GI13/4C60: Advanced Topics on Machine Learning

Part 2: Kernel-based Learning Algorithms

Massimiliano Pontil

http://www.cs.ucl.ac.uk/staff/M.Pontil/

About these lectures

Theme: We discuss two important kernel-based methods: kernel principal component analysis and support vector machines (SVMs). The first method serves to illustrate simple operations with kernels (such as centering the data and computing projections in a feature space of the kernel). A SVM implements the idea of optimal margin classifier in a feature space and is based on a convex optimization problem similar to that of ridge regression.

Math required: Calculus, elements of linear algebra, convex functions, Supervised Learning (GI01/4C55).

Outline

- Kernel principal component analysis
  - Linear projections
  - Singular value decomposition and PCA (review)
- Support vector machines
  - Optimal separating hyperplane
  - Soft margin separation
  - Extensions

Centering the data

We wish to translate the coordinate system $\mathbf{x} \mapsto \hat{\mathbf{x}} = \mathbf{x} - \boldsymbol{\mu}$ such that the average of the $\hat{\mathbf{x}}_i$ is zero.

Clearly, this is achieved for $\boldsymbol{\mu} = \frac{1}{\ell} \sum_{i=1}^{\ell} \mathbf{x}_i$

We also have

$$\begin{aligned} \hat{\mathbf{x}}'\hat{\mathbf{t}} &= \left(\mathbf{x} - \frac{1}{\ell} \sum_{i=1}^{\ell} \mathbf{x}_i\right)' \left(\mathbf{t} - \frac{1}{\ell} \sum_{i=1}^{\ell} \mathbf{x}_i\right) \\ &= \mathbf{x}'\mathbf{t} - \frac{1}{\ell} \sum_{i=1}^{\ell} \mathbf{x}\mathbf{x}_i - \frac{1}{\ell} \sum_{i=1}^{\ell} \mathbf{t}\mathbf{x}_i + \frac{1}{\ell^2} \sum_{i,j=1}^{\ell} \mathbf{x}_i'\mathbf{x}_j \end{aligned}$$

Centering the data in a feature space

If we are working with a feature map ϕ of kernel K we have

$$\hat{\phi}(\mathbf{x}) = \phi(\mathbf{x}) - \frac{1}{\ell}\sum_{i=1}^{\ell}\phi(\mathbf{x}_i)$$

and the 'centered kernel', $\hat{K}(\mathbf{x},\mathbf{t}) = \langle\hat{\phi}(\mathbf{x}),\hat{\phi}(\mathbf{t})\rangle$, is

$$\hat{K}(\mathbf{x},\mathbf{t}) = K(\mathbf{x},\mathbf{t}) - \frac{1}{\ell}\sum_{i=1}^{\ell}(K(\mathbf{x},\mathbf{x}_i) + K(\mathbf{t},\mathbf{x}_i)) + \frac{1}{\ell^2}\sum_{i,j=1}^{\ell}K(\mathbf{x}_i,\mathbf{x}_j)$$

The kernel matrix $\mathbf{K}$ becomes

$$\hat{\mathbf{K}} = \mathbf{K} - \frac{1}{\ell}\mathbf{1}\mathbf{1}'\mathbf{K} - \frac{1}{\ell}\mathbf{K}\mathbf{1}\mathbf{1}' + \frac{1}{\ell^2}\mathbf{1}\mathbf{K}\mathbf{1}'\mathbf{1}\mathbf{1}'$$

where $\mathbf{1}$ is the $\ell-$dimensional vector with all entries equal to 1.

Linear projection

By a projection we mean a linear operator $P : \mathbb{R}^n \to \mathbb{R}^n$ such that: (i) $P = P^2$ and (ii) $\langle P(\mathbf{x}), \mathbf{x} - P(\mathbf{x})\rangle = 0\ \forall \mathbf{x} \in \mathbb{R}^n$. **Remark:** In general a projection only requires property (i). If also (ii) hold we speak of orthogonal projection

The orthogonal projection to P, denoted by $P^{\perp}$ is defined as $P^{\perp}(\mathbf{x}) = \mathbf{x} - P(\mathbf{x})$. We have that: $dim(P^{\perp}) = n - \dim(P)$

Exercise: Show that $P^{\perp}$ is a projection.

Projections satisfy Pythagoras's Theorem:
$\|P_U^{\perp}(\mathbf{x})\| = \|\mathbf{x} - P_U(\mathbf{x})\| = \|\mathbf{x}\| - \|P_U(\mathbf{x})\|$

Remark: We can equivalently speak on an $n \times n$ projection matrix $\mathbf{P}$ such that a) $\mathbf{P}\mathbf{P} = \mathbf{P}$, b) $\mathbf{P}' = \mathbf{P}$ (which is equivalent to (ii)). One can also show that $\mathbf{P}$ is symmetric.

Linear projections (cont.)

If $\mathcal{M}$ is a linear subspace of $\mathbb{R}$, the projection $P_{\mathcal{M}}$ on $\mathcal{M}$ is defined as

$$P_{\mathcal{M}}(\mathbf{x}) = \text{argmin}_{\mathbf{z}}\{\|\mathbf{z}-\mathbf{x}\| : \mathbf{z} \in \mathcal{M}\}$$

In particular if $\mathcal{M} = span\{\mathbf{u}_1, \ldots, \mathbf{u}_t\}$ and the vector $\mathbf{u}_i$ are orthonormal we have

$$P_{\mathcal{M}}(\mathbf{x}) = \mathbf{U}\mathbf{U}'\mathbf{x}$$

($\mathbf{U}\mathbf{U}'$ the projection matrix associated to $P_{\mathcal{M}}$) where $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_t]$ and

$$P_{\mathcal{M}}^{\perp}(\mathbf{x}) = (\mathbf{I} - \mathbf{U}\mathbf{U}')\mathbf{x}, \quad \mathbf{x} \in \mathbb{R}^n$$

Remark: Careful! if $\mathbf{u}$ is a unit vector we also denote the projection of $\mathbf{x}$ on $\mathbf{u}$ as $P_{\mathbf{u}}(\mathbf{x}) = \mathbf{u}'\mathbf{x}$ (this is a scalar!).

Singular valued decomposition (review)

Let $\mathbf{X}' = [\mathbf{x}_1, \cdots, \mathbf{x}_\ell]$ be an $n \times \ell$ matrix, $\ell\mathbf{C} = \mathbf{X}'\mathbf{X}$ and $\mathbf{K} = \mathbf{X}\mathbf{X}'$. All these matrices have the same rank $t \leq \min(n, \ell)$. Singular value decomposition (SVD) establishes that

$$\ell\mathbf{C} = \mathbf{U}\Lambda_n\mathbf{U}' \quad \mathbf{K} = \mathbf{X}\mathbf{X}' = \mathbf{V}\Lambda_\ell\mathbf{V}'$$

where $\mathbf{U}, \mathbf{V}$ are orthogonal matrices, $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_n]$, $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_\ell]$,

$$\Lambda_n = diag(\lambda_1, \ldots, \lambda_t, \mathbf{0}_{n-t}), \qquad \Lambda_\ell = diag(\lambda_1, \ldots, \lambda_t, \mathbf{0}_{\ell-t})$$

and $\lambda_1 \geq \cdots \geq \lambda_t > 0$. Moreover, we have

$$\sqrt{\lambda_j}\mathbf{u}_j = \mathbf{X}'\mathbf{v}_j$$

that is, $\mathbf{X}' = \mathbf{U}\Sigma\mathbf{V}'$, where Σ is the $n \times \ell$ matrix with leading diagonal entries $\sigma_j = \sqrt{\lambda_j}$.

Projection of a new point

We rewrite the above equation, $\sqrt{\lambda_j}\mathbf{u}_j = \mathbf{X}'\mathbf{v}_j$, as

$$\mathbf{u}_j = \frac{1}{\sqrt{\lambda_j}} \sum_{i=1}^{\ell} (\mathbf{v}_j)_i \mathbf{x}_i = \sum_{i=1}^{\ell} \alpha_i^j \mathbf{x}_i, \quad j \in \mathbb{N}_t$$

where we have defined the 'dual variables' $\alpha_i^j = \frac{(\mathbf{v}_j)_i}{\sqrt{\lambda_j}}$.

Thus, the projection on $\mathbf{u}_j$ of a point $\mathbf{x}$ is given by

$$P_{\mathbf{u}_j}(\mathbf{x}) = \mathbf{u}_j'\mathbf{x} = \sum_{i=1}^{\ell} \alpha_i^j \mathbf{x}_i \mathbf{x}$$

Projections in a feature space

All the above observations hold true in a feature space. Given the data matrix $\mathbf{X}' = [\boldsymbol{\phi}(\mathbf{x}_1), \cdots, \boldsymbol{\phi}(\mathbf{x}_\ell)]$ the projection on $\mathbf{u}_j$ in the feature space is given by

$$P_{\mathbf{u}_j}(\boldsymbol{\phi}(\mathbf{x})) = \langle \mathbf{u}_j, \boldsymbol{\phi}(\mathbf{x}) \rangle = \sum_{i=1}^{\ell} \alpha_i^j \langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}) \rangle = \sum_{i=1}^{\ell} \alpha_i^j K(\mathbf{x}_i, \mathbf{x})$$

where $\alpha_i^j = \frac{(\mathbf{v}_j)_i}{\sqrt{\lambda_j}}$ and, recall, $\lambda_j, \mathbf{v}_j$ are the eigen-values/vectors of the kernel matrix $\mathbf{K}$

These observations will later lead us to kernel-PCA...

PCA (review)

Suppose our datapoints $\mathbf{x}_1, \ldots, \mathbf{x}_\ell$ have zero mean (are 'centered'). The direction $\mathbf{w}$ of maximal variance is the unique solution to the optimization problem

$$\min_{\mathbf{w}} \left\{ \mathbf{w}'\mathbf{C}\mathbf{w} : \mathbf{w}'\mathbf{w} = 1 \right\} = \min_{\mathbf{w}} \left\{ \frac{\mathbf{w}'\mathbf{C}\mathbf{w}}{\mathbf{w}'\mathbf{w}} \right\}$$

Indeed, the variance of the projection along the direction $\mathbf{w}$ is given by

$$\frac{1}{\ell}\sum_{i=1}^{\ell}(\mathbf{w}'\mathbf{x}_i)^2 = \frac{1}{\ell}\sum_{i=1}^{\ell}\mathbf{w}'\mathbf{x}_i\mathbf{x}_i'\mathbf{w} = \mathbf{w}'\left(\frac{1}{\ell}\sum_{i=1}^{\ell}\mathbf{x}_i\mathbf{x}_i'\right)\mathbf{w} = \mathbf{w}'\mathbf{C}\mathbf{w}$$

PCA (review – cont.)

The solution to the above optimization problem is given by the eigenvector corresponding to the largest eigenvalue of $\mathbf{C}$

$$\mathbf{C}\mathbf{w} = \lambda\mathbf{w}$$

Remark: we assume for simplicity that the positive eigenvalues of C are distinct.

We can repeat this operation to find directions of ordered maximum variance. Let $\lambda_1, \mathbf{w}_1$ be the largest eigenvector and $\mathbf{w}_1$ the corresponding eigenvalue. The direction of maximum variance in the space orthogonal to $\mathbf{w}_1$ is the eigenvector of the second eigenvalue λ_2 of $\mathbf{C}$ (indeed, if we deflate the data according to $\mathbf{w}_1$ we obtain the matrix $\mathbf{C} - \lambda_1\mathbf{w}_1\mathbf{w}_1'$ (verify this!)...)

PCA (review – cont.)

Thus, the set of r orthogonal directions of maximum variance consists of the r eigenvectors of matrix $\mathbf{C}$ corresponding to the eigenvalues $\lambda_1 > \cdots > \lambda_r \geq 0$. Their total variance is $\sum_{i=1}^{r} \lambda_r$

The ratio

$$\frac{\sum_{i=1}^{r} \lambda_i}{\sum_{i=1}^{\ell} \lambda_i} = \frac{\sum_{i=1}^{r} \lambda_i}{trace(\mathbf{K})}$$

measures the percentage of the overall variance captured by these first r eigenvectors (note that this is 1 for $r \geq rank(\mathbf{X})$

Remark: $\mathbf{X}'\mathbf{X} = \ell\mathbf{C}$ has the same eigenvectors as $\mathbf{C}$ with eigenvalues scaled by ℓ. The quantity $\ell\lambda_i$ is the sum of the projection of the data along the eigenvector $\mathbf{w}_i$

PCA: an alternative characterization

The following result is expected. Its proof is left as an exercise

Theorem: The orthogonal projection in the space spanned by the first r eigenvectors of the correlation matrix $\mathbf{C}$ is the solution to the following optimization problem:

$$\max_{P} \left\{ \frac{1}{\ell} \sum_{i=1}^{\ell} \|P(\mathbf{x}_i)\|^2 : rank(P) = k \right\}$$

Moreover, the maximum equals to $\sum_{i=1}^{r} \lambda_i$

Remark: By Pythagoras's Theorem this maximization problem is equivalent to minimize $\frac{1}{\ell} \sum_{i=1}^{\ell} \|P^{\perp}(\mathbf{x}_i)\|^2$

PCA algorithm (summary)

- Input: $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_\ell\}$, $r \in \mathbb{N}$

1. Compute data mean: $\boldsymbol{\mu} = \frac{1}{\ell} \sum_{i=1}^{\ell} \mathbf{x}_i$
2. Compute covariance: $C = \frac{1}{\ell} \sum_{i=1}^{\ell} (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})'$
3. Compute first r eigen-values/vectors of $\mathbf{C}$: $(\lambda_i, \mathbf{u}_i), i \in \mathbb{N}_r$
4. Let $\tilde{\mathbf{x}}_i = \mathbf{U}_r \mathbf{x}_i$, $i \in \mathbb{N}_\ell$, $\mathbf{U}_r = [\mathbf{u}_1, \ldots, \mathbf{u}_r]$

- Output: $\widehat{S} = \{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_\ell\}$

Remark: In practice r needs to be tuned, say choose r so that 95% of the variance is captured (by cross-validation) and step 3 is replaced by running SVD only once.

Kernel PCA

- Input: $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_\ell\}$, kernel function K, $r \in \mathbb{N}$

1. Compute kernel matrix : $\mathbf{K} = (K(\mathbf{x}_i, \mathbf{x}_j) : i, j \in \mathbb{N}_\ell)$
2. Center kernel matrix: $\mathbf{K} \mapsto \mathbf{K} - \frac{1}{\ell}\mathbf{1}\mathbf{1}'\mathbf{K} - \frac{1}{\ell}\mathbf{K}\mathbf{1}\mathbf{1}' + \frac{1}{\ell^2}\mathbf{1}\mathbf{K}\mathbf{1}'\mathbf{1}\mathbf{1}'$
3. Compute first r eigenvalues/vectors of $\mathbf{K}$: $(\lambda_i, \mathbf{w}_i), i \in \mathbb{N}_r$
4. Set $\boldsymbol{\alpha}^j = \frac{\mathbf{v}_j}{\sqrt{\lambda_j}}$, $j \in \mathbb{N}_r$
5. Let $\tilde{\mathbf{x}}_k = \left(\sum_{i=1}^{\ell} \alpha_i^j K(\mathbf{x}_i, \mathbf{x}_k) \right)_{j=1}^{r}$

- Output: $\widehat{S} = \{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_\ell\}$

Remark: A new point $\mathbf{x}$ is transformed to $\left(\sum_{i=1}^{\ell} \alpha_i^j K(\mathbf{x}_i, \mathbf{x}) \right)_{j=1}^{r}$

Some remarks

- The above algorithm can also be applied to not centered data. The advantage of working with centered data is that, in this case, the variance along any fixed direction is minimized if we subtract the mean (we remove irrelevant variance due to the shift of the data)
- Kernel-PCA was original proposed by Schölkopf et al. where they shown the advantage of this method over PCA for optical character recognition and de-noising.

Support vector machines (SVMs)

- Linearly separable data and optimal separating hyperplane (*aka* hard-margin linear SVM)
- Generalized optimal separating hyperplane (*aka* soft-margin linear SVM)
- Non-linear extension and SVM
- Variations on the theme
- Connection to regularization

Separating hyperplane

Let $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{\ell} \in \mathbb{R}^n \times \{-1, 1\}$ be a training set.

By hyperplane we mean the set $H_{\mathbf{w},b} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{w}'\mathbf{x} + b = 0\}$

We assume that the data are linearly separable, that is, there exist $\mathbf{w} \in \mathbb{R}^n$ and $b \in \mathbb{R}$ such that

$$y_i(\mathbf{w}'\mathbf{x}_i + b) > 0, \quad i \in \mathbb{N}_\ell \tag{1}$$

we call $H_{\mathbf{w},b}$ a *separating hyperplane*

Remark: Note that we require the inequality in eq.(1) to be strict

Separating hyperplane (cont.)

The distance $\rho_{\mathbf{x}}(\mathbf{w}, b)$ of a point $\mathbf{x}$ from a hyperplane $H_{\mathbf{w},b}$ is

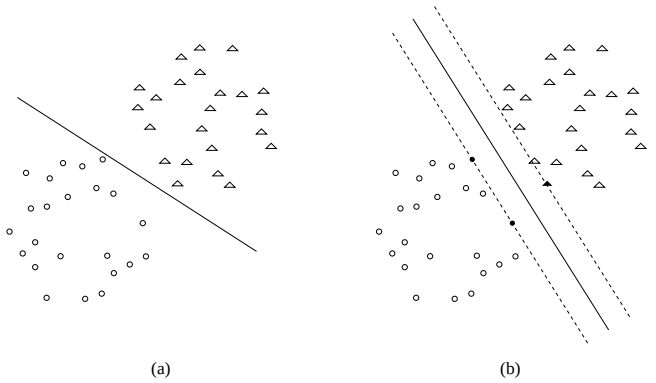
$$\rho_{\mathbf{x}}(\mathbf{w}, b) := \frac{|\mathbf{w}'\mathbf{x} + b|}{\|\mathbf{w}\|}$$

If $H_{\mathbf{w},b}$ separates the training set S we define its *margin* as

$$\rho_S(\mathbf{w}, b) := \min_{i=1}^{\ell} \rho_{\mathbf{x}_i}(\mathbf{w}, b)$$

If $H_{\mathbf{w},b}$ is a hyperplane (separating or not) we also define the *margin of a point* $\mathbf{x}$ as $\mathbf{w}'\mathbf{x} + b$ (note that this can be positive or negative)

Optimal separating hyperplane



The **optimal separating hyperplane** (**OSH**) is the separating hyperplane with maximum margin. It solves the optimization problem

$$\max_{\mathbf{w},b} \rho_S(\mathbf{w},b) = \max_{\mathbf{w},b} \left\{ \min_i \left\{ \frac{y_i(\mathbf{w}'\mathbf{x}_i + b)}{\|\mathbf{w}\|} \right\} : y_i(\mathbf{w}'\mathbf{x}_i + b) \geq 0 \right\}$$

Choosing a parameterization

A hyperplane is parameterized by coefficients $(\mathbf{w}, b)$, but this choice is not unique (rescaling with a positive constant gives the same hyperplane). Two possible ways to fix the parametrization of a separating hyperplane:

- *Normalized hyperplane:* set $\|\mathbf{w}\| = 1$, in which case $\rho_{\mathbf{x}}(\mathbf{w}, b) = |\mathbf{w}'\mathbf{x} + b|$ and $\rho_S(\mathbf{w}, b) = \min_{i=1}^{\ell} y_i(\mathbf{w}'\mathbf{x}_i + b)$.

- *Canonical hyperplane:* choose $\|\mathbf{w}\|$ such that $\rho_S(\mathbf{w}, b) = \frac{1}{\|\mathbf{w}\|}$, ie. we require that $\min_{i=1}^{\ell} y_i(\mathbf{w}'\mathbf{x}_i + b) = 1$ (a data-dependent parameterization).

We will mainly work with the second parametrization

Optimal separating hyperplane

- If we work with normalized hyperplanes we have

$$\rho_S(\mathbf{w},b) = \max_{\mathbf{w},b}\left\{\min_i\left\{y_i(\mathbf{w}'\mathbf{x}_i+b)\right\} : y_i(\mathbf{w}'\mathbf{x}_i+b)\geq 0, \|\mathbf{w}\|=1\right\}$$

- If we work with canonical hyperplanes, instead, we have

$$\begin{aligned}\rho_S(\mathbf{w},b) &= \max_{\mathbf{w},b}\left\{\frac{1}{\|\mathbf{w}\|} : \min_i\{y_i(\mathbf{w}'\mathbf{x}_i+b)\}=1, y_i(\mathbf{w}'\mathbf{x}_i+b)\geq 0\right\}\\ &= \max_{\mathbf{w},b}\left\{\frac{1}{\|\mathbf{w}\|} : y_i(\mathbf{w}'\mathbf{x}_i+b)\geq 1\right\}\\ &= \frac{1}{\min_{\mathbf{w},b}\{\|\mathbf{w}\| : y_i(\mathbf{w}'\mathbf{x}_i+b)\geq 1\}}\end{aligned}$$

Optimal separating hyperplane (cont.)

We choose to work with canonical hyperplanes and, so, look at the optimization problem

Problem **P1**	
Minimize	$\frac{1}{2}\mathbf{w}'\mathbf{w}$
subject to	$y_i(\mathbf{w}'\mathbf{x}_i+b)\geq 1,\ i=1,\ldots,\ell.$

The quantity $1/\|\mathbf{w}\|$ is the **margin** of the OSH.

Saddle point

The solution of problem **P1** is equivalent to determine the **saddle point** of the Lagrangian function

$$L = \frac{1}{2}\mathbf{w}'\mathbf{w} - \sum_{i=1}^{\ell} \alpha_i \left\{ y_i(\mathbf{w}'\mathbf{x}_i + b) - 1 \right\} \tag{2}$$

where $\alpha_i \geq 0$ are the Lagrange multipliers.

L has a minimum wrt. $(\mathbf{w}, b)$ and a maximum wrt. α. Differentiating w.r.t $\mathbf{w}$ and b we obtain:

$$\begin{aligned} \frac{\partial L}{\partial b} &= \sum_{i=1}^{\ell} y_i \alpha_i = 0 \\ \frac{\partial L}{\partial \mathbf{w}} &= \mathbf{w} - \sum_{i=1}^{\ell} \alpha_i y_i \mathbf{x}_i = 0 \;\Rightarrow \mathbf{w} = \sum_{i=1}^{\ell} \alpha_i y_i \mathbf{x}_i \end{aligned} \tag{3}$$

Dual problem

Substituting eq.(3) in eq.(2) leads to the **dual problem**

Problem **P2**

Maximize $\quad -\frac{1}{2}\boldsymbol{\alpha}'\mathbf{A}\boldsymbol{\alpha} + \sum \alpha_i$

subject to $\quad \sum y_i \alpha_i = 0$
$\quad\quad\quad\quad\quad \alpha_i \geq 0, \quad i = 1, \ldots, \ell.$

where $\mathbf{A}$ is an $\ell \times \ell$ matrix $\mathbf{A} = (y_i y_j \mathbf{x}_i' \mathbf{x}_j : i, j \in \mathbb{N}_\ell)$.

Note that the complexity of this problem depend on ℓ, not on the number of input components n (same as ridge regression)

Kuhn-Tucker conditions and support vectors

If $\bar{\alpha}$ is a solution of the dual problem then the solution $(\bar{\mathbf{w}}, \bar{b})$ of the primal problem is given by

$$\bar{\mathbf{w}} = \sum_{i=1}^{\ell} \bar{\alpha}_i y_i \mathbf{x}_i,$$

Note that $\bar{\mathbf{w}}$ is a linear combination of only the $\mathbf{x}_i$ for which $\bar{\alpha}_i > 0$. These $\mathbf{x}_i$ are termed **support vectors** (SV).

Parameter $\bar{b}$ can be determined by looking at the Kuhn-Tucker conditions

$$\bar{\alpha}_i \left(y_i(\bar{\mathbf{w}}' \mathbf{x}_i + \bar{b}) - 1 \right) = 0.$$

Specifically if $\mathbf{x}_j$ is a SV we have that

$$\bar{b} = y_j - \bar{\mathbf{w}}' \mathbf{x}_j.$$

Some remarks

- The fact that that the optimal separating hyperplane is determined only by the support vectors is most remarkable. Usually, the support vectors are a small subset of the training data.

- All the information contained in the data set is summarized by the support vectors: The whole data set could be replaced by only these points and the **same** hyperplane would be found.

- A new point $\mathbf{x}$ is classified as $\operatorname{sgn}\left(\sum_{i=1}^{\ell} \bar{\alpha}_i \mathbf{x}_i' \mathbf{x} + \bar{b} \right)$

Linearly nonseparable case

If the data is not linearly separable (or one simply ignores whether this is the case) the previous analysis can be generalized by looking at the problem

Problem **P3**

Minimize $\frac{1}{2}\mathbf{w}'\mathbf{w} + C\sum_{i=1}^{\ell}\xi_i$

subject to $y_i(\mathbf{w}'\mathbf{x}_i + b) \geq 1 - \xi_i,$
$\xi_i \geq 0, \quad i = 1, \ldots, \ell$

The idea is to introduce the slack variables ξ_i to relax the separation constraints ($\xi_i > 0 \Rightarrow \mathbf{x}_i$ has margin less than 1)

New dual problem

A saddle point analysis (similar to that above) leads to the dual problem

Problem **P4**

Maximize $Q(\alpha) = -\frac{1}{2}\boldsymbol{\alpha}^{\top} A\boldsymbol{\alpha} + \sum \alpha_i$

subject to $\sum y_i\alpha_i = 0$
$0 \leq \alpha_i \leq C, \quad i = 1, \ldots, \ell.$

This is like problem **P2** except that now we have 'box constraints' on α_i. If the data is linearly separable, by choosing C large enough we obtain the OSH.

Nonseparable case (cont)

Again we have

$$\bar{\mathbf{w}} = \sum_{i=1}^{\ell} \bar{\alpha}_i y_i \mathbf{x}_i,$$

while $\bar{b}$ can be determined from $\bar{\boldsymbol{\alpha}}$, solution of the problem **P4**, and from the new Kuhn-Tucker conditions

$$\begin{aligned} \bar{\alpha}_i \left(y_i(\bar{\mathbf{w}}'\mathbf{x}_i + \bar{b}) - 1 + \bar{\xi}_i \right) &= 0 \quad (*) \\ (C - \bar{\alpha}_i)\bar{\xi}_i &= 0 \quad (**) \end{aligned}$$

Again, points for which $\bar{\alpha}_i > 0$ are termed **support vectors**.

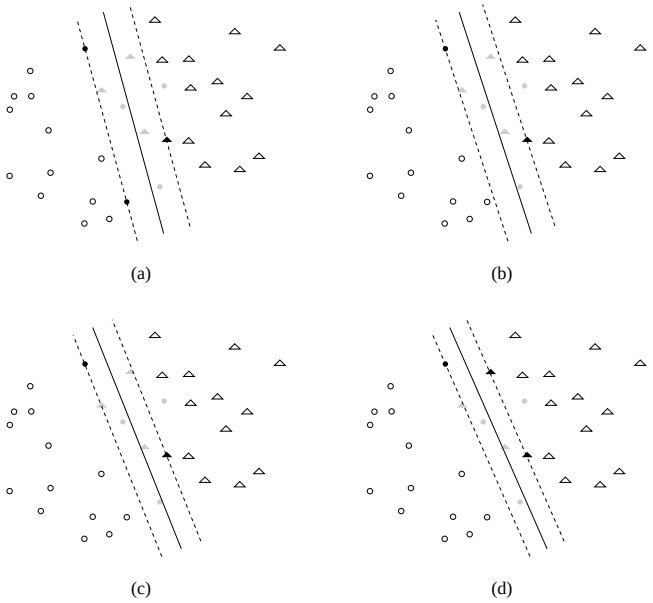
A closer look at the KKT conditions

Equation (*) tell us that if

- $y_i(\mathbf{w}'\mathbf{x}_i + b) > 1 \Rightarrow \bar{\alpha}_i = 0$ (not a SV)
- $y_i(\mathbf{w}'\mathbf{x}_i + b) = 1 \Rightarrow \bar{\alpha}_i \in [0, C]$ (if $\alpha_i > 0$ a SV 'on the margin')
- $y_i(\mathbf{w}'\mathbf{x}_i + b) < 1 \Rightarrow \bar{\alpha}_i = C$ (a SV with positive slack ξ_i)

Remark: Conversely, from eqs.(*),(**) if $\bar{\alpha}_i = 0$ then $y_i(\mathbf{w}'\mathbf{x}_i + b) \geq 1, \xi_i = 0$; if $\bar{\alpha}_i \in (0, C)$ then $y_i(\mathbf{w}'\mathbf{x}_i + b) = 1, \xi_i = 0$; if $\bar{\alpha}_i = C$ then $y_i(\mathbf{w}'\mathbf{x}_i + b) \leq 1, \xi_i \geq 0$. In practice, however, all inequalities are strict.

The role of the parameter C



Optimal separating hyperplane for four increasing values of C. Margin decreases with C while the training error increases.

The role of the parameter C (cont.)

The parameter C controls the trade-off between margin $\frac{1}{\|\mathbf{w}\|}$ and the training error $\sum_{i=1}^{\ell} \xi_i$.

The parameters $\bar{\alpha}_i$ (and, so, $\bar{\mathbf{w}}, \bar{b}$) are piecewise continuous functions of C (we skip the proof of this)

Support Vector Machines (SVMs)

The above analysis holds true if we work with a feature map $\phi : \mathcal{X} \to \mathcal{W}$. We simply replace $\mathbf{x}$ by $\phi(x)$ and $\mathbf{x}'\mathbf{t}$ by $\langle \phi(x), \phi(t) \rangle = K(x,t)$.

An SVM with kernel K is the function

$$f(x) = \sum_{i=1}^{\ell} y_i \alpha_i K(x_i, x) + b, \quad x \in \mathcal{X}$$

where the parameters α_i solve problem **P4** with $\mathbf{A} = (y_i y_j K(x_i, x_j) : i, j \in N_\ell)$ and b is obtained as discussed above

A new point $x \in \mathcal{X}$ is classified as $\text{sgn}(f(x))$

Regularization formulation

The SVM formulation above is equivalent to the problem

$$E_\lambda(\mathbf{w}, b) = \sum_{i=1}^{\ell} \max(1 - y_i(\langle \mathbf{w}, \phi(x_i) \rangle + b), 0) + \lambda \|\mathbf{w}\|^2$$

with $\lambda = \frac{1}{2C}$

In fact, we have

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \left\{ C \sum_{i=1}^{\ell} \xi_i + \frac{1}{2}\|\mathbf{w}\|^2 : y_i(\langle \mathbf{w}, \phi(x_i) \rangle + b) \geq 1 - \xi_i, \xi_i \geq 0 \right\} =$$

$$\min_{\mathbf{w},b} \left\{ \min_{\boldsymbol{\xi}} \left\{ C \sum_{i=1}^{\ell} \xi_i + \frac{1}{2}\|\mathbf{w}\|^2 : \xi_i \geq 1 - y_i(\langle \mathbf{w}, \phi(x_i) \rangle + b), \xi_i \geq 0 \right\} \right\} =$$

$$\min_{\mathbf{w},b} \left\{ C \max(1 - y_i(\langle \mathbf{w}, \phi(x_i) \rangle + b), 0) + \frac{1}{2}\|\mathbf{w}\|^2 \right\} = C E_{\frac{1}{2C}}(\mathbf{w}, b)$$

Solution methods

The above optimization problems are Quadratic Programming (QP) problems. Several methods (eg, interior point's) from convex optimization exist for solving QP problems...

If we work with a non-linear kernel, the number of underline features, N, is typically much larger (or infinite) than the number of examples. Thus, we need to solve the dual problem.

In some datamining applications, instead, $\ell \gg N$ and it is more efficient to solve the primal problem.

Decomposition of the dual problem

For large datasets (say $\ell > 10^5$) it is practically impossible to solve the dual problem with standard optimization techniques (matrix $\mathbf{A}$ is dense!)

A typical approach is to iteratively optimize wrt. an 'active set' $\mathcal{A}$ of dual variables. Set $\boldsymbol{\alpha} = 0$, choose $q \leq \ell$ and a subet $\mathcal{A}$ of variables. We repeat till convergence the steps

- Optimize $Q(\alpha)$ wrt. the variables in $\mathcal{A}$
- Remove one variables from $\mathcal{A}$ which satisfy the KKT conditions and add one variable, if any, which 'violate' the KKT conditions. If not such variable exists stop

One can show that after each iteration Q increases

Removing b

If we are looking for a hyperplane which passes through the origin we do not need to optimize wrt. b (set $b \equiv 0$).

In this case we have the simplified dual problem

Problem **P4'**

Maximize $\quad -\frac{1}{2}\sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j K(x_i, x_j) + \sum \alpha_i$

subject to $\quad 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, \ell.$

(the constraint $\sum_{i=1}^{\ell} \alpha_i = 0$ disappears)

b or not b?

In general, unless we are sure that a threshold b does not influence the classification, we should also learn b

A simple way is to introduce a small regularization on b, i.e. to set $K'(x,t) = K(x,t) + \lambda_0$ and solve problem **P4'** using this new kernel

In the limit $\lambda_0 \to \infty$, the regularization on b is removed and we get the additional constraint $\sum_{i=1}^{\ell} \alpha_i y_i = 0$ (see why?), so, we are back to problem **P4**

Conditionally positive semidefinite kernels

Note that in order to have a solution to problem $\mathbf{P4}$ the kernel only needs to be *conditionally positive semidefinite* (cpsd)

Definition: A kernel K is cpsd if for every $\ell > 0$ and $x_1, \ldots, x_m \in \mathbb{R}^n$, $\sum_{i,j=1}^{\ell} c_i c_j K(x_i, x_j) \geq 0$ if $\sum_{i=1}^{\ell} c_i = 0$

Example: The kernel $K(\mathbf{x}, \mathbf{t}) = -\|\mathbf{x} - \mathbf{t}\|^2$ is conditionally positive semidefinite but **not** positive semidefinite.

In fact, for every $\mathbf{x}_i \in \mathbb{R}^n, c_i \in \mathbb{R}, i \in \mathbb{N}_\ell$ such that $\sum_i c_i = 0$ we have that

$$\begin{aligned} -\sum_{i,j} c_i c_j \|\mathbf{x}_i - \mathbf{x}_j\|^2 &= -2\sum_i c_i \sum_j c_i \|\mathbf{x}_j\|^2 + 2\sum_{i,j} c_i c_j \mathbf{x}_i' \mathbf{x}_j \\ &= 2\sum_{i,j} c_i c_j \mathbf{x}_i' \mathbf{x}_j = 2\left\|\sum_i c_i \mathbf{x}_i\right\|^2 \geq 0 \end{aligned}$$

On the other hand, if all c_i are non-negative we get '$\leq$'

SVM regression

SVM's can be developed for regression as well. Here we choose the loss $= |y - f(\mathbf{x})|_\epsilon = \max(|y - f(\mathbf{x})| - \epsilon, 0)$

Minimize	$\frac{1}{2}\mathbf{w}'\mathbf{w} + C\sum_{i=1}^{\ell}(\xi_i + \xi_i^*)$
subject to	$\mathbf{w}'\mathbf{x}_i + b - y_i \leq \epsilon + \xi_i,$
	$y_i - \mathbf{w}'\mathbf{x}_i - b \leq \epsilon + \xi_i^*,$
	$\xi_i, \xi_i^* \geq 0, \quad i = 1, \ldots, \ell$

The common feature of the SVM loss functions (both for classification and regression) is that they *scale sensitive* loss functions: errors below a certain resolution do not count.

Bibliography

Lectures available at:
http://www.cs.ucl.ac.uk/staff/M.Pontil/courses/index-ATML05.htm

- Kernel PCA: See Shawe-Taylor and Cristianini's book, Chapter 5.1, 6.1 and 6.2. Original paper: B. Schölkopf A.J. Smola and K.R. Müller Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation 10(5), pp. 1299-1319, 1998.

- SVM: Originally proposed by B. Boser, I. Guyon, and V. Vapnik. In 5-th Annual Workshop on Computational Learning Theory, pages 144–152, ACM. 1992 and extended to non-separable case by C. Cortes and V. Vapnik. Support-Vector Networks. Machine Learning, 20, 1995.

  Lecture are based on: M. Pontil and A. Verri. Properties of support vector machines. Neural Computation, 10, pp. 955–974, 1998.

  See aslo: Shawe-Taylor and Cristianini's book, Chapter 7.2