Multiclass Classification

Pontus Giselsson

Learning goals

- Know multiclass logistic regression and SVM and their purpose
- Understand the logistic regression cost function
- Understand dual multiclass SVM formulations
- Be able to predict class beloning from dual SVM solutions

What is multiclass classification?

- We have previously seen binary classification
  - Two classes (cats and dogs)
  - Each sample belongs to one class (has one label)
- Multiclass classification
  - K classes with $K \geq 3$ (cats, dogs, rabbits, horses)
  - Each sample belongs to one class (has one label)
  - (Not to confuse with multilabel classification with ≥ 2 labels)

Multiclass classification from binary classification

- 1-vs-1: Train binary classifiers between all classes
  - Example:
    - cat-vs-dog,
    - cat-vs-rabbit
    - cat-vs-horse
    - dog-vs-rabbit
    - dog-vs-horse
    - rabbit-vs-horse
  - Prediction: Pick, e.g., the one that wins the most classifications
  - Number of classifiers: $\frac{K(K-1)}{2}$
- 1-vs-all: Train each class against the rest
  - Example
    - cat-vs-(dog,rabbit,horse)
    - dog-vs-(cat,rabbit,horse)
    - rabbit-vs-(cat,dog,horse)
    - horse-vs-(cat,dog,rabbit)
  - Prediction: Pick, e.g., the one that wins with highest margin
  - Number of classifiers: K
  - Always skewed number of samples in the two classes

Multiclass classification

- Labeled training data $\{(x_i, y_i)\}_{i=1}^N$
- $K \geq 3$ classes and class labels (responses) $y \in \{1, \ldots, K\}$
- Training problem, find model parameters θ that solve

$$\underset{\theta}{\text{minimize}} \sum_{i=1}^{N} L(m(x_i; \theta), y_i)$$

- Prediction: Based on model output
- We will cover:
  - Multiclass logistic regression
  - Two multiclass SVM versions

Multiclass Logistic Regression

Multiclass logistic regression

- K classes in $\{1, \ldots, K\}$ and data/labels $(x, y) \in \mathcal{X} \times \mathcal{Y}$
- Labels: $y \in \mathcal{Y} = \{e_1, \ldots, e_K\}$ where $\{e_j\}$ coordinate basis
  - Example, $K = 5$ class 2: $y = e_2 = [0, 1, 0, 0, 0]^T$
- Objective: Find θ such that $\sigma(m(x; \theta)) - y \approx 0$ where
  - $\sigma : \mathbb{R}^K \to \mathrm{conv}(\mathcal{Y})$ is a fixed-function
  - $m : \mathcal{X} \to \mathbb{R}^K$ has K regression models, one per class:

$$m(x; \theta) = \begin{bmatrix} m_1(x; \theta_1) \\ \vdots \\ m_K(x; \theta_K) \end{bmatrix} = \begin{bmatrix} w_1^T x + b_1 \\ \vdots \\ w_K^T x + b_K \end{bmatrix}$$

- Want to find θ and select σ such that:
  - $m_j(x; \theta_j) \gg 0$, if label $y = e_j$ and $m_j(x; \theta_j) \ll 0$ if $y \neq e_j$
  - $\sigma_j(m(x; \theta)) \approx 1$, if label $y = e_j$ and $\sigma_j(m(x; \theta)) \approx 0$ if $y \neq e_j$

Multiclass logistic regression – σ

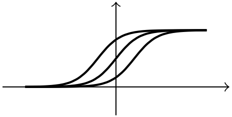
- For $\mathcal{Y} = \{e_1, \ldots, e_K\}$, $\mathrm{conv}(\mathcal{Y}) = \Delta_K$, where

$$\Delta_K = \{v \in \mathbb{R}^K : v_i \geq 0 \text{ and } \mathbf{1}^T v = 1\}$$

  is probability simplex on $\mathbb{R}^K$, we want $\sigma : \mathbb{R}^K \to \Delta_K$
- The *softmax* function $\sigma : \mathbb{R}^K \to \Delta_K$ with $u = (u_1, \ldots, u_K)$

$$\sigma(u) = \frac{1}{\sum_{j=1}^K e^{u_j}} \begin{bmatrix} e^{u_1} \\ \vdots \\ e^{u_K} \end{bmatrix}$$

  satisfies this and is gradient of convex function
  - Graph for $\sigma_1(u_1)$ for some fixed $u_2, \ldots, u_K$



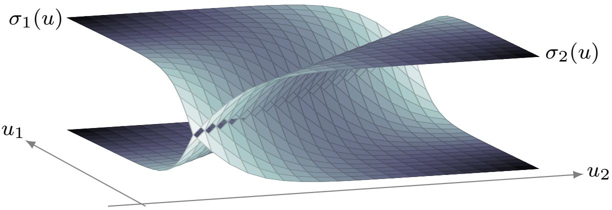
  - Model $m_j(x; \theta_j) \to \infty$, other outputs fixed $\Rightarrow \sigma(m(x; \theta)) \to e_j$

Two-class logistic regression – σ

- Let $u = (u_1, u_2)$ and use $\sigma : \mathbb{R}^2 \to \Delta^2$:

$$\sigma(u) = \frac{1}{e^{u_1} + e^{u_2}} \begin{bmatrix} e^{u_1} \\ e^{u_2} \end{bmatrix}$$

  that satisfies
  - $m_1(x;\theta_1) \to \infty$ and $m_2(x;\theta_2)$ fixed $\Rightarrow \sigma(m(x;\theta)) \to (1,0)$
  - $m_2(x;\theta_2) \to \infty$ and $m_1(x;\theta_1)$ fixed $\Rightarrow \sigma(m(x;\theta)) \to (0,1)$
- Will see this two-class version can give standard logistic regression



Multiclass logistic regression – Loss function

- Primitive function of softmax

$$\left(\int \sigma(v)dv\right)(u) = \log\left(\sum_{j=1}^{K} e^{u_j}\right)$$

- Same cost construction as for logistic regression,

$$\begin{aligned} L(u,y) &= \left(\int \sigma(v)dv\right)(u) - u^T y \\ &= \log\left(\sum_{j=1}^{K} e^{u_j}\right) - u^T y \\ &= \log\left(\sum_{j=1}^{K} e^{u_j}\right) - \sum_{j=1}^{K} \mathbb{I}(y_j = 1)u_j \end{aligned}$$

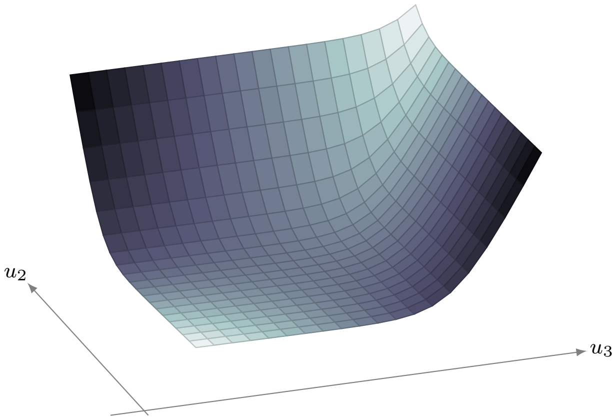
with last step since $y \in \{e_1, \ldots, e_K\}$

Multiclass logistic loss function – Example

- Multiclass logistic loss for $K = 3$, $u_1 = 1$, $y = e_1$

$$L((1, u_2, u_3), 1) = \log(e^1 + e^{u_2} + e^{u_3}) - 1$$

- Increasing model outputs u_2 or u_3 gives higher cost



Multiclass logistic regression – Training problem

- Affine data model $m(x;\theta) = w^T x + b$ with

$$w = [w_1, \ldots, w_K] \in \mathbb{R}^{n\times K}, b = [b_1, \ldots, b_K]^T \in \mathbb{R}^K$$

- One data model per class
- Training problem:

$$\begin{aligned}\underset{\theta}{\text{minimize}} \sum_{i=1}^{N} L(m(x_i;\theta), y_i) \\ = \sum_{i=1}^{N} \log\left(\sum_{j=1}^{K} e^{w_j^T x_i + b_j}\right) - \sum_{j=1}^{K} \mathbb{I}(y_j = 1)(w_j^T x_i + b_j)\end{aligned}$$

- Problem is convex since affine model is used

Multiclass logistic regression – Prediction

- Assume model is trained and want to predict label for new data x
- $\sigma(m(x;\theta))$ outputs probability of class belonging for all K classes
- The jth output, $\sigma_j(m(x;\theta_j))$, is probablity for class j
- Predict label of x based on highest probability

Reduces to standard logistic regression

- Consider two-class version and let
  - $\Delta u = u_1 - u_2$, $\Delta w = w_1 - w_2$, and $\Delta b = b_1 - b_2$
  - $\Delta u = m_{\text{bin}}(x;\theta) = m_1(x;\theta_1) - m_2(x;\theta_2) = \Delta w^T x + \Delta b$
  - $y_{\text{bin}} = 1$ if $y = (1,0)$ and $y_{\text{bin}} = 0$ if $y = (0,1)$
  - $\sigma_{\text{bin}}(\Delta u) = \frac{1}{1+e^{-\Delta u}}$
- Loss L is equivalent to nominal, but with different variables

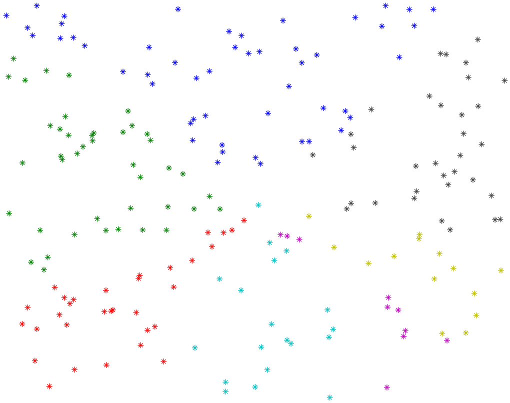
$$
\begin{aligned}
L(u,y) &= \log(e^{u_1} + e^{u_2}) - y_1 u_1 - y_2 u_2 \\
&= \log\left(1 + e^{u_1 - u_2}\right) + \log(e^{u_2}) - y_1 u_1 - y_2 u_2 \\
&= \log\left(1 + e^{\Delta u}\right) - y_1 u_1 - (y_2 - 1) u_2 \\
&= \log\left(1 + e^{\Delta u}\right) - y_{\text{bin}} \Delta u
\end{aligned}
$$

- σ is equivalent to nominal, but with different input

$$
\begin{aligned}
\sigma(u) &= \frac{1}{e^{u_1} + e^{u_2}} \begin{bmatrix} e^{u_1} \\ e^{u_2} \end{bmatrix} = \begin{bmatrix} 1/(1 + e^{u_2 - u_1}) \\ 1/(1 + e^{u_1 - u_2}) \end{bmatrix} = \begin{bmatrix} 1/(1 + e^{-\Delta u}) \\ 1/(1 + e^{\Delta u}) \end{bmatrix} \\
&= \begin{bmatrix} \sigma_{\text{bin}}(\Delta u) \\ 1 - \sigma_{\text{bin}}(\Delta u) \end{bmatrix}
\end{aligned}
$$

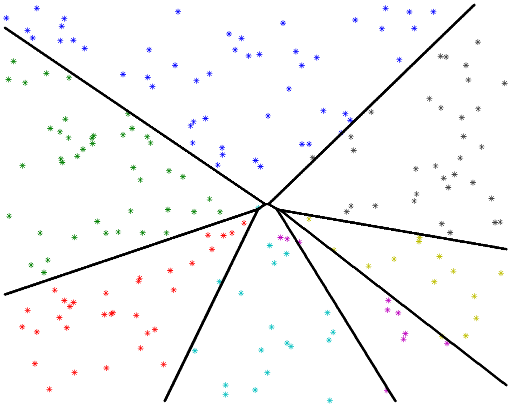
Multiclass logistic regression – Example

- Problem with 7 classes



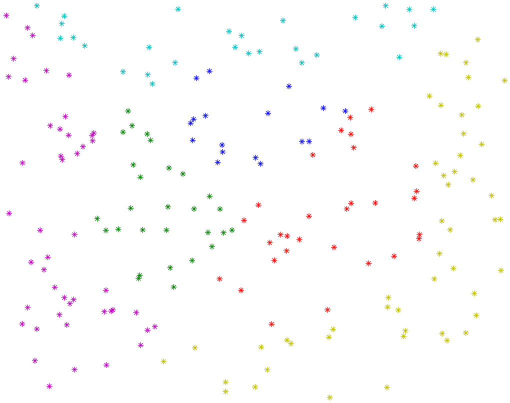
Multiclass logistic regression – Example

- Problem with 7 classes and affine multiclass model



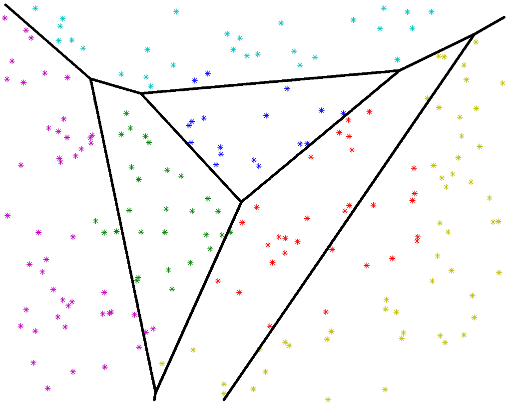
Multiclass logistic regression – Example

- Same data, new labels in 6 classes



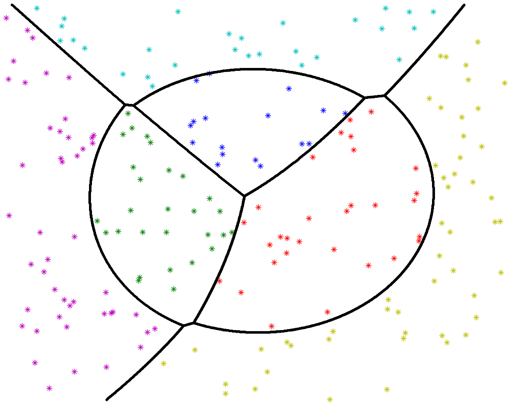
Multiclass logistic regression – Example

- Same data, new labels in 6 classes, affine model



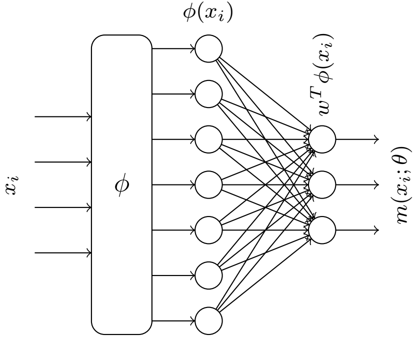
Multiclass logistic regression – Example

- Same data, new labels in 6 classes, quadratic model



Features

- Used quadratic features in last example
- Same procedure as before:
  - replace data vector x_i with feature vector $\phi(x_i)$
  - run classification method with feature vectors as inputs



Regularization

- Tikhonov regularization to avoid overfitting
- Penalize all w_i vectors (not bias terms b_i):

$$\underset{\theta}{\text{minimize}} \sum_{i=1}^{N} \left(\log \left(\sum_{j=1}^{K} e^{w_j^T x_i + b_j} \right) - y^T m(x_i; \theta) \right) + \frac{\lambda}{2} \sum_{j=1}^{K} \|w_j\|_2^2$$

Multiclass SVM

Deriving multiclass SVM

- Rewrite binary SVM with two models instead of one
- Generalize this model in two different ways

Rewrite binary SVM

- Introduce one model model per class label ($y \in \{1, 2\}$):

$$m_1(x; \theta_1) = w_1^T x + b_1 \qquad m_2(x; \theta_2) = w_2^T x + b_2$$

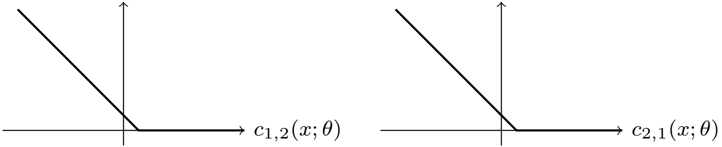
- We want class i to satisfy $m_i(x; \theta) > 0$ for $i = 1, 2$
- Define *confidence* for each class in relation to the other:

$$c_{1,2}(x; \theta) = m_1(x; \theta_1) - m_2(x; \theta_2)$$
$$c_{2,1}(x; \theta) = m_2(x; \theta_2) - m_1(x; \theta_1)$$

  - $c_{1,2}(x_i; \theta) \gg 0$ confident that $y_i = 1$
  - $c_{2,1}(x_i; \theta) \gg 0$ condident that $y_i = 2$
  - Note that $c_{1,2}(x_i; \theta) = -c_{2,1}(x_i; \theta)$

SVM loss

- Penalize confidence for the two classes using hinge loss



cost if $y = 1$ cost if $y = 2$

- Find parameters θ to have high confidence (low cost) for $y = i$
- $c_{1,2} = -c_{2,1}$: high confidence in one gives low confidence in other
- Let $y_i^c = \{1, 2\} \backslash y_i$ be complement and define training problem:

$$\underset{\theta}{\text{minimize}} \sum_{i=1}^{N} \max(0, 1 - c_{y_i, y_i^c}(x;\theta))$$

- Convex: sum of convex functions composed with affine mappings

Equivalent to standard formulation

- Training problem can be written as

$$\begin{aligned}\underset{\theta}{\text{minimize}} \sum_{i=1}^{N} \max(0, 1 - m_{y_i}(x_i; \theta_{y_i}) + m_{y_i^c}(x_i; \theta_{y_i^c})) \\ = \sum_{i=1}^{N} \max(0, 1 - (w_{y_i}^T x_i + b_{y_i}) + (w_{y_i^c}^T x_i + b_{y_i^c}))\end{aligned}$$

- Change of variables $\theta = \theta_{y_2} - \theta_{y_1}$ gives equivalent problem

$$\begin{aligned}\underset{\theta}{\text{minimize}} \sum_{i=1}^{N} \max(0, 1 - 2(y_i - 1.5)(w^T x_i + b)) \\ = \sum_{i=1}^{N} \max(0, 1 - 2(y_i - 1.5) m(x; \theta))\end{aligned}$$

i.e., SVM hinge loss since labels $2(y_i - 1.5)\{1, 2\} = \{-1, 1\}$

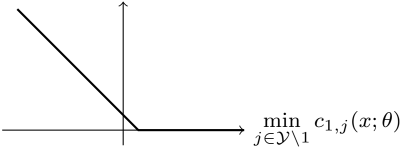
Multiclass SVM

- In binary SVM, confidence of label w.r.t. to complement
- In multiclass SVM, complement is more than one class
- Will compare to complement in two different ways

Multiclass SVM – Max version

- Multiclass SVM labels $y \in \mathcal{Y} = \{1, \ldots, K\}$
- Max version: Hinge on smallest confidence w.r.t. other classes

$$\max(0, 1 - \min_{j \in \mathcal{Y} \setminus y} c_{y,j}(x; \theta))$$



cost if $y = 1$

- Loss can be written as

$$\begin{aligned}
\max(0, 1 - \min_{j \in \mathcal{Y} \setminus y} c_{y,j}(x; \theta)) &= \max_{j \in \mathcal{Y} \setminus y} (0, 1 - c_{y,j}(x_i; \theta)) \\
&= \max_{j \in \mathcal{Y} \setminus y} (0, 1 - m_y(x; \theta_y) + m_j(x; \theta_j))
\end{aligned}$$

Multiclass Max-SVM

- Define loss

$$L(u, y) = \max_{j \in \mathcal{Y} \setminus y} (0, 1 - u_y + u_j)$$

- Let $m = (m_1, \ldots, m_K)$ and define training problem

$$\underset{\theta}{\text{minimize}} \sum_{i=1}^{N} L(m(x_i; \theta), y_i) = \sum_{i=1}^{N} \max_{j \in \mathcal{Y} \setminus y_i} (0, 1 - m_{y_i}(x_i; \theta_{y_i}) + m_j(x_i; \theta_j))$$

- Prediction: Predict class belonging of new data x:

  let $c_y := \min_{j \in \mathcal{Y} \setminus y} c_{y,j}(x; \theta)$ and select $y \in \mathcal{Y}$ that gives largest c_y

  i.e., select the one with highest minimum confidence

Multiclass SVM – Sum version

- Multiclass SVM labels $y \in \mathcal{Y} = \{1, \ldots, K\}$
- Sum version: Sum hinge on confidence w.r.t. all other classes

$$\sum_{j \in \mathcal{Y} \setminus y} \max(0, 1 - c_{y,j}(x;\theta))$$

- Loss can be written as

$$\sum_{j \in \mathcal{Y} \setminus y} \max(0, 1 - c_{y,j}(x;\theta)) = \sum_{j \in \mathcal{Y} \setminus y} \max(0, 1 - m_y(x;\theta_y) + m_j(x;\theta_j))$$

Multiclass Sum-SVM

- Define loss

$$L(u, y) = \sum_{j \in \mathcal{Y} \setminus y} \max(0, 1 - u_y + u_j)$$

- Let $m = (m_1, \ldots, m_K)$ and define training problem

$$\begin{aligned}
\underset{\theta}{\text{minimize}} \sum_{i=1}^{N} L(m(x_i; \theta), y_i) \\
= \sum_{i=1}^{N} \sum_{j \in \mathcal{Y} \setminus y_i} \max(0, 1 - m_{y_i}(x_i; \theta_{y_i}) + m_j(x_i; \theta_j))
\end{aligned}$$

- Prediction: Predict class belonging of new data x:

let $c_y := \sum_{j \in \mathcal{Y} \setminus y} c_{y,j}(x; \theta)$ and select $y \in \mathcal{Y}$ that gives largest c_y

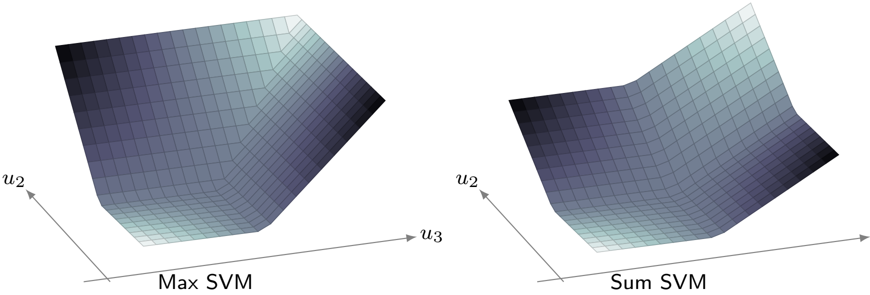
i.e., select the one with highest average confidence

Comparing Max-SVM and Sum-SVM losses

- Multiclass Max-SVM and Sum-SVM for $K = 3$, $u_1 = 1$, $y = 1$

$$L((1, u_2, u_3), 1) = \max(0, u_2, u_3)$$
$$L((1, u_2, u_3), 1) = \max(0, u_2) + \max(0, u_3)$$

- Max-SVM similar to multiclass logistic loss, but sharp corners



Tikhonov regularized versions

- State versions with feature maps ϕ and without bias terms b_j
- Regularized multiclass Max-SVM training problem

$$\underset{\theta}{\text{minimize}} \sum_{i=1}^{N} \max_{j \in \mathcal{Y} \setminus y_i} (0, 1 - \phi(x_i)^T w_{y_i} + \phi(x_i)^T w_j) + \tfrac{\lambda}{2} \sum_{j=1}^{K} \|w_j\|_2^2$$

- Regularized multiclass Sum-SVM training problem

$$\underset{\theta}{\text{minimize}} \sum_{i=1}^{N} \sum_{j \in \mathcal{Y} \setminus y_i} \max(0, 1 - \phi(x_i)^T w_{y_i} + \phi(x_i)^T w_j) + \tfrac{\lambda}{2} \sum_{j=1}^{K} \|w_j\|_2^2$$

Dual problems and Kernel methods

- Multiclass SVM problems best solved via dual formulations
- Can exploit Kernel trick in dual also in multiclass setting

Dual of Max-SVM – Primal reformulation

Max-SVM with K classes can be written as

$$\underset{w}{\text{minimize}} \underbrace{\sum_{i=1}^{N} \max(c - M_i X_i w)}_{f(MXw)} + \underbrace{\tfrac{\lambda}{2}\|w\|_2^2}_{g(w)},$$

where

$$M_i = \begin{bmatrix} 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ -1 & & & 1 & & & \\ & \ddots & & \vdots & & & \\ & & -1 & 1 & & & \\ & & & 1 & -1 & & \\ & & & \vdots & & \ddots & \\ & & & 1 & & & -1 \end{bmatrix} \in \mathbb{R}^{K \times K} \qquad c = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^K$$

$$X_i = \begin{bmatrix} \phi(x_i)^T & & \\ & \ddots & \\ & & \phi(x_i)^T \end{bmatrix} \in \mathbb{R}^{K \times pK} \qquad w = \begin{bmatrix} w_1 \\ \vdots \\ w_K \end{bmatrix} \in \mathbb{R}^{pK}$$

where i:th column in M_i (except for row 1) is filled with 1s

Dual of Max-SVM – Primal reformulation

- Further

$$M = \begin{bmatrix} M_1 & & \\ & \ddots & \\ & & M_N \end{bmatrix} \in \mathbb{R}^{NK \times NK} \quad X = \begin{bmatrix} X_1 \\ \vdots \\ X_N \end{bmatrix} \in \mathbb{R}^{NK \times pK}$$

  and $M_i X_i = U_i M X$, where $U_i = [0 \ \cdots \ 0 \ I \ 0 \ \cdots \ 0] \in \mathbb{R}^{K \times NK}$
- The function $f : \mathbb{R}^{NK} \to \mathbb{R}$ satisfies $f(u) = \sum_{i=1}^{N} f_i(u_i)$, where
  - $f_i(u_i) = \max(c - u_i)$
  - $u = (u_1, \ldots, u_N) \in \mathbb{R}^{KN}$ and $u_i \in \mathbb{R}^K$

Dual of Max-SVM

- Can be shown that $\max_i(\cdot)^*(\mu_i) = \iota_{\Delta_K}(\mu_i)$, therefore

$$\begin{aligned} f_i^*(\mu_i) &= \sup_{u_i}(\mu_i^T u_i - \max_i(c - u_i)) = \sup_{v_i}(\mu_i^T(c - v_i) - \max_i(v_i)) \\ &= \sup_{v_i}((-\mu_i)^T v_i - \max_i(v_i)) + \mu_i^T c \\ &= \iota_{\Delta_K}(-\mu_i) + \mu_i^T c \end{aligned}$$

  where Δ_K is probability simplex in $\mathbb{R}^K$

- Further, conjugate of separable functions are separable:

$$f^*(\mu) = \sum_{i=1}^N f_i^*(\mu_i) = \sum_{i=1}^N \iota_{\Delta_K}(-\mu_i) + \mu_i^T c.$$

- Conjugate of $g(w) = \frac{\lambda}{2}\|w\|_2^2$ satisfies $g^*(\nu) = \frac{1}{2\lambda}\|\nu\|_2^2$ and

$$g^*(-(MX)^T\mu) = \frac{1}{2\lambda}\mu^T M X X^T M^T \mu$$

Dual of Max-SVM, primal recovery, class belonging

- The dual problem $\underset{\mu}{\text{minimize}}\, f^*(\mu) + g^*(-(MX)^T\mu)$ is

$$\begin{array}{ll} \underset{\mu}{\text{minimize}} & \sum_{i=1}^{N}(c^T\mu_i) + \frac{1}{2\lambda}\mu^T MXX^T M^T\mu \\ \text{subject to} & -\mu_i \in \Delta_K \text{ for all } i \in \{1,\ldots,N\} \end{array}$$

- g^* differentiable; recover primal solution from optimality condition

$$w = \partial g^*(-M^T\mu) = -\tfrac{1}{\lambda}X^T M^T\mu$$

- Predict class belonging y from largest $c_y = \min_{j\in\mathcal{Y}\setminus y} c_{y,j}(x;\theta)$:

$$\begin{aligned} c_{y,j}(x;\theta) &= m_y(x;\theta_y) - m_j(x;\theta_j) = \phi(x)^T w_y - \phi(x)^T w_j \\ &= -\tfrac{1}{\lambda}(\phi(x)^T(X^T M^T\mu)_y - \phi(x)^T(X^T M^T\mu)_j) \end{aligned}$$

where $(\cdot)_y$ and $(\cdot)_j$ refer to the y:th and j:th blocks

Dual with Kernel matrix

- The matrix multiplication XX^T is

$$XX^T = \begin{bmatrix} X_1 \\ \vdots \\ X_N \end{bmatrix} \begin{bmatrix} X_1^T, \cdots, X_N^T \end{bmatrix} = \begin{bmatrix} X_1 X_1^T & \cdots & X_1 X_N^T \\ \vdots & \ddots & \vdots \\ X_N X_1^T & \cdots & X_N X_N^T \end{bmatrix}$$
$$= \begin{bmatrix} \phi(x_1)^T \phi(x_1) I_K & \cdots & \phi(x_1)^T \phi(x_N) I_K \\ \vdots & \ddots & \vdots \\ \phi(x_N)^T \phi(x_1) I_K & \cdots & \phi(x_N)^T \phi(x_N) I_K \end{bmatrix}$$
$$= K \otimes I_K$$

where $\otimes$ is the Kronecker product and K is the Kernel matrix

$$K = \begin{bmatrix} \phi(x_1)^T \phi(x_1) & \cdots & \phi(x_1)^T \phi(x_N) \\ \vdots & \ddots & \vdots \\ \phi(x_N)^T \phi(x_1) & \cdots & \phi(x_N)^T \phi(x_N) \end{bmatrix}$$

- Can replace XX^T by this Kernel matrix in dual problem

Dual from Kernel operator

- Can implicitly define features using Kernel trick (see SVM lecture)
- Let $\kappa : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ be Kernel operator and $[K]_{ij} = \kappa(x_i, x_j)$
- Then, dual problem:

$$\begin{array}{ll} \underset{\mu}{\text{minimize}} & \sum_{i=1}^{N}(c^T\mu_i) + \frac{1}{2\lambda}\mu^T M(K \otimes I_K)M^T\mu \\ \text{subject to} & -\mu_i \in \Delta_K \text{ for all } i \in \{1,\ldots,N\} \end{array}$$

solves primal problem with potentially infinite number of variables

$$\underset{\theta}{\text{minimize}} \sum_{i=1}^{N} \max_{j \in \mathcal{Y} \setminus y_i} (0, 1 - \langle \phi(x_i), w_{y_i} \rangle + \langle \phi(x_i), w_j \rangle) + \frac{\lambda}{2} \sum_{j=1}^{K} \|w_j\|^2$$

Class prediction

Predict class belonging y from largest $c_y = \sum_{j \in \mathcal{Y} \setminus y} c_{y,j}(x;\theta)$:

$$c_{y,j}(x;\theta) = -\frac{1}{\lambda}(\phi(x)^T(X^T M^T \mu)_y - \phi(x)^T(X^T M^T \mu)_j)$$

where

$$\begin{aligned}
\phi(x)^T(X^T M^T \mu)_j &= \phi(x)^T \left(\left[\begin{bmatrix} \phi(x_1) & & \\ & \ddots & \\ & & \phi(x_1) \end{bmatrix} \quad \cdots \quad \begin{bmatrix} \phi(x_N) & & \\ & \ddots & \\ & & \phi(x_N) \end{bmatrix} \right] M^T \mu \right)_j \\
&= \phi(x)^T \left(\left[\begin{bmatrix} \phi(x_1) & & \\ & \ddots & \\ & & \phi(x_1) \end{bmatrix} \quad \cdots \quad \begin{bmatrix} \phi(x_N) & & \\ & \ddots & \\ & & \phi(x_N) \end{bmatrix} \right] \begin{bmatrix} M_1^T \mu_1 \\ \vdots \\ M_N^T \mu_N \end{bmatrix} \right)_j \\
&= \phi(x)^T \left(\begin{bmatrix} \sum_{i=1}^N \phi(x_i)(M_i^T \mu_i)_1 \\ \vdots \\ \sum_{i=1}^N \phi(x_i)(M_i^T \mu_i)_K \end{bmatrix} \right)_j \\
&= \sum_{i=1}^N \phi(x)^T \phi(x_i)(M_i^T \mu_i)_j \\
&= \sum_{i=1}^N \kappa(x, x_i)(M_i^T \mu_i)_j
\end{aligned}$$

Can be decided by evaluating Kernel operator

Dual of Sum-SVM – Primal reformulation

Sum-SVM with K classes can be written as

$$\underset{w}{\text{minimize}} \underbrace{\sum_{i=1}^{N} \mathbf{1}^T \max(1 - D_i X_i w)}_{f(DXw)} + \underbrace{\tfrac{\lambda}{2}\|w\|_2^2}_{g(w)},$$

where

$$D_i = \begin{bmatrix} -1 & & & 1 & & & \\ & \ddots & & \vdots & & & \\ & & -1 & 1 & & & \\ & & & 1 & -1 & & \\ & & & \vdots & & \ddots & \\ & & & 1 & & & -1 \end{bmatrix} \in \mathbb{R}^{(K-1)\times K} \quad w = \begin{bmatrix} w_1 \\ \vdots \\ w_K \end{bmatrix} \in \mathbb{R}^{pK}$$

$$X_i = \begin{bmatrix} \phi(x_i)^T & & \\ & \ddots & \\ & & \phi(x_i)^T \end{bmatrix} \in \mathbb{R}^{K \times pK}$$

where i:th column in D_i is filled with 1s

Dual of Sum-SVM – Primal reformulation

- Further

$$D = \begin{bmatrix} D_1 & & \\ & \ddots & \\ & & D_N \end{bmatrix} \in \mathbb{R}^{N(K-1) \times NK} \quad X = \begin{bmatrix} X_1 \\ \vdots \\ X_N \end{bmatrix} \in \mathbb{R}^{NK \times pK}$$

  and $D_i X_i = U_i D X$, where
  $U_i = [0 \ \cdots \ 0 \ I \ 0 \ \cdots \ 0] \in \mathbb{R}^{(K-1) \times N(K-1)}$
- The function $f : \mathbb{R}^{N(K-1)} \to \mathbb{R}$ satisfies $f(u) = \sum_{i=1}^{N} f_i(u_i)$;
  - $f_i(u_i) = \mathbf{1}^T \max(0, 1 - u_i)$ is sum of hinge losses
  - $u = (u_1, \ldots, u_N) \in \mathbb{R}^{(K-1)N}$ and $u_i \in \mathbb{R}^{K-1}$

Dual of Sum-SVM

- Conjugate of sum of hinge losses f_i satisfies

$$f_i^*(\mu_i) = \mu_i^T \mathbf{1} + \iota_{[-1,0]}(\mu_i)$$

- Further, conjugate of separable functions are separable:

$$f^*(\mu) = \sum_{i=1}^{N} f_i^*(\mu_i) = \sum_{i=1}^{N} (\iota_{[-1,0]}(-\mu_i) + \mathbf{1}^T \mu_i) = \iota_{[-1,0]}(-\mu) + \mathbf{1}^T \mu$$

- Conjugate of $g(w) = \frac{\lambda}{2}\|w\|_2^2$ satisfies $g^*(\nu) = \frac{1}{2\lambda}\|\nu\|_2^2$ and

$$g^*(-(DX)^T\mu) = \frac{1}{2\lambda}\mu^T DXX^T D^T \mu$$

Dual of Sum-SVM, primal recovery, class belonging

- The dual problem $\underset{\mu}{\text{minimize}}\, f^*(\mu) + g^*(-(DX)^T\mu)$ is

$$\begin{array}{ll} \underset{\mu}{\text{minimize}} & 1^T\mu + \frac{1}{2\lambda}\mu^T DXX^T D^T\mu \\ \text{subject to} & -1 \leq \mu \leq 0 \end{array}$$

- g^* differentiable; recover primal solution from optimality condition

$$w = \partial g^*(-D^T\mu) = -\tfrac{1}{\lambda}X^T D^T\mu$$

- Predict class belonging y from largest $c_y = \sum_{j\in\mathcal{Y}\setminus y} c_{y,j}(x;\theta)$:

$$\begin{aligned} c_{y,j}(x;\theta) &= m_y(x;\theta_y) - m_j(x;\theta_j) = \phi(x)^T w_y - \phi(x)^T w_j \\ &= -\tfrac{1}{\lambda}(\phi(x)^T(X^T D^T\mu)_y - \phi(x)^T(X^T D^T\mu)_j) \end{aligned}$$

where $(\cdot)_y$ and $(\cdot)_j$ refer to the y:th and j:th blocks

Dual with Kernel matrix

- The matrix multiplication XX^T is

$$
\begin{aligned}
XX^T = \begin{bmatrix} X_1 \\ \vdots \\ X_N \end{bmatrix} \left[X_1^T, \cdots, X_N^T\right] &= \begin{bmatrix} X_1X_1^T & \cdots & X_1X_N^T \\ \vdots & \ddots & \vdots \\ X_NX_1^T & \cdots & X_NX_N^T \end{bmatrix} \\
&= \begin{bmatrix} \phi(x_1)^T\phi(x_1)I_K & \cdots & \phi(x_1)^T\phi(x_N)I_K \\ \vdots & \ddots & \vdots \\ \phi(x_N)^T\phi(x_1)I_K & \cdots & \phi(x_N)^T\phi(x_N)I_K \end{bmatrix} \\
&= K \otimes I_K
\end{aligned}
$$

where $\otimes$ is the Kronecker product and K is the Kernel matrix

$$
K = \begin{bmatrix} \phi(x_1)^T\phi(x_1) & \cdots & \phi(x_1)^T\phi(x_N) \\ \vdots & \ddots & \vdots \\ \phi(x_N)^T\phi(x_1) & \cdots & \phi(x_N)^T\phi(x_N) \end{bmatrix}
$$

- Can replace XX^T by this Kernel matrix in dual problem

Dual from Kernel operator

- Can implicitly define features using Kernel trick (see SVM lecture)
- Let $\kappa : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ be Kernel operator and $[K]_{ij} = \kappa(x_i, x_j)$
- Then, dual problem:

$$\begin{array}{ll} \underset{\mu}{\text{minimize}} & \mathbf{1}^T \mu + \frac{1}{2\lambda} \mu^T D(K \otimes I_K) D^T \mu \\ \text{subject to} & -1 \leq \mu \leq 0 \end{array}$$

solves primal problem with potentially infinite number of variables

$$\underset{\theta}{\text{minimize}} \sum_{i=1}^{N} \sum_{j \in \mathcal{Y} \setminus y_i} (0, 1 - \langle \phi(x_i), w_{y_i} \rangle + \langle \phi(x_i), w_j \rangle) + \frac{\lambda}{2} \sum_{j=1}^{K} \|w_j\|^2$$

Class prediction

Predict class belonging y from largest $c_y = \min_{j \in \mathcal{Y} \setminus y} c_{y,j}(x;\theta)$:

$$c_{y,j}(x;\theta) = -\frac{1}{\lambda}(\phi(x)^T (X^T D^T \mu)_y - \phi(x)^T (X^T D^T \mu)_j)$$

where

$$\begin{aligned}
\phi(x)^T (X^T D^T \mu)_j &= \phi(x)^T \left(\begin{bmatrix} \begin{bmatrix} \phi(x_1) & & \\ & \ddots & \\ & & \phi(x_1) \end{bmatrix} & \cdots & \begin{bmatrix} \phi(x_N) & & \\ & \ddots & \\ & & \phi(x_N) \end{bmatrix} \end{bmatrix} D^T \mu \right)_j \\
&= \phi(x)^T \left(\begin{bmatrix} \begin{bmatrix} \phi(x_1) & & \\ & \ddots & \\ & & \phi(x_1) \end{bmatrix} & \cdots & \begin{bmatrix} \phi(x_N) & & \\ & \ddots & \\ & & \phi(x_N) \end{bmatrix} \end{bmatrix} \begin{bmatrix} D_1^T \mu_1 \\ \vdots \\ D_N^T \mu_N \end{bmatrix} \right)_j \\
&= \phi(x)^T \left(\begin{bmatrix} \sum_{i=1}^N \phi(x_i)(D_i^T \mu_i)_1 \\ \vdots \\ \sum_{i=1}^N \phi(x_i)(D_i^T \mu_i)_K \end{bmatrix} \right)_j \\
&= \sum_{i=1}^N \phi(x)^T \phi(x_i)(D_i^T \mu_i)_j \\
&= \sum_{i=1}^N \kappa(x, x_i)(D_i^T \mu_i)_j
\end{aligned}$$

Can be decided by evaluating Kernel operator