tfrac{m}{2}||(W - A)||_\infty \leq \lambda\}$$

Interior point method – O(n^6) operations/iter

Block coordinate ascent

Update one row and one column of the dual matrix W at each step

$$W = \begin{bmatrix} W_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix}$$

$$\max_{W \succ 0} \{ \tfrac{m}{2} \ln(\det(W)) - mp/2 : \text{ s.t. } \tfrac{m}{2} ||W - A||_\infty \leq \lambda \}$$

$$\text{lndet} W = \ln(\det(W_{11})(w_{22} - {w_{12}}^T W_{11}^{-1} w_{12}))$$

Block coordinate ascent subproblem

Update one row and one column of the dual matrix W at each step

$$W = \begin{bmatrix} W_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix}$$

$$\begin{array}{ll} \max_{w_{12}, w_{22}} & \ln(w_{22} - {w_{12}}^T W_{11}^{-1} w_{12})) \\ \text{s.t.} & \|w_{12} - a_{12}\|_\infty \leq \frac{2}{m}\lambda, |w_{22} - a_{22}| \leq \frac{2}{m}\lambda \end{array}$$

$$\min_{w_{12}} \{ w_{12}^\top W_{11}^{-1} w_{12} : \quad \text{s.t.} \quad \|w_{12} - a_{12}\|_\infty \leq \frac{2}{m}\lambda,$$

Subproblem reformulation

$$\min_{w_{12}}\{w_{12}^\top W_{11}^{-1} w_{12} : \quad \text{s.t.} \quad \|w_{12} - a_{12}\|_\infty \leq \frac{2}{m}\lambda,$$

$$w_{12} = W_{11}\beta$$

$$\min_{\beta}\{\beta^\top W_{11}\beta : \quad \text{s.t.} \quad \|W_{11}\beta - a_{12}\|_\infty \leq \frac{2}{m}\lambda\}$$

Remember Lasso!

Primal-Dual pair of problems

$$\min \quad \frac{1}{2}||Ax - b||^2 + \lambda||x||_1$$

$$\begin{aligned} \min \quad & \frac{1}{2}x^\top A^\top Ax \\ s.t. \quad & \|A^\top(Ax - b)\|_\infty \leq \lambda \end{aligned}$$

Dual subproblem

$$\min_{w_{12}}\{w_{12}^\top W_{11}^{-1} w_{12} : \quad \text{s.t.} \quad \|w_{12} - a_{12}\|_\infty \leq \frac{2}{m}\lambda,$$

$$w_{12} = W_{11}\beta$$

$$\min_{\beta}\{\beta^\top W_{11}\beta : \quad \text{s.t.} \quad \|W_{11}\beta - a_{12}\|_\infty \leq \frac{2}{m}\lambda\}$$

$$\min_{\beta}\{\|W_{11}^{1/2}\beta - W_{11}^{-1/2} a_{12}\|^2 + \frac{4}{m}\lambda\|\beta\|_1$$

The dual subproblem is the Lasso problem

Remember coordinate descent for Lasso

$$\min_{x_i} \quad \frac{1}{2}\|Ax - b\|^2 + \lambda\|x\|_1$$

Choose one variable x_i and column A_i.
Let $\bar{x}$ and $\bar{A}$ correspond to the fixed part

$$\min_{x_i} \quad \frac{1}{2}(A_i x_i + \bar{A}\bar{x} - b)^2 + \lambda|x_i|$$

Soft-thresholding operator

$$\min_{x_i} \frac{1}{2}(x_i - r)^2 + \lambda|x| \rightarrow x_i = \begin{cases} r - \lambda & \text{if } r > \lambda \\ 0 & \text{if } -\lambda \leq r \leq \lambda \\ r + \lambda & \text{if } r < -\lambda \end{cases}$$

$$r = -{A_i}^\top(\bar{A}\bar{x} - b)/||A_i||^2, \;\; \lambda \rightarrow \lambda/||A_i||^2$$

Remember coordinate descent for Lasso

$$\min_{x_i} \quad \frac{1}{2}\|W_{11}^{1/2}\beta - W_{11}^{-1/2}a_{12}\|^2 + \lambda\|\beta\|_1$$

$$\min_{\beta_i} \frac{1}{2}(\beta_i - r)^2 + \lambda|x| \to \beta_i = \begin{cases} r-\lambda & \text{if } r > \lambda \\ 0 & \text{if } -\lambda \leq r \leq \lambda \\ r+\lambda & \text{if } r < -\lambda \end{cases}$$

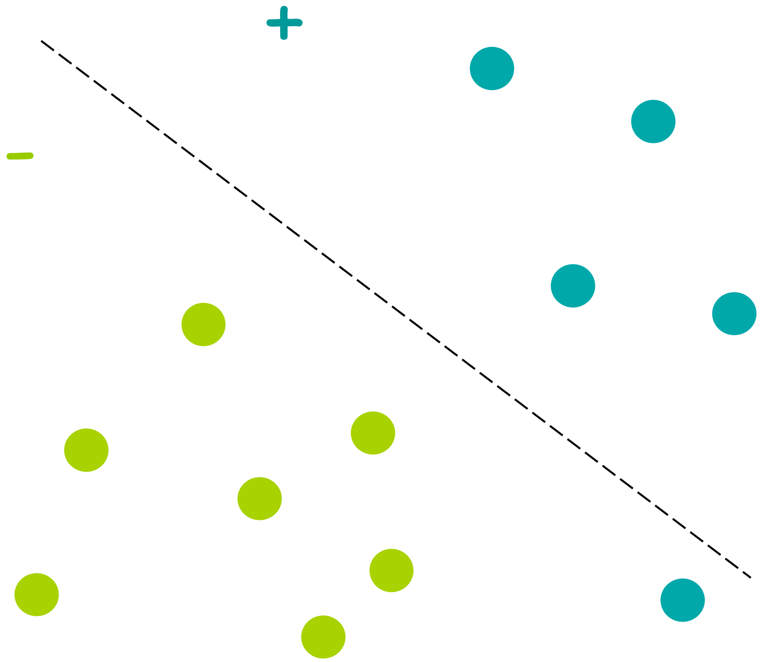
$$r = -((W_{11})_i{}^\top \bar{\beta} - (a_{12})_i)/(W_{11})_{ii}, \;\; \lambda \to \lambda/(W_{11})_{ii}$$

No need to compute $W^{1/2}$

Multiple Kernel Learning

Modified from Gert Lanckriet’s (UCSD) slides

Support Vector Machines



Kernel SVM

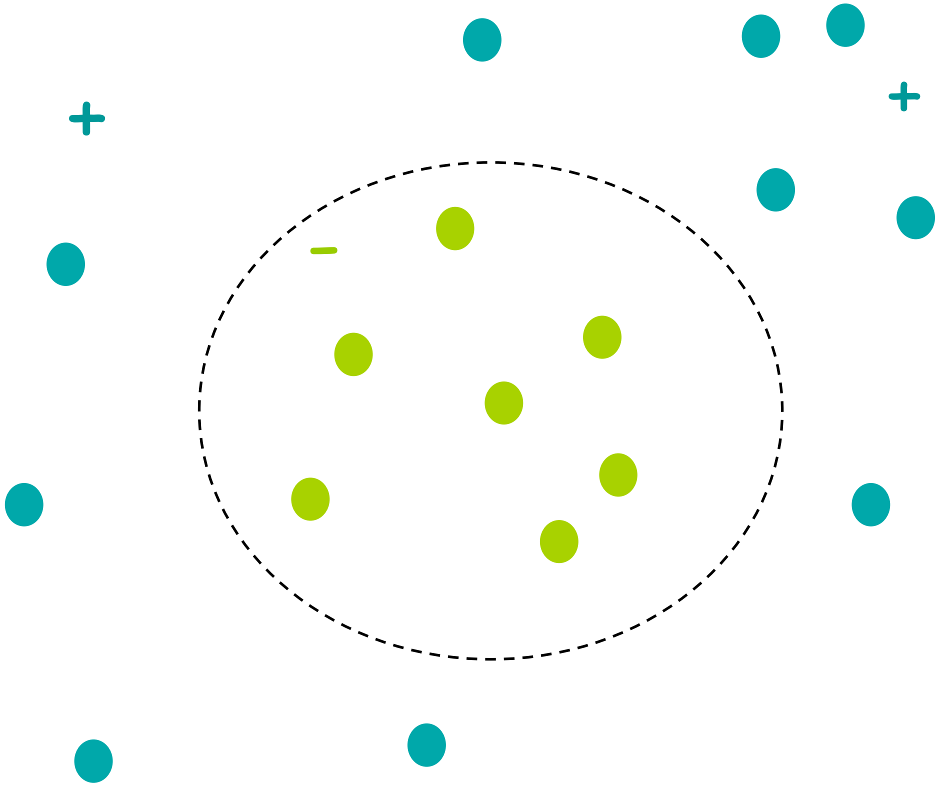
$Q_{ij} = y_i y_j x_i^{>} x_j \quad ! \quad Q_{ij} = y_i y_j \phi(x_i)^{>} \phi(x_j) = y_i y_j K(x_i, x_j)$

Kernel operation: $K(x_i, x_j) = \phi(x_i)^{>} \phi(x_j)$

Examples:

2 $K(x_i, x_j) = \exp^{i\ jjx_i\ i\ x_j\ jj^2 = 2\text{¾}^2}$

2 $K(x_i, x_j) = (x_i^{>} x_j / a_1 + a_2)^d$



Maximal margin classification

- Training: **convex** optimization problem (**QP**)
- **Dual** problem:

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \boxed{\phi(x_i)^T \phi(x_j)} \quad \text{s.t.} \quad \sum_{i=1}^{n} \alpha_i y_i = 0,\ 0 \leqslant \alpha_i \leqslant C$$

$$\Big\downarrow \quad K_{ij} = \phi(x_i)^T \phi(x_j)$$

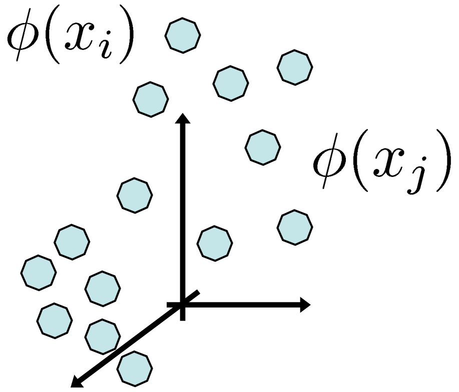
$$\max_{\alpha} \quad \alpha^\top e - \frac{1}{2} \alpha^\top D_y \boxed{K} D_y \alpha \quad \text{s.t.} \quad \alpha^\top y = 0,\ 0 \leqslant \alpha \leqslant C$$

- **Optimality** condition: $\boxed{w = \sum_{i=1}^{n} \alpha_i y_i \phi(x_i)}$

Kernel-based learning

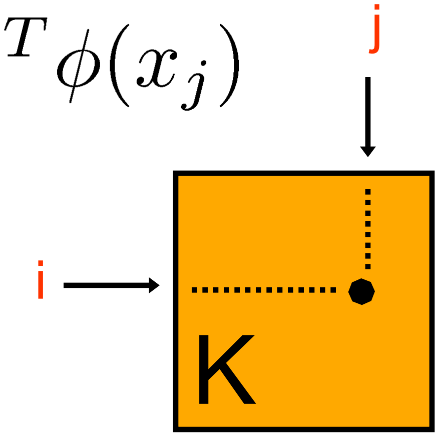
Embed data

IMPLICITLY: Inner product measures similarity



$$k(x_i, x_j) = \phi(x_i)^T \phi(x_j)$$

$$K_{ij} = \phi(x_i)^T \phi(x_j)$$



Property: Any **symmetric positive definite** matrix specifies a kernel matrix & every kernel matrix is **symmetric positive definite**

Optimizing over the kernel

- **Primal** problem:

$$\begin{array}{rl}\boxed{\min_{K\succeq 0}\min_{\alpha}} & \frac{1}{2}\alpha^\top D_y \boxed{K} D_y \alpha + C\sum_{i=1}^{n}\xi_i \\ \text{s.t.} & D_y \boxed{K} D_y \alpha + y\beta + s - \xi = -e, \\ & 0 \leqslant \alpha_i \leqslant C,\ \xi \geqslant 0\end{array}$$

- Can we do this?

Optimizing over the kernel?

- **Primal** problem:

$$
\begin{array}{ll}
\boxed{\min_{K \succeq 0} \min_{\alpha}} & \frac{1}{2}\alpha^\top D_y \boxed{K} D_y \alpha + C \sum_{i=1}^{n} \xi_i \\
\text{s.t.} & D_y \boxed{K} D_y \alpha + y\beta + s - \xi = -e, \\
 & 0 \leqslant \alpha_i \leqslant C, \xi \geqslant 0
\end{array}
$$

- **Consider** $K = yy^\top \succeq 0$

$$
K(x_i, x_j) = \begin{cases} 1 & \text{if } x_i,\ x_j \text{ in the same class} \\ -1 & \text{if } x_i,\ x_j \text{ in different classes} \end{cases}
$$

Classification using the kernel

- Training:

$$\max_{\alpha} \quad \alpha^\top 1 - \frac{1}{2}\alpha^\top D_y \boxed{K} D_y \alpha \quad \text{s.t.} \quad \alpha^\top y = 0,\ 0 \leqslant \alpha \leqslant C$$

- Classification rule: classify new data point x:

$$\begin{aligned} f(\phi(x)) &= \operatorname{sign}\left(w^T \phi(x) + b\right) \\ &= \operatorname{sign}\left(\sum_{i=1}^{n} \alpha_i y_i \phi(x_i)^T \phi(x) + b\right) \end{aligned}$$

Classification using the kernel

- Training:

$$\max_{\alpha} \quad \alpha^\top 1 - \frac{1}{2}\alpha^\top D_y \boxed{K} D_y \alpha \quad \text{s.t.} \quad \alpha^\top y = 0, \; 0 \leqslant \alpha \leqslant C$$

- Classification rule: classify new data point x:

$$\begin{aligned} f(\phi(x)) &= \operatorname{sign}\left(w^T \phi(x) + b\right) \\ &= \operatorname{sign}\left(\sum_{i=1}^{n} \alpha_i y_i \boxed{\phi(x_i)^T \phi(x)} + b\right) \\ & \qquad\qquad\qquad\qquad \Big\downarrow \\ &= \operatorname{sign}\left(\sum_{i=1}^{n} \alpha_i y_i \boxed{k(x_i, x)} + b\right) \end{aligned}$$

Optimizing over the kernel?

- **Primal** problem:

$$
\begin{aligned}
\boxed{\min_{K\succeq 0}\min_{\alpha}} \quad & \frac{1}{2}\alpha^\top D_y \boxed{K} D_y \alpha + C\sum_{i=1}^{n}\xi_i \\
\text{s.t.} \quad & D_y \boxed{K} D_y \alpha + y\beta + s - \xi = -e, \\
& 0 \leqslant \alpha_i \leqslant C, \xi \geqslant 0
\end{aligned}
$$

- Need additional conditions on K

When the unlabeled data is given

- **Primal** problem:

$$
\begin{aligned}
\boxed{\min_{K\succeq 0}\min_{\alpha}} \quad & \frac{1}{2}\alpha^\top D_y \boxed{K_{tr}} D_y\alpha + C\sum_{i=1}^{n}\xi_i \\
\text{s.t.} \quad & D_y \boxed{K_{tr}} D_y\alpha + y\beta + s - \xi = -e, \\
& 0 \leqslant \alpha_i \leqslant C,\ \ \xi \geqslant 0
\end{aligned}
$$

K_{tr}	$K_{\mathsf{ts;tr}}$
$K_{\mathsf{tr;ts}}$	K_{ts}

- Still need more conditions

Kernel methods with heterogeneous data

1

- First focus on every single source k of information individually
- Extract relevant information from source j into K_j

→ Focus on kernel design for specific types of information

2

- Design algorithm that learns the optimal K, by “mixing” any number of kernel matrices K_j, for a given learning problem

→ Homogeneous, standardized input

→ Flexibility

→ Can ignore information irrelevant for learning task

Classification with multiple kernels

- Consider a convex sets of kernels

$$K = \sum_{j=1}^{m} \eta_j K_{j,tr}$$

$$\sum_{j=1}^{m} \eta_j = c$$

$$\sum_{j=1}^{m} \eta_j K_j \succeq 0, \;\; \eta \geq 0$$

Can reformulate this as an SOCP

Convex combination of kernels

$$K_{tr} = \sum_{j=1}^{m} \eta_j K_{j,tr}$$

$$\sum_j \eta_j = c$$

$$K_j \succeq 0, \;\; \eta \geq 0$$

$$\min_{\eta_j \geqslant 0, \sum_j \eta_j = c} \left(\max_{\alpha, \alpha^\top y = 0} \quad \alpha^\top e - \frac{1}{2} \alpha^\top D_y \left(\sum_j \eta_j K_j \right) D_y \alpha \quad \text{s.t.} \quad 0 \leqslant \alpha \leqslant C \right)$$

Convex combination of kernels

$$\min_{\eta_j \geqslant 0, \sum_j \eta_j = c} \left(\max_{\alpha, \alpha^\top y = 0} \quad \alpha^\top e - \frac{1}{2}\alpha^\top \left(\sum_j \eta_j K_j \right) \alpha \quad \text{s.t.} \quad 0 \leqslant \alpha \leqslant C \right)$$

Omit D_y for simplicity

Because both problems are convex and have strictly feasible solutions

$$\left(\max_{\alpha, \alpha^\top y = 0} \quad \alpha^\top e - \max_{\eta_j \geqslant 0, \sum_j \eta_j = c} \frac{1}{2}\alpha^\top \left(\sum_j \eta_j K_j \right) \alpha \quad \text{s.t.} \quad 0 \leqslant \alpha \leqslant C \right)$$

Optimum of the linear function is achieved at the corners

$$\left(\max_{\alpha, \alpha^\top y = 0} \quad \alpha^\top e - c \max_j \frac{1}{2}\alpha^\top \left(K_j \right) \alpha \quad \text{s.t.} \quad 0 \leqslant \alpha \leqslant C \right)$$

Convex combination of kernels

$$\left(\max_{\alpha,\alpha^\top y=0} \quad \alpha^\top e - c \max_j \frac{1}{2}\alpha^\top \left(K_j\right) \alpha \quad \text{s.t.} \quad 0 \leqslant \alpha \leqslant C\right)$$

$$\Updownarrow$$

$$\begin{array}{ll}
\max\limits_{t,\alpha} & \alpha^\top e - ct \\
\text{s.t.} & t \geq \frac{1}{2}\alpha^\top K_j \alpha \\
 & y^\top \alpha = 0 \\
 & 0 \leqslant \alpha \leqslant C
\end{array}$$

Convex combination of kernels

$$\left(\max_{\alpha,\alpha^\top y=0} \quad \alpha^\top e - c\max_{j}\frac{1}{2}\alpha^\top\left(K_j\right)\alpha \quad \text{s.t.} \quad 0 \leqslant \alpha \leqslant C\right)$$

$$\Updownarrow$$

$$\begin{aligned}
\max_{t,\alpha} \qquad & \alpha^\top e - ct \\
\text{s.t.} \qquad & t \geq \frac{1}{2}\alpha^\top K_j \alpha \\
& y^\top \alpha = 0 \\
& 0 \leqslant \alpha \leqslant C
\end{aligned}$$

This is a QCQP

Convex combination of kernels

$$\left(\max_{\alpha,\alpha^\top y=0} \quad \alpha^\top e - c \max_j \frac{1}{2}\alpha^\top \left(K_j\right)\alpha \quad \text{s.t.} \quad 0 \leqslant \alpha \leqslant C\right)$$

A first order method

$$\begin{array}{ll} \max\limits_{t,\alpha} & \alpha^\top e - ct \\ \text{s.t.} & t \geq \frac{1}{2}\alpha^\top K_j \alpha \\ & y^\top \alpha = 0 \\ & 0 \leqslant \alpha \leqslant C \end{array}$$

An ASM

An IPM

Multiple kernels: primal problem

$$x \mapsto \begin{pmatrix} \phi_1(x) \\ \vdots \\ \phi_m(x) \end{pmatrix} \leftrightarrow w = \begin{pmatrix} w_1 \\ \vdots \\ w_m \end{pmatrix}$$

- Primal problem

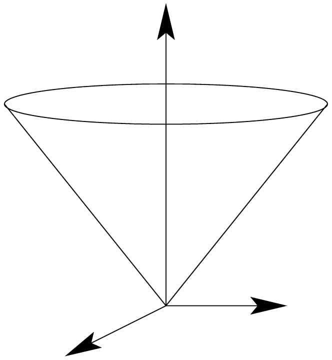
$$\begin{aligned} \text{min}_{w,b} \quad & \frac{1}{2}\Big(\sum_j ||w_j||_2\Big)^2 + C\sum_i \xi_i \\ s.t. \quad & y_i(w^\top \phi(x_i) + b) \geq 1 - \xi_i, \qquad i = 1, \ldots, n \\ & \xi_i \geq 0, \qquad\qquad\qquad\qquad\quad i = 1, \ldots, n \end{aligned}$$

Multiple kernels: dual problem

- Reformulation as an **SOCP**

$$\min_{w,b,t} \ \frac{1}{2}\left(\sum_j t_j\right)^2 + C\sum_i \xi_i \quad \text{s.t.} \quad \forall j, ||w_j||_2 \leqslant t_j$$

- **Constraint of type** $\boxed{||u||_2 \leqslant t}$
  - Second-order cone (Lorentz cone, “ice-cream cone”)



  - Self-dual cone

Multiple kernels: dual problem

- Dual problem

$$\max_{\alpha} \ \alpha^\top 1 - \frac{1}{2} \max_{j} \alpha^\top K_j \alpha \quad \text{s.t.} \quad \alpha^\top y = 0, \ 0 \leqslant \alpha \leqslant C$$

- KKT conditions
  - α is the solution of the SVM with **$K = \Sigma_j \eta_j K_j$**
    - η_j's: from conic duality
    - **equivalent** to previously obtained **QCQP** (for combining kernels)
  - **“Support vectors”**: x_i for which $\alpha_i > 0$
  - **“Support kernels”**: K_j for which $\eta_j > 0$

➔ **SKM: Support kernel machine**