18. Dual SVMs

- SVMs
- Kernel SVMs

Support vector machines

Hard Margin SVM duality

$$\begin{array}{ll} \underset{\theta}{\text{minimize}} & \frac{1}{2}\|\theta\|_2^2 \\ \text{subject to} & y_i x_i^T \theta \geq 1,\ i = 1, ..., m \end{array}$$

Lagrangian

Dual problem

Hard Margin SVM duality

$$\text{(P)} \quad \min_{\theta} \quad \tfrac{1}{2}\|\theta\|_2^2 \qquad\qquad \text{(D)} \quad \max_{u} \quad -\tfrac{1}{2}\sum_{i=1}^{m}(y_i x_i^T u)^2 + u^T \mathbf{1}$$
$$\text{st} \quad y_i x_i^T \theta \geq 1,\ i = 1, ..., m \qquad\qquad \text{st} \quad u \geq 0$$

Primal recovery

Complementary slackness

Hard Margin SVM duality summary

$$\text{(P)} \quad \min_{\theta} \ \tfrac{1}{2}\|\theta\|_2^2 \quad \text{st} \ \ Z\theta \geq \mathbf{1} \qquad\qquad \text{(D)} \quad \max_{u} \ -\tfrac{1}{2}\|Z^T u\|_2^2 + u^T\mathbf{1} \quad \text{st} \ \ u \geq 0$$

- $Z = \begin{bmatrix} x_1 y_1 & \cdots & x_m y_m \end{bmatrix}^T$
- Primal recovery from dual optimal: $\theta^* = Z^T u^*$
- Complementary slackness

$$u_i > 0 \text{ or } y_i z_i > 1 \text{ but never both.}$$

- Dual recovery of support vectors:

$$\{i : u_i > 0\} \supseteq \{i : y_i x_i = 1\} = \text{ support vectors}$$

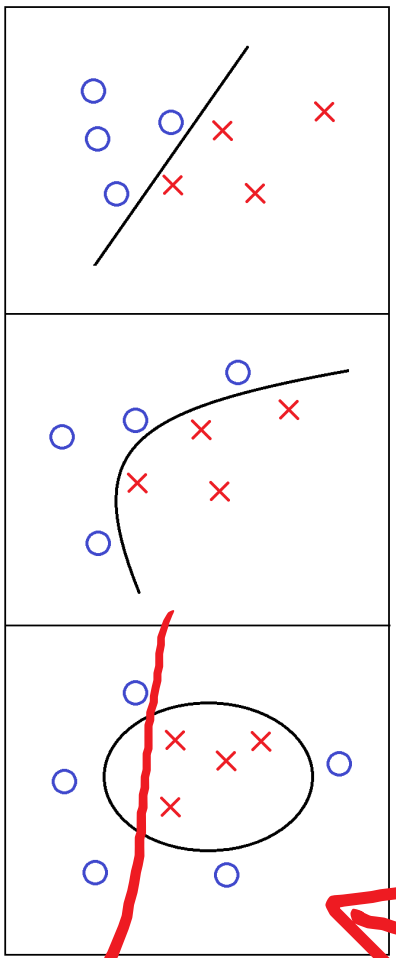
Hard margin SVM computation

$$\text{(P)} \quad \begin{array}{ll} \min\limits_{\theta} & \frac{1}{2}\|\theta\|_2^2 \\ \text{st} & Z\theta \geq \mathbf{1} \end{array} \qquad \text{(D)} \quad \begin{array}{ll} \max\limits_{u} & -\frac{1}{2}\|Z^T u\|_2^2 + u^T \mathbf{1} \\ \text{st} & u \geq 0 \end{array}$$

- $Z = \begin{bmatrix} x_1 y_1 & \cdots & x_m y_m \end{bmatrix}^T$
- Gradient of primal vs gradient of dual
  - Slightly more computation, requires one pass through data
- Projection on feasibility sets
  - Very difficult for primal, very simple for dual
- In general, solving dual SVM is preferred over solving primal SVM
- Suggested exercise: repeat all of this for soft margin SVM!

Kernel SVM

Separating hyperplanes



- Goal: find θ where $y_i x_i^T \theta \geq 0$ for all training samples i
- Q: What if no such θ exists?
- A1: soft margins! But is that fudging too much?
- A2: nonlinear separators

$$\theta \in \mathbb{R}^s : y_i \phi(x_i)^T \theta \geq 0 \; \forall i, \quad \phi : \mathbb{R}^d \to \mathbb{R}^q$$

Here usually $q \gg d$

representation lifting function

Example: Polynomial function basis

- $x = (x_1, x_2, x_3) \in \mathbb{R}^3$
- Polynomial basis of order 2

$$\phi(x) = (x_1^2, x_2^2, x_3^2, x_1x_2, x_2x_3, x_1x_3) \in \mathbb{R}^5$$

- Polynomial basis of order 3

$$\phi(x) = (x_1^3, x_2^3, x_3^3, x_1^2x_2, x_2^2x_3, x_3^2x_1, x_1^2x_3, x_2^2x_1, x_3^2x_2) \in \mathbb{R}^9$$

- If $d = 100$, $q = 6$, $\phi(x) \in \mathbb{R}^{1.6 \text{ billion}}$.

Wait, but do we really need that much?

$$\min_{\substack{\theta \in \mathbb{R}^q, \\ s \in \mathbb{R}^m}} \quad \frac{1}{2}\|\theta\|_2^2 + \lambda s^T \mathbf{1}$$
$$\text{st} \quad y_i \phi(x_i)^T \theta + s_i \geq 1$$
$$s \geq 0$$

$$\max_{u \in \mathbb{R}^m} \quad u^T \mathbf{1} - \frac{1}{2} \sum_{i,j} u_i u_j y_i y_j \phi(x_i)^T \phi(x_j)$$
$$\text{st} \quad 0 \leq u \leq \lambda$$

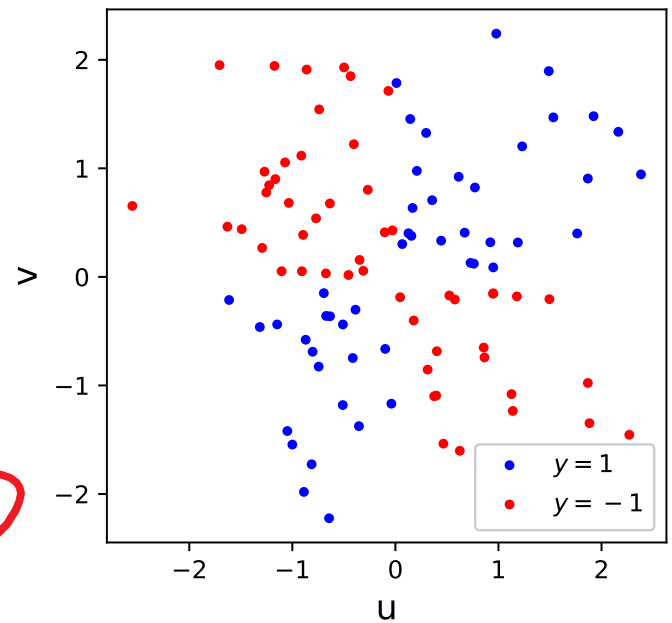
- Primal variable $\theta \in \mathbb{R}^{1.6 \text{ billion}}$, can't even store it!
- Dual variable $u \in \mathbb{R}^m$, size doesn't change based on function basis
- Kernel trick: construct $K \in \mathbb{R}^{m,m}$ where

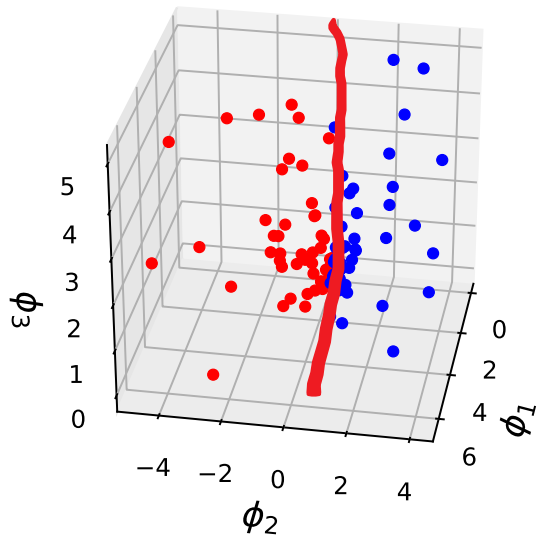
$$K_{ij} = \phi(x_i)^T \phi(x_j)$$

and never form anything using storage $O(q)$!

Lifted spaces example: 3-D polynomials

$$
\begin{aligned}
K(u,v) &= (u^T v)^2 \\
&= (u[1]^2 v[1]^2 + 2u[1]v[1]u[2]v[2] + u[2]^2 v[2]^2) \\
&= \underbrace{(u[1]^2, \sqrt{2}u[1]u[2], u[2]^2)^T}_{\phi(u)} \underbrace{(v[1]^2, \sqrt{2}v[1]v[2], v[2]^2)}_{\phi(v)}
\end{aligned}
$$





Kernels

- A Kernel function $\mathcal{K} : \mathbb{R}^{d \times d} \to \mathbb{R}$ measures similarity between two vectors
- **Examples**
  - Linear kernel: $\mathcal{K}(u, v) = u^T v$
  - Polynomial kernel of degree q: $\mathcal{K}(u, v) = (u^T v)^q$
  - Polynomial kernel of degree up to q: $\mathcal{K}(u, v) = (u^T v + 1)^q$
  - Radial basis function $\mathcal{K}(u, v) := \exp\left\{-\frac{\|u-v\|^2}{\gamma}\right\}$
- The Kernel matrix $K \in \mathbb{R}^{m \times m}$ evaluates $\mathcal{K}$ over training samples

$$K_{i,j} = \mathcal{K}(x_i, x_j), \qquad i, j = 1, ..., m$$

- If K is PSD, then there exists ϕ where $\mathcal{K}(u, v) = \phi(u)^T \phi(v)$

Kernel SVM

- Primal and dual SVM

$$\begin{array}{ll} \min\limits_{\substack{\theta\in\mathbb{R}^s,\\ s\in\mathbb{R}^m}} & \frac{1}{2}\|\theta\|_2^2 + \lambda s^T\mathbf{1} \\ \text{st} & y_i\phi(x_i)^T\theta \geq 1 - s_i \\ & s \geq 0 \end{array}$$

$$\begin{array}{ll} \max\limits_{u\in\mathbb{R}^m} & u^T\mathbf{1} - \sum\limits_{i,j=1}^{m} u_i u_j y_i y_j K_{ij} \\ \text{st} & 0 \leq u \leq \lambda \end{array}$$

- Recovery of primal solution given dual: as before,

$$\theta = \sum_{i=1}^{m} u_i y_i \phi(x_i)$$ (Never actually formed)

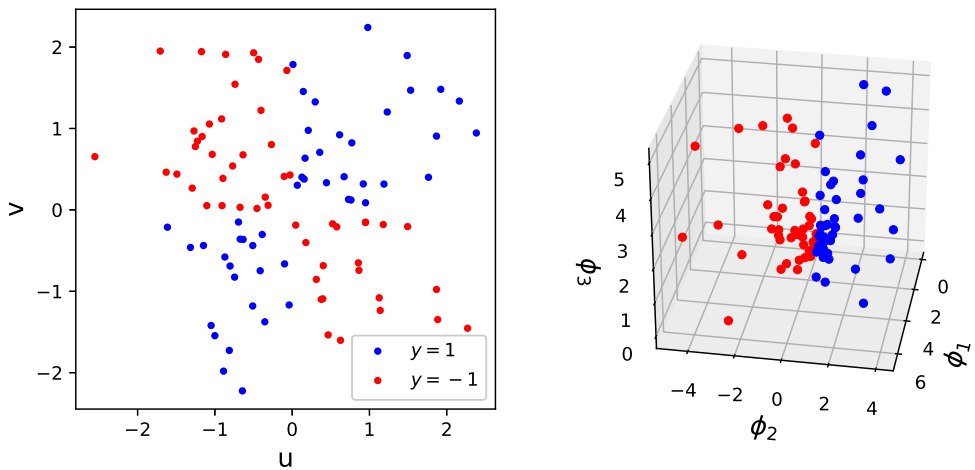
- Classification: given new x,

$$y = \mathbf{sign}(\theta^T\phi(x)) = \mathbf{sign}\left(\sum_{i=1}^{m} \underbrace{y_i u_i}_{\substack{\text{sample}\\ \text{weight}}} \underbrace{\phi(x_i)^T\phi(x)}_{=\mathcal{K}(x_i,x)}\right)$$

Example: Polynomial kernel

$$\mathcal{K}(x_1, x_2) = (x_1^T x_2 + c)^d$$

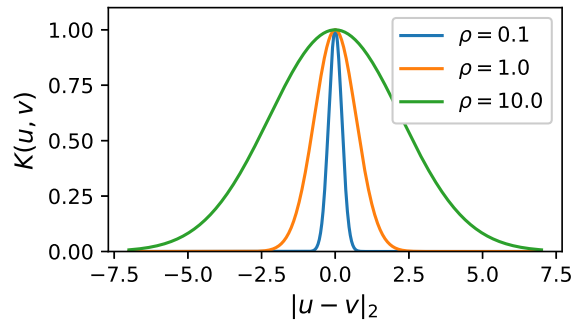
Separation for $d = 2$



- It can be shown that $\mathcal{K}(x_1, x_2) = \phi(x_1)^T \phi(x_2)$
  - $\phi(x_1) \in \mathbb{R}^q$, $q =$ number of terms in expansion
- Therefore, PSD kernel.

Example: Radial basis function

$$\mathcal{K}(x_1, x_2) = \exp\left(-\frac{\|x_1 - x_2\|_2^2}{\rho}\right)$$



- Hyper parameter ρ: hard to tune, can start with

$$\rho = d \cdot \text{sample var}(x_i) = \text{mean}_{x} \|x - \text{mean}_{x'} x'\|^2 \text{ or } \text{mean}_{x,x'} \|x - x'\|^2$$

- Overfitting: If $\rho \to 0$, $K \to cI$ and training error $= 0$ (turns into 1-NN)

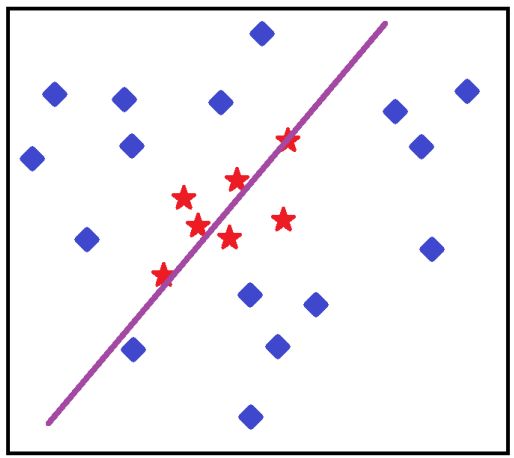
$$\text{pred}(x_k) = \textbf{sign}\left(\sum_{i=1}^{m} y_i u_i \underbrace{\mathcal{K}(x_i, x_k)}_{0 \text{ if } i \neq k}\right) = \textbf{sign}(cy_k) = y_k$$

- PSD? Yes, not easy to prove.

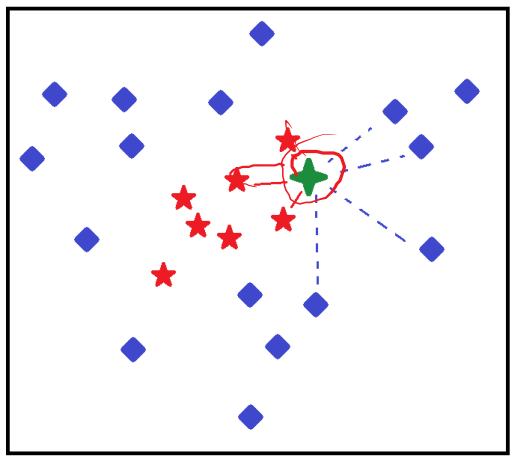
Kernel SVM: Another view of how it works (RBF)

$$\text{new } x, \qquad y = \mathbf{sign}\left(\sum_{i=1}^{m} y_i u_i \mathcal{K}(x_i, x)\right)$$

- sample weight: how much ith training sample influences decision boundary
- Kernel evaluation filters label to be determined by x_i closest to x



no linear classifier



classify new point based on
proximity to classified points

Distance-based kernels

$$\mathcal{K}(u, v) = \exp\left(-\frac{\mathbf{dist}(u, v)}{\rho}\right),$$

- Earth mover's distance to measure the distance between distributions

$$\mathbf{dist}_{\mathsf{EMD}}(U, V) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{m} p_U(u_i)p_V(v_j)d(u_i, v_j)}{\sum_{i=1}^{m}\sum_{j=1}^{m} p_U(u_i)p_V(v_j)}$$

- Chi-2 distance to measure distance between histograms

$$\mathbf{dist}_{\mathsf{hist}}(u, v) = \sum_{i=1}^{d} \frac{(u_i - v_i)^2}{u_i + v_i}$$

- Commonly used to compare histogram samples (e.g. bag of features)
- Cross validation to tune ρ, start with $\rho = \underset{u,v}{\text{mean}}\mathbf{dist}(u, v)$

Jianguo Zhang, Marcin Marszalek, Svetlana Lazebnik, Cordelia Schmid 2007

Example: Tanh kernel

$$\mathcal{K}(x_1, x_2) = \tanh(\eta x_1^T x_2 + \mu)$$

- Cross validation to tune hyperparameters η, μ
- PSD? no! Take $\eta = 0$, $\mu = 0$, $x_1 = [-\epsilon]$, $x_2 = [100/\epsilon]$. Then

$$K = \tanh\left(\begin{bmatrix} \epsilon^2 & -100 \\ -100 & 100^2/\epsilon^2 \end{bmatrix}\right) \approx \begin{bmatrix} 0 & -1 \\ -1 & 1 \end{bmatrix}$$

- Then dual problem is not convex! May not reach global minimum
- What's the upside? Doesn't overfit so easily compared to RBF

Example: Fisher kernel

- Recall that $\log(P(X|\theta))$ is the log-likelihood of observing X given θ
- Often, something we maximize
- Take

$$G(X) = \nabla_\theta \log(P(X|\theta)), \qquad H(X) = \nabla^2_\theta \log(P(X|\theta))$$
$$\mathcal{I} = \mathbb{E}_X[-H], \qquad \mathcal{K}(u, v) = G(u)^T \mathcal{I}^{-1} G(v)$$

- $\mathcal{I}$ is the Fisher information matrix,
- $\mathcal{K}(u, v)$ is the Fisher Kernel

Interpretations

- Large $\mathcal{I}$ hints at "peakyness" if distribution $\rightarrow$ more certainty of "correctness"
- For multivariate normal distribution, relation to conditioning by inverse variance

In practice

Example: Fisher kernel

$$\mathcal{I} = \mathbb{E}_X[-H], \qquad \mathcal{K}(u, v) = G(u)^T \mathcal{I}^{-1} G(v)$$

Interpretations

- Large $\mathcal{I}$ hints at "peakyness" if distribution $\rightarrow$ more certainty of "correctness"
- For multivariate normal distribution, relation to conditioning by inverse variance

In practice

- Fisher information matrix estimated through training samples
- If "quick estimates", $\mathcal{I}_{\text{sample}} = -\frac{1}{m} \sum_{i=1}^{m} H(x_i)$ may not be invertible

Kernel matrix approximations

- Dual SVM has kernel matrix $K \in \mathbb{R}^{m,m}$ where $m = \#$ training samples
- When m is very large, forming and using K is computationally difficult
- Effective Kernel approximations are key for computational scalability

Approaches

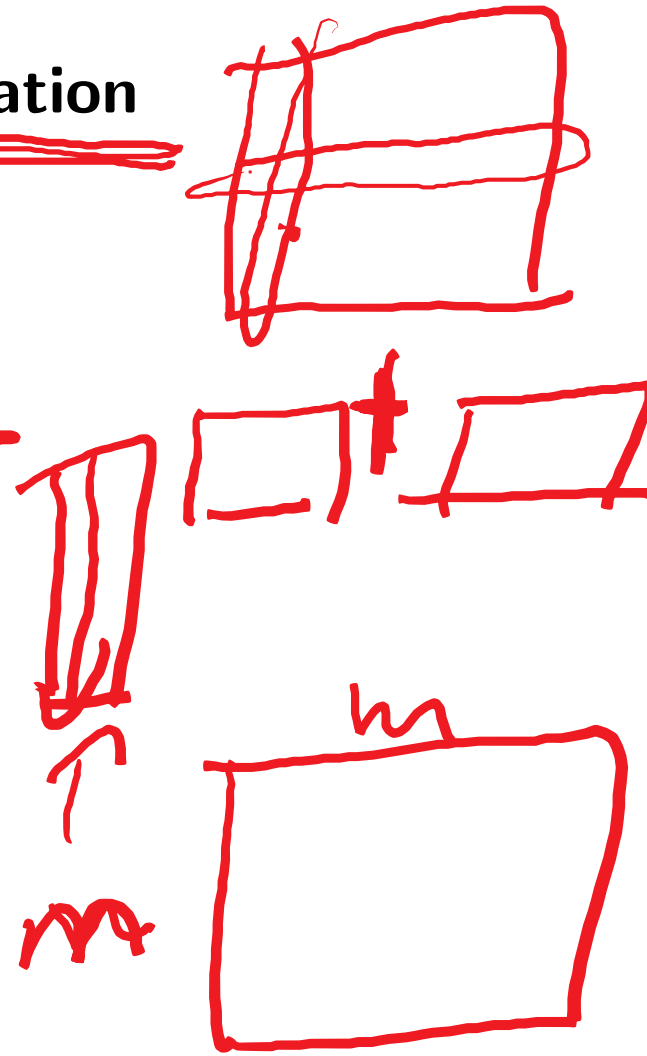
- Sparsification
- Sketching: Pick a random skinny matrix $U \in \mathbb{R}^{m \times q}$. Then

$$z^T K z \approx z^T U \underbrace{U^T K U}_{q \times q} U^T z$$

- Low rank approximation: find V where

$$K \approx V V^T, \qquad z^T K z = \|V^T z\|_F^2$$

Nystrom low-rank approximation



Procedure:

- Pick $\mathcal{S}$ indices from $\{1, ..., m\}$ randomly, $|\mathcal{S}| \ll m$
- Form $C = K_{:,\mathcal{S}}$, $W = K_{\mathcal{S},\mathcal{S}}$
- $\hat{K} = CW^{\dagger}C^{T}$, storage $O(s^2 + 2sm) \ll O(m^2)$

Picking indices active area of research

- One way: leverage scores $\Pr(i \in \mathcal{S}) \propto K_{ii} + \rho$
- In practice, uniform sampling often works well

Drineas, Mahoney '05, Cohen, Musco, Musco '17; Kumar, Mohri, Talwalker '09 ...