Introduction to Machine Learning (67577)
Lecture 13

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 ?

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 ?

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,

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

  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,

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

  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,

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

  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
   - ℓ_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
   - ℓ_1 norm
2. Feature Manipulation and Normalization
3. Feature Learning

Filters

- 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.:

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

$$-\min_{a,b\in\mathbb{R}} \sum_{i=1}^{m} \ell(av_i + b, y_i)$$

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)

$$\frac{|\langle \mathbf{v} - \bar{v}, \mathbf{y} - \bar{y} \rangle|}{\|\mathbf{v} - \bar{v}\| \, \|\mathbf{y} - \bar{y}\|}$$

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)

$$\frac{|\langle \mathbf{v} - \bar{v}, \mathbf{y} - \bar{y} \rangle|}{\|\mathbf{v} - \bar{v}\| \, \|\mathbf{y} - \bar{y}\|}$$

  - Spearman's rho: Apply Pearson's coefficient on the ranking of $\mathbf{v}$

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

$$-\min_{a,b\in\mathbb{R}} \sum_{i=1}^{m} \ell(av_i + b, y_i)$$

  - Pearson correlation coefficient: (obtained by minimizing squared loss)

$$\frac{|\langle \mathbf{v} - \bar{v}, \mathbf{y} - \bar{y} \rangle|}{\|\mathbf{v} - \bar{v}\| \, \|\mathbf{y} - \bar{y}\|}$$

  - Spearman's rho: Apply Pearson's coefficient on the ranking of $\mathbf{v}$
  - 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 $\mathbf{v}$ alone is useless for predicting $\mathbf{y}$

Weakness of Filters

- If Pearson's coefficient is zero then $\mathbf{v}$ alone is useless for predicting $\mathbf{y}$
- This doesn't mean that $\mathbf{v}$ is a bad feature — maybe with other features it is very useful

Weakness of Filters

- If Pearson's coefficient is zero then $\mathbf{v}$ alone is useless for predicting $\mathbf{y}$
- This doesn't mean that $\mathbf{v}$ 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
   - ℓ_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 $i \notin I$ 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 $i \notin I$ and learn a predictor based on $I \cup i$
- Choose the i that led to best predictor and update $I = I \cup \{i\}$
- Example: Orthogonal Matching Pursuit

Orthogonal Matching Pursuit (OMP)

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

Orthogonal Matching Pursuit (OMP)

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

Orthogonal Matching Pursuit (OMP)

- Let $X \in \mathbb{R}^{m,d}$ be a data matrix (instances in rows). Let $\mathbf{y} \in \mathbb{R}^m$ 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

$$j_t = \operatorname*{argmin}_j \min_{\mathbf{w} \in \mathbb{R}^t} \|X_{I_{t-1} \cup \{j\}} \mathbf{w} - \mathbf{y}\|^2 .$$

Orthogonal Matching Pursuit (OMP)

- Let $X \in \mathbb{R}^{m,d}$ be a data matrix (instances in rows). Let $\mathbf{y} \in \mathbb{R}^m$ 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

$$j_t = \operatorname*{argmin}_j \min_{\mathbf{w} \in \mathbb{R}^t} \left\| X_{I_{t-1} \cup \{j\}} \mathbf{w} - \mathbf{y} \right\|^2 .$$

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

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

Orthogonal Matching Pursuit (OMP)

- Let $X \in \mathbb{R}^{m,d}$ be a data matrix (instances in rows). Let $\mathbf{y} \in \mathbb{R}^m$ 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

$$j_t = \operatorname*{argmin}_j \min_{\mathbf{w} \in \mathbb{R}^t} \|X_{I_{t-1} \cup \{j\}} \mathbf{w} - \mathbf{y}\|^2 .$$

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

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

- Let θ_t be a minimizer of the right-hand side

Orthogonal Matching Pursuit (OMP)

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

Orthogonal Matching Pursuit (OMP)

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

$$\min_{\boldsymbol{\theta},\alpha} \|V_{t-1}\boldsymbol{\theta} + \alpha\mathbf{u}_j - \mathbf{y}\|^2$$

Orthogonal Matching Pursuit (OMP)

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

$$\begin{aligned}
&\min_{\boldsymbol{\theta},\alpha} \|V_{t-1}\boldsymbol{\theta} + \alpha\mathbf{u}_j - \mathbf{y}\|^2 \\
&= \min_{\boldsymbol{\theta},\alpha} \left[\|V_{t-1}\boldsymbol{\theta} - \mathbf{y}\|^2 + \alpha^2\|\mathbf{u}_j\|^2 + 2\alpha\langle \mathbf{u}_j, V_{t-1}\boldsymbol{\theta} - \mathbf{y}\rangle\right]
\end{aligned}$$

Orthogonal Matching Pursuit (OMP)

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

$$\begin{aligned}
&\min_{\boldsymbol{\theta},\alpha} \|V_{t-1}\boldsymbol{\theta} + \alpha\mathbf{u}_j - \mathbf{y}\|^2 \\
&= \min_{\boldsymbol{\theta},\alpha} \left[\|V_{t-1}\boldsymbol{\theta} - \mathbf{y}\|^2 + \alpha^2\|\mathbf{u}_j\|^2 + 2\alpha\langle \mathbf{u}_j, V_{t-1}\boldsymbol{\theta} - \mathbf{y}\rangle\right] \\
&= \min_{\boldsymbol{\theta},\alpha} \left[\|V_{t-1}\boldsymbol{\theta} - \mathbf{y}\|^2 + \alpha^2\|\mathbf{u}_j\|^2 + 2\alpha\langle \mathbf{u}_j, -\mathbf{y}\rangle\right]
\end{aligned}$$

Orthogonal Matching Pursuit (OMP)

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

$$
\begin{aligned}
&\min_{\boldsymbol{\theta},\alpha} \|V_{t-1}\boldsymbol{\theta} + \alpha\mathbf{u}_j - \mathbf{y}\|^2 \\
&= \min_{\boldsymbol{\theta},\alpha} \left[\|V_{t-1}\boldsymbol{\theta} - \mathbf{y}\|^2 + \alpha^2\|\mathbf{u}_j\|^2 + 2\alpha\langle \mathbf{u}_j, V_{t-1}\boldsymbol{\theta} - \mathbf{y}\rangle\right] \\
&= \min_{\boldsymbol{\theta},\alpha} \left[\|V_{t-1}\boldsymbol{\theta} - \mathbf{y}\|^2 + \alpha^2\|\mathbf{u}_j\|^2 + 2\alpha\langle \mathbf{u}_j, -\mathbf{y}\rangle\right] \\
&= \min_{\boldsymbol{\theta}} \left[\|V_{t-1}\boldsymbol{\theta} - \mathbf{y}\|^2\right] + \min_{\alpha} \left[\alpha^2\|\mathbf{u}_j\|^2 - 2\alpha\langle \mathbf{u}_j, \mathbf{y}\rangle\right]
\end{aligned}
$$

Orthogonal Matching Pursuit (OMP)

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

$$
\begin{aligned}
&\min_{\boldsymbol{\theta},\alpha} \|V_{t-1}\boldsymbol{\theta} + \alpha \mathbf{u}_j - \mathbf{y}\|^2 \\
&= \min_{\boldsymbol{\theta},\alpha} \left[\|V_{t-1}\boldsymbol{\theta} - \mathbf{y}\|^2 + \alpha^2\|\mathbf{u}_j\|^2 + 2\alpha\langle \mathbf{u}_j, V_{t-1}\boldsymbol{\theta} - \mathbf{y}\rangle\right] \\
&= \min_{\boldsymbol{\theta},\alpha} \left[\|V_{t-1}\boldsymbol{\theta} - \mathbf{y}\|^2 + \alpha^2\|\mathbf{u}_j\|^2 + 2\alpha\langle \mathbf{u}_j, -\mathbf{y}\rangle\right] \\
&= \min_{\boldsymbol{\theta}} \left[\|V_{t-1}\boldsymbol{\theta} - \mathbf{y}\|^2\right] + \min_{\alpha} \left[\alpha^2\|\mathbf{u}_j\|^2 - 2\alpha\langle \mathbf{u}_j, \mathbf{y}\rangle\right] \\
&= \left[\|V_{t-1}\boldsymbol{\theta}_{t-1} - \mathbf{y}\|^2\right] + \min_{\alpha} \left[\alpha^2\|\mathbf{u}_j\|^2 - 2\alpha\langle \mathbf{u}_j, \mathbf{y}\rangle\right]
\end{aligned}
$$

Orthogonal Matching Pursuit (OMP)

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

$$
\begin{aligned}
&\min_{\boldsymbol{\theta},\alpha} \|V_{t-1}\boldsymbol{\theta} + \alpha \mathbf{u}_j - \mathbf{y}\|^2 \\
&= \min_{\boldsymbol{\theta},\alpha} \left[\|V_{t-1}\boldsymbol{\theta} - \mathbf{y}\|^2 + \alpha^2\|\mathbf{u}_j\|^2 + 2\alpha\langle \mathbf{u}_j, V_{t-1}\boldsymbol{\theta} - \mathbf{y}\rangle\right] \\
&= \min_{\boldsymbol{\theta},\alpha} \left[\|V_{t-1}\boldsymbol{\theta} - \mathbf{y}\|^2 + \alpha^2\|\mathbf{u}_j\|^2 + 2\alpha\langle \mathbf{u}_j, -\mathbf{y}\rangle\right] \\
&= \min_{\boldsymbol{\theta}} \left[\|V_{t-1}\boldsymbol{\theta} - \mathbf{y}\|^2\right] + \min_{\alpha} \left[\alpha^2\|\mathbf{u}_j\|^2 - 2\alpha\langle \mathbf{u}_j, \mathbf{y}\rangle\right] \\
&= \left[\|V_{t-1}\boldsymbol{\theta}_{t-1} - \mathbf{y}\|^2\right] + \min_{\alpha} \left[\alpha^2\|\mathbf{u}_j\|^2 - 2\alpha\langle \mathbf{u}_j, \mathbf{y}\rangle\right] \\
&= \|V_{t-1}\boldsymbol{\theta}_{t-1} - \mathbf{y}\|^2 - \frac{(\langle \mathbf{u}_j, \mathbf{y}\rangle)^2}{\|\mathbf{u}_j\|^2}
\end{aligned}
$$

Orthogonal Matching Pursuit (OMP)

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

$$
\begin{aligned}
&\min_{\boldsymbol{\theta},\alpha} \|V_{t-1}\boldsymbol{\theta} + \alpha\mathbf{u}_j - \mathbf{y}\|^2 \\
&= \min_{\boldsymbol{\theta},\alpha} \left[\|V_{t-1}\boldsymbol{\theta} - \mathbf{y}\|^2 + \alpha^2\|\mathbf{u}_j\|^2 + 2\alpha\langle \mathbf{u}_j, V_{t-1}\boldsymbol{\theta} - \mathbf{y}\rangle\right] \\
&= \min_{\boldsymbol{\theta},\alpha} \left[\|V_{t-1}\boldsymbol{\theta} - \mathbf{y}\|^2 + \alpha^2\|\mathbf{u}_j\|^2 + 2\alpha\langle \mathbf{u}_j, -\mathbf{y}\rangle\right] \\
&= \min_{\boldsymbol{\theta}} \left[\|V_{t-1}\boldsymbol{\theta} - \mathbf{y}\|^2\right] + \min_{\alpha} \left[\alpha^2\|\mathbf{u}_j\|^2 - 2\alpha\langle \mathbf{u}_j, \mathbf{y}\rangle\right] \\
&= \left[\|V_{t-1}\boldsymbol{\theta}_{t-1} - \mathbf{y}\|^2\right] + \min_{\alpha} \left[\alpha^2\|\mathbf{u}_j\|^2 - 2\alpha\langle \mathbf{u}_j, \mathbf{y}\rangle\right] \\
&= \|V_{t-1}\boldsymbol{\theta}_{t-1} - \mathbf{y}\|^2 - \frac{(\langle \mathbf{u}_j, \mathbf{y}\rangle)^2}{\|\mathbf{u}_j\|^2}
\end{aligned}
$$

- It follows that we should select the feature $j_t = \operatorname{argmax}_j \frac{(\langle \mathbf{u}_j,\mathbf{y}\rangle)^2}{\|\mathbf{u}_j\|^2}$.

Orthogonal Matching Pursuit (OMP)

```
Orthogonal Matching Pursuit (OMP)
```

input:
data matrix $X \in \mathbb{R}^{m,d}$, labels vector $\mathbf{y} \in \mathbb{R}^m$,
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 $\mathbf{u}_j = X_j - VV^\top X_j$
let $j_t = \operatorname{argmax}_{j \notin I_t : \|\mathbf{u}_j\| > 0} \frac{(\langle \mathbf{u}_j, \mathbf{y} \rangle)^2}{\|\mathbf{u}_j\|^2}$
update $I_{t+1} = I_t \cup \{j_t\}$

output I_{T+1}

Gradient-based Greedy Selection

- Let $R(\mathbf{w})$ be the empirical risk as a function of $\mathbf{w}$

Gradient-based Greedy Selection

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

$$\operatorname*{argmin}_{j} \min_{\mathbf{w}:\mathrm{supp}(\mathbf{w})=I\cup\{i\}} R(\mathbf{w})$$

Gradient-based Greedy Selection

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

$$\operatorname*{argmin}_j \min_{\mathbf{w}:\mathrm{supp}(\mathbf{w})=I\cup\{i\}} R(\mathbf{w})$$

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

$$\operatorname*{argmin}_j \min_{\eta\in\mathbb{R}} R(\mathbf{w} + \eta\mathbf{e}_j)$$

Gradient-based Greedy Selection

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

$$\operatorname*{argmin}_j \min_{\mathbf{w}:\mathrm{supp}(\mathbf{w})=I\cup\{i\}} R(\mathbf{w})$$

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

$$\operatorname*{argmin}_j \min_{\eta\in\mathbb{R}} R(\mathbf{w} + \eta\mathbf{e}_j)$$

- An even simpler approach is to choose the feature which minimizes the above for infinitesimal η, namely,

$$\operatorname*{argmin}_j |\nabla_j R(\mathbf{w})|$$

AdaBoost as Forward Greedy Selection

- It is possible to show (left as an exercise), that the AdaBoost algorithm is in fact Forward Greedy Selection for the objective function

$$R(\mathbf{w}) = \log\left(\sum_{i=1}^{m} \exp\left(-y_i \sum_{j=1}^{d} w_j h_j(\mathbf{x}_j)\right)\right) .$$

Outline

1. Feature Selection
   - Filters
   - Greedy selection
   - ℓ_1 norm
2. Feature Manipulation and Normalization
3. Feature Learning

Sparsity Inducing Norms

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

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

Sparsity Inducing Norms

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

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

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

Sparsity Inducing Norms

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

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

- Replace the non-convex constraint, $\|\mathbf{w}\|_0 \le k$, with a convex constraint, $\|\mathbf{w}\|_1 \le k_1$.
- Why ℓ_1 ?

Sparsity Inducing Norms

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

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

- Replace the non-convex constraint, $\|\mathbf{w}\|_0 \le k$, with a convex constraint, $\|\mathbf{w}\|_1 \le k_1$.
- Why ℓ_1 ?
  - "Closest" convex surrogate

Sparsity Inducing Norms

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

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

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

Sparsity Inducing Norms

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

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

- Replace the non-convex constraint, $\|\mathbf{w}\|_0 \le k$, with a convex constraint, $\|\mathbf{w}\|_1 \le k_1$.
- Why ℓ_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, ℓ_1 "induces" sparse solutions

ℓ_1 regularization

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

$$\min_{\mathbf{w}} \left(L_S(\mathbf{w}) + \lambda \|\mathbf{w}\|_1 \right)$$

ℓ_1 regularization

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

$$\min_{\mathbf{w}} \left(L_S(\mathbf{w}) + \lambda \|\mathbf{w}\|_1 \right)$$

- For Squared-Loss this is the Lasso method

ℓ_1 regularization

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

$$\min_{\mathbf{w}} \left(L_S(\mathbf{w}) + \lambda \|\mathbf{w}\|_1 \right)$$

- For Squared-Loss this is the Lasso method
- ℓ_1 norm often induces sparse solutions. Example:

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

ℓ_1 regularization

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

$$\min_{\mathbf{w}} \left(L_S(\mathbf{w}) + \lambda\|\mathbf{w}\|_1\right)$$

- For Squared-Loss this is the Lasso method
- ℓ_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"

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

ℓ_1 regularization

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

$$\min_{\mathbf{w}} \left(L_S(\mathbf{w}) + \lambda \|\mathbf{w}\|_1 \right)$$

- For Squared-Loss this is the Lasso method
- ℓ_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"

$$w = \mathrm{sign}(x) \left[|x| - \lambda \right]_+$$

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

ℓ_1 regularization

- One dimensional Lasso:

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

ℓ_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:

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

ℓ_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 $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^m x_i y_i$, then the optimal solution is

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

ℓ_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 $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^m x_i y_i$, then the optimal solution is
$$w = \operatorname{sign}(\langle \mathbf{x}, \mathbf{y} \rangle) \left[|\langle \mathbf{x}, \mathbf{y} \rangle| / m - \lambda \right]_+ .$$
- Sparsity: $w = 0$ unless the correlation between $\mathbf{x}$ and $\mathbf{y}$ is larger than λ.

ℓ_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 $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^m x_i y_i$, then the optimal solution is

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

- Sparsity: $w = 0$ unless the correlation between $\mathbf{x}$ and $\mathbf{y}$ is larger than λ.
- Exercise: Show that the ℓ_2 norm doesn't induce a sparse solution for this case

Outline

1. Feature Selection
   - Filters
   - Greedy selection
   - ℓ_1 norm
2. Feature Manipulation and Normalization
3. Feature Learning

Feature Manipulation and Normalization

- 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:

$$\operatorname*{argmin}_{\mathbf{w}} \left[\frac{1}{m} \|X\mathbf{w} - \mathbf{y}\|^2 + \lambda \|\mathbf{w}\|^2 \right] = (2\lambda m I + X^\top X)^{-1} X^\top \mathbf{y} \ .$$

Example: The effect of Normalization

- Consider 2-dim ridge regression problem:

$$\operatorname*{argmin}_{\mathbf{w}} \left[\frac{1}{m} \|X\mathbf{w} - \mathbf{y}\|^2 + \lambda \|\mathbf{w}\|^2 \right] = (2\lambda m I + X^\top X)^{-1} X^\top \mathbf{y} \ .$$

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

Example: The effect of Normalization

- Consider 2-dim ridge regression problem:

$$\operatorname*{argmin}_{\mathbf{w}} \left[\frac{1}{m} \|X\mathbf{w} - \mathbf{y}\|^2 + \lambda \|\mathbf{w}\|^2 \right] = (2\lambda m I + X^\top X)^{-1} X^\top \mathbf{y} \ .$$

- Suppose: $y \sim U(\pm 1)$, $\alpha \sim U(\pm 1)$, $x_1 = y + \alpha/2$, $x_2 = 0.0001y$
- Best weight vector is $\mathbf{w}^\star = [0; 10000]$, and $L_{\mathcal{D}}(\mathbf{w}^\star) = 0$.

Example: The effect of Normalization

- Consider 2-dim ridge regression problem:

$$\operatorname*{argmin}_{\mathbf{w}} \left[\frac{1}{m} \|X\mathbf{w} - \mathbf{y}\|^2 + \lambda \|\mathbf{w}\|^2 \right] = (2\lambda m I + X^\top X)^{-1} X^\top \mathbf{y} \ .$$

- Suppose: $y \sim U(\pm 1)$, $\alpha \sim U(\pm 1)$, $x_1 = y + \alpha/2$, $x_2 = 0.0001y$
- Best weight vector is $\mathbf{w}^\star = [0; 10000]$, and $L_{\mathcal{D}}(\mathbf{w}^\star) = 0$.
- However, the objective of ridge regression at $\mathbf{w}^\star$ is $\lambda 10^8$ while the objective of ridge regression at $\mathbf{w} = [1; 0]$ is likely to be close to $0.25 + \lambda \quad \Rightarrow \quad$ we'll choose wrong solution if λ is not too small

Example: The effect of Normalization

- Consider 2-dim ridge regression problem:

$$\underset{\mathbf{w}}{\operatorname{argmin}} \left[\frac{1}{m} \|X\mathbf{w} - \mathbf{y}\|^2 + \lambda \|\mathbf{w}\|^2 \right] = (2\lambda m I + X^\top X)^{-1} X^\top \mathbf{y} \ .$$

- Suppose: $y \sim U(\pm 1)$, $\alpha \sim U(\pm 1)$, $x_1 = y + \alpha/2$, $x_2 = 0.0001y$
- Best weight vector is $\mathbf{w}^\star = [0; 10000]$, and $L_{\mathcal{D}}(\mathbf{w}^\star) = 0$.
- However, the objective of ridge regression at $\mathbf{w}^\star$ is $\lambda 10^8$ while the objective of ridge regression at $\mathbf{w} = [1; 0]$ is likely to be close to $0.25 + \lambda \quad \Rightarrow \quad$ we'll choose wrong solution if λ is not too small
- Crux of the problem: features have completely different scale while ℓ_2 regularization treats them equally

Example: The effect of Normalization

- Consider 2-dim ridge regression problem:

$$\underset{\mathbf{w}}{\operatorname{argmin}} \left[\frac{1}{m} \|X\mathbf{w} - \mathbf{y}\|^2 + \lambda \|\mathbf{w}\|^2 \right] = (2\lambda m I + X^\top X)^{-1} X^\top \mathbf{y} \ .$$

- Suppose: $y \sim U(\pm 1)$, $\alpha \sim U(\pm 1)$, $x_1 = y + \alpha/2$, $x_2 = 0.0001y$
- Best weight vector is $\mathbf{w}^\star = [0; 10000]$, and $L_{\mathcal{D}}(\mathbf{w}^\star) = 0$.
- However, the objective of ridge regression at $\mathbf{w}^\star$ is $\lambda 10^8$ while the objective of ridge regression at $\mathbf{w} = [1; 0]$ is likely to be close to $0.25 + \lambda \quad \Rightarrow \quad$ we'll choose wrong solution if λ is not too small
- Crux of the problem: features have completely different scale while ℓ_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

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

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 $a \to \infty$

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 $L_{\mathcal{D}}(w^*) \to 0.5$

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 $L_{\mathcal{D}}(w^*) \to 0.5$
- But, if we apply "clipping", $x \mapsto \mathrm{sign}(x) \min\{1, |x|\}$, then $L_{\mathcal{D}}(1) = 0$

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 $L_{\mathcal{D}}(w^*) \to 0.5$
- But, if we apply "clipping", $x \mapsto \mathrm{sign}(x) \min\{1, |x|\}$, then $L_{\mathcal{D}}(1) = 0$
- "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, $y \sim U(\pm 1)$, $a \gg 1$, 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 $L_{\mathcal{D}}(w^*) \to 0.5$
- But, if we apply "clipping", $x \mapsto \mathrm{sign}(x) \min\{1, |x|\}$, then $L_{\mathcal{D}}(1) = 0$
- "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 $\mathbf{f} = (f_1, \ldots, f_m) \in \mathbb{R}^m$ 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: $f_i \leftarrow f_i - \bar{f}$.

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

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

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: $f_i \leftarrow \mathrm{sign}(f_i) \max\{b, |f_i|\}$

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: $f_i \leftarrow \mathrm{sign}(f_i) \max\{b, |f_i|\}$
- 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: $f_{\max} = \max_i f_i$, $f_{\min} = \min_i f_i$, $f_i \leftarrow \frac{f_i - f_{\min}}{f_{\max} - f_{\min}}$.
- 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: $f_i \leftarrow \mathrm{sign}(f_i)\,\max\{b, |f_i|\}$
- 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 $\mathbf{f} = (f_1, \ldots, f_m) \in \mathbb{R}^m$ 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: $f_i \leftarrow \mathrm{sign}(f_i)\, \max\{b, |f_i|\}$
- 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: $f_i \mapsto (\mathbb{1}_{[f_i=1]}, \ldots, \mathbb{1}_{[f_i=k]})$

Outline

1. Feature Selection
   - Filters
   - Greedy selection
   - ℓ_1 norm
2. Feature Manipulation and Normalization
3. Feature Learning

Feature Learning

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

Feature Learning

- Goal: learn a feature mapping, $\psi : \mathcal{X} \to \mathbb{R}^d$, 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, $\psi : \mathcal{X} \to \mathbb{R}^d$, 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 $\psi : \mathcal{X} \to \{0,1\}^k$ that captures "visual words", e.g., $(\psi(x))_i$ captures something like "there is an eye in the image" ?

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 $\psi : \mathcal{X} \to \{0,1\}^k$ that captures "visual words", e.g., $(\psi(x))_i$ captures something like "there is an eye in the image" ?
- Using clustering: A clustering function $c : \mathcal{X} \to \{1,\ldots,k\}$ yields the mapping $\psi(x)_i = 1$ iff x belongs to cluster i

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 $\psi : \mathcal{X} \to \{0,1\}^k$ that captures "visual words", e.g., $(\psi(x))_i$ captures something like "there is an eye in the image" ?
- Using clustering: A clustering function $c : \mathcal{X} \to \{1,\ldots,k\}$ yields the mapping $\psi(x)_i = 1$ iff x belongs to cluster i
- Sparse auto-encoders: Given $\mathbf{x} \in \mathbb{R}^d$ and dictionary matrix $D \in \mathbb{R}^{d,k}$, let

$$\psi(\mathbf{x}) = \operatorname*{argmin}_{\mathbf{v} \in \mathbb{R}^k} \|\mathbf{x} - D\mathbf{v}\| \quad \text{s.t.} \quad \|\mathbf{v}\|_0 \le s$$

Summary

- Feature selection
- Feature normalization and manipulations
- Feature learning