

Linear methods for supervised learning

- LDA
- Logistic regression
- Naïve Bayes
- PLA
- Maximum margin hyperplanes
- Soft-margin hyperplanes
- Least squares regression
- Ridge regression
- ...

Nonlinear feature maps

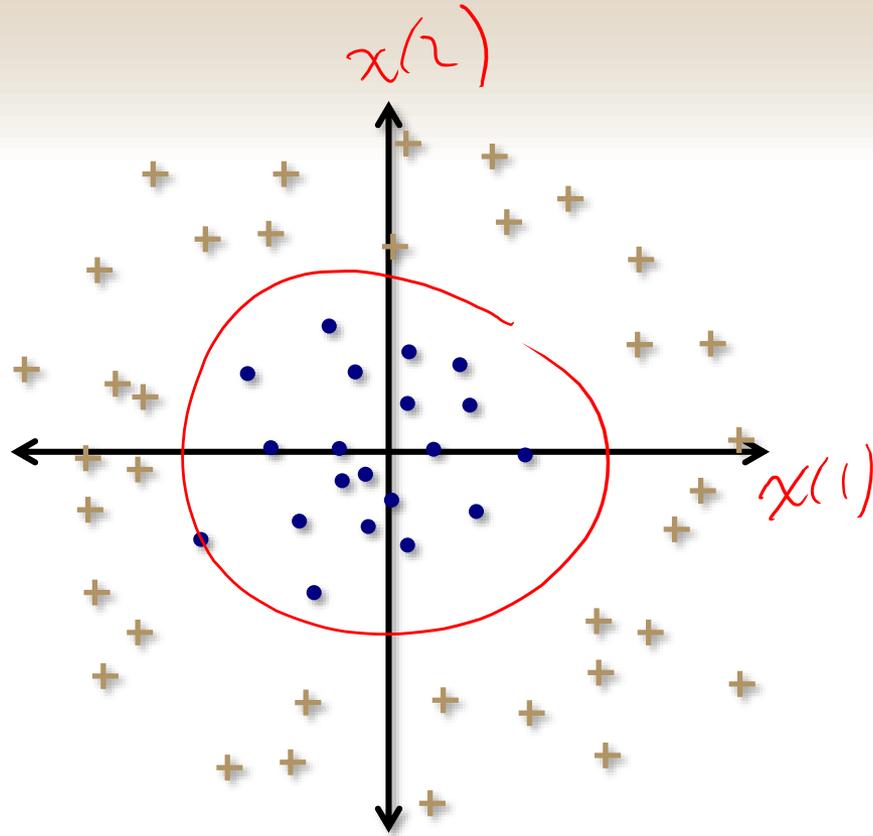
Sometimes linear methods (in both regression and classification) just don't work

One way to create nonlinear estimators or classifiers is to first transform the data via a nonlinear feature map

$$\Phi : \mathbb{R}^d \rightarrow \mathbb{R}^p$$

After applying Φ , we can then try applying a linear method to the transformed data $\Phi(\mathbf{x}_1), \dots, \Phi(\mathbf{x}_n)$

Classification



This data set is not linearly separable

Consider the mapping

$$\Phi(\mathbf{x}) = \begin{bmatrix} 1 \\ x(1) \\ x(2) \\ x(1)x(2) \\ x(1)^2 \\ x(2)^2 \end{bmatrix}$$

The dataset *is* linearly separable after applying this feature map: $\mathbf{w} = [-1, 0, 0, 0, 1, 1]^T$

$$x(1)^2 + x(2)^2 \geq 1$$

Issues with nonlinear feature maps

Suppose we transform our data via

$$\mathbf{x} = \begin{bmatrix} x(1) \\ \vdots \\ x(d) \end{bmatrix} \xrightarrow{\Phi} \Phi(\mathbf{x}) = \begin{bmatrix} \Phi^{(1)}(x) \\ \vdots \\ \Phi^{(p)}(x) \end{bmatrix}$$

where $p \gg d$

- If $p \geq n$, then this can lead to problems with overfitting
 - you can **always** find a separating hyperplane
- In addition, however, when p is very large, there can be an increased **computational** burden

The “kernel trick”

Fortunately, there is a clever way to get around this computational challenge by exploiting two facts:

- Many machine learning algorithms only involve the data through *inner products*
- For many interesting feature maps Φ , the function

$$\underline{k(\mathbf{x}, \mathbf{x}')} := \underline{\langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle} = \underline{\Phi(\mathbf{x}')^T \Phi(\mathbf{x})}$$

has a simple, closed form expression that can be evaluated *without explicitly calculating* $\Phi(\mathbf{x})$ and $\Phi(\mathbf{x}')$

$$\langle \cdot, \cdot \rangle = \begin{array}{l} \text{standard} \\ \text{dot product} \end{array}$$

Kernel-based classifiers

Algorithms we've seen that only need to compute inner products:

- nearest-neighbor classifiers

$$\|\mathbf{x} - \mathbf{x}_i\|_2^2 = \langle \mathbf{x} - \mathbf{x}_i, \mathbf{x} - \mathbf{x}_i \rangle$$

$k(x - x_i, x - x_i)$

Kernel-based classifiers

Algorithms we've seen that only need to compute inner products:

- maximum margin hyperplanes

$$(\mathbf{w}^*, b^*) = \arg \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|_2^2 \quad \leftarrow$$

$$\text{s.t. } \underline{y_i(\mathbf{w}^T \mathbf{x}_i + b)} \geq 1 \text{ for } i = 1, \dots, n$$

it is a fact that the optimal \mathbf{w} can be expressed as

$$\mathbf{w} = \sum_{i=1}^n a_i \mathbf{x}_i \text{ for some choice of } a_i$$

$$\|\mathbf{w}\|_2^2 = \left\| \sum_{i=1}^n a_i \mathbf{x}_i \right\|_2^2 = \underline{\sum_{i,j=1}^n a_i a_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle}$$

$$\mathbf{w}^T \mathbf{x}_i = \left(\sum_{j=1}^n a_j \mathbf{x}_j \right)^T \mathbf{x}_i = \sum_{j=1}^n a_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle$$

Quadratic kernel

$$\begin{aligned}(\mathbf{u}^T \mathbf{v})^2 &= \left(\sum_{i=1}^d u(i)v(i) \right)^2 \\ &= \left(\sum_{i=1}^d u(i)v(i) \right) \left(\sum_{j=1}^d u(j)v(j) \right) \\ &= \sum_{i=1}^d \sum_{j=1}^d u(i)v(i)u(j)v(j) \\ &= \sum_{i=1}^d u(i)^2 v(i)^2 + \sum_{\substack{i \neq j \\ \text{---}}} \underbrace{u(i)u(j)} \cdot \underbrace{v(i)v(j)}\end{aligned}$$

Quadratic kernel

$$\sum_{i=1}^d u(i)^2 v(i)^2 + \sum_{i \neq j} u(i)u(j) \cdot v(i)v(j) \stackrel{?}{=} \langle \Phi(\mathbf{u}), \Phi(\mathbf{v}) \rangle$$

What is Φ and what is the dimension p of the corresponding feature space?

$$\Phi(\mathbf{u}) = [u(1)^2, \dots, u(d)^2, \dots,$$

$$\sqrt{2}u(1)u(2), \dots, \sqrt{2}u(d-1)u(d)]^T$$

$$p = d + \frac{d(d-1)}{2} \approx d^2$$

Nonhomogeneous quadratic kernel

$$k(\mathbf{u}, \mathbf{v}) = (\mathbf{u}^T \mathbf{v} + 1)^2 = \underbrace{(\mathbf{u}^T \mathbf{v})^2} + \underbrace{2\mathbf{u}^T \mathbf{v}} + 1$$

In this case, the corresponding $\Phi(\mathbf{u})$ is similar to the homogenous quadratic kernel, but it also replicates \mathbf{u}

$$\Phi(\mathbf{u}) = \left[\underbrace{1, \sqrt{2}u(1), \dots, \sqrt{2}u(d)}_{\text{replicates } \mathbf{u}}, \underbrace{u(1)^2, \dots, u(d)^2}_{\text{quadratic terms}}, \dots, \underbrace{\sqrt{2}u(1)u(2), \dots, \sqrt{2}u(d-1)u(d)}_{\text{cross terms}} \right]^T$$

Polynomial kernels

In general

$$k(\mathbf{u}, \mathbf{v}) = (\mathbf{u}^T \mathbf{v})^m$$
$$= \sum_{\substack{\text{partitions} \\ (j_1, \dots, j_d)}} \binom{m}{j_1, \dots, j_d} u(1)^{j_1} v(1)^{j_1} \dots u(d)^{j_d} v(d)^{j_d}$$

Multinomial theorem

$$(x_1 + \dots + x_d)^m = \sum_{\substack{\text{partitions} \\ (j_1, \dots, j_d)}} \binom{m}{j_1, \dots, j_d} x_1^{j_1} \dots x_d^{j_d}$$

multinomial coef

$j_1 + \dots + j_d = m$

$$\Phi(\mathbf{u}) = \left[\dots, \sqrt{\binom{m}{j_1, \dots, j_d}} u(1)^{j_1} \dots u(d)^{j_d}, \dots \right]^T$$

Inner product kernel

Definition. An *inner product kernel* is a mapping

$$k : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$$

for which there exists an inner product space \mathcal{F} and a mapping $\Phi : \mathbb{R}^d \rightarrow \mathcal{F}$ such that

$$k(\mathbf{u}, \mathbf{v}) = \langle \Phi(\mathbf{u}), \Phi(\mathbf{v}) \rangle_{\mathcal{F}}$$

for all $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$

Given a function $k(\mathbf{u}, \mathbf{v})$, how can we tell when it is an inner product kernel?

- Mercer's theorem
- Positive semidefinite property

Positive semidefinite kernels

We say that $k(\mathbf{u}, \mathbf{v})$ is a **positive semidefinite** kernel if

- k is symmetric
- for all n and all $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$, the **Gram matrix** \mathbf{K} defined by

$$K(i, j) = k(\mathbf{x}_i, \mathbf{x}_j)$$

*n x n
matrix*

is positive semidefinite, i.e., $\mathbf{x}^T \mathbf{K} \mathbf{x} \geq 0$ for all \mathbf{x}

Theorem

k is an inner product kernel if and only if k is a positive semidefinite kernel

Proof: (Future homework)

Examples: Polynomial kernels

Homogeneous polynomial kernel

$$k(\mathbf{u}, \mathbf{v}) = (\mathbf{u}^T \mathbf{v})^m \quad m = 1, 2, \dots$$

Inhomogenous polynomial kernel

$$k(\mathbf{u}, \mathbf{v}) = (\mathbf{u}^T \mathbf{v} + c)^m \quad m = 1, 2, \dots$$
$$c > 0$$

Φ maps to the set of all monomials of degree $\leq m$

Examples: Gaussian/RBF kernels

Gaussian / Radial basis function (RBF) kernel:

$$\underline{k(\mathbf{u}, \mathbf{v})} = (2\pi\sigma^2)^{-d/2} \exp\left(-\frac{\|\mathbf{u} - \mathbf{v}\|^2}{2\sigma^2}\right)$$

$$k(\mathbf{u}, \mathbf{v}) = \exp\left(-\frac{\|\mathbf{u} - \mathbf{v}\|^2}{2\sigma^2}\right)$$

One can show that k is a positive semidefinite kernel, but what is \mathcal{F} ?

$$\Phi: \mathbb{R}^d \rightarrow \mathcal{F}$$

\mathcal{F} is *infinite dimensional!*

Kernels in action: SVMs

In order to understand both

- how to kernelize the maximum margin optimization problem
- how to actually solve this optimization problem with a practical algorithm

we will need to spend a bit of time learning about constrained optimization



This stuff is super useful, even outside of this context

Constrained optimization

A general constrained optimization problem has the form

$$\min_{\mathbf{x}} f(\mathbf{x})$$

such that
subject to

$$\text{s.t. } \underline{g_i(\mathbf{x})} \leq 0 \quad i = 1, \dots, m$$
$$h_i(\mathbf{x}) = 0 \quad i = 1, \dots, \underline{p}$$

where $\mathbf{x} \in \mathbb{R}^d$

We call $f(\mathbf{x})$ the ***objective function***

If \mathbf{x} satisfies all the constraints, we say that \mathbf{x} is ***feasible***

We will assume that $f(\mathbf{x})$ is defined for all feasible \mathbf{x}

The Lagrangian

It is possible to convert a constrained optimization problem into an unconstrained one via the *Lagrangian function*

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & \underline{g_i(\mathbf{x})} \leq 0 \quad i = 1, \dots, m \\ & h_i(\mathbf{x}) = 0 \quad i = 1, \dots, p \end{aligned}$$

In this case, the Lagrangian is given by

$$L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) := \underline{f(\mathbf{x})} + \sum_{i=1}^m \lambda_i \underline{g_i(\mathbf{x})} + \sum_{i=1}^p \underline{\nu_i h_i(\mathbf{x})}$$
$$\lambda_i \geq 0$$

Lagrangian duality

The Lagrangian provides a way to *bound* or *solve* the original optimization problem via a different optimization problem

If we have the Lagrangian

$$L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) := f(\mathbf{x}) + \sum_{i=1}^m \lambda_i g_i(\mathbf{x}) + \sum_{i=1}^p \nu_i h_i(\mathbf{x})$$

$\boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_m]^T$ and $\boldsymbol{\nu} = [\nu_1, \dots, \nu_p]^T$ are called *Lagrange multipliers* or *dual variables*

The *(Lagrange) dual function* is

$$L_D(\boldsymbol{\lambda}, \boldsymbol{\nu}) = \min_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu})$$


The dual optimization problem

We can then define the *dual optimization problem*

$$\begin{aligned} \max_{\lambda, \nu} \quad & L_D(\lambda, \nu) \\ \text{s.t.} \quad & \lambda_i \geq 0 \quad i = 1, \dots, m \end{aligned}$$

Why do we constrain $\lambda_i \geq 0$?

Interesting Fact: No matter the choice of f , g_i , or h_i , $L_D(\lambda, \nu)$ is always **concave**

The primal optimization problem

The *primal function* is

$$L_P(\mathbf{x}) := \max_{\lambda, \nu: \lambda_i \geq 0} L(\mathbf{x}, \lambda, \nu)$$

and the *primal optimization problem* is

$$\min_{\mathbf{x}} L_P(\mathbf{x}) = \min_{\mathbf{x}} \max_{\lambda, \nu: \lambda_i \geq 0} L(\mathbf{x}, \lambda, \nu)$$

Contrast this with the dual optimization problem

$$\max_{\lambda, \nu: \lambda_i \geq 0} \min_{\mathbf{x}} L(\mathbf{x}, \lambda, \nu)$$


What is so “primal” about the primal?

$$\min_{\mathbf{x}} L_P(\mathbf{x}) = \min_{\mathbf{x}} \max_{\lambda, \nu: \lambda_i \geq 0} L(\mathbf{x}, \lambda, \nu)$$

Suppose \mathbf{x} is feasible

$$\begin{aligned} L_P(\mathbf{x}) &= \max_{\lambda, \nu: \lambda_i \geq 0} f(\mathbf{x}) + \sum_{i=1}^m \lambda_i g_i(\mathbf{x}) + \sum_{i=1}^p \nu_i h_i(\mathbf{x}) \\ &= f(\mathbf{x}) \end{aligned}$$

Suppose \mathbf{x} is not feasible

$$L_P(\mathbf{x}) = \infty$$

The primal encompasses the original problem (i.e., they have the same solution), yet it is **unconstrained**

Weak duality

Theorem

$$\begin{aligned} d^* &:= \max_{\lambda, \nu: \lambda_i \geq 0} \min_{\mathbf{x}} L(\mathbf{x}, \lambda, \nu) \\ &\leq \min_{\mathbf{x}} \max_{\lambda, \nu: \lambda_i \geq 0} L(\mathbf{x}, \lambda, \nu) =: p^* \end{aligned}$$

Proof

Let $\tilde{\mathbf{x}}$ be feasible. Then for any λ, ν with $\lambda_i \geq 0$,

$$L(\tilde{\mathbf{x}}, \lambda, \nu) = f(\tilde{\mathbf{x}}) + \sum_i \lambda_i g_i(\tilde{\mathbf{x}}) + \sum_i \nu_i h_i(\tilde{\mathbf{x}}) \leq f(\tilde{\mathbf{x}})$$

$$\begin{aligned} \text{Hence } L_D(\lambda, \nu) &= \min_{\mathbf{x}} L(\mathbf{x}, \lambda, \nu) \\ &\leq \min_{\text{feasible } \tilde{\mathbf{x}}} L(\tilde{\mathbf{x}}, \lambda, \nu) \\ &\leq \min_{\text{feasible } \tilde{\mathbf{x}}} f(\tilde{\mathbf{x}}) = p^* \end{aligned}$$

Weak duality

Theorem

$$\begin{aligned} d^* &:= \max_{\lambda, \nu: \lambda_i \geq 0} \min_{\mathbf{x}} L(\mathbf{x}, \lambda, \nu) \\ &\leq \min_{\mathbf{x}} \max_{\lambda, \nu: \lambda_i \geq 0} L(\mathbf{x}, \lambda, \nu) =: p^* \end{aligned}$$

Proof

Thus, for any λ, ν with $\lambda_i \geq 0$, we have

$$\min_{\mathbf{x}} L(\mathbf{x}, \lambda, \nu) \leq p^*$$

Since this holds for any λ, ν , we can take the max to obtain

$$d^* = \max_{\lambda, \nu: \lambda_i \geq 0} \min_{\mathbf{x}} L(\mathbf{x}, \lambda, \nu) \leq p^*$$

Duality gap

We have just shown that for any optimization problem, we can solve the dual (which is always a concave maximization problem), and obtain a lower bound d^* on p^* (the optimal value of the objective function for the original problem)

The difference $p^* - d^*$ is called the **duality gap**

In general, all we can say about the duality gap is that $p^* - d^* \geq 0$, but sometimes we are lucky...

If $p^* = d^*$, we say that **strong duality** holds

The KKT conditions

Assume that f, g_i, h_i are all differentiable

The Karush-Kuhn-Tucker (KKT) conditions are a set of properties that must hold under strong duality

The KKT conditions will allow us to translate between solutions of the primal and dual optimization problems

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} L_P(\mathbf{x})$$



$$(\boldsymbol{\lambda}^*, \boldsymbol{\nu}^*) = \arg \max_{\boldsymbol{\lambda}, \boldsymbol{\nu}: \lambda_i \geq 0} L_D(\boldsymbol{\lambda}, \boldsymbol{\nu})$$

The KKT conditions

$$1. \nabla f(\mathbf{x}^*) + \sum_{i=1}^m \lambda_i^* \nabla g_i(\mathbf{x}^*) + \sum_{i=1}^p \nu_i^* \nabla h_i(\mathbf{x}^*) = 0$$

$$2. g_i(\mathbf{x}^*) \leq 0, \quad i = 1, \dots, m$$

$$3. h_i(\mathbf{x}^*) = 0, \quad i = 1, \dots, p$$

$$4. \lambda_i^* \geq 0, \quad i = 1, \dots, m$$

$$5. \lambda_i^* g_i(\mathbf{x}^*) = 0 \quad i = 1, \dots, m$$

(complementary slackness)

KKT conditions and convex problems

Theorem

If the original problem is convex, meaning that f and the g_i are convex and the h_i are affine, and $(\tilde{\mathbf{x}}, \tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\nu}})$ satisfy the KKT conditions, then $\tilde{\mathbf{x}}$ is primal optimal, $(\tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\nu}})$ is dual optimal, and the duality gap is zero ($p^* = d^*$).

Moreover, if $p^* = d^*$, \mathbf{x}^* is primal optimal, and $(\boldsymbol{\lambda}^*, \boldsymbol{\nu}^*)$ is dual optimal, then the KKT conditions hold.

Proof: See notes.

KKT conditions: Necessity

Theorem

If $p^* = d^*$, \mathbf{x}^* is primal optimal, and $(\boldsymbol{\lambda}^*, \boldsymbol{\nu}^*)$ is dual optimal, then the KKT conditions hold.

Proof

- 2 and 3 hold because \mathbf{x}^* must be feasible
- 4 holds by the definition of the dual problem
- To prove 5, note that from strong duality, we have

$$\begin{aligned} f(\mathbf{x}^*) &= L_D(\boldsymbol{\lambda}^*, \boldsymbol{\nu}^*) \\ &= \min_{\mathbf{x}} f(\mathbf{x}) + \sum_i \lambda_i^* g_i(\mathbf{x}) + \sum_i \nu_i^* h_i(\mathbf{x}) \\ &\leq f(\mathbf{x}^*) + \sum_i \lambda_i^* g_i(\mathbf{x}^*) + \sum_i \nu_i^* h_i(\mathbf{x}^*) \\ &\leq f(\mathbf{x}^*) \quad (\text{by 2, 3, and 4}) \end{aligned}$$

KKT conditions: Necessity

- This shows that the two inequalities are actually equalities, and hence the equality of the last two lines implies

$$f(\mathbf{x}^*) + \sum_i \lambda_i^* g_i(\mathbf{x}^*) + \sum_i \nu_i^* h_i(\mathbf{x}^*) = f(\mathbf{x}^*)$$

which implies that $\sum_i \lambda_i^* g_i(\mathbf{x}^*) = 0$, which is 5

- Equality of the 2nd and 3rd lines implies that \mathbf{x}^* is a minimizer of $L(\mathbf{x}, \boldsymbol{\lambda}^*, \boldsymbol{\nu}^*)$ with respect to \mathbf{x} , and hence

$$\nabla L(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\nu}^*) = 0$$

which establishes 1

KKT conditions: Sufficiency

Theorem

If the original problem is convex, meaning that f and the g_i are convex and the h_i are affine, and $(\tilde{\mathbf{x}}, \tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\nu}})$ satisfy the KKT conditions, then $\tilde{\mathbf{x}}$ is primal optimal, $(\tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\nu}})$ is dual optimal, and the duality gap is zero.

Proof

- 2 and 3 imply that $\tilde{\mathbf{x}}$ is feasible
- 4 implies that $L(\mathbf{x}, \tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\nu}})$ is convex in \mathbf{x} , and so 1 means that $\tilde{\mathbf{x}}$ is a minimizer of $L(\mathbf{x}, \tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\nu}})$
- Thus $L_D(\tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\nu}}) = L(\tilde{\mathbf{x}}, \tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\nu}})$
$$= f(\tilde{\mathbf{x}}) + \sum_i \tilde{\lambda}_i g_i(\tilde{\mathbf{x}}) + \sum_i \tilde{\nu}_i h_i(\tilde{\mathbf{x}})$$
$$= f(\tilde{\mathbf{x}}) \quad (\text{by feasibility and 5})$$

KKT conditions: Sufficiency

- If $L_D(\tilde{\lambda}, \tilde{\nu}) = f(\tilde{\mathbf{x}})$ then $d^* = \max_{\lambda, \nu: \lambda_i \geq 0} L_D(\lambda, \nu)$
 $\geq L_D(\tilde{\lambda}, \tilde{\nu})$
 $= f(\tilde{\mathbf{x}})$
 $\geq p^*$
- But earlier we proved that $d^* \leq p^*$, and hence it must actually be that $d^* = p^*$, and thus we have
 - zero duality gap
 - $\tilde{\mathbf{x}}$ is primal optimal
 - $(\tilde{\lambda}, \tilde{\nu})$ is dual optimal

KKT conditions: The bottom line

If a constrained optimization problem is

- differentiable
- convex

then the KKT conditions are necessary and sufficient for primal/dual optimality (with zero duality gap)

In this case, we can use the KKT conditions to find a solution to our optimization problem

i.e., if we find $(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\nu}^*)$ satisfying the conditions, we have found solutions to both the primal and dual problems