10-725: Optimization **Fall 2012**

Lecture 21: November 8

Lecturer: Geoff Gordon/Ryan Tibshirani *Scribes: Martin Azizyan, Dwijaya Wijaya*

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

21.1 Maximum variance unfolding

Maximum variance unfolding (MVU, a.k.a. semidefinite embedding) is yet another example of a problem that can be expressed as a semidefinite program. The goal of maximum variance unfolding is as follows: given $\mathbf{x}_1, ..., \mathbf{x}_T \in \mathbb{R}^n$, find $\mathbf{y}_1, ..., \mathbf{y}_T \in \mathbb{R}^k$ ($k \ll n$) such that $\|\mathbf{y}_i - \mathbf{y}_j\| \approx \|\mathbf{x}_i - \mathbf{x}_j\|$ for $(i, j) \in E$ for some given edge set E.

MVU reduces to PCA if E contains all pairs of points, i.e. when we are trying to preserve all distances. However, PCA only has a good solution (i.e. one that preserves distances well) when x_i lie near a k-dimensional subspace of $\mathbb{R}^n$.

If we constrain E to contain e.g. only pairs of nearby points, then maximum variance unfolding finds a non-linear embedding of the points, meaning that we can preserve the local geometry of non-linear manifolds. For instance, the data in Figure 21.1 can be viewed as a 1 dimensional manifold embedded in $\mathbb{R}^2$, and finding the maximum variance unfolding with $k = 1$ would map the data points to some interval in $\mathbb{R}$.

Maximum variance unfolding will proceed in two steps:

- First, we find $\mathbf{z}_1, ..., \mathbf{z}_T \in \mathbb{R}^n$ with
$$\|\mathbf{z}_i - \mathbf{z}_j\| = \|\mathbf{x}_i - \mathbf{x}_j\| \ \ \forall (i, j) \in E$$
and $\text{var}(\mathbf{z})$ as large as possible.
- Next, we use PCA to get $\mathbf{y}_i$ from $\mathbf{z}_i$.

In essence, we maximize the variance of $\mathbf{z}_i$ in order to "stretch out" the manifold so that it is nearly linear. As we will see, this step is a semidefinite program.

Precisely, the optimization problem in the first step above is:

$$\begin{aligned} \max_{\mathbf{z}} \ & \text{tr}(\text{cov}(\mathbf{z})) \\ \text{s.t.} \ & \|\mathbf{z}_i - \mathbf{z}_j\| = \|\mathbf{x}_i - \mathbf{x}_j\| \ \ \forall (i, j) \in E. \end{aligned}$$

(Note that $\text{tr}(\text{cov}(\mathbf{z})) = \frac{1}{T} \sum_{i=1}^{T} \|\mathbf{z}_i - \overline{\mathbf{z}}\|^2$, where $\overline{\mathbf{z}} = \frac{1}{T} \sum_{i=1}^{T} \mathbf{z}_i$).

21.1.1 MVU as a semidefinite program

In order to show that this is a semidefinite program, we transform the problem as follows. Define $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T) \in \mathbb{R}^{n \times T}$, $\mathbf{Z} = (\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_T) \in \mathbb{R}^{n \times T}$, $\mathbf{P} = \mathbf{X}^T\mathbf{X}$, and $\mathbf{Q} = \mathbf{Z}^T\mathbf{Z}$. Our new optimization

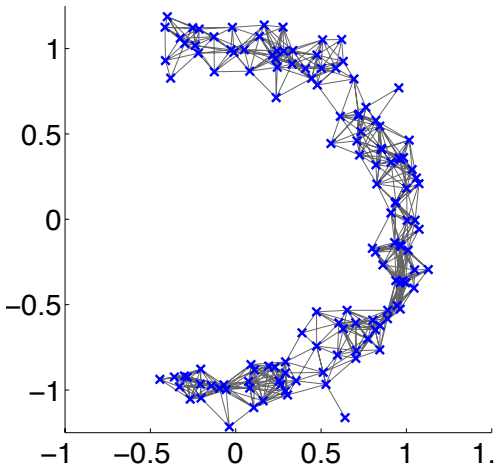

Figure 21.1: A non-linear manifold with neighborhood graph.

variable will be $\mathbf{Q}$ (subject to $\mathbf{Q} \succeq 0$, in order to be a valid matrix of inner products). We will recover an equivalent embedding $\mathbf{Z}$ by factoring $\mathbf{Q}$ (e.g. using a Cholesky decomposition). We now express the objectives and constraints in terms of $\mathbf{Q}$.

Consider the constraint

$$\|\mathbf{z}_i - \mathbf{z}_j\| = \|\mathbf{x}_i - \mathbf{x}_j\|.$$

Squaring both sides and expanding the norms, we obtain the equivalent constraint

$$\mathbf{z}_i^T\mathbf{z}_i - 2\mathbf{z}_i^T\mathbf{z}_j + \mathbf{z}_j^T\mathbf{z}_j = \mathbf{x}_i^T\mathbf{x}_i - 2\mathbf{x}_i^T\mathbf{x}_j + \mathbf{x}_j^T\mathbf{x}_j.$$

We can easily rewrite this in terms of the matrices $\mathbf{P}$ and $\mathbf{Q}$; the equality constraints are now

$$\mathbf{Q}_{ii} - 2\mathbf{Q}_{ij} + \mathbf{Q}_{jj} = \mathbf{P}_{ii} - 2\mathbf{P}_{ij} + \mathbf{P}_{jj} \quad \forall (i,j) \in E$$

which are linear constraints, so, together with the constraint $\mathbf{Q} \succeq 0$, we still have semidefinite constraints.

Now we derive an expression for $\mathrm{tr}(\mathrm{cov}(\mathbf{z}))$ in terms of $\mathbf{Q}$:

$$\begin{aligned}
\mathrm{cov}(\mathbf{z}) &= \frac{1}{T}\sum_{i=1}^{T}(\mathbf{z}_i - \overline{\mathbf{z}})(\mathbf{z}_i - \overline{\mathbf{z}})^T \\
&= \frac{1}{T}\sum_{i=1}^{T}\mathbf{z}_i\mathbf{z}_i^T - \overline{\mathbf{z}}\overline{\mathbf{z}}^T \\
&= \frac{1}{T}\mathbf{Z}\mathbf{Z}^T - \frac{1}{T^2}\mathbf{Z}\mathbf{1}\mathbf{1}^T\mathbf{Z}^T
\end{aligned}$$

and the trace is

$$\begin{aligned}
\mathrm{tr}(\mathrm{cov}(\mathbf{z})) &= \frac{1}{T}\mathrm{tr}(\mathbf{Z}\mathbf{Z}^T) - \frac{1}{T^2}\mathrm{tr}(\mathbf{Z}\mathbf{1}\mathbf{1}^T\mathbf{Z}^T) \\
&= \frac{1}{T}\mathrm{tr}(\mathbf{Z}^T\mathbf{Z}) - \frac{1}{T^2}\mathrm{tr}(\mathbf{Z}^T\mathbf{Z}\mathbf{1}\mathbf{1}^T) \\
&= \frac{1}{T}\mathrm{tr}(\mathbf{Q}) - \frac{1}{T^2}\mathrm{tr}(\mathbf{Q}\mathbf{1}\mathbf{1}^T).
\end{aligned}$$

So the full SDP is

$$\begin{aligned}\max_{\mathbf{Q}} \ & \frac{1}{T}\mathrm{tr}(\mathbf{Q}) - \frac{1}{T^2}\mathrm{tr}(\mathbf{Q}\mathbf{1}\mathbf{1}^T) \\ \text{s.t.} \ & \mathbf{Q}_{ii} - 2\mathbf{Q}ij + \mathbf{Q}_{jj} = \mathbf{P}_{ii} - 2\mathbf{P}ij + \mathbf{P}_{jj} \ \ \forall (i,j) \in E \\ & \mathbf{Q} \succeq 0.\end{aligned}$$

21.2 Duality for quadratic programs (QPs) and cone programs (CPs)

Consider a quadratic cone program:

$$\begin{aligned}\min_{\mathbf{x}} \ & \mathbf{c}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} \\ \text{s.t.} \ & \mathbf{A}\mathbf{x} + \mathbf{b} \in K \\ & \mathbf{x} \in L\end{aligned}$$

where cones K and L encode all equality, inequality, and generalized inequality constraints. We assume K and L are closed and convex. This is just a cone program if $\mathbf{H} = 0$, and a quadratic program if e.g. K and L are the non-negative orthants.

Let $\mathbf{y} \in K^*$ and $\mathbf{s} \in L^*$. By definition, $\mathbf{y}^T(\mathbf{A}\mathbf{x} + \mathbf{b}) \geq 0$ and $\mathbf{s}^T\mathbf{x} \geq 0$. Then

$$\begin{aligned}\mathbf{c}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} &\geq \mathbf{c}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} - \mathbf{y}^T(\mathbf{A}\mathbf{x} + \mathbf{b}) - \mathbf{s}^T\mathbf{x} \\ &\geq \min_{\mathbf{z}} \mathbf{c}^T\mathbf{z} + \frac{1}{2}\mathbf{z}^T\mathbf{H}\mathbf{z} - \mathbf{y}^T(\mathbf{A}\mathbf{z} + \mathbf{b}) - \mathbf{s}^T\mathbf{z}.\end{aligned}$$

We can compute the minimum in the last line above by differentiating w.r.t. $\mathbf{z}$ and setting the gradient of the quadratic to 0:

$$\mathbf{0} = \mathbf{c} + \mathbf{H}\mathbf{z} - \mathbf{A}^T\mathbf{y} - \mathbf{s}$$

i.e.

$$\mathbf{H}\mathbf{z} = \mathbf{s} + \mathbf{A}^T\mathbf{y} - \mathbf{c}$$

which is going to be a constraint in the dual program. Substituting this back into the inequality above,

$$\begin{aligned}\mathbf{c}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} &\geq (\mathbf{c} - \mathbf{A}^T\mathbf{y} - \mathbf{s})^T\mathbf{z} + \frac{1}{2}\mathbf{z}^T\mathbf{H}\mathbf{z} - \mathbf{y}^T\mathbf{b} \\ &= -\frac{1}{2}\mathbf{z}^T\mathbf{H}\mathbf{z} - \mathbf{y}^T\mathbf{b}\end{aligned}$$

and the dual program is (**scribe's note:** I'm not sure if the following optimization problem should be with respect to $\mathbf{z}$ as well)

$$\begin{aligned}\max_{\mathbf{y},\mathbf{s},\mathbf{z}} \ & -\frac{1}{2}\mathbf{z}^T\mathbf{H}\mathbf{z} - \mathbf{y}^T\mathbf{b} \\ \text{s.t.} \ & \mathbf{0} = \mathbf{c} + \mathbf{H}\mathbf{z} - \mathbf{A}^T\mathbf{y} - \mathbf{s} \\ & \mathbf{s} \in L^* \\ & \mathbf{y} \in K^*.\end{aligned}$$

We can eliminate the variable $\mathbf{s}$ by rewriting the first constraint as $\mathbf{s} = \mathbf{c} + \mathbf{H}\mathbf{z} - \mathbf{A}^T\mathbf{y}$ and combining it with the second constraint, so the dual program becomes

$$\begin{aligned} \max_{\mathbf{y},\mathbf{z}} \quad & -\frac{1}{2}\mathbf{z}^T\mathbf{H}\mathbf{z} - \mathbf{y}^T\mathbf{b} \\ \text{s.t.} \quad & \mathbf{H}\mathbf{z} + \mathbf{c} - \mathbf{A}^T\mathbf{y} \in L^* \\ & \mathbf{y} \in K^*. \end{aligned}$$

21.2.1 KKT conditions

Recall the primal and dual quadratic cone problems:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{c}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} + \mathbf{b} \in K \\ & \mathbf{x} \in L \end{aligned}$$

$$\begin{aligned} \max_{\mathbf{y},\mathbf{z}} \quad & -\frac{1}{2}\mathbf{z}^T\mathbf{H}\mathbf{z} - \mathbf{y}^T\mathbf{b} \\ \text{s.t.} \quad & \mathbf{H}\mathbf{z} + \mathbf{c} - \mathbf{A}^T\mathbf{y} \in L^* \\ & \mathbf{y} \in K^*. \end{aligned}$$

The primal feasibility conditions are

$$\mathbf{A}\mathbf{x} + \mathbf{b} \in K \quad \text{and} \quad \mathbf{x} \in L.$$

The dual feasibility conditions are

$$\mathbf{H}\mathbf{z} + \mathbf{c} - \mathbf{A}^T\mathbf{y} \in L^* \quad \text{and} \quad \mathbf{y} \in K^*.$$

The last set of constraints is equality of the primal and dual objective values:

$$\mathbf{c}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} = -\frac{1}{2}\mathbf{z}^T\mathbf{H}\mathbf{z} - \mathbf{y}^T\mathbf{b}.$$

We will now transform this into a more interpretable form.

First, we rewrite the constraint as

$$\frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} + \frac{1}{2}\mathbf{z}^T\mathbf{H}\mathbf{z} + \mathbf{c}^T\mathbf{x} + \mathbf{y}^T\mathbf{b} = \mathbf{0}.$$

We would like to "complete the square", i.e. combine the quadratic terms into a single term $\frac{1}{2}(\mathbf{x}-\mathbf{z})^T\mathbf{H}(\mathbf{x}-\mathbf{z})$, so we add and subtract $\mathbf{x}^T\mathbf{H}\mathbf{z}$ from the left hand side:

$$\frac{1}{2}(\mathbf{x}-\mathbf{z})^T\mathbf{H}(\mathbf{x}-\mathbf{z}) + \mathbf{x}^T\mathbf{H}\mathbf{z} + \mathbf{c}^T\mathbf{x} + \mathbf{y}^T\mathbf{b} = \mathbf{0}.$$

By also adding and subtracting $\mathbf{x}^T\mathbf{A}^T\mathbf{y}$ and collecting some terms, we obtain the form

$$\frac{1}{2}(\mathbf{x}-\mathbf{z})^T\mathbf{H}(\mathbf{x}-\mathbf{z}) + (\mathbf{A}\mathbf{x}+\mathbf{b})^T\mathbf{y} + \mathbf{x}^T(\mathbf{c} + \mathbf{H}\mathbf{z} - \mathbf{A}^T\mathbf{y}) = \mathbf{0}.$$

Observe that $\frac{1}{2}(\mathbf{x}-\mathbf{z})^T\mathbf{H}(\mathbf{x}-\mathbf{z}) \geq 0$ since it is a PSD quadratic term, $(\mathbf{Ax}+\mathbf{b})^T\mathbf{y} \geq 0$ since $\mathbf{Ax}+\mathbf{b} \in K$ and $\mathbf{y} \in K^*$, and $\mathbf{x}^T(\mathbf{c}+\mathbf{Hz}-\mathbf{A}^T\mathbf{y}) \geq 0$ since $\mathbf{x} \in L$ and $\mathbf{c}+\mathbf{Hz}-\mathbf{A}^T\mathbf{y} \in L^*$. Hence, all three terms must equal 0, and the full set of KKT conditions are:

$$\begin{aligned}
&\mathbf{Ax}+\mathbf{b} \in K \quad \text{and} \quad \mathbf{x} \in L \quad \text{(primal feasibility)},\\
&\mathbf{Hz}+\mathbf{c}-\mathbf{A}^T\mathbf{y} \in L^* \quad \text{and} \quad \mathbf{y} \in K^* \quad \text{(dual feasibility)},\\
&(\mathbf{Ax}+\mathbf{b})^T\mathbf{y} = \mathbf{0} \quad \text{and} \quad \mathbf{x}^T(\mathbf{c}+\mathbf{Hz}-\mathbf{A}^T\mathbf{y}) = \mathbf{0} \quad \text{(comp. slackness)},\\
&\mathbf{Hx} = \mathbf{Hz}.
\end{aligned}$$

21.2.2 Support Vector Machines (Separable Case)

This is one of the most important quadratic problems in Machine Learning, where duality makes a big difference, where in some problems it is much easier to solve the dual than the primal or vice versa; depending on the relative size of the dimensionality of the problem and number of examples.

SVM is a classification problem. Assuming separable case, where we can classify the data without errors, the task is to find a classification surface that separates the positive from negative data points.

Given $\mathbf{x}_i \in \mathbb{R}^n$ and $\mathbf{y}_i \in \{-1, 1\}$, the classification surface can be expressed as $\mathbf{wx}-\mathbf{b}=0$, where $\mathbf{w}$ is a vector in the direction of the normal to the classification surface and $\mathbf{b}$ controls the intersect.

If particular if we define $\overline{\mathbf{w}} = \frac{\mathbf{w}}{\|\mathbf{w}\|}$ and $\overline{\mathbf{b}} = \frac{\mathbf{b}}{\|\mathbf{w}\|}$, $\overline{\mathbf{w}}$ will be a unit vector in the direction of the normal (i.e. orthogonal to the classification surface), and the $\overline{\mathbf{b}}$ will be the distance along the direction of the normal, from the origin to the classification surface (Figure 21.2).

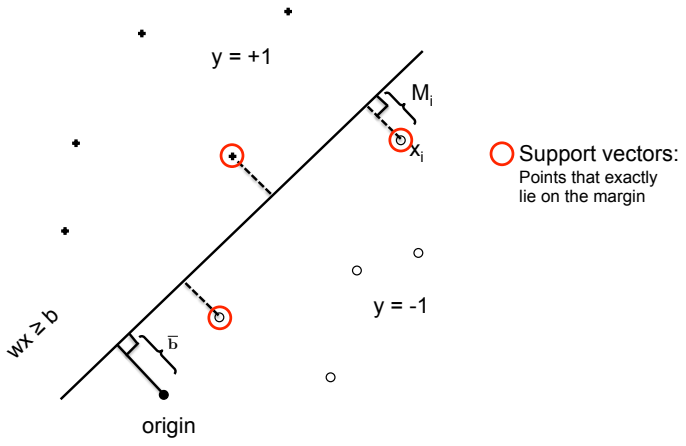

Figure 21.2: SVM

In SVM, we are concerned about the margin by which we separate the positive from negative examples. The margin for the i^{th} example is defined as $\mathbf{M}_i = \mathbf{y}_i\left(\overline{\mathbf{w}}\mathbf{x}_i - \overline{\mathbf{b}}\right)$, which is the distance along the direction of the normal, from the point $\mathbf{x}_i$ to the classification surface. The problem is then to find the hyperplane that separates the data points with maximum possible margin:

$$\begin{aligned} \max \ & \mathbf{M} \\ \text{s.t.} \ & \mathbf{M} \leq \mathbf{y}_i \left(\overline{\mathbf{w}} \mathbf{x}_i - \overline{\mathbf{b}} \right) \quad \forall i \end{aligned}$$

But this optimization problem does not have a convex constraint (due to $\overline{\mathbf{w}}$ that is constrained to be in a unit sphere that is not a convex constraint). Defining $\mathbf{v} = \frac{\overline{\mathbf{w}}}{\mathbf{M}}$ and $\mathbf{d} = \frac{\overline{\mathbf{b}}}{\mathbf{M}}$, we have $\overline{\mathbf{w}} = \mathbf{M}\mathbf{v}$, $\overline{\mathbf{b}} = \mathbf{M}\mathbf{d}$, and $\|\mathbf{v}\| = \frac{1}{\mathbf{M}}$. Hence the optimization problem can be rewritten as:

$$\begin{aligned} \max \ & \frac{1}{\|\mathbf{v}\|} \\ \text{s.t.} \ & 1 \leq \mathbf{y}_i \left(\mathbf{v}\mathbf{x}_i - \mathbf{d} \right) \quad \forall i \end{aligned}$$

But this has non convex objective. Since $\frac{1}{\|\mathbf{v}\|}$ is monotone decreasing in $\|\mathbf{v}\|$ and $\|\mathbf{v}\|^2$ is monotone increasing in $\|\mathbf{v}\|$, making a monotone transformation of the objective function:

$$\begin{aligned} \min \ & \|\mathbf{v}\|^2 \\ \text{s.t.} \ & 1 \leq \mathbf{y}_i \left(\mathbf{v}\mathbf{x}_i - \mathbf{d} \right) \quad \forall i \end{aligned}$$

which is a quadratic program of convex objective and linear constraints.

For non separable case, we can introduce slack variables $\mathbf{s}_i \geq 0$ in the optimization problem:

$$\begin{aligned} \min \ & \|\mathbf{v}\|^2 + \mathbf{C} \sum_i \mathbf{s}_i \\ \text{s.t.} \ & 1 - \mathbf{s}_i \leq \mathbf{y}_i \left(\mathbf{v}\mathbf{x}_i - \mathbf{d} \right) \quad \forall i \end{aligned}$$

Depending on the value of $\mathbf{C}$, we trade-off between making the margin as wide as possible versus making as few mistakes (i.e. low slacks) as possible. To make slacks vector sparse (few mistakes), tricks such as $L1$-penalty on the vector are used.