Statistical Machine Learning (BE4M33SSU)
Lecture 11: Structured Output Support Vector Machines

Czech Technical University in Prague
V. Franc

Linear classifier

Two-class linear classifier:

- $\mathcal{X}$ is a set of observations and $\mathcal{Y} = \{+1, -1\}$ is a set of hidden labels
- $\phi\colon \mathcal{X} \to \mathbb{R}^n$ - feature map embedding observations from $\mathcal{X}$ to $\mathbb{R}^n$
- Two-class linear classifier $h\colon \mathcal{X} \to \mathcal{Y}$

$$h(x; \boldsymbol{w}, b) = \operatorname{sign}(\langle \boldsymbol{w}, \boldsymbol{\phi}(x) \rangle + b) = \begin{cases} +1 & \text{if} \quad \langle \boldsymbol{w}, \boldsymbol{\phi}(x) \rangle + b \geq 0 \\ -1 & \text{if} \quad \langle \boldsymbol{w}, \boldsymbol{\phi}(x) \rangle + b < 0 \end{cases}$$

A generic linear classifier:

- $\mathcal{X}$ is set of observations and $\mathcal{Y}$ is a finite set
- $\phi\colon \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^n$ is a joint feature map embedding $\mathcal{X} \times \mathcal{Y}$ to $\mathbb{R}^n$
- Generic linear classifier $h\colon \mathcal{X} \to \mathcal{Y}$

$$h(x; \boldsymbol{w}) \in \operatorname*{Argmax}_{y \in \mathcal{Y}} \langle \boldsymbol{w}, \boldsymbol{\phi}(x, y) \rangle$$

Example: multi-class linear classifier

- $\mathcal{X}$ is a set of observations and $\mathcal{Y} = \{1, \ldots, Y\}$ is a set of class labels
- Multi-class linear classifier $h\colon \mathcal{X} \to \mathcal{Y}$

$$h(x; \boldsymbol{w}) \in \operatorname*{Argmax}_{y \in \mathcal{Y}} \langle \boldsymbol{w}_y, \boldsymbol{\phi}(x) \rangle$$

where $\boldsymbol{\phi}\colon \mathcal{X} \to \mathbb{R}^d$ is a feature map $\boldsymbol{w} = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_Y) \in \mathbb{R}^{d \cdot Y}$ are parameters.

- We can write the score function as

$$\langle \boldsymbol{w}_y, \boldsymbol{\phi}(x) \rangle = \langle \boldsymbol{w}, \boldsymbol{\phi}(x, y) \rangle$$

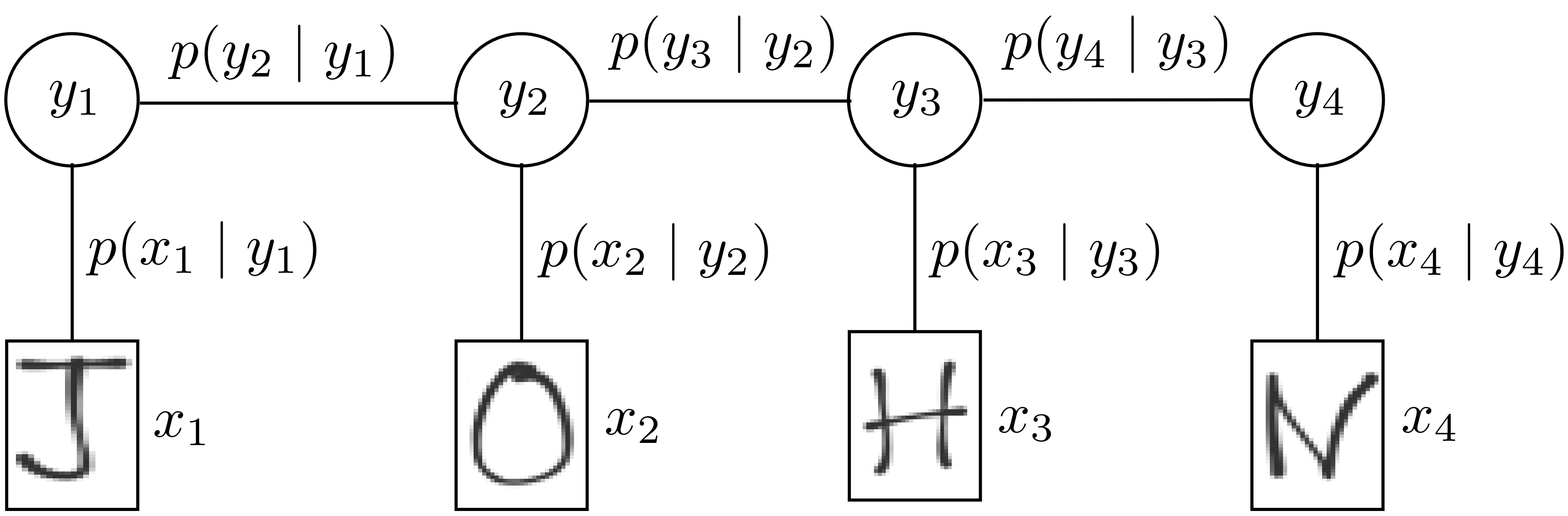
where $\boldsymbol{\phi}\colon \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^{d \cdot Y}$ is

$$\boldsymbol{\phi}(x, y) = (0, \ldots, \underbrace{\boldsymbol{\phi}(x)}_{y-\text{th slot}}, \ldots, 0)$$

Example: sequence classifier for OCR

- $\boldsymbol{x} = (x_1, \dots, x_L) \in \mathcal{I}^L$ - sequence of images with characters
- $\boldsymbol{y} = (y_1, \dots, y_L) \in \mathcal{A}^L$ - seq. of chars. from $\mathcal{A} = \{A, \dots, Z\}$
- $p(x_i \mid y_i)$ - appearance model for characters
- $p(y_i \mid y_{i-1})$ - language model
- Finding the most probable sequence of characters:

$$\hat{\boldsymbol{y}} \in \operatorname*{Argmax}_{\boldsymbol{y} \in \mathcal{A}^L} \underbrace{\left(p(y_1) \prod_{i=2}^{L} p(y_i \mid y_{i-1}) \prod_{i=1}^{L} p(x_i \mid y_i) \right)}_{p(x_1, \dots, x_L, y_1, \dots, y_L)}$$



Example: sequence classifier for OCR

- The MAP estimate from HMM:

$$\hat{\boldsymbol{y}} \in \operatorname*{Argmax}_{\boldsymbol{y}\in\mathcal{A}^L} \left(\log p(y_1) + \sum_{i=2}^{L} \log p(y_i \mid y_{i-1}) + \sum_{i=1}^{L} \log p(x_i \mid y_i) \right)$$

- Let us assume the following parametrization:

$$\begin{aligned} \log p(y_1) &= \langle \boldsymbol{w}, \boldsymbol{\phi}(y_1) \rangle \\ \log p(y_i \mid y_{i-1}) &= \langle \boldsymbol{w}, \boldsymbol{\phi}(y_{i-1}, y_i) \rangle \\ \log p(x_i | y_i) &= \langle \boldsymbol{w}, \boldsymbol{\phi}(x_i, y_i) \rangle \end{aligned}$$

- The MAP estimate becomes a linear classifier:

$$\hat{\boldsymbol{y}} = \operatorname*{Argmax}_{(y_1,\ldots,y_k)\in\mathcal{A}^L} \left\langle \boldsymbol{w}, \underbrace{\boldsymbol{\phi}(y_1) + \sum_{i=2}^{L} \boldsymbol{\phi}(y_{i-1}, y_i) + \sum_{i=1}^{L} \boldsymbol{\phi}(x_i, y_i)}_{\boldsymbol{\phi}(\boldsymbol{x},\boldsymbol{y})} \right\rangle$$

Learning by Emprical Risk Minimization

- $\ell\colon \mathcal{Y} \times \mathcal{Y} \to [0, \infty)$ - loss such that $\ell(y, y') = 0$ iff $y = y'$.
- Find parameters $\boldsymbol{w}$ of $h(x; \boldsymbol{w})$ which minimize the expected risk

$$R(\boldsymbol{w}) = \mathbb{E}_{(x,y)\sim p}\Big(\ell(y, h(x; \boldsymbol{w}))\Big)$$

- The Empirical Risk Minimization principle leads to solving

$$\boldsymbol{w}^* \in \operatorname*{Argmin}_{\boldsymbol{w} \in \mathbb{R}^n} R_{\mathcal{T}^m}(\boldsymbol{w})$$

where the empirical risk is

$$R_{\mathcal{T}^m}(\boldsymbol{w}) = \frac{1}{m} \sum_{i=1}^{m} \ell(y^i, h(x^i; \boldsymbol{w}))$$

and $\mathcal{T}^m = \{(x^i, y^i) \in (\mathcal{X} \times \mathcal{Y}) \mid i = 1, \ldots, m\}$ are training examples drawn from i.i.d. with distribution $p(x, y)$.

Learning linear classifier from separable examples

- A correctly classified example (x^i, y^i), that is,

$$y^i = h(x^i; \boldsymbol{w}) = \operatorname*{Argmax}_{y \in \mathcal{Y}} \langle \boldsymbol{w}, \boldsymbol{\phi}(x^i, y) \rangle$$

implies

$$\langle \boldsymbol{\phi}(x^i, y^i), \boldsymbol{w} \rangle > \langle \boldsymbol{\phi}(x^i, y), \boldsymbol{w} \rangle \,, \qquad \forall y \in \mathcal{Y} \setminus \{y^i\}$$

Definition 1. *The examples* $\mathcal{T}^m = \{(x^i, y^i) \in (\mathcal{X} \times \mathcal{Y}) \mid i = 1, \dots, m\}$ *are linearly separable w.r.t. joint feature map* $\boldsymbol{\phi}\colon \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^n$ *if there exists* $\boldsymbol{w} \in \mathbb{R}^n$ *such that*

$$\langle \boldsymbol{\phi}(x^i, y^i), \boldsymbol{w} \rangle > \langle \boldsymbol{\phi}(x^i, y), \boldsymbol{w} \rangle \,, \qquad \forall i \in \{1, \dots, m\}, y \in \mathcal{Y} \setminus \{y^i\}$$

(Generic) Perceptron algorithm

- **Task:** given a set of points $\{\boldsymbol{a}^i \in \mathbb{R}^n \mid i = 1, 2, \ldots, K\}$ we want to find $\boldsymbol{w} \in \mathbb{R}^n$ such that

$$\langle \boldsymbol{w}, \boldsymbol{a}^i \rangle > 0 \,, \qquad \forall i \in \{1, 2, \ldots, K\} \tag{1}$$

- **Perceptron:**

  1. $\boldsymbol{w} \leftarrow \boldsymbol{0}$
  2. Find a violating $\langle \boldsymbol{w}, \boldsymbol{a}^i \rangle \leq 0$, $i \in \{1, 2, \ldots, K\}$
  3. If there is no violating inequality return $\boldsymbol{w}$ otherwise update

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \boldsymbol{a}^i$$

  and go to step 2.

- If the set of inequalities (1) is solvable then the Perceptron algorithm exits in a finite number of steps which does not depend on m.

Structured Output Perceptron

- Learning $h(x;\boldsymbol{w}) \in \operatorname{Argmax}_{y\in\mathcal{Y}}\langle \boldsymbol{w}, \boldsymbol{\phi}(x,y)\rangle$ from examples $\mathcal{T}^m = \{(x^i,y^i) \in (\mathcal{X}\times\mathcal{Y}) \mid i=1,\ldots,m\}$ leads to solving

$$\langle \boldsymbol{\phi}(x^i,y^i),\boldsymbol{w}\rangle - \langle \boldsymbol{\phi}(x^i,y),\boldsymbol{w}\rangle > 0 \,, \qquad \forall i \in \{1,\ldots,m\}, y \in \mathcal{Y}\setminus\{y^i\}$$

- **Structured Output Perceptron:**

1. $\boldsymbol{w} \leftarrow \mathbf{0}$
2. Find a misclassified example $(x^i,y^i) \in \mathcal{T}^m$ such that

$$y^i \neq \hat{y}^i \in \operatorname*{Argmax}_{y\in\mathcal{Y}}\langle \boldsymbol{w}, \boldsymbol{\phi}(x^i,y)\rangle \qquad \text{prediction problem}$$

3. If there is no misclassified example return $\boldsymbol{w}$ otherwise update

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \boldsymbol{\phi}(x^i,y^i) - \boldsymbol{\phi}(x^i,\hat{y}^i) \qquad \text{parameter update}$$

and go to step 2.

Structured Output SVM

- Learning $h(x;\boldsymbol{w}) \in \mathrm{Argmax}_{y\in\mathcal{Y}}\langle \boldsymbol{w}, \boldsymbol{\phi}(x,y)\rangle$ from examples $\mathcal{T}^m = \{(x^i,y^i)\in(\mathcal{X}\times\mathcal{Y}) \mid i=1,\ldots,m\}$ by ERM leads to

$$\boldsymbol{w}^* \in \operatorname*{Argmin}_{\boldsymbol{w}\in\mathbb{R}^n} R_{\mathcal{T}^m}(\boldsymbol{w}) \quad \text{where} \quad R_{\mathcal{T}^m}(\boldsymbol{w}) = \frac{1}{m}\sum_{i=1}^m \ell(y^i, h(x^i;\boldsymbol{w}))$$

- The SO-SVM approximates the ERM by a convex problem

$$\boldsymbol{w}^* \in \operatorname*{Argmin}_{\boldsymbol{w}\in\mathcal{W}_r} R^{\psi}(\boldsymbol{w}) \quad \text{where} \quad R^{\psi}(\boldsymbol{w}) = \frac{1}{m}\sum_{i=1}^m \psi(x^i, y^i, \boldsymbol{w})$$

where

- $\mathcal{W}_r \subseteq \mathbb{R}^n$ - convex feasible set; e.g. $\mathcal{W}_r = \{\boldsymbol{w}\in\mathbb{R}^n \mid \|\boldsymbol{w}\| \leq r\}$
- $\psi\colon \mathcal{X}\times\mathcal{Y}\times\mathbb{R}^n \to \mathbb{R}$ - convex proxy approximating the true loss ℓ

Margin rescaling loss

- We require that the score of the correct label y^i is higher than the score of the incorrect label y by margin proportional to the loss $\ell(y^i, y)$:

$$\langle \boldsymbol{w}, \phi(x^i, y^i) \rangle \geq \langle \boldsymbol{w}, \phi(x^i, y) \rangle + \ell(y^i, y)\,, \qquad \forall y \in \mathcal{Y} \setminus \{y^i\}$$

- The margin rescaling loss

$$\psi(x^i, y^i, \boldsymbol{w}) = \max\Big\{0, \max_{y \in \mathcal{Y} \setminus \{y^i\}} \{\ell(y^i, y) + \langle \boldsymbol{w}, \boldsymbol{\phi}(x^i, y) \rangle - \langle \boldsymbol{w}, \boldsymbol{\phi}(x^i, y^i) \rangle\}\Big\}$$

- Upper bounds of the true loss:

$$y^i \neq \hat{y} = h(x^i; \boldsymbol{w}) \in \operatorname*{Argmax}_{y \in \mathcal{Y}} \langle \boldsymbol{w}, \boldsymbol{\phi}(x^i, y) \rangle$$

implies $\langle \boldsymbol{w}, \boldsymbol{\phi}(x^i, \hat{y}) \rangle - \langle \boldsymbol{w}, \boldsymbol{\phi}(x^i, y^i) \rangle \geq 0$ and hence

$$\psi(x^i, y^i, \boldsymbol{w}) \geq \ell(y^i, h(x^i, \boldsymbol{w}))\,, \qquad \forall \boldsymbol{w} \in \mathbb{R}^n$$

SO-SVM with margin-rescaling loss

- Using shortcuts $\ell_i(y) = \ell(y^i, y)$ and $\boldsymbol{\phi}_i(y) = \boldsymbol{\phi}(x^i, y) - \boldsymbol{\phi}(x^i, y^i)$ we can simplify the margin rescaling loss:

$$\begin{aligned}\psi(x^i, y^i, \boldsymbol{w}) &= \max\{0, \max_{y\in\mathcal{Y}\setminus\{y^i\}} \{\ell(y^i, y) + \langle \boldsymbol{w}, \boldsymbol{\phi}(x^i, y)\rangle - \langle \boldsymbol{w}, \boldsymbol{\phi}(x^i, y^i)\rangle\}\} \\ &= \max_{y\in\mathcal{Y}}\{\ell(y^i, y) + \langle \boldsymbol{w}, \boldsymbol{\phi}(x^i, y)\rangle - \langle \boldsymbol{w}, \boldsymbol{\phi}(x^i, y^i)\rangle\} \\ &= \max_{y\in\mathcal{Y}}\{\ell_i(y) + \langle \boldsymbol{w}, \boldsymbol{\phi}_i(y)\rangle\}\end{aligned}$$

- The SO-SVM leads to a convex constrained optimization problem:

$$\boldsymbol{w}^* \in \operatorname*{Argmin}_{\boldsymbol{w}\in\mathcal{W}_r} R^\psi(\boldsymbol{w}) \quad \text{where} \quad R^\psi(\boldsymbol{w}) = \frac{1}{m}\sum_{i=1}^m \max_{y\in\mathcal{Y}}\{\ell_i(y) + \langle \boldsymbol{w}, \boldsymbol{\phi}_i(y)\rangle\}$$

and $\mathcal{W}_r = \{\boldsymbol{w} \in \mathbb{R}^n \mid \|\boldsymbol{w}\| \leq r\}$ is a convex set.

SO-SVM as a convex quadratic program

- The SO-SVM problem can be written as unconstrained problem:

$$\boldsymbol{w}^* \in \operatorname*{Argmin}_{\boldsymbol{w}\in\mathbb{R}^n} \left(\frac{\lambda}{2}\|\boldsymbol{w}\|^2 + R^{\psi}(\boldsymbol{w}) \right)$$

- After introducing slack variables it can be further rewritten as a constrained quadratic program:

$$\boldsymbol{w}^* = \operatorname*{argmin}_{\boldsymbol{w}\in\mathbb{R}^n, \boldsymbol{\xi}\in\mathbb{R}^m} \left(\frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \frac{1}{m}\sum_{i=1}^{m} \xi_i \right)$$

subject to

$$\xi_i \geq \ell_i(y) + \langle \boldsymbol{w}, \boldsymbol{\phi}_i(y) \rangle \,, \quad \forall i \in \{1,\ldots,m\}, \forall y \in \mathcal{Y}$$

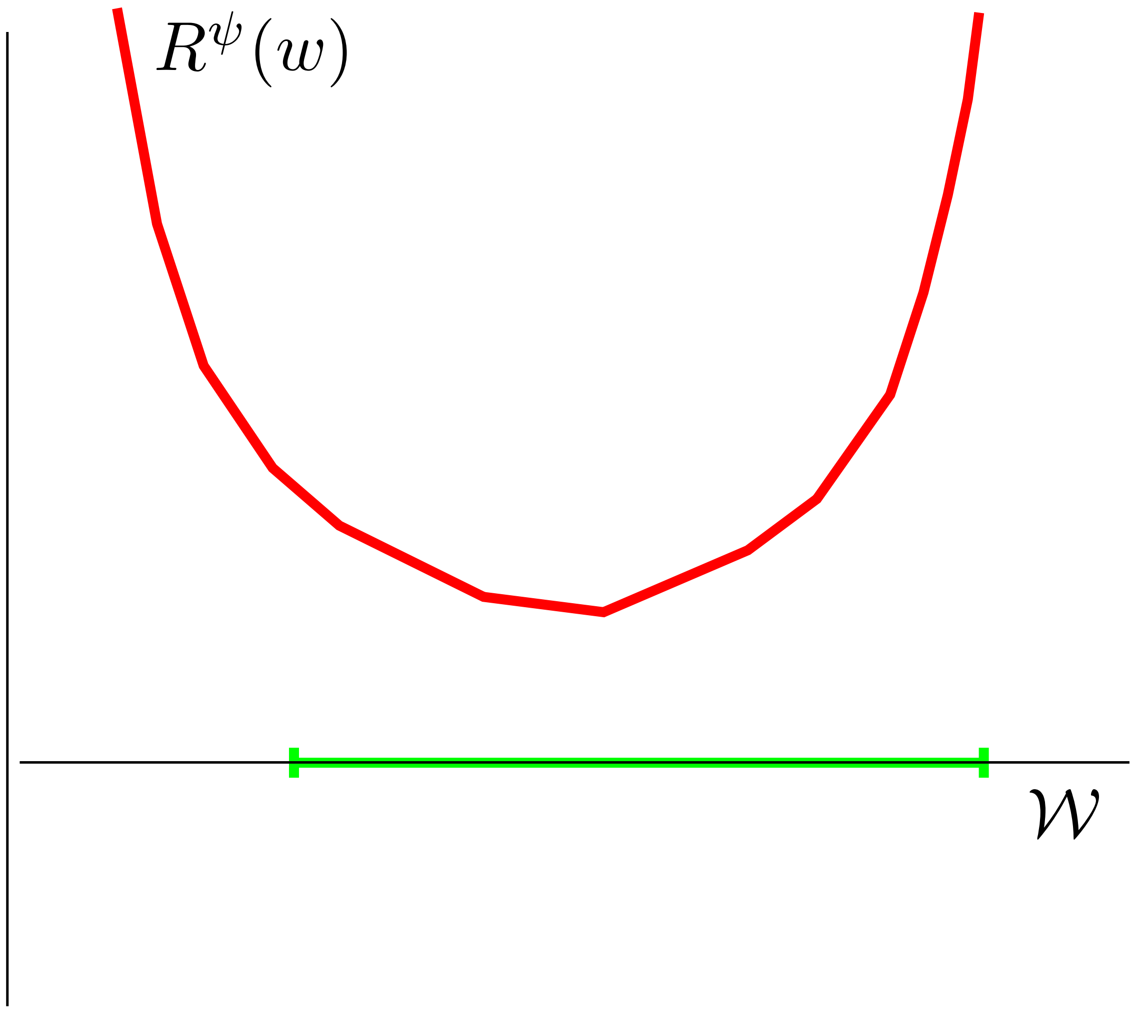
- Note that the QP has $m|\mathcal{Y}|$ linear constaints !

Cutting plane algorithm

$$R^\psi(\boldsymbol{w}) = \frac{1}{m}\sum_{i=1}^{m} \max_{\hat{y}^i \in \mathcal{Y}} (\ell_i(\hat{y}^i) + \langle \boldsymbol{\phi}_i(\hat{y}^i), \boldsymbol{w}\rangle) = \max_{\substack{\hat{y}^1 \in \mathcal{Y} \\ \vdots \\ \hat{y}^m \in \mathcal{Y}}} \frac{1}{m}\sum_{i=1}^{m} (\ell_i(\hat{y}^i) + \langle \boldsymbol{\phi}_i(\hat{y}^i), \boldsymbol{w}\rangle)$$

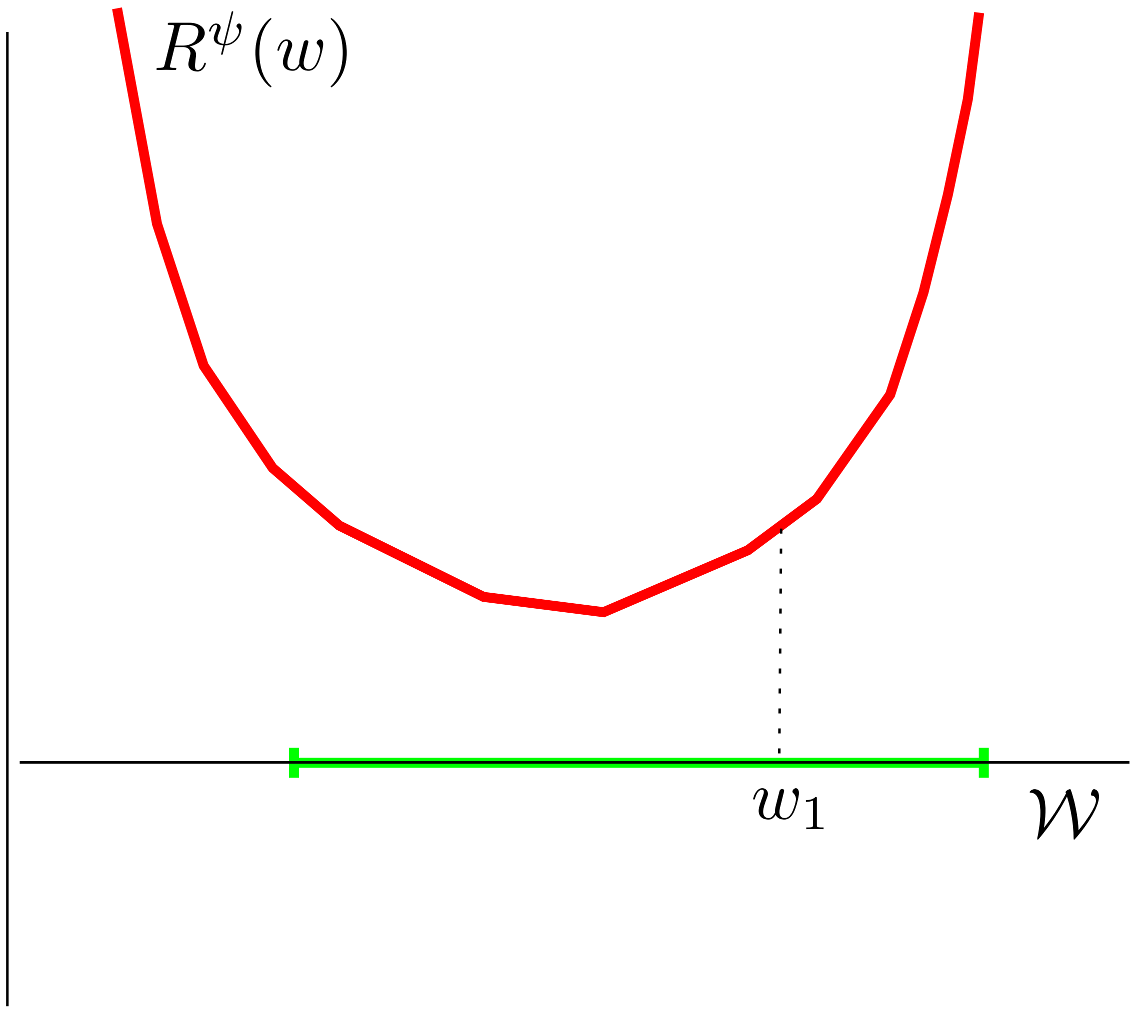
Cutting plane algorithm

$$R^{\psi}(\boldsymbol{w}) = \frac{1}{m}\sum_{i=1}^{m} \max_{\hat{y}^i \in \mathcal{Y}} (\ell_i(\hat{y}^i) + \langle \boldsymbol{\phi}_i(\hat{y}^i), \boldsymbol{w}\rangle) = \max_{\substack{\hat{y}^1 \in \mathcal{Y} \\ \vdots \\ \hat{y}^m \in \mathcal{Y}}} \frac{1}{m}\sum_{i=1}^{m} (\ell_i(\hat{y}^i) + \langle \boldsymbol{\phi}_i(\hat{y}^i), \boldsymbol{w}\rangle)$$



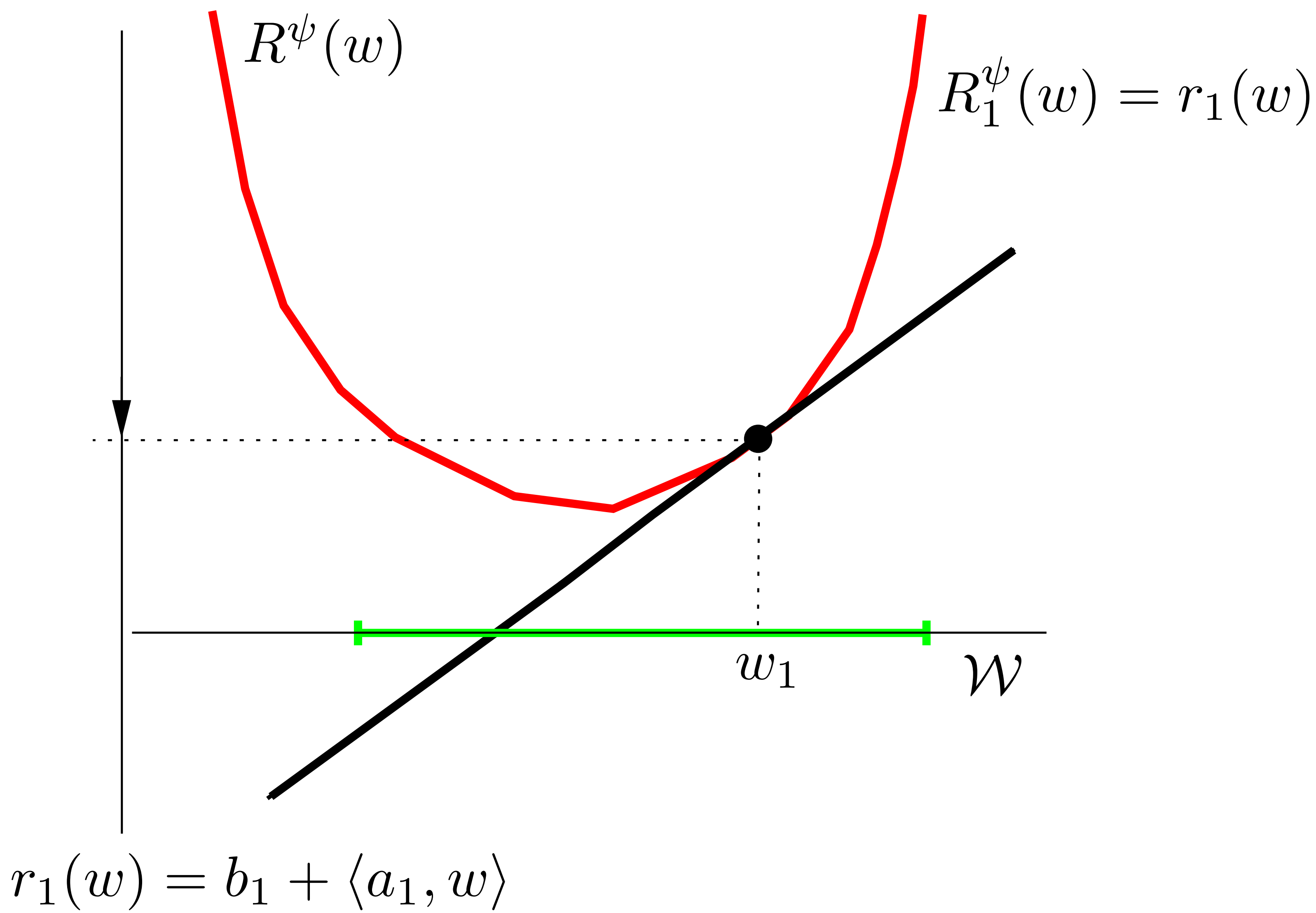
Cutting plane algorithm

$$R^\psi(\boldsymbol{w}) = \frac{1}{m}\sum_{i=1}^{m} \max_{\hat{y}^i \in \mathcal{Y}} (\ell_i(\hat{y}^i) + \langle \boldsymbol{\phi}_i(\hat{y}^i), \boldsymbol{w} \rangle) = \max_{\substack{\hat{y}^1 \in \mathcal{Y} \\ \vdots \\ \hat{y}^m \in \mathcal{Y}}} \frac{1}{m}\sum_{i=1}^{m} (\ell_i(\hat{y}^i) + \langle \boldsymbol{\phi}_i(\hat{y}^i), \boldsymbol{w} \rangle)$$



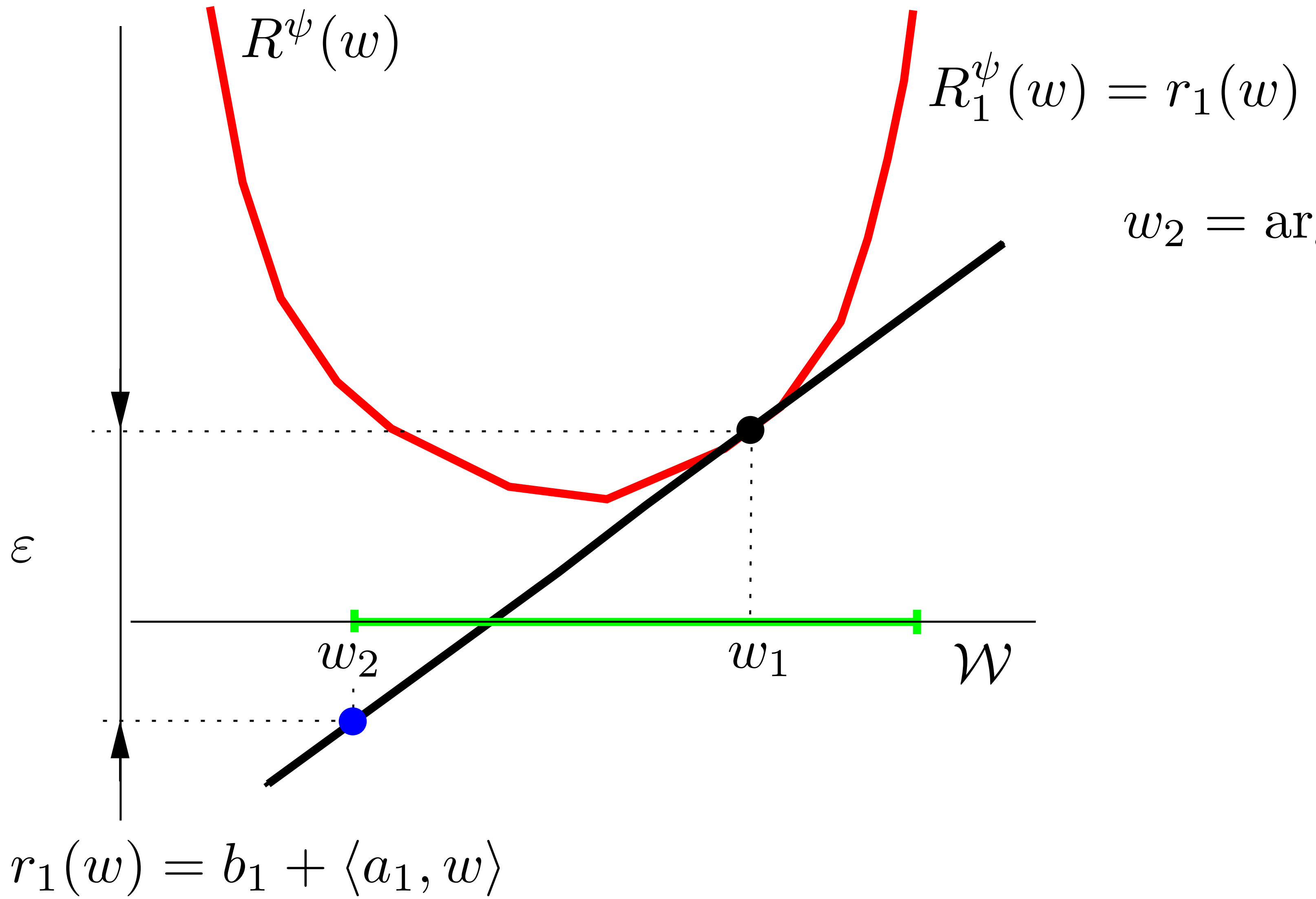
Cutting plane algorithm

$$R^{\psi}(\boldsymbol{w}) = \frac{1}{m}\sum_{i=1}^{m} \max_{\hat{y}^i \in \mathcal{Y}} (\ell_i(\hat{y}^i) + \langle \boldsymbol{\phi}_i(\hat{y}^i), \boldsymbol{w} \rangle) = \max_{\substack{\hat{y}^1 \in \mathcal{Y} \\ \vdots \\ \hat{y}^m \in \mathcal{Y}}} \frac{1}{m}\sum_{i=1}^{m} (\ell_i(\hat{y}^i) + \langle \boldsymbol{\phi}_i(\hat{y}^i), \boldsymbol{w} \rangle)$$



Cutting plane algorithm

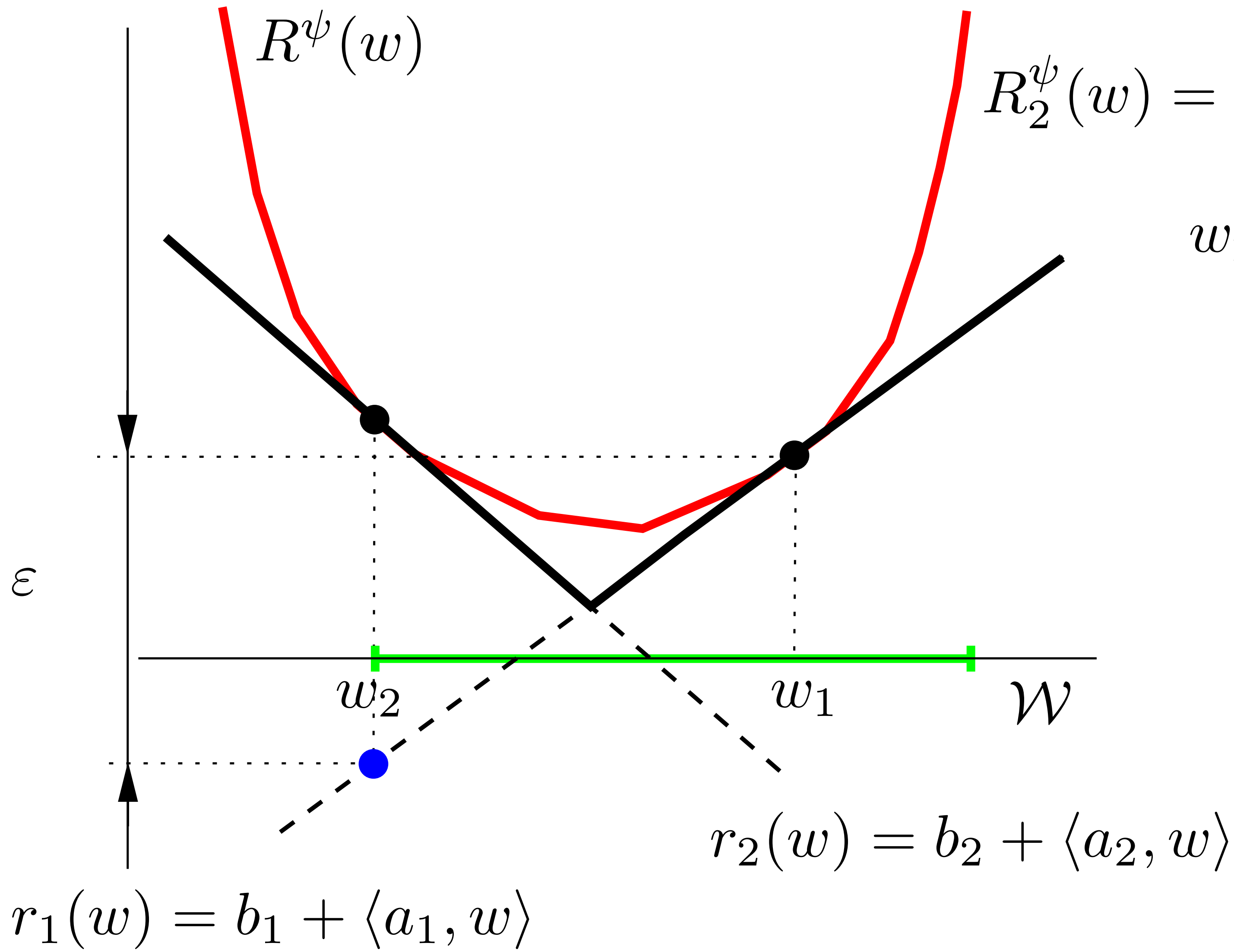
$$R^{\psi}(\boldsymbol{w}) = \frac{1}{m}\sum_{i=1}^{m} \max_{\hat{y}^i \in \mathcal{Y}} (\ell_i(\hat{y}^i) + \langle \boldsymbol{\phi}_i(\hat{y}^i), \boldsymbol{w}\rangle) = \max_{\substack{\hat{y}^1 \in \mathcal{Y} \\ \vdots \\ \hat{y}^m \in \mathcal{Y}}} \frac{1}{m}\sum_{i=1}^{m} (\ell_i(\hat{y}^i) + \langle \boldsymbol{\phi}_i(\hat{y}^i), \boldsymbol{w}\rangle)$$



$$w_2 = \operatorname{argmin}_{w \in \mathcal{W}} R_1^{\psi}(w)$$

Cutting plane algorithm

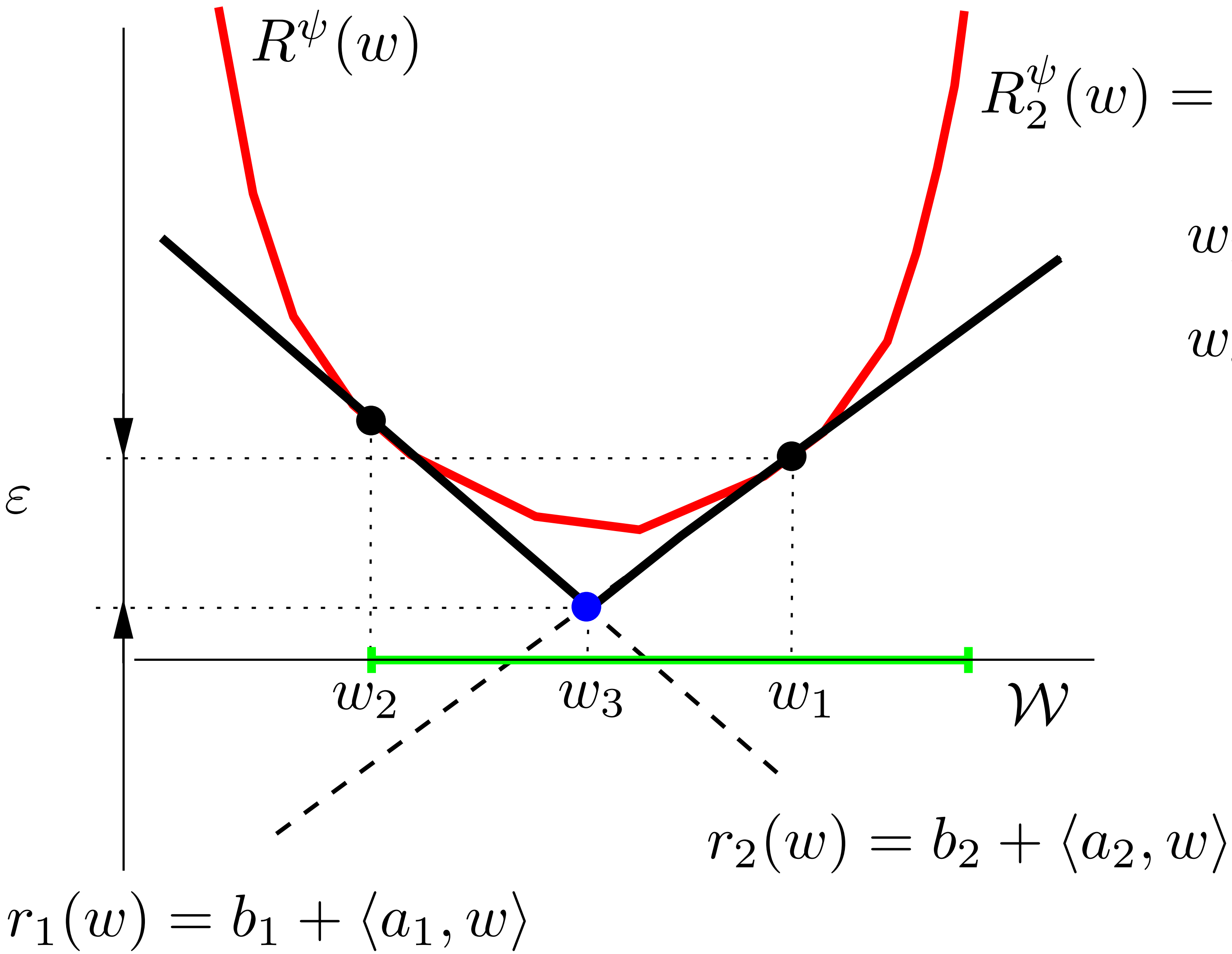
$$R^{\psi}(\boldsymbol{w}) = \frac{1}{m}\sum_{i=1}^{m} \max_{\hat{y}^i \in \mathcal{Y}} (\ell_i(\hat{y}^i) + \langle \boldsymbol{\phi}_i(\hat{y}^i), \boldsymbol{w}\rangle) = \max_{\substack{\hat{y}^1 \in \mathcal{Y} \\ \vdots \\ \hat{y}^m \in \mathcal{Y}}} \frac{1}{m}\sum_{i=1}^{m} (\ell_i(\hat{y}^i) + \langle \boldsymbol{\phi}_i(\hat{y}^i), \boldsymbol{w}\rangle)$$



$w_2 = \operatorname{argmin}_{w\in\mathcal{W}} R_1^{\psi}(w)$

Cutting plane algorithm

$$R^{\psi}(\boldsymbol{w}) = \frac{1}{m}\sum_{i=1}^{m} \max_{\hat{y}^i\in\mathcal{Y}} (\ell_i(\hat{y}^i) + \langle \boldsymbol{\phi}_i(\hat{y}^i), \boldsymbol{w}\rangle) = \max_{\substack{\hat{y}^1\in\mathcal{Y}\\ \vdots \\ \hat{y}^m\in\mathcal{Y}}} \frac{1}{m}\sum_{i=1}^{m} (\ell_i(\hat{y}^i) + \langle \boldsymbol{\phi}_i(\hat{y}^i), \boldsymbol{w}\rangle)$$

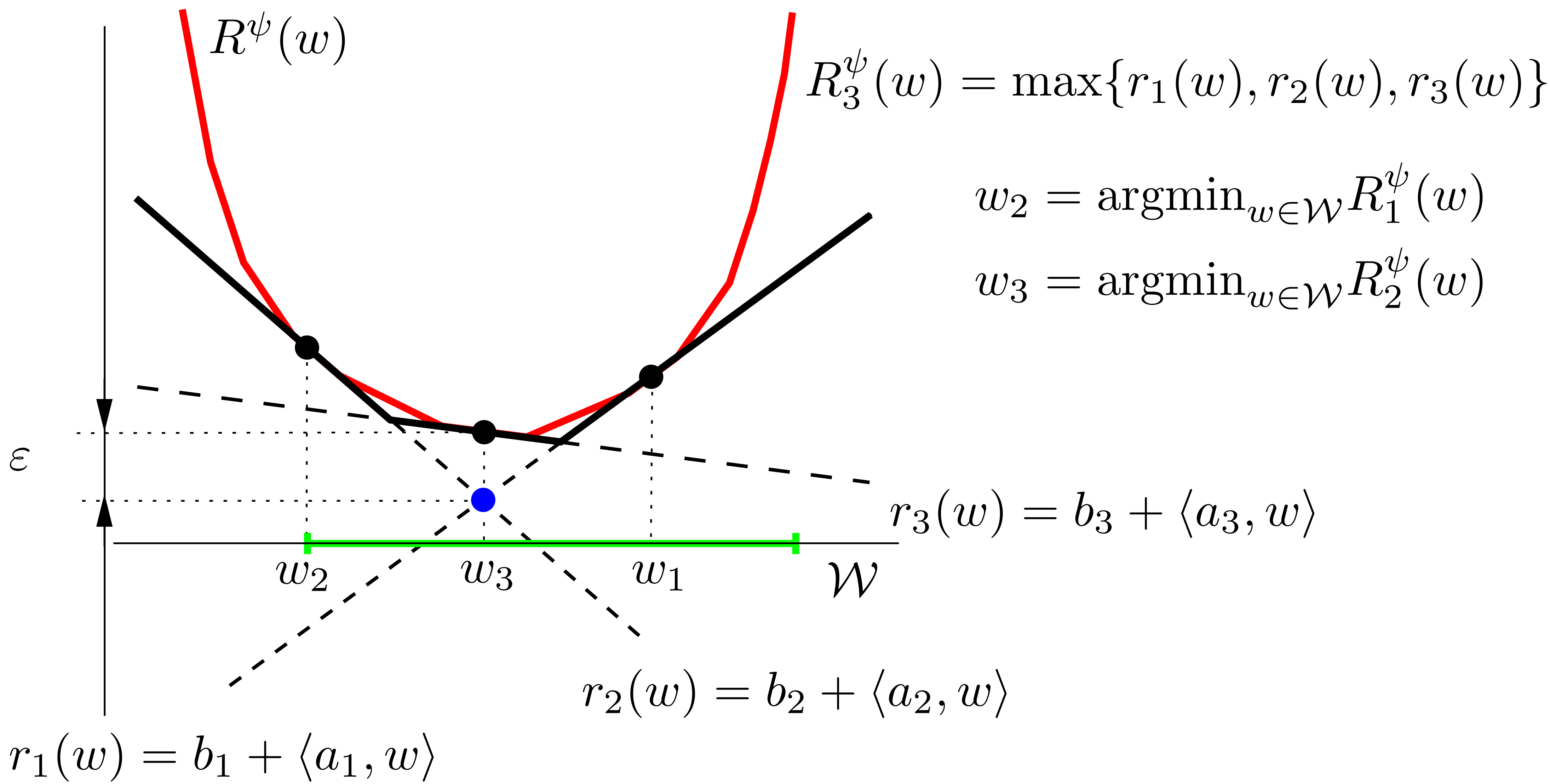


$$w_2 = \text{argmin}_{w\in\mathcal{W}} R^{\psi}_1(w)$$

$$w_3 = \text{argmin}_{w\in\mathcal{W}} R^{\psi}_2(w)$$

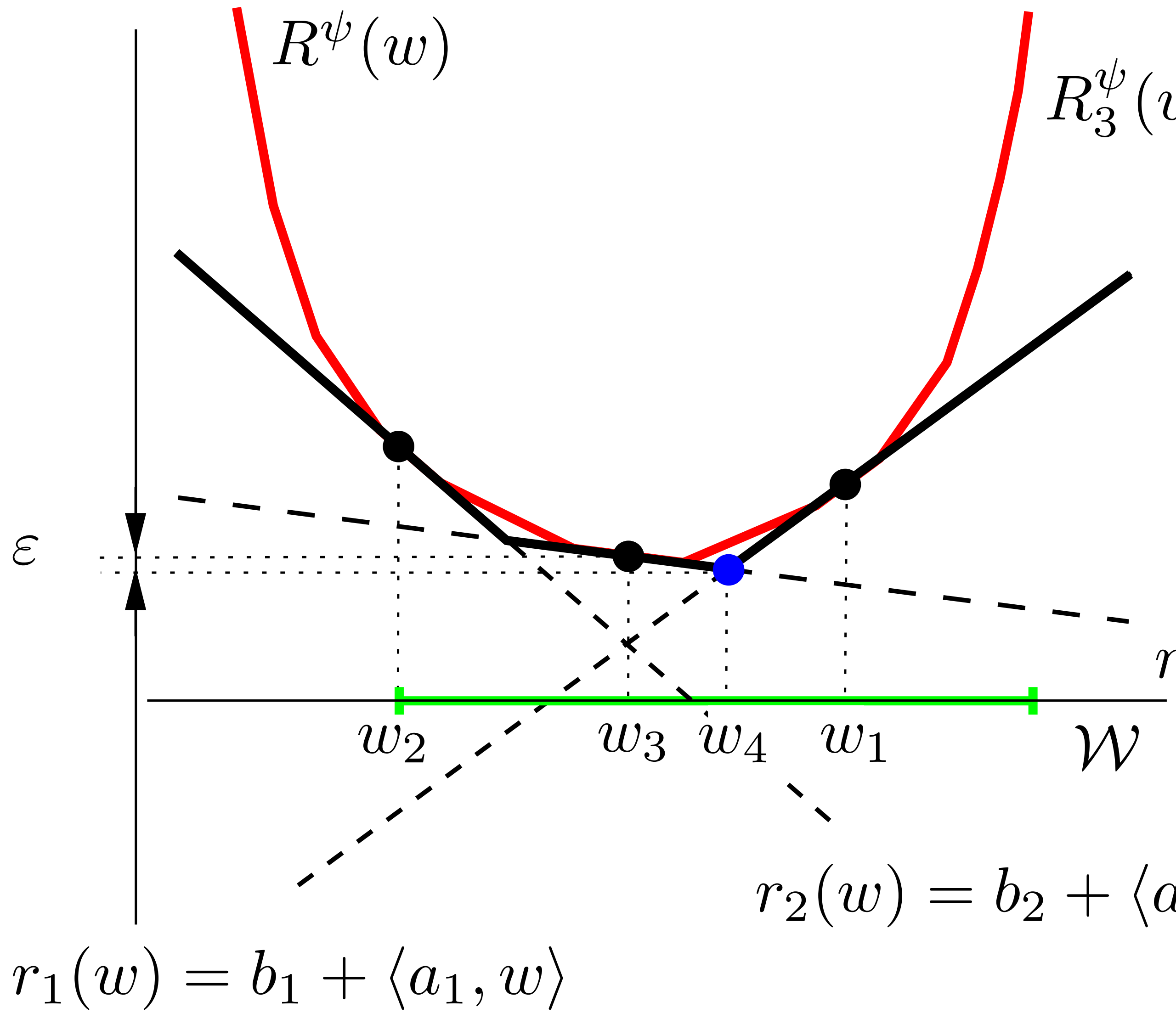
Cutting plane algorithm

$$R^{\psi}(\boldsymbol{w}) = \frac{1}{m}\sum_{i=1}^{m} \max_{\hat{y}^i \in \mathcal{Y}} (\ell_i(\hat{y}^i) + \langle \boldsymbol{\phi}_i(\hat{y}^i), \boldsymbol{w}\rangle) = \max_{\substack{\hat{y}^1 \in \mathcal{Y} \\ \vdots \\ \hat{y}^m \in \mathcal{Y}}} \frac{1}{m}\sum_{i=1}^{m} (\ell_i(\hat{y}^i) + \langle \boldsymbol{\phi}_i(\hat{y}^i), \boldsymbol{w}\rangle)$$



Cutting plane algorithm

$$R^\psi(\boldsymbol{w}) = \frac{1}{m}\sum_{i=1}^m \max_{\hat{y}^i\in\mathcal{Y}}(\ell_i(\hat{y}^i) + \langle \boldsymbol{\phi}_i(\hat{y}^i), \boldsymbol{w}\rangle) = \max_{\substack{\hat{y}^1\in\mathcal{Y}\\ \vdots \\ \hat{y}^m\in\mathcal{Y}}} \frac{1}{m}\sum_{i=1}^m (\ell_i(\hat{y}^i) + \langle \boldsymbol{\phi}_i(\hat{y}^i), \boldsymbol{w}\rangle)$$



$$w_2 = \mathrm{argmin}_{w\in\mathcal{W}} R_1^\psi(w)$$

$$w_3 = \mathrm{argmin}_{w\in\mathcal{W}} R_2^\psi(w)$$

$$w_4 = \mathrm{argmin}_{w\in\mathcal{W}} R_3^\psi(w)$$

Cutting plane algorithm

1. $\boldsymbol{w}_1 \in \mathcal{W}$, $t \leftarrow 1$

2. Compute a new cutting plane and the objective value:

$$\boldsymbol{a}_t = \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{\phi}_i(\hat{y}^i)\,, \quad b_t = \frac{1}{m}\sum_{i=1}^{m}\ell_i(\hat{y}^i)\,, \quad R^{\psi}(\boldsymbol{w}_t) = b_t + \langle \boldsymbol{w}_t, \boldsymbol{a}_t\rangle$$

   where $\hat{y}^i$ is a solutions of loss augmented prediction problem:

$$\hat{y}^i = \operatorname*{argmax}_{y\in\mathcal{Y}}\big(\ell_i(y) + \langle \boldsymbol{w}, \boldsymbol{\phi}_i(y)\rangle\big) = \operatorname*{argmax}_{y\in\mathcal{Y}}\big(\ell(y^i, y) + \langle \boldsymbol{w}, \boldsymbol{\phi}(x^i, y)\rangle\big)$$

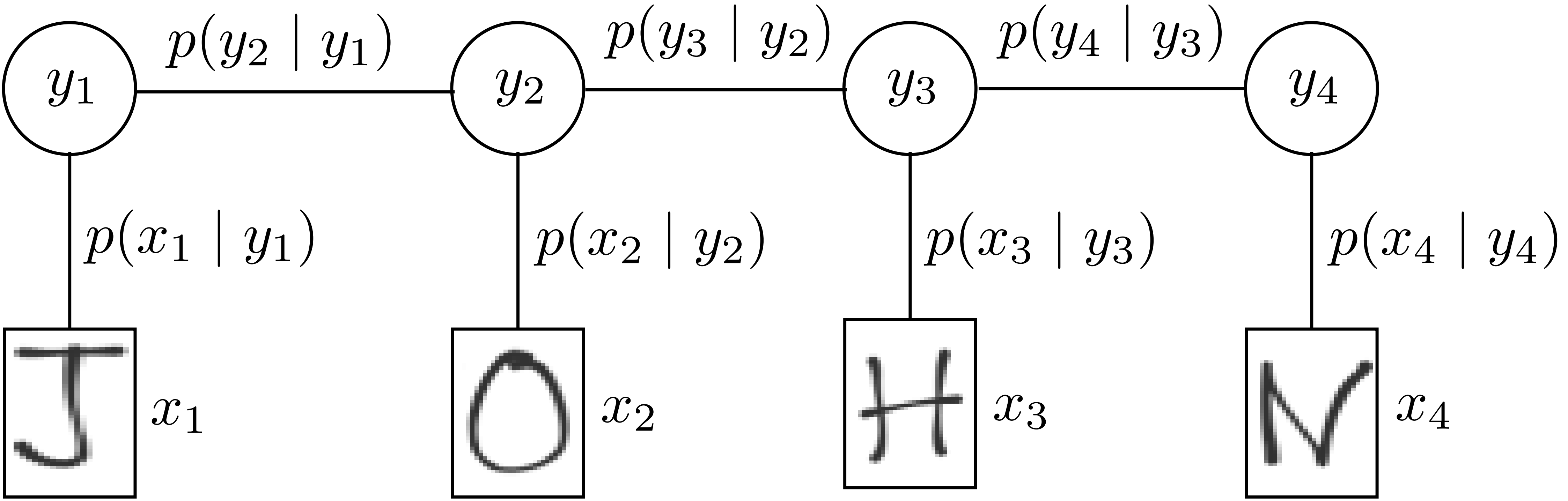
3. Solve a reduced problem

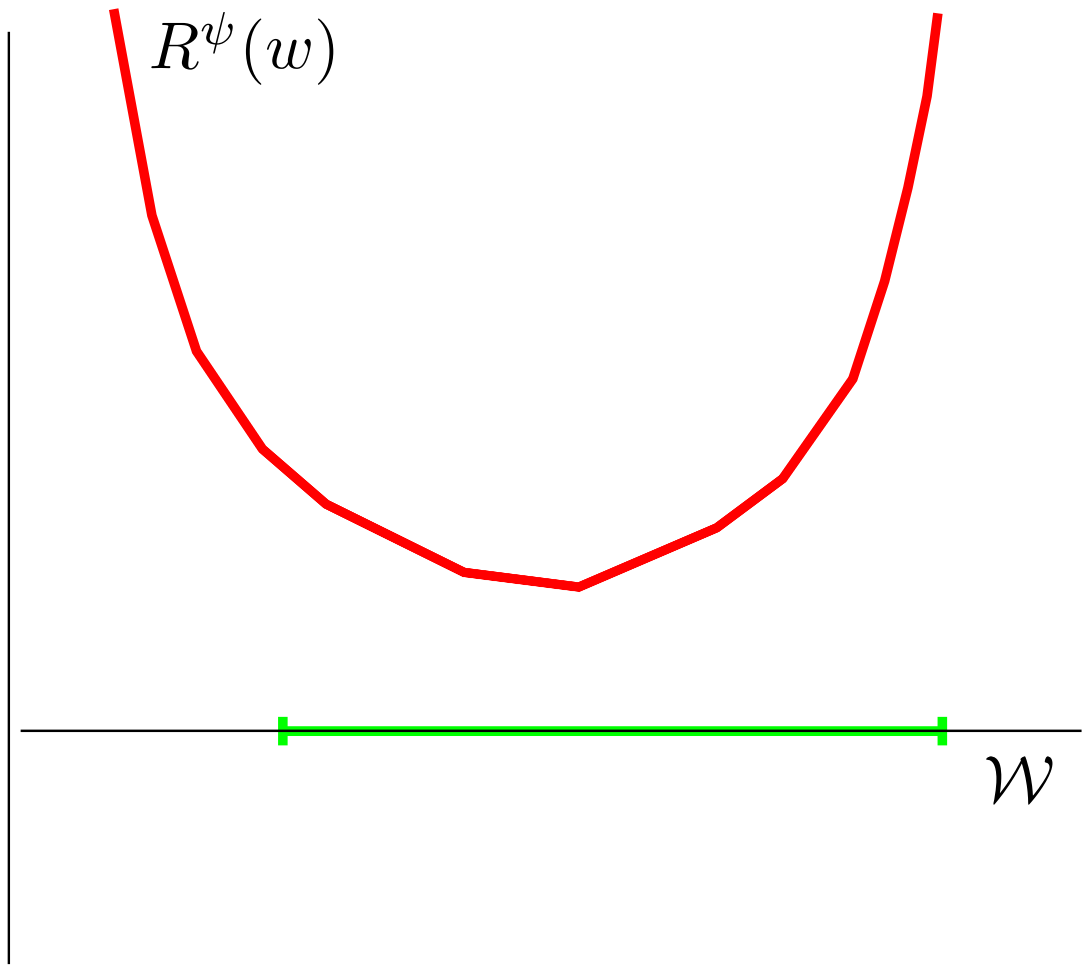
$$\boldsymbol{w}_{t+1} = \operatorname*{argmin}_{\boldsymbol{w}\in\mathcal{W}} R_t^{\psi}(\boldsymbol{w})\,, \quad \text{where} \quad R_t^{\psi}(\boldsymbol{w}) = \max_{i=1,\ldots,t}(b_i + \langle \boldsymbol{w}, \boldsymbol{a}_i\rangle)$$

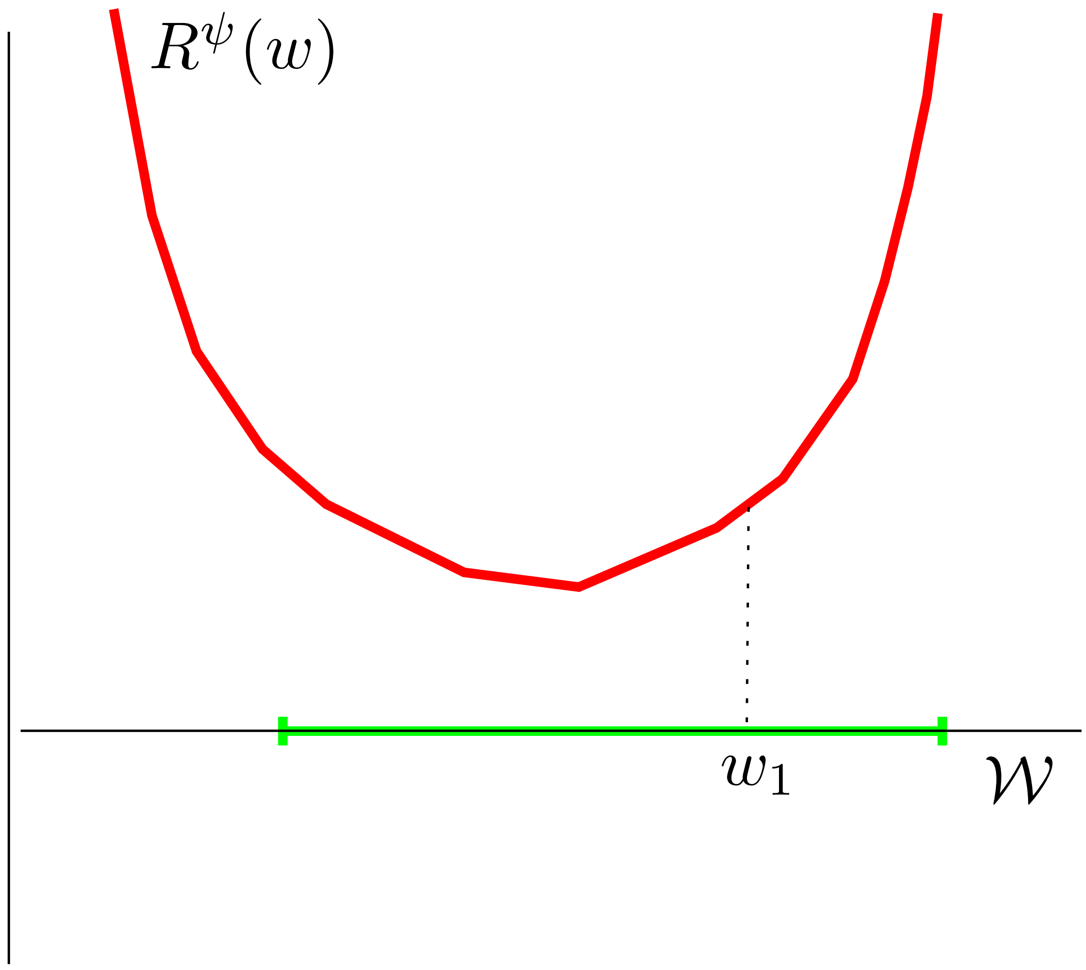
4. If $\min_{i=1,\ldots,t+1} R(\boldsymbol{w}_t) - R^{\psi}(\boldsymbol{w}_{t+1}) \leq \varepsilon$ exit else $t \leftarrow t+1$ and go to 2.

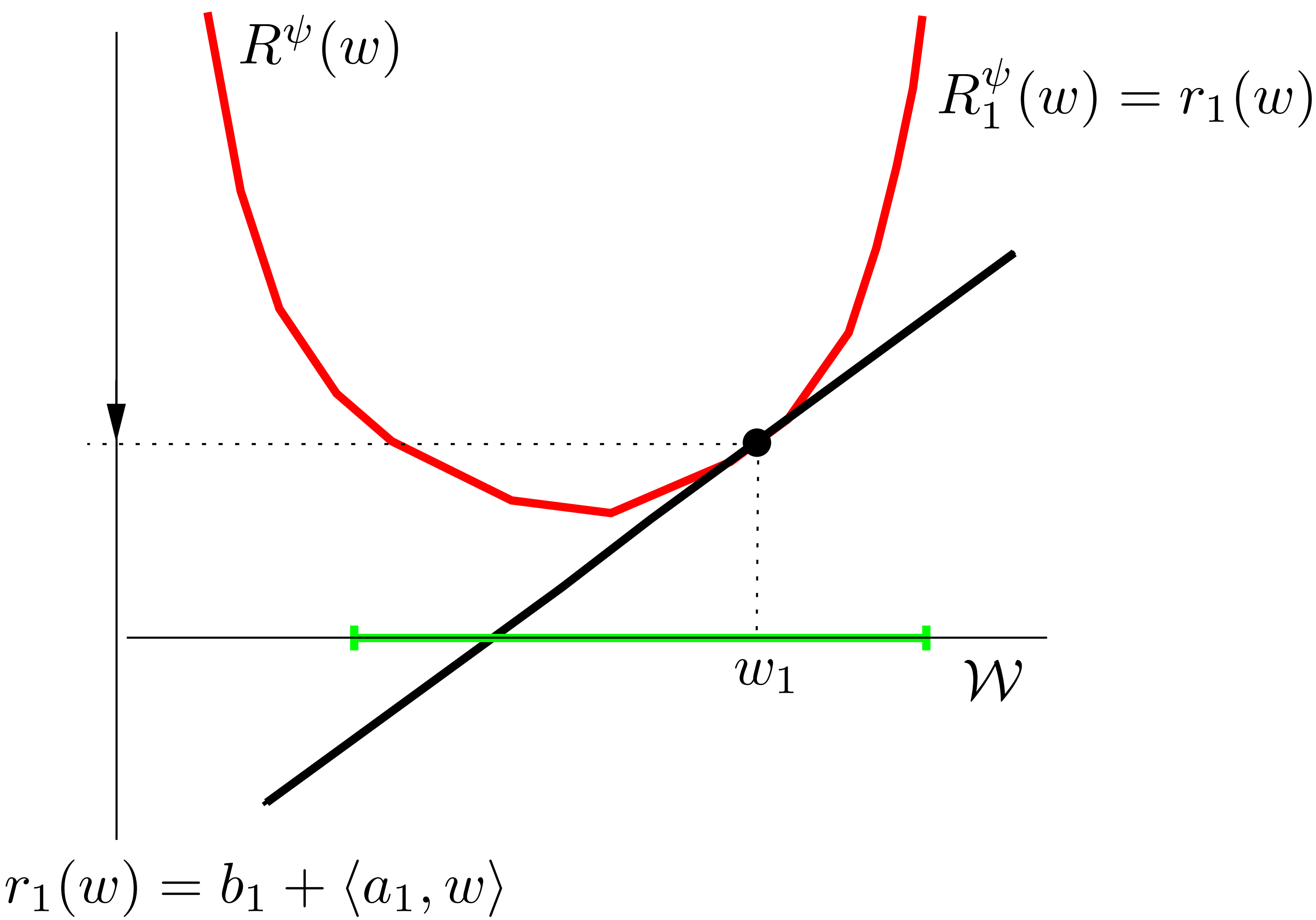
Summary

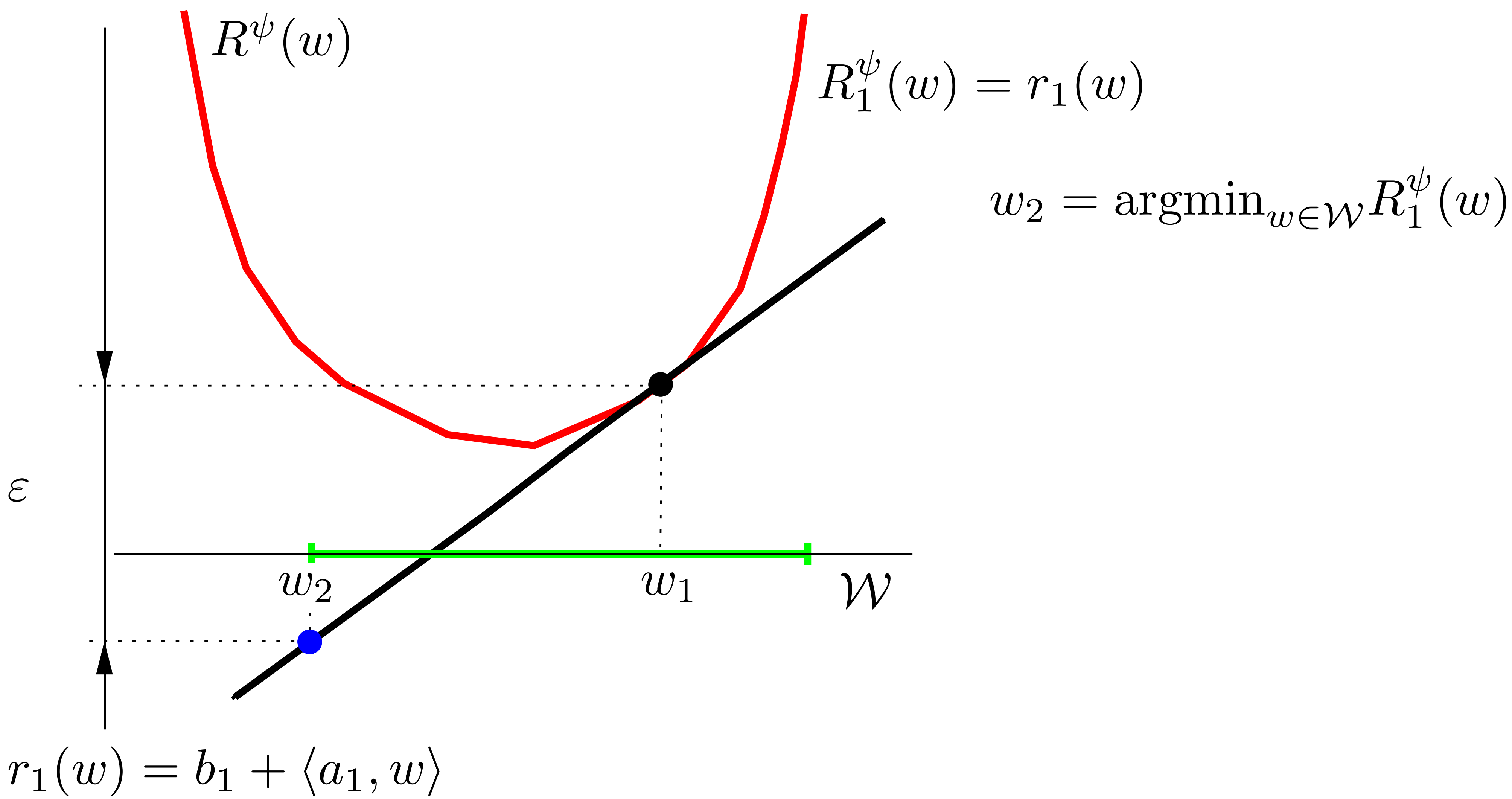
- Generalized linear classifier
- Structured Output Perceptron
- Structured Output Support Vector Machines
- Cutting Plane Algorithm


$p(y_2 \mid y_1)$
$p(y_3 \mid y_2)$
$p(y_4 \mid y_3)$
y_1
y_2
y_3
y_4
$p(x_1 \mid y_1)$
$p(x_2 \mid y_2)$
$p(x_3 \mid y_3)$
$p(x_4 \mid y_4)$
x_1
x_2
x_3
x_4

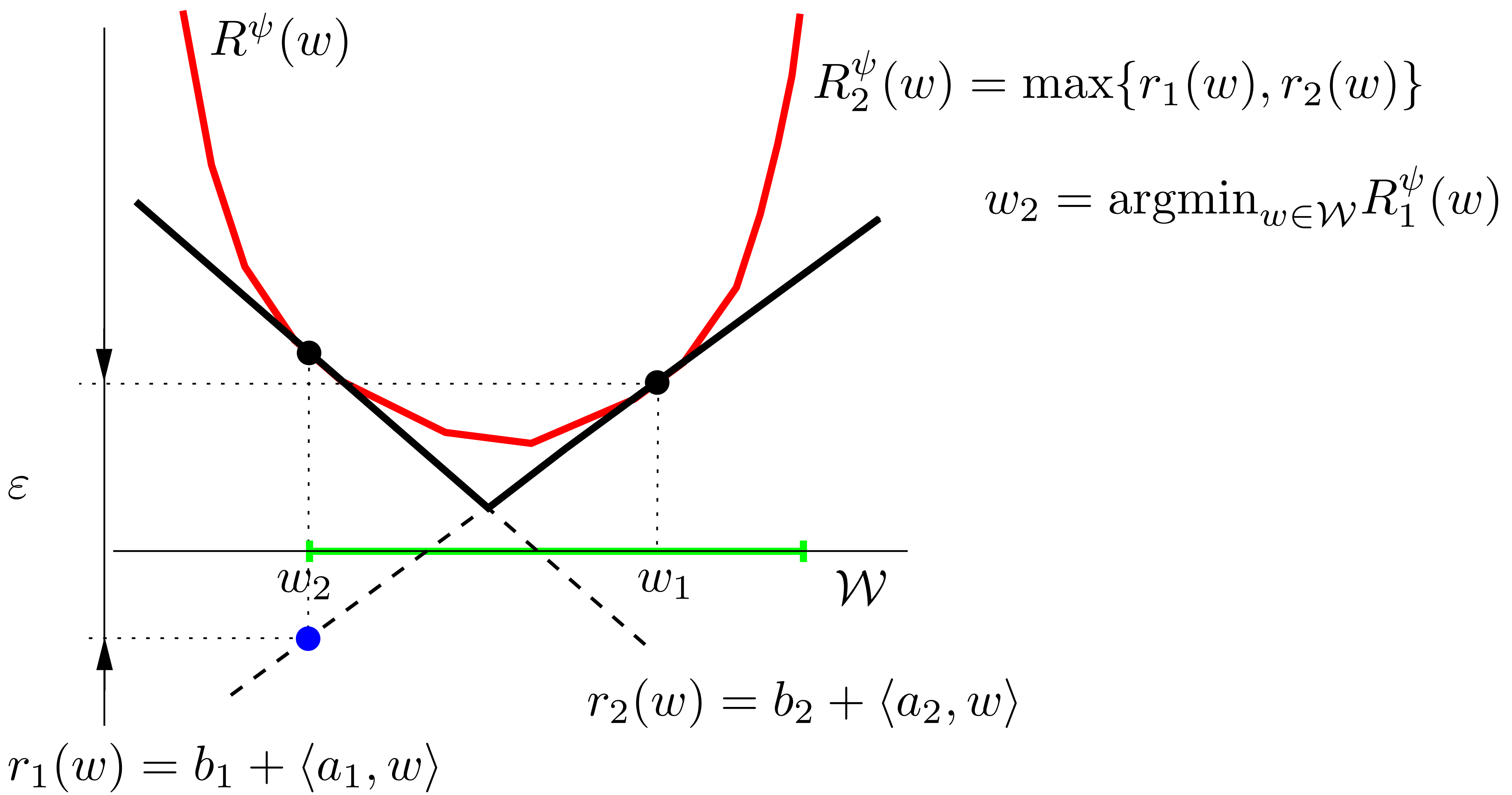

$R^{\psi}(w)$
$\mathcal{W}$

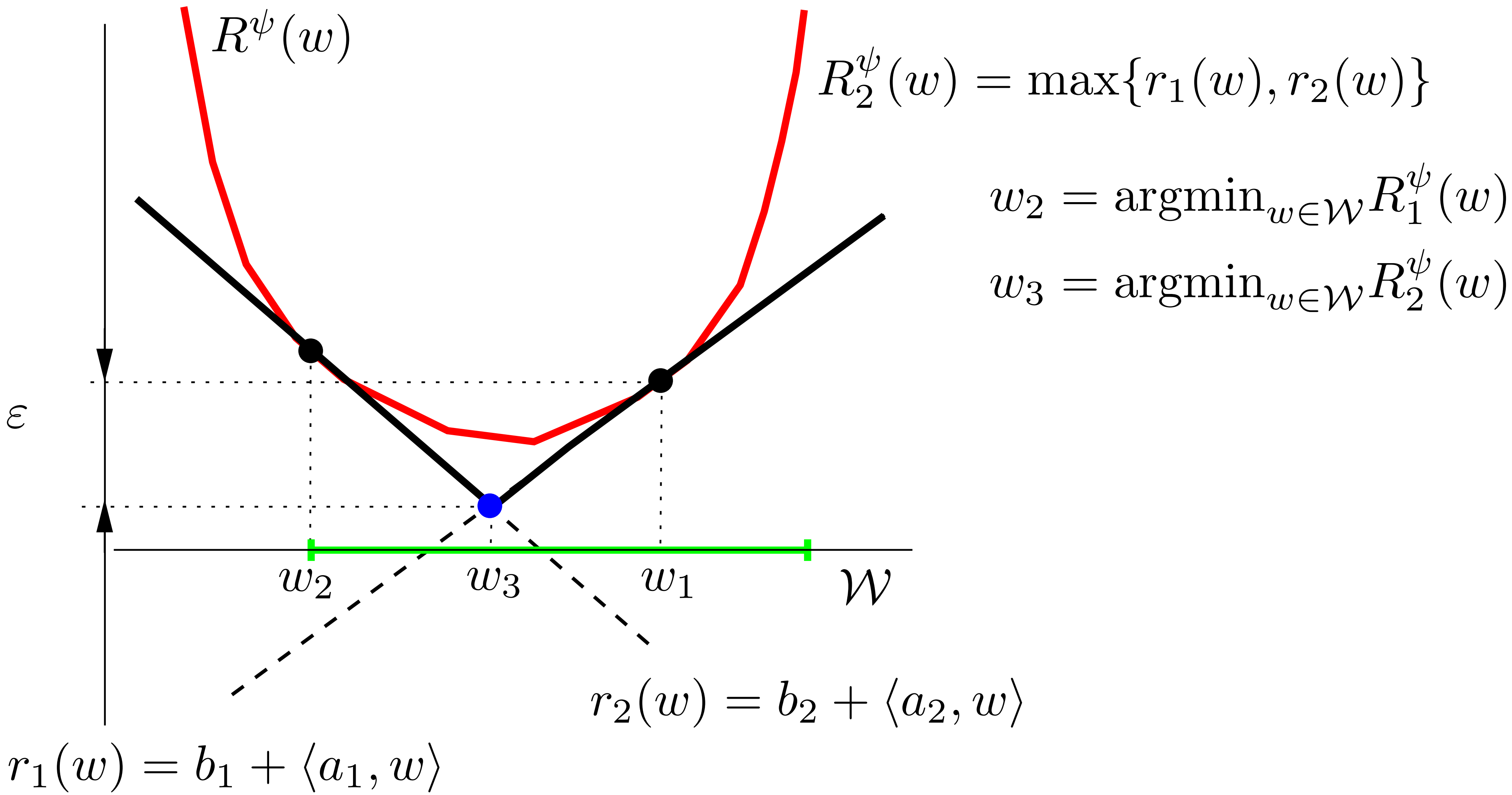

$R^{\psi}(w)$
w_1
$\mathcal{W}$


$R^{\psi}(w)$
$R_1^{\psi}(w) = r_1(w)$
w_1
$\mathcal{W}$
$r_1(w) = b_1 + \langle a_1, w \rangle$

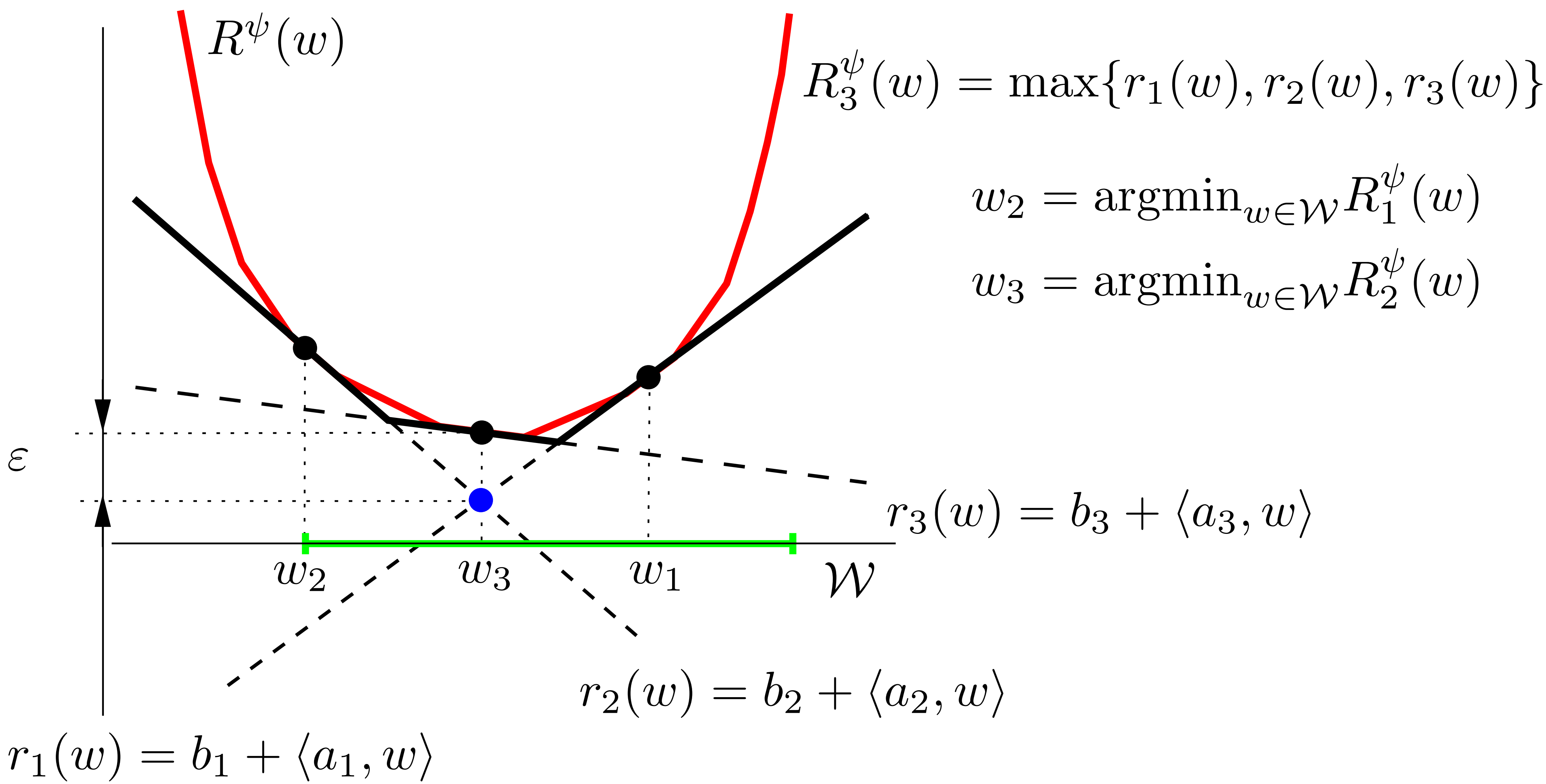


$R^{\psi}(w)$
$R_1^{\psi}(w) = r_1(w)$
$w_2 = \mathrm{argmin}_{w \in \mathcal{W}} R_1^{\psi}(w)$
ε
w_2
w_1
$\mathcal{W}$
$r_1(w) = b_1 + \langle a_1, w \rangle$

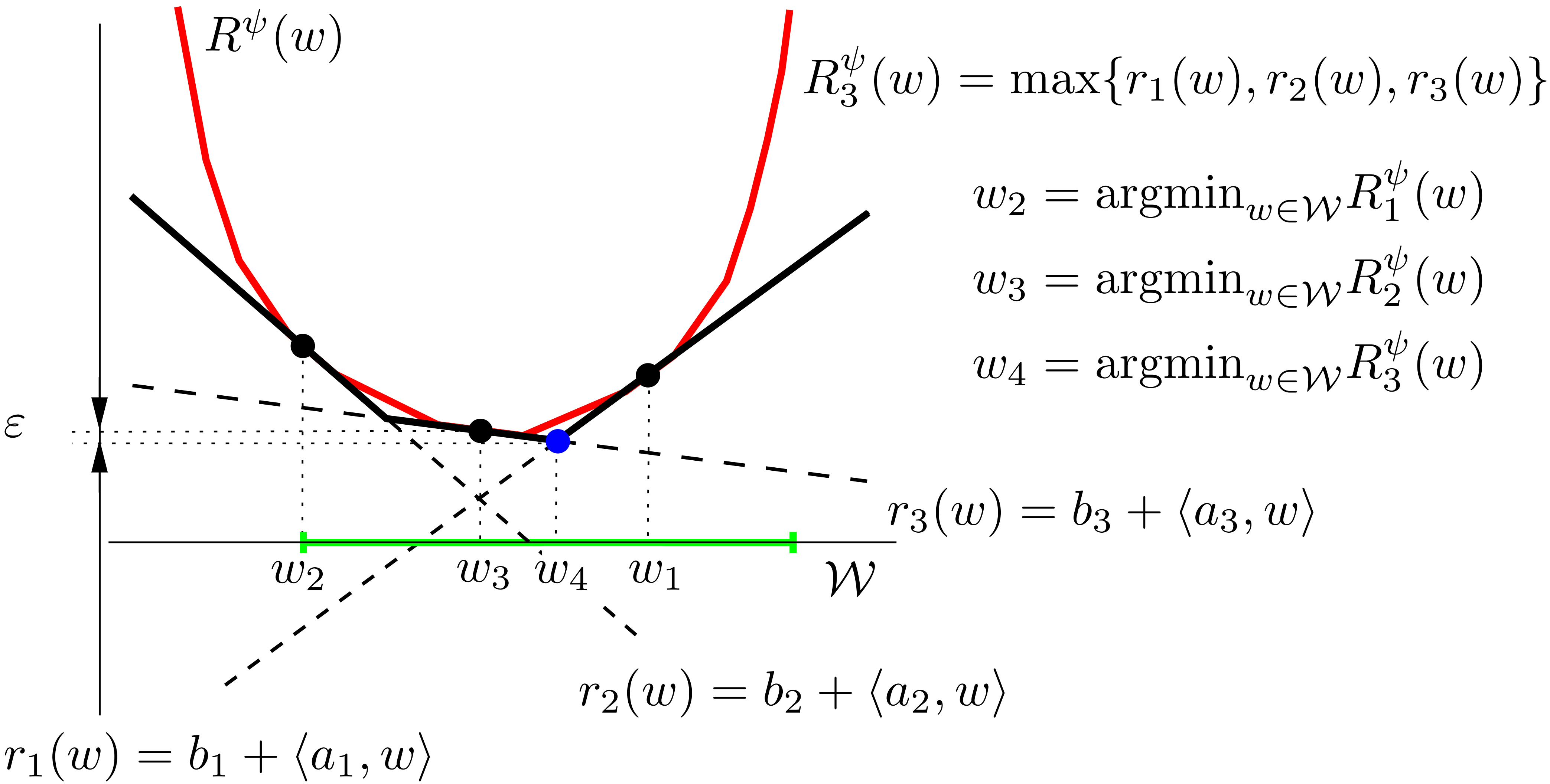

$R^{\psi}(w)$
$R_2^{\psi}(w) = \max\{r_1(w), r_2(w)\}$
$w_2 = \operatorname{argmin}_{w \in \mathcal{W}} R_1^{\psi}(w)$
ε
w_2
w_1
$\mathcal{W}$
$r_2(w) = b_2 + \langle a_2, w \rangle$
$r_1(w) = b_1 + \langle a_1, w \rangle$



$R^{\psi}(w)$
$R^{\psi}_2(w) = \max\{r_1(w), r_2(w)\}$
$w_2 = \mathrm{argmin}_{w \in \mathcal{W}} R^{\psi}_1(w)$
$w_3 = \mathrm{argmin}_{w \in \mathcal{W}} R^{\psi}_2(w)$
ε
w_2
w_3
w_1
$\mathcal{W}$
$r_2(w) = b_2 + \langle a_2, w \rangle$
$r_1(w) = b_1 + \langle a_1, w \rangle$



$R^{\psi}(w)$
$R_3^{\psi}(w) = \max\{r_1(w), r_2(w), r_3(w)\}$
$w_2 = \mathrm{argmin}_{w \in \mathcal{W}} R_1^{\psi}(w)$
$w_3 = \mathrm{argmin}_{w \in \mathcal{W}} R_2^{\psi}(w)$
ε
$r_3(w) = b_3 + \langle a_3, w \rangle$
w_2
w_3
w_1
$\mathcal{W}$
$r_2(w) = b_2 + \langle a_2, w \rangle$
$r_1(w) = b_1 + \langle a_1, w \rangle$


$R^\psi(w)$
$R_3^\psi(w) = \max\{r_1(w), r_2(w), r_3(w)\}$
$w_2 = \mathrm{argmin}_{w\in\mathcal{W}} R_1^\psi(w)$
$w_3 = \mathrm{argmin}_{w\in\mathcal{W}} R_2^\psi(w)$
$w_4 = \mathrm{argmin}_{w\in\mathcal{W}} R_3^\psi(w)$
ε
$r_3(w) = b_3 + \langle a_3, w\rangle$
w_2
w_3
w_4
w_1
$\mathcal{W}$
$r_2(w) = b_2 + \langle a_2, w\rangle$
$r_1(w) = b_1 + \langle a_1, w\rangle$