Shai Shalev-Shwartz

School of CS and Engineering,
The Hebrew University of Jerusalem

Features
Feature Selection

- How to represent real-world objects (e.g. Papaya) as a feature vector?

Even if we have a representation as a feature vector, maybe there's a "better" representation? What is "better"? depends on the hypothesis class:

Example: regression problem, \( x_1 \sim U[-1, 1], y = x_2^2 \), \( x_2 \sim U[y - 0.01, y + 0.01] \). Which feature is better, \( x_1 \) or \( x_2 \)?

If the hypothesis class is linear regressors, we should prefer \( x_2 \). If the hypothesis class is quadratic regressors, we should prefer \( x_1 \). No-free-lunch...
Feature Selection

- How to represent real-world objects (e.g. Papaya) as a feature vector?
- Even if we have a representation as a feature vector, maybe there’s a “better” representation?

What is “better”? depends on the hypothesis class:

Example: regression problem, $x_1 \sim U[-1,1]$, $y = x_2^1$, $x_2 \sim U[y-0.01, y+0.01]$. Which feature is better, $x_1$ or $x_2$?

If the hypothesis class is linear regressors, we should prefer $x_2$. If the hypothesis class is quadratic regressors, we should prefer $x_1$.

No-free-lunch ...
Feature Selection

- How to represent real-world objects (e.g. Papaya) as a feature vector?
- Even if we have a representation as a feature vector, maybe there’s a “better” representation?
- What is “better”? depends on the hypothesis class:
  Example: regression problem,
  \[ x_1 \sim U[-1,1], \quad y = x_1^2, \quad x_2 \sim U[y - 0.01, y + 0.01] \]

Which feature is better, \( x_1 \) or \( x_2 \)?
Feature Selection

- How to represent real-world objects (e.g. Papaya) as a feature vector?
- Even if we have a representation as a feature vector, maybe there’s a “better” representation?
- What is “better”? depends on the hypothesis class:

  Example: regression problem,

\[
x_1 \sim U[-1, 1], \quad y = x_1^2, \quad x_2 \sim U[y - 0.01, y + 0.01]
\]

Which feature is better, \(x_1\) or \(x_2\)?

- If the hypothesis class is linear regressors, we should prefer \(x_2\). If the hypothesis class is quadratic regressors, we should prefer \(x_1\).
Feature Selection

- How to represent real-world objects (e.g. Papaya) as a feature vector?
- Even if we have a representation as a feature vector, maybe there’s a “better” representation?
- What is “better”? depends on the hypothesis class:
  Example: regression problem,

\[ x_1 \sim U[-1, 1], \quad y = x_1^2, \quad x_2 \sim U[y - 0.01, y + 0.01] \]

Which feature is better, \( x_1 \) or \( x_2 \)?
- If the hypothesis class is linear regressors, we should prefer \( x_2 \). If the hypothesis class is quadratic regressors, we should prefer \( x_1 \).
- No-free-lunch ...
Outline

1. Feature Selection
   - Filters
   - Greedy selection
   - $\ell_1$ norm

2. Feature Manipulation and Normalization

3. Feature Learning
Feature Selection

- $\mathcal{X} = \mathbb{R}^d$
- We’d like to learn a predictor that only relies on $k \ll d$ features
- Why?
  - Can reduce estimation error
  - Reduces memory and runtime (both at train and test time)
  - Obtaining features may be costly (e.g. medical applications)
Feature Selection

- Optimal approach: try all subsets of $k$ out of $d$ features and choose the one which leads to best performing predictor.
Feature Selection

- Optimal approach: try all subsets of $k$ out of $d$ features and choose the one which leads to best performing predictor.
- Problem: runtime is $d^k$... can formally prove hardness in many situations.
Feature Selection

- Optimal approach: try all subsets of $k$ out of $d$ features and choose the one which leads to best performing predictor
- Problem: runtime is $d^k$... can formally prove hardness in many situations
- We describe three computationally efficient heuristics (some of them come with some types of formal guarantees, but this is beyond the scope)
Outline

1 Feature Selection
   • Filters
     • Greedy selection
     • $\ell_1$ norm

2 Feature Manipulation and Normalization

3 Feature Learning
Filter method: assess individual features, independently of other features, according to some quality measure, and select $k$ features with highest score
Filters

- **Filter method**: assess individual features, independently of other features, according to some quality measure, and select $k$ features with highest score
- **Score function**: Many possible score functions. E.g.:
Filter method: assess individual features, independently of other features, according to some quality measure, and select $k$ features with highest score.

Score function: Many possible score functions. E.g.:
- Minimize loss: Rank features according to
  \[- \min_{a,b \in \mathbb{R}} \sum_{i=1}^{m} \ell(a v_i + b, y_i)\]
Filter method: assess individual features, independently of other features, according to some quality measure, and select $k$ features with highest score

Score function: Many possible score functions. E.g.:

- Minimize loss: Rank features according to
  \[
  - \min_{a,b \in \mathbb{R}} \sum_{i=1}^{m} \ell(av_i + b, y_i)
  \]

- Pearson correlation coefficient: (obtained by minimizing squared loss)
  \[
  \frac{|\langle \mathbf{v} - \bar{v}, \mathbf{y} - \bar{y} \rangle|}{\| \mathbf{v} - \bar{v} \| \| \mathbf{y} - \bar{y} \|}
  \]
Filters

- **Filter method:** assess individual features, independently of other features, according to some quality measure, and select $k$ features with highest score

- **Score function:** Many possible score functions. E.g.:
  - **Minimize loss:** Rank features according to
    \[
    - \min_{a,b \in \mathbb{R}} \sum_{i=1}^{m} \ell( av_i + b, y_i )
    \]
  - **Pearson correlation coefficient:** (obtained by minimizing squared loss)
    \[
    \frac{|\langle v - \bar{v}, y - \bar{y} \rangle|}{\| v - \bar{v} \| \| y - \bar{y} \|}
    \]
  - **Spearman’s rho:** Apply Pearson’s coefficient on the ranking of $v$
Filters

- **Filter method:** assess individual features, independently of other features, according to some quality measure, and select \( k \) features with highest score
- **Score function:** Many possible score functions. E.g.:
  - **Minimize loss:** Rank features according to
    \[
    - \min_{a,b \in \mathbb{R}} \sum_{i=1}^{m} \ell(a v_i + b, y_i)
    \]
  - **Pearson correlation coefficient:** (obtained by minimizing squared loss)
    \[
    \frac{|\langle v - \bar{v}, y - \bar{y} \rangle|}{\| v - \bar{v} \| \| y - \bar{y} \|}
    \]
  - **Spearman’s rho:** Apply Pearson’s coefficient on the ranking of \( v \)
  - **Mutual information:** \( \sum p(v_i, y_i) \log(p(v_i, y_i)/(p(v_i)p(y_i))) \)
Weakness of Filters

- If Pearson’s coefficient is zero then \( v \) alone is useless for predicting \( y \)
Weakness of Filters

• If Pearson’s coefficient is zero then \( v \) alone is useless for predicting \( y \)
• This doesn’t mean that \( v \) is a bad feature — maybe with other features it is very useful
Weakness of Filters

- If Pearson’s coefficient is zero then $v$ alone is useless for predicting $y$.
- This doesn’t mean that $v$ is a bad feature — maybe with other features it is very useful.
- Example:

  $$ y = x_1 + 2x_2, \quad x_1 \sim U[\pm 1], \quad x_2 = (z - x_1)/2, \quad z \sim U[\pm 1] $$

  Then, Pearson of $x_1$ is zero, but no function can predict $y$ without $x_1$. 
Outline

1. **Feature Selection**
   - Filters
   - Greedy selection
   - $\ell_1$ norm

2. **Feature Manipulation and Normalization**

3. **Feature Learning**
Forward Greedy Selection

- Start with empty set of features $I = \emptyset$
Forward Greedy Selection

- Start with empty set of features $I = \emptyset$
- At each iteration, go over all $i \not\in I$ and learn a predictor based on $I \cup i$
Forward Greedy Selection

- Start with empty set of features $I = \emptyset$
- At each iteration, go over all $i \notin I$ and learn a predictor based on $I \cup i$
- Choose the $i$ that led to best predictor and update $I = I \cup \{i\}$
Forward Greedy Selection

- Start with empty set of features $I = \emptyset$
- At each iteration, go over all $i \not\in I$ and learn a predictor based on $I \cup i$
- Choose the $i$ that led to best predictor and update $I = I \cup \{i\}$
- Example: Orthogonal Matching Pursuit
Orthogonal Matching Pursuit (OMP)

- Let $X \in \mathbb{R}^{m,d}$ be a data matrix (instances in rows). Let $y \in \mathbb{R}^m$ be the targets vector.
Orthogonal Matching Pursuit (OMP)

- Let $X \in \mathbb{R}^{m,d}$ be a data matrix (instances in rows). Let $y \in \mathbb{R}^{m}$ be the targets vector.
- Let $X_{i}$ denote the $i$'th column of $X$ and let $X_{I}$ be the matrix whose columns are $\{X_{i} : i \in I\}$. 

An efficient implementation: let $V_{t}$ be a matrix whose columns are an orthonormal basis of the columns of $X_{I_{t}}$. Clearly, $\min_{w} \|X_{I_{t}}w - y\|_{2} = \min_{\theta} \|V_{t}\theta - y\|_{2}$.

Shai Shalev-Shwartz (Hebrew U)
Orthogonal Matching Pursuit (OMP)

- Let $X \in \mathbb{R}^{m,d}$ be a data matrix (instances in rows). Let $y \in \mathbb{R}^{m}$ be the targets vector.
- Let $X_i$ denote the $i$'th column of $X$ and let $X_I$ be the matrix whose columns are $\{X_i : i \in I\}$.
- At iteration $t$, we add the feature

$$j_t = \arg\min_j \min_{w \in \mathbb{R}^t} \|X_{I_{t-1} \cup \{j\}} w - y\|^2.$$
Orthogonal Matching Pursuit (OMP)

- Let $X \in \mathbb{R}^{m,d}$ be a data matrix (instances in rows). Let $y \in \mathbb{R}^m$ be the targets vector.
- Let $X_i$ denote the $i$’th column of $X$ and let $X_I$ be the matrix whose columns are $\{X_i : i \in I\}$.
- At iteration $t$, we add the feature

$$ j_t = \arg\min_j \min_{w \in \mathbb{R}^t} \|X_{I_{t-1}\cup\{j\}}w - y\|^2. $$

- An efficient implementation: let $V_t$ be a matrix whose columns are orthonormal basis of the columns of $X_{I_t}$. Clearly,

$$ \min_{w} \|X_{I_t}w - y\|^2 = \min_{\theta \in \mathbb{R}^t} \|V_t\theta - y\|^2. $$
Orthogonal Matching Pursuit (OMP)

- Let $X \in \mathbb{R}^{m,d}$ be a data matrix (instances in rows). Let $y \in \mathbb{R}^m$ be the targets vector.
- Let $X_i$ denote the $i'$th column of $X$ and let $X_I$ be the matrix whose columns are $\{X_i : i \in I\}$.
- At iteration $t$, we add the feature

$$j_t = \arg\min_j \min_{w \in \mathbb{R}^t} \|X_{I_{t-1} \cup \{j\}} w - y\|^2.$$

- An efficient implementation: let $V_t$ be a matrix whose columns are orthonormal basis of the columns of $X_{I_t}$. Clearly,

$$\min_w \|X_{I_t} w - y\|^2 = \min_{\theta \in \mathbb{R}^t} \|V_t \theta - y\|^2.$$

- Let $\theta_t$ be a minimizer of the right-hand side
Orthogonal Matching Pursuit (OMP)

- Given $V_{t-1}$ and $\theta_{t-1}$, we write for every $j$, $X_j = V_{t-1}V_{t-1}^\top X_j + u_j$, where $u_j$ is orthogonal to $V_j$. Then:

$$\min_{\theta, \alpha} \| V_{t-1} \theta + \alpha u_j - y \|_2^2 = \min_{\theta, \alpha} \left[ \| V_{t-1} \theta - y \|_2^2 + \alpha^2 \| u_j \|_2^2 + 2 \alpha \langle u_j, V_{t-1} \theta - y \rangle \right] = \min_{\theta} \| V_{t-1} \theta - y \|_2^2 + \min_{\alpha} \left[ \alpha^2 \| u_j \|_2^2 - 2 \alpha \langle u_j, y \rangle \right] = \| V_{t-1} \theta - y \|_2^2 - \left( \langle u_j, y \rangle \right)^2 \| u_j \|_2^2.$$
Orthogonal Matching Pursuit (OMP)

- Given $V_{t-1}$ and $\theta_{t-1}$, we write for every $j$, $X_j = V_{t-1}V_{t-1}^T X_j + u_j$, where $u_j$ is orthogonal to $V_j$. Then:
  \[
  \min_{\theta, \alpha} \|V_{t-1}\theta + \alpha u_j - y\|^2
  \]

Shai Shalev-Shwartz (Hebrew U)
IML Lecture 13
Features 12 / 28
Orthogonal Matching Pursuit (OMP)

- Given \( V_{t-1} \) and \( \theta_{t-1} \), we write for every \( j \), \( X_j = V_{t-1} V_{t-1}^\top X_j + u_j \), where \( u_j \) is orthogonal to \( V_j \). Then:

\[
\min_{\theta,\alpha} \| V_{t-1} \theta + \alpha u_j - y \|^2
\]

\[
= \min_{\theta,\alpha} \left[ \| V_{t-1} \theta - y \|^2 + \alpha^2 \| u_j \|^2 + 2\alpha \langle u_j, V_{t-1} \theta - y \rangle \right]
\]
Orthogonal Matching Pursuit (OMP)

Given $V_{t-1}$ and $\theta_{t-1}$, we write for every $j$, $X_j = V_{t-1}V_{t-1}^T X_j + u_j$, where $u_j$ is orthogonal to $V_j$. Then:

$$
\min_{\theta, \alpha} \|V_{t-1}\theta + \alpha u_j - y\|^2
$$

$$
= \min_{\theta, \alpha} \left[ \|V_{t-1}\theta - y\|^2 + \alpha^2 \|u_j\|^2 + 2\alpha \langle u_j, V_{t-1}\theta - y \rangle \right]
$$

$$
= \min_{\theta, \alpha} \left[ \|V_{t-1}\theta - y\|^2 + \alpha^2 \|u_j\|^2 + 2\alpha \langle u_j, -y \rangle \right]
$$
Orthogonal Matching Pursuit (OMP)

- Given $V_{t-1}$ and $\theta_{t-1}$, we write for every $j$, $X_j = V_{t-1}V_{t-1}^T X_j + u_j$, where $u_j$ is orthogonal to $V_j$. Then:

$$\min_{\theta, \alpha} \| V_{t-1} \theta + \alpha u_j - y \|^2$$

$$= \min_{\theta, \alpha} \left[ \| V_{t-1} \theta - y \|^2 + \alpha^2 \| u_j \|^2 + 2\alpha \langle u_j, V_{t-1} \theta - y \rangle \right]$$

$$= \min_{\theta, \alpha} \left[ \| V_{t-1} \theta - y \|^2 + \alpha^2 \| u_j \|^2 + 2\alpha \langle u_j, -y \rangle \right]$$

$$= \min_{\theta} \left[ \| V_{t-1} \theta - y \|^2 \right] + \min_{\alpha} \left[ \alpha^2 \| u_j \|^2 - 2\alpha \langle u_j, y \rangle \right]$$
Orthogonal Matching Pursuit (OMP)

- Given $V_{t-1}$ and $\theta_{t-1}$, we write for every $j$, $X_j = V_{t-1} V_{t-1}^\top X_j + u_j$, where $u_j$ is orthogonal to $V_j$. Then:

$$\begin{align*}
\min_{\theta, \alpha} \| V_{t-1} \theta + \alpha u_j - y \|^2 \\
= \min_{\theta, \alpha} \left[ \| V_{t-1} \theta - y \|^2 + \alpha^2 \| u_j \|^2 + 2\alpha \langle u_j, V_{t-1} \theta - y \rangle \right] \\
= \min_{\theta, \alpha} \left[ \| V_{t-1} \theta - y \|^2 + \alpha^2 \| u_j \|^2 + 2\alpha \langle u_j, -y \rangle \right] \\
= \min_{\theta} \left[ \| V_{t-1} \theta - y \|^2 \right] + \min_{\alpha} \left[ \alpha^2 \| u_j \|^2 - 2\alpha \langle u_j, y \rangle \right] \\
= \left[ \| V_{t-1} \theta_{t-1} - y \|^2 \right] + \min_{\alpha} \left[ \alpha^2 \| u_j \|^2 - 2\alpha \langle u_j, y \rangle \right]
\end{align*}$$
Orthogonal Matching Pursuit (OMP)

- Given $V_{t-1}$ and $\theta_{t-1}$, we write for every $j$, $X_j = V_{t-1} V_{t-1}^\top x_j + u_j$, where $u_j$ is orthogonal to $V_j$. Then:

$$
\min_{\theta,\alpha} \| V_{t-1} \theta + \alpha u_j - y \|^2 \\
= \min_{\theta,\alpha} \left[ \| V_{t-1} \theta - y \|^2 + \alpha^2 \| u_j \|^2 + 2\alpha \langle u_j, V_{t-1} \theta - y \rangle \right] \\
= \min_{\theta,\alpha} \left[ \| V_{t-1} \theta - y \|^2 + \alpha^2 \| u_j \|^2 + 2\alpha \langle u_j, -y \rangle \right] \\
= \min_{\theta} \left[ \| V_{t-1} \theta - y \|^2 \right] + \min_{\alpha} \left[ \alpha^2 \| u_j \|^2 - 2\alpha \langle u_j, y \rangle \right] \\
= \left[ \| V_{t-1} \theta_{t-1} - y \|^2 \right] + \min_{\alpha} \left[ \alpha^2 \| u_j \|^2 - 2\alpha \langle u_j, y \rangle \right] \\
= \| V_{t-1} \theta_{t-1} - y \|^2 - \frac{\langle u_j, y \rangle^2}{\| u_j \|^2}
$$
Orthogonal Matching Pursuit (OMP)

Given $V_{t-1}$ and $\theta_{t-1}$, we write for every $j$, $X_j = V_{t-1}V_{t-1}^\top X_j + u_j$, where $u_j$ is orthogonal to $V_j$. Then:

$$\min_{\theta, \alpha} \| V_{t-1} \theta + \alpha u_j - y \|^2$$

$$= \min_{\theta, \alpha} \left[ \| V_{t-1} \theta - y \|^2 + \alpha^2 \| u_j \|^2 + 2\alpha \langle u_j, V_{t-1} \theta - y \rangle \right]$$

$$= \min_{\theta, \alpha} \left[ \| V_{t-1} \theta - y \|^2 + \alpha^2 \| u_j \|^2 + 2\alpha \langle u_j, -y \rangle \right]$$

$$= \min_{\theta} \left[ \| V_{t-1} \theta - y \|^2 \right] + \min_{\alpha} \left[ \alpha^2 \| u_j \|^2 - 2\alpha \langle u_j, y \rangle \right]$$

$$= \left[ \| V_{t-1} \theta_{t-1} - y \|^2 \right] + \min_{\alpha} \left[ \alpha^2 \| u_j \|^2 - 2\alpha \langle u_j, y \rangle \right]$$

$$= \| V_{t-1} \theta_{t-1} - y \|^2 - \frac{(\langle u_j, y \rangle)^2}{\| u_j \|^2}$$

It follows that we should select the feature $j_t = \arg\max_j \frac{(\langle u_j, y \rangle)^2}{\| u_j \|^2}$.
Orthogonal Matching Pursuit (OMP)

input:
- data matrix $X \in \mathbb{R}^{m,d}$, labels vector $y \in \mathbb{R}^m$,
- budget of features $T$

initialize: $I_1 = \emptyset$

for $t = 1, \ldots, T$

- use SVD to find an orthonormal basis $V \in \mathbb{R}^{m,t-1}$ of $X_{I_t}$
  (for $t = 1$ set $V$ to be the all zeros matrix)

foreach $j \in [d] \setminus I_t$ let $u_j = X_j - V V^\top X_j$

let $j_t = \arg\max_{j \notin I_t: \|u_j\| > 0} \frac{(\langle u_j, y \rangle)^2}{\|u_j\|^2}$

update $I_{t+1} = I_t \cup \{j_t\}$

output $I_{T+1}$
Gradient-based Greedy Selection

- Let $R(w)$ be the empirical risk as a function of $w$
Gradient-based Greedy Selection

- Let $R(w)$ be the empirical risk as a function of $w$
- For the squared loss, $R(w) = \frac{1}{m} \|Xw - y\|^2$, we can easily solve the problem

$$\arg\min_j \min_{w: \text{supp}(w) = I \cup \{i\}} R(w)$$
Gradient-based Greedy Selection

- Let $R(w)$ be the empirical risk as a function of $w$
- For the squared loss, $R(w) = \frac{1}{m} \|Xw - y\|^2$, we can easily solve the problem

$$\min_j \min_{w: \text{supp}(w) = I \cup \{i\}} R(w)$$

- For general $R$, this may be expensive. An approximation is to only optimize $w$ over the new feature:

$$\min_j \min_{\eta \in \mathbb{R}} R(w + \eta e_j)$$
Gradient-based Greedy Selection

- Let $R(w)$ be the empirical risk as a function of $w$.
- For the squared loss, $R(w) = \frac{1}{m} \|Xw - y\|^2$, we can easily solve the problem:
  \[
  \arg\min_j \min_{w: \text{supp}(w) = I \cup \{i\}} R(w)
  \]
- For general $R$, this may be expensive. An approximation is to only optimize $w$ over the new feature:
  \[
  \arg\min_j \min_{\eta \in \mathbb{R}} R(w + \eta e_j)
  \]
- An even simpler approach is to choose the feature which minimizes the above for infinitesimal $\eta$, namely,
  \[
  \arg\min_j |\nabla_j R(w)|
  \]
It is possible to show (left as an exercise), that the AdaBoost algorithm is in fact Forward Greedy Selection for the objective function

\[ R(w) = \log \left( \sum_{i=1}^{m} \exp \left( -y_i \sum_{j=1}^{d} w_j h_j(x_j) \right) \right). \]
Outline

1 Feature Selection
   - Filters
   - Greedy selection
     - $\ell_1$ norm

2 Feature Manipulation and Normalization

3 Feature Learning
Minimizing the empirical risk subject to a budget of $k$ features can be written as:

$$\min_{\mathbf{w}} L_S(\mathbf{w}) \quad \text{s.t.} \quad \|\mathbf{w}\|_0 \leq k,$$

Replace the non-convex constraint, $\|\mathbf{w}\|_0 \leq k$, with a convex constraint, $\|\mathbf{w}\|_1 \leq k$. Why $\ell_1$?

"Closest" convex surrogate

If $\|\mathbf{w}\|_1$ is small, can construct $\tilde{\mathbf{w}}$ with $\|\tilde{\mathbf{w}}\|_0$ small and similar value of $L_S$. Often, $\ell_1$ "induces" sparse solutions.
Minimizing the empirical risk subject to a budget of \( k \) features can be written as:

\[
\min_w L_S(w) \quad \text{s.t.} \quad \|w\|_0 \leq k,
\]

Replace the non-convex constraint, \( \|w\|_0 \leq k \), with a convex constraint, \( \|w\|_1 \leq k_1 \).

Why \( \ell_1 \)?

"Closest" convex surrogate

If \( \|w\|_1 \) is small, can construct \( \tilde{w} \) with \( \|\tilde{w}\|_0 \) small and similar value of \( L_S \). Often, \( \ell_1 \) "induces" sparse solutions.
Minimizing the empirical risk subject to a budget of \( k \) features can be written as:

\[
\min_w L_S(w) \quad \text{s.t.} \quad \|w\|_0 \leq k,
\]

Replace the non-convex constraint, \( \|w\|_0 \leq k \), with a convex constraint, \( \|w\|_1 \leq k_1 \).

Why \( \ell_1 \)?
Minimizing the empirical risk subject to a budget of $k$ features can be written as:

$$\min_{\mathbf{w}} L_S(\mathbf{w}) \quad \text{s.t.} \quad \|\mathbf{w}\|_0 \leq k,$$

Replace the non-convex constraint, $\|\mathbf{w}\|_0 \leq k$, with a convex constraint, $\|\mathbf{w}\|_1 \leq k_1$.

Why $\ell_1$?
- “Closest” convex surrogate
Minimizing the empirical risk subject to a budget of \( k \) features can be written as:

\[
\min_w L_S(w) \quad \text{s.t.} \quad \|w\|_0 \leq k,
\]

Replace the non-convex constraint, \( \|w\|_0 \leq k \), with a convex constraint, \( \|w\|_1 \leq k_1 \).

Why \( \ell_1 \)?

“Closest” convex surrogate

If \( \|w\|_1 \) is small, can construct \( \tilde{w} \) with \( \|\tilde{w}\|_0 \) small and similar value of \( L_S \)
Sparsity Inducing Norms

- Minimizing the empirical risk subject to a budget of \( k \) features can be written as:

\[
\min_w L_S(w) \quad \text{s.t.} \quad \|w\|_0 \leq k,
\]

- Replace the non-convex constraint, \( \|w\|_0 \leq k \), with a convex constraint, \( \|w\|_1 \leq k_1 \).

- Why \( \ell_1 \)?
  - “Closest” convex surrogate
  - If \( \|w\|_1 \) is small, can construct \( \tilde{w} \) with \( \|\tilde{w}\|_0 \) small and similar value of \( L_S \)
  - Often, \( \ell_1 \) “induces” sparse solutions
Instead of constraining $\|w\|_1$ we can regularize:

$$\min_w (L_S(w) + \lambda \|w\|_1)$$

For Squared-Loss this is the Lasso method. $\ell_1$ norm often induces sparse solutions. Example:

$$\min_{w \in \mathbb{R}} \left( \frac{1}{2} w^2 - xw + \lambda |w| \right)$$

Easy to verify that the solution is "soft thresholding" $w = \text{sign}(x) \left( |x| - \lambda \right) +$.

Sparsity: $w = 0$ unless $|x| > \lambda$. 

Shai Shalev-Shwartz (Hebrew U)
\( \ell_1 \) regularization

- Instead of constraining \( \|w\|_1 \) we can regularize:
  \[
  \min_w (L_S(w) + \lambda \|w\|_1)
  \]

- For Squared-Loss this is the Lasso method
$\ell_1$ regularization

- Instead of constraining $\|w\|_1$ we can regularize:

$$\min_w (L_S(w) + \lambda \|w\|_1)$$

- For Squared-Loss this is the Lasso method
- $\ell_1$ norm often induces sparse solutions. Example:

$$\min_{w \in \mathbb{R}} \left( \frac{1}{2} w^2 - xw + \lambda |w| \right) .$$
\( \ell_1 \) regularization

- Instead of constraining \( \|w\|_1 \) we can regularize:
  \[
  \min_w (L_S(w) + \lambda \|w\|_1)
  \]

- For Squared-Loss this is the Lasso method
- \( \ell_1 \) norm often induces sparse solutions. Example:
  \[
  \min_{w \in \mathbb{R}} \left( \frac{1}{2} w^2 - xw + \lambda |w| \right).
  \]

- Easy to verify that the solution is “soft thresholding”
  \[
  w = \text{sign}(x) \left[ |x| - \lambda \right]_+
  \]
Instead of constraining $\|w\|_1$ we can regularize:

$$\min_w (L_S(w) + \lambda\|w\|_1)$$

For Squared-Loss this is the Lasso method. \ell_1 norm often induces sparse solutions. Example:

$$\min_{w \in \mathbb{R}} \left( \frac{1}{2}w^2 - xw + \lambda|w| \right).$$

East to verify that the solution is “soft thresholding”

$$w = \text{sign}(x) [x - \lambda]_+$$

Sparsity: $w = 0$ unless $|x| > \lambda$
\( \ell_1 \) regularization

**One dimensional Lasso:**

\[
\argmin_{w \in \mathbb{R}^m} \left( \frac{1}{2m} \sum_{i=1}^{m} (x_i w - y_i)^2 + \lambda |w| \right).
\]
**$\ell_1$ regularization**

- **One dimensional Lasso:**

$$\operatorname*{argmin}_{w \in \mathbb{R}^m} \left( \frac{1}{2m} \sum_{i=1}^{m} (x_i w - y_i)^2 + \lambda |w| \right).$$

- **Rewrite:**

$$\operatorname*{argmin}_{w \in \mathbb{R}^m} \left( \frac{1}{2} \left( \frac{1}{m} \sum_{i} x_i^2 \right) w^2 - \left( \frac{1}{m} \sum_{i=1}^{m} x_i y_i \right) w + \lambda |w| \right).$$

Assume $\frac{1}{m} \sum_{i} x_i^2 = 1$, and denote $\langle x, y \rangle = \sum_{i} x_i y_i$, then the optimal solution is

$$w = \text{sign}(\langle x, y \rangle) \left( \frac{|\langle x, y \rangle|}{m} - \lambda \right) + w.$$
One dimensional Lasso:

$$\text{argmin}_{w \in \mathbb{R}^m} \left( \frac{1}{2m} \sum_{i=1}^{m} (x_i w - y_i)^2 + \lambda |w| \right) .$$

Rewrite:

$$\text{argmin}_{w \in \mathbb{R}^m} \left( \frac{1}{2} \left( \frac{1}{m} \sum_i x_i^2 \right) w^2 - \left( \frac{1}{m} \sum_{i=1}^{m} x_i y_i \right) w + \lambda |w| \right) .$$

Assume $\frac{1}{m} \sum_i x_i^2 = 1$, and denote $\langle x, y \rangle = \sum_{i=1}^{m} x_i y_i$, then the optimal solution is

$$w = \text{sign}(\langle x, y \rangle) \left[ ||\langle x, y \rangle||/m - \lambda \right]_+ .$$

Sparsity: $w = 0$ unless the correlation between $x$ and $y$ is larger than $\lambda$. 

Exercise: Show that the $\ell_2$ norm doesn't induce a sparse solution for this case.
One dimensional Lasso:

$$\text{argmin}_{w \in \mathbb{R}^m} \left( \frac{1}{2m} \sum_{i=1}^{m} (x_i w - y_i)^2 + \lambda |w| \right).$$

Rewrite:

$$\text{argmin}_{w \in \mathbb{R}^m} \left( \frac{1}{2} \left( \frac{1}{m} \sum_i x_i^2 \right) w^2 - \left( \frac{1}{m} \sum_{i=1}^{m} x_i y_i \right) w + \lambda |w| \right).$$

Assume $\frac{1}{m} \sum_i x_i^2 = 1$, and denote $\langle x, y \rangle = \sum_{i=1}^{m} x_i y_i$, then the optimal solution is

$$w = \text{sign}(\langle x, y \rangle) \left[ ||\langle x, y \rangle||/m - \lambda \right]_+.$$

Sparsity: $w = 0$ unless the correlation between $x$ and $y$ is larger than $\lambda$. 
\( l_1 \) regularization

- **One dimensional Lasso:**

\[
\text{argmin}_{w \in \mathbb{R}^m} \left( \frac{1}{2m} \sum_{i=1}^{m} (x_iw - y_i)^2 + \lambda |w| \right).
\]

- **Rewrite:**

\[
\text{argmin}_{w \in \mathbb{R}^m} \left( \frac{1}{2} \left( \frac{1}{m} \sum_i x_i^2 \right) w^2 - \left( \frac{1}{m} \sum_{i=1}^{m} x_iy_i \right) w + \lambda |w| \right).
\]

- Assume \( \frac{1}{m} \sum_i x_i^2 = 1 \), and denote \( \langle x, y \rangle = \sum_{i=1}^{m} x_iy_i \), then the optimal solution is

\[
w = \text{sign}(\langle x, y \rangle) \left[ ||\langle x, y \rangle||/m - \lambda \right]_+.
\]

- **Sparsity:** \( w = 0 \) unless the correlation between \( x \) and \( y \) is larger than \( \lambda \).

- **Exercise:** Show that the \( \ell_2 \) norm doesn’t induce a sparse solution for this case.
Outline

1. Feature Selection
   - Filters
   - Greedy selection
   - $\ell_1$ norm

2. Feature Manipulation and Normalization

3. Feature Learning
Simple transformations that we apply on each of our original features
Feature Manipulation and Normalization

- Simple transformations that we apply on each of our original features
- May decrease the approximation or estimation errors of our hypothesis class, or can yield a faster algorithm
Feature Manipulation and Normalization

- Simple transformations that we apply on each of our original features
- May decrease the approximation or estimation errors of our hypothesis class, or can yield a faster algorithm
- As in feature selection, there are no absolute “good” and “bad” transformations — need prior knowledge
Example: The effect of Normalization

Consider 2-dim ridge regression problem:

$$\arg\min_w \left[ \frac{1}{m} \| Xw - y \|_2^2 + \lambda \| w \|_2^2 \right] = (2\lambda m I + X^\top X)^{-1} X^\top y .$$

Suppose:

- $y \sim U(\pm 1)$
- $\alpha \sim U(\pm 1)$
- $x_1 = y + \alpha / 2$
- $x_2 = 0$

Best weight vector is $w^\star = [0; 10000]$, and $L_D(w^\star) = 0$.

However, the objective of ridge regression at $w^\star$ is $\lambda 10^8$ while the objective of ridge regression at $w = [1; 0]$ is likely to be close to $0.25 + \lambda$.

$\Rightarrow$ we'll choose wrong solution if $\lambda$ is not too small.

Crux of the problem: features have completely different scale while $\ell_2$ regularization treats them equally.

Simple solution: normalize features to have the same range (dividing by max, or by standard deviation).
Example: The effect of Normalization

- Consider 2-dim ridge regression problem:

\[
\arg\min_w \left[ \frac{1}{m} \| Xw - y \|^2 + \lambda \| w \|^2 \right] = (2\lambda m I + X^\top X)^{-1} X^\top y.
\]

- Suppose: \( y \sim U(\pm 1), \alpha \sim U(\pm 1), x_1 = y + \alpha/2, x_2 = 0.0001y \)

Best weight vector is \( w^\star = [0; 10000] \), and \( L_D(w^\star) = 0 \).

However, the objective of ridge regression at \( w^\star \) is \( \lambda 10^8 \) while the objective of ridge regression at \( w = [1; 0] \) is likely to be close to \( 0^{25 + \lambda} \). We'll choose the wrong solution if \( \lambda \) is not too small.

Crux of the problem: features have completely different scale while \( \ell_2 \) regularization treats them equally.

Simple solution: normalize features to have the same range (dividing by max, or by standard deviation).
Consider 2-dim ridge regression problem:

$$\arg\min_w \left[ \frac{1}{m} \| Xw - y \|^2 + \lambda \| w \|^2 \right] = (2\lambda m I + X^\top X)^{-1} X^\top y.$$ 

Suppose: \( y \sim U(\pm 1), \alpha \sim U(\pm 1), x_1 = y + \alpha/2, x_2 = 0.0001y \)

Best weight vector is \( w^* = [0; 10000] \), and \( L_D(w^*) = 0 \).
Example: The effect of Normalization

- Consider 2-dim ridge regression problem:

\[
\arg\min_w \left[ \frac{1}{m} \| Xw - y \|^2 + \lambda \| w \|^2 \right] = (2\lambda m I + X^\top X)^{-1} X^\top y .
\]

- Suppose: \( y \sim U(\pm 1), \alpha \sim U(\pm 1), x_1 = y + \alpha/2, x_2 = 0.0001y \)

- Best weight vector is \( w^* = [0; 10000] \), and \( L_D(w^*) = 0 \).

- However, the objective of ridge regression at \( w^* \) is \( \lambda 10^8 \) while the objective of ridge regression at \( w = [1; 0] \) is likely to be close to \( 0.25 + \lambda \) \( \Rightarrow \) we’ll choose wrong solution if \( \lambda \) is not too small.
Consider 2-dim ridge regression problem:

$$\arg\min_w \left[ \frac{1}{m} \|Xw - y\|^2 + \lambda \|w\|^2 \right] = (2\lambda m I + X^\top X)^{-1} X^\top y .$$

Suppose: $y \sim U(\pm 1)$, $\alpha \sim U(\pm 1)$, $x_1 = y + \alpha/2$, $x_2 = 0.0001y$

Best weight vector is $w^* = [0; 10000]$, and $L_D(w^*) = 0$.

However, the objective of ridge regression at $w^*$ is $\lambda 10^8$ while the objective of ridge regression at $w = [1; 0]$ is likely to be close to $0.25 + \lambda \Rightarrow$ we’ll choose wrong solution if $\lambda$ is not too small.

Crux of the problem: features have completely different scale while $\ell_2$ regularization treats them equally
Example: The effect of Normalization

- Consider 2-dim ridge regression problem:

\[
\arg\min_w \left[ \frac{1}{m} \| Xw - y \|^2 + \lambda \| w \|^2 \right] = (2\lambda m I + X^\top X)^{-1} X^\top y .
\]

- Suppose: \( y \sim U(\pm 1), \alpha \sim U(\pm 1), x_1 = y + \alpha/2, x_2 = 0.0001y \)

- Best weight vector is \( w^* = [0; 10000] \), and \( L_D(w^*) = 0 \).

- However, the objective of ridge regression at \( w^* \) is \( \lambda 10^8 \) while the objective of ridge regression at \( w = [1; 0] \) is likely to be close to \( 0.25 + \lambda \) \( \Rightarrow \) we’ll choose wrong solution if \( \lambda \) is not too small

- Crux of the problem: features have completely different scale while \( \ell_2 \) regularization treats them equally

- Simple solution: normalize features to have the same range (dividing by max, or by standard deviation)
Example: The effect of Transformations

Consider 1-dim regression problem, \( y \sim U(\pm 1) \), \( a \gg 1 \), and

\[
x = \begin{cases} 
y & \text{w.p. } (1 - 1/a) \\
ay & \text{w.p. } 1/a 
\end{cases}
\]

It is easy to show that \( w^* = 2a - 1 \) so \( w^* \to 0 \) as \( a \to \infty \).

It follows that \( L_D(w^*) \to 0 \).

But, if we apply "clipping", \( x \mapsto \text{sign}(x) \min\{1, |x|\} \), then \( L_D(1) = 0 \).

"Prior knowledge": features that get values larger than a predefined threshold value give us no additional useful information, and therefore we can clip them to the predefined threshold.

Of course, this "prior knowledge" can be wrong and it is easy to construct examples for which clipping hurts performance.
Example: The effect of Transformations

- Consider 1-dim regression problem, $y \sim U(\pm1)$, $a \gg 1$, and

\[
x = \begin{cases} 
y & \text{w.p. } (1 - 1/a) \\ ay & \text{w.p. } 1/a \end{cases}
\]

- It is easy to show that $w^* = \frac{2a-1}{a^2+a-1}$ so $w^* \to 0$ as $a \to \infty$
Example: The effect of Transformations

- Consider 1-dim regression problem, $y \sim U(\pm 1)$, $a \gg 1$, and

\[
x = \begin{cases} 
  y & \text{w.p. } (1 - 1/a) \\
  ay & \text{w.p. } 1/a 
\end{cases}
\]

- It is easy to show that $w^* = \frac{2a-1}{a^2+a-1}$ so $w^* \to 0$ as $a \to \infty$

- It follows that $L_D(w^*) \to 0.5$
Example: The effect of Transformations

- Consider 1-dim regression problem, \( y \sim U(\pm 1) \), \( a \gg 1 \), and

\[
x = \begin{cases} 
  y & \text{w.p. } (1 - 1/a) \\
  ay & \text{w.p. } 1/a 
\end{cases}
\]

- It is easy to show that \( w^* = \frac{2a - 1}{a^2 + a - 1} \) so \( w^* \to 0 \) as \( a \to \infty \)

- It follows that \( L_D(w^*) \to 0.5 \)

- But, if we apply “clipping”, \( x \mapsto \text{sign}(x) \min\{1, |x|\} \), then \( L_D(1) = 0 \)
Example: The effect of Transformations

- Consider 1-dim regression problem, \( y \sim U(\pm1), \ a \gg 1, \) and

\[
x = \begin{cases} 
    y & \text{w.p. } (1 - 1/a) \\
    ay & \text{w.p. } 1/a 
\end{cases}
\]

It is easy to show that \( w^* = \frac{2a-1}{a^2+a-1} \) so \( w^* \to 0 \) as \( a \to \infty \)

- It follows that \( L_D(w^*) \to 0.5 \)

- But, if we apply “clipping”, \( x \mapsto \text{sign}(x) \min\{1, |x|\} \), then \( L_D(1) = 0 \)

- “Prior knowledge”: features that get values larger than a predefined threshold value give us no additional useful information, and therefore we can clip them to the predefined threshold.
Example: The effect of Transformations

- Consider 1-dim regression problem, \( y \sim U(\pm1), a \gg 1 \), and

\[
x = \begin{cases} 
y & \text{w.p. } (1 - 1/a) \\
y a & \text{w.p. } 1/a \\
\end{cases}
\]

- It is easy to show that \( w^* = \frac{2a-1}{a^2+a-1} \) so \( w^* \to 0 \) as \( a \to \infty \)

- It follows that \( L_D(w^*) \to 0.5 \)

- But, if we apply “clipping”, \( x \mapsto \text{sign}(x) \min\{1, |x|\} \), then \( L_D(1) = 0 \)

- “Prior knowledge”: features that get values larger than a predefined threshold value give us no additional useful information, and therefore we can clip them to the predefined threshold.

- Of course, this “prior knowledge” can be wrong and it is easy to construct examples for which clipping hurts performance
Some Examples of Feature Transformations

- Denote $f = (f_1, \ldots, f_m) \in \mathbb{R}^m$ the values of the feature and $\bar{f}$ the empirical mean.
Some Examples of Feature Transformations

- Denote $\mathbf{f} = (f_1, \ldots, f_m) \in \mathbb{R}^m$ the values of the feature and $\bar{f}$ the empirical mean.
- **Centering**: $f_i \leftarrow f_i - \bar{f}$. 

Unary representation for categorical features: $f_i \mapsto (1 \begin{array}{c} f_i = 1 \\ \vdots \\ 1 \end{array}, \ldots, 1 \begin{array}{c} f_i = k \end{array})$. 
Some Examples of Feature Transformations

- Denote $\mathbf{f} = (f_1, \ldots, f_m) \in \mathbb{R}^m$ the values of the feature and $\bar{f}$ the empirical mean.
- Centering: $f_i \leftarrow f_i - \bar{f}$.
- Unit Range: $f_{\text{max}} = \max_i f_i$, $f_{\text{min}} = \min_i f_i$, $f_i \leftarrow \frac{f_i - f_{\text{min}}}{f_{\text{max}} - f_{\text{min}}}$.
Some Examples of Feature Transformations

- Denote $\mathbf{f} = (f_1, \ldots, f_m) \in \mathbb{R}^m$ the values of the feature and $\bar{f}$ the empirical mean.

- Centering: $f_i \leftarrow f_i - \bar{f}$.

- Unit Range: $f_{\text{max}} = \max_i f_i$, $f_{\text{min}} = \min_i f_i$, $f_i \leftarrow \frac{f_i - f_{\text{min}}}{f_{\text{max}} - f_{\text{min}}}$.

- Standardization: $\nu = \frac{1}{m} \sum_{i=1}^{m} (f_i - \bar{f})^2$, $f_i \leftarrow \frac{f_i - \bar{f}}{\sqrt{\nu}}$.
Denote $\mathbf{f} = (f_1, \ldots, f_m) \in \mathbb{R}^m$ the values of the feature and $\bar{f}$ the empirical mean.

- **Centering:** $f_i \leftarrow f_i - \bar{f}$.
- **Unit Range:** $f_{\text{max}} = \max_i f_i$, $f_{\text{min}} = \min_i f_i$, $f_i \leftarrow \frac{f_i - f_{\text{min}}}{f_{\text{max}} - f_{\text{min}}}$.
- **Standardization:** $\nu = \frac{1}{m} \sum_{i=1}^{m} (f_i - \bar{f})^2$, $f_i \leftarrow \frac{f_i - \bar{f}}{\sqrt{\nu}}$.
- **Clipping:** $f_i \leftarrow \text{sign}(f_i) \max\{b, |f_i|\}$.
Some Examples of Feature Transformations

- Denote $\mathbf{f} = (f_1, \ldots, f_m) \in \mathbb{R}^m$ the values of the feature and $\bar{f}$ the empirical mean.
- Centering: $f_i \leftarrow f_i - \bar{f}$.
- Unit Range: $f_{\max} = \max_i f_i$, $f_{\min} = \min_i f_i$, $f_i \leftarrow \frac{f_i - f_{\min}}{f_{\max} - f_{\min}}$.
- Standardization: $\nu = \frac{1}{m} \sum_{i=1}^{m} (f_i - \bar{f})^2$, $f_i \leftarrow \frac{f_i - \bar{f}}{\sqrt{\nu}}$.
- Clipping: $f_i \leftarrow \text{sign}(f_i) \max\{b, |f_i|\}$.
- Sigmoidal transformation: $f_i \leftarrow \frac{1}{1 + \exp(b f_i)}$. 
Some Examples of Feature Transformations

- Denote $\mathbf{f} = (f_1, \ldots, f_m) \in \mathbb{R}^m$ the values of the feature and $\bar{f}$ the empirical mean.

- Centering: $f_i \leftarrow f_i - \bar{f}$.

- Unit Range: $f_{\text{max}} = \max_i f_i$, $f_{\text{min}} = \min_i f_i$, $f_i \leftarrow \frac{f_i - f_{\text{min}}}{f_{\text{max}} - f_{\text{min}}}$.

- Standardization: $\nu = \frac{1}{m} \sum_{i=1}^{m} (f_i - \bar{f})^2$, $f_i \leftarrow \frac{f_i - \bar{f}}{\sqrt{\nu}}$.

- Clipping: $f_i \leftarrow \text{sign}(f_i) \max\{b, |f_i|\}$.

- Sigmoidal transformation: $f_i \leftarrow \frac{1}{1 + \exp(b f_i)}$

- Logarithmic transformation: $f_i \leftarrow \log(b + f_i)$
Some Examples of Feature Transformations

- Denote $f = (f_1, \ldots, f_m) \in \mathbb{R}^m$ the values of the feature and $\bar{f}$ the empirical mean.
- Centering: $f_i \leftarrow f_i - \bar{f}$.
- Unit Range: $f_{\text{max}} = \max_i f_i$, $f_{\text{min}} = \min_i f_i$, $f_i \leftarrow \frac{f_i - f_{\text{min}}}{f_{\text{max}} - f_{\text{min}}}$.
- Standardization: $\nu = \frac{1}{m} \sum_{i=1}^{m} (f_i - \bar{f})^2$, $f_i \leftarrow \frac{f_i - \bar{f}}{\sqrt{\nu}}$.
- Clipping: $f_i \leftarrow \text{sign}(f_i) \max\{b, |f_i|\}$
- Sigmoidal transformation: $f_i \leftarrow \frac{1}{1 + \exp(b f_i)}$
- Logarithmic transformation: $f_i \leftarrow \log(b + f_i)$
- Unary representation for categorical features: $f_i \mapsto (1_{f_i = 1}, \ldots, 1_{f_i = k})$
Outline

1. Feature Selection
   - Filters
   - Greedy selection
   - $\ell_1$ norm

2. Feature Manipulation and Normalization

3. Feature Learning
**Goal:** learn a feature mapping, $\psi : \mathcal{X} \rightarrow \mathbb{R}^d$, so that a linear predictor on top of $\psi(x)$ will yield a good hypothesis class.
Feature Learning

- **Goal:** learn a feature mapping, $\psi : \mathcal{X} \rightarrow \mathbb{R}^d$, so that a linear predictor on top of $\psi(x)$ will yield a good hypothesis class.

- **Example:** we can think on the first layers of a neural network as $\psi(x)$ and the last layer as the linear predictor applied on top of it.
Feature Learning

- **Goal:** learn a feature mapping, $\psi : \mathcal{X} \rightarrow \mathbb{R}^d$, so that a linear predictor on top of $\psi(x)$ will yield a good hypothesis class.

- **Example:** we can think on the first layers of a neural network as $\psi(x)$ and the last layer as the linear predictor applied on top of it.

- We will describe an unsupervised learning approach for feature learning called **Dictionary learning**.
Dictionary Learning

- Motivation: recall the description of a document as a “bag-of-words”: $\psi(x) \in \{0, 1\}^k$ where coordinate $i$ of $\psi(x)$ determines if word $i$ appears in the document or not.
Dictionary Learning

- Motivation: recall the description of a document as a “bag-of-words”: \( \psi(x) \in \{0, 1\}^k \) where coordinate \( i \) of \( \psi(x) \) determines if word \( i \) appears in the document or not

- What is the dictionary in general? For example, what will be a good dictionary for visual data? Can we learn \( \psi : \mathcal{X} \rightarrow \{0, 1\}^k \) that captures “visual words”, e.g., \( (\psi(x))_i \) captures something like “there is an eye in the image”? 


Shai Shalev-Shwartz (Hebrew U)
Dictionary Learning

Motivation: recall the description of a document as a “bag-of-words”: $\psi(x) \in \{0, 1\}^k$ where coordinate $i$ of $\psi(x)$ determines if word $i$ appears in the document or not.

What is the dictionary in general? For example, what will be a good dictionary for visual data? Can we learn $\psi: \mathcal{X} \rightarrow \{0, 1\}^k$ that captures “visual words”, e.g., $(\psi(x))_i$ captures something like “there is an eye in the image”?

Using clustering: A clustering function $c: \mathcal{X} \rightarrow \{1, \ldots, k\}$ yields the mapping $\psi(x)_i = 1$ iff $x$ belongs to cluster $i$. 

Shai Shalev-Shwartz (Hebrew U)
Dictionary Learning

- Motivation: recall the description of a document as a “bag-of-words”: \( \psi(x) \in \{0, 1\}^k \) where coordinate \( i \) of \( \psi(x) \) determines if word \( i \) appears in the document or not

- What is the dictionary in general? For example, what will be a good dictionary for visual data? Can we learn \( \psi : \mathcal{X} \rightarrow \{0, 1\}^k \) that captures “visual words”, e.g., \( (\psi(x))_i \) captures something like “there is an eye in the image”?

- Using clustering: A clustering function \( c : \mathcal{X} \rightarrow \{1, \ldots, k\} \) yields the mapping \( \psi(x)_i = 1 \) iff \( x \) belongs to cluster \( i \)

- Sparse auto-encoders: Given \( x \in \mathbb{R}^d \) and dictionary matrix \( D \in \mathbb{R}^{d,k} \), let

\[
\psi(x) = \arg\min_{v \in \mathbb{R}^k} \|x - Dv\| \quad \text{s.t.} \quad \|v\|_0 \leq s
\]
Summary

- Feature selection
- Feature normalization and manipulations
- Feature learning